Université de Lille
Inria Lille-Nord Europe

École doctorale **ED Régionale SPI 72**

Unité de recherche **Équipe-projet MΘDAL**

Thèse présentée par **Quentin Grimonprez**

Soutenue le **14 décembre 2016**

En vue de l'obtention du grade de docteur de l'Université de Lille

Discipline **Mathématiques appliquées**

Spécialité **Statistique**

Titre de la thèse

Sélection de groupes de variables corrélées en grande dimension

Thèse dirigée par	Julien Jacques	directeur
	Alain Celisse	co-encadrant
	Guillemette Marot	co-encadrante

Composition du jury

Rapporteurs	Faicel Chamroukhi	professeur à l'Université de Caen	
	Julien Chiquet	chargé de recherche HDR au MIA Paris	
Examinateurs	Agathe Guilloux	professeure à l'Université d'Évry	
	Étienne Birmelé	chargé de recherche HDR à l'Université Paris Descartes	président du jury
Directeurs de thèse	Julien Jacques	professeur à l'Université de Lyon	
	Alain Celisse	MCF à l'Université de Lille, Sciences et Technologies	
	Guillemette Marot	MCF à l'Université de Lille, Droit et Santé	

Mots clés : group-lasso, classification ascendante hiérarchique, variables corrélées, test hiérarchique, grande dimension, sélection de variables

Keywords: group-lasso, hierarchical clustering, correlated variables, hierarchical testing, high dimension, variable selection

Cette thèse a été préparée dans les laboratoires suivants.

Équipe-projet MΘDAL
Inria Lille-Nord Europe
40 avenue Halley
59650 Villeneuve d'Ascq
France

☎ (+33) 03 59 57 78 00
Site https://modal.lille.inria.fr/

Laboratoire Paul Painlevé
CNRS U.M.R. 8524
59655 Villeneuve d'Ascq Cedex
France

☎ (+33) 03 20 43 48 50
Site https://math.univ-lille1.fr/

La statistique est la première des sciences inexactes.

Edmond et Jules de Goncourt

Il ne faut pas utiliser les statistiques comme les ivrognes utilisent les réverbères : pour s'appuyer et non s'éclairer.

Andrew Lang

Sélection de groupes de variables corrélées en grande dimension

Résumé

Le contexte de cette thèse est la sélection de variables en grande dimension à l'aide de procédures de régression régularisée en présence de redondance entre variables explicatives. Parmi les variables candidates, on suppose que seul un petit nombre est réellement pertinent pour expliquer la réponse. Dans ce cadre de grande dimension, les approches classiques de type Lasso voient leurs performances se dégrader lorsque la redondance croît, puisqu'elles ne tiennent pas compte de cette dernière. Regrouper au préalable ces variables peut pallier ce défaut, mais nécessite usuellement la calibration de paramètres supplémentaires. L'approche proposée combine regroupement et sélection de variables dans un souci d'interprétabilité et d'amélioration des performances. D'abord une Classification Ascendante Hiérarchique (CAH) fournit à chaque niveau une partition des variables en groupes. Puis le Group-lasso est utilisé à partir de l'ensemble des groupes de variables des différents niveaux de la CAH à paramètre de régularisation fixé. Choisir ce dernier fournit alors une liste de groupe candidats issus potentiellement de différents niveaux. Le choix final des groupes est obtenu via une procédure de tests multiples.
La procédure proposée exploite la structure hiérarchique de la CAH et des pondérations dans le Group-lasso. Cela permet de réduire considérablement la complexité algorithmique induite par la flexibilité liée à la possibilité de choisir des groupes issus de différents niveaux de la CAH.

Mots clés : group-lasso, classification ascendante hiérarchique, variables corrélées, test hiérarchique, grande dimension, sélection de variables

Abstract

This thesis takes place in the context of variable selection in the high dimensional setting using penalized regression in presence of redundancy between explanatory variables. Among all variables, we suppose that only a few number is relevant for predicting the response variable. In this high dimensional setting, performance of classical lasso-based approaches decreases when redundancy increases as they do not take it into account. Firstly aggregating variables can overcome this problem but generally requires calibration of additional parameters.
The proposed approach combines variables aggregation and selection in order to improve interpretability and performance. First, a hierarchical clustering procedure provides at each level a partition of the variables into groups. Then the Group-lasso is used with the set of groups of variables from the different levels of the hierarchical clustering and a fixed regularization parameter. Choosing this parameter provides a list of candidates groups potentially coming from different levels. The final choice of groups is done by a multiple testing procedure.
The proposed procedure exploits the hierarchical structure from hierarchical clustering and some weights in Group-lasso. This allows to greatly reduce the algorithm complexity induced by the possibility to choose groups coming from different levels of the hierarchical clustering.

Keywords: group-lasso, hierarchical clustering, correlated variables, hierarchical testing, high dimension, variable selection

Équipe-projet MΘDAL
Inria Lille-Nord Europe – 40 avenue Halley – 59650 Villeneuve d'Ascq – France

Remerciements

Je voudrais tout d'abord remercier Alain Celisse, Guillemette Marot et Julien Jacques pour m'avoir encadré durant ces années de thèse. Je remercie Faicel Chamroukhi et Julien Chiquet d'avoir accepter de rapporter ma thèse et pour leurs remarques constructives qui ont permis d'améliorer la qualité de ce manuscrit. Je remercie également Étienne Birmelé et Agathe Guilloux d'avoir acceptés d'être memres de mon jury.

Je remercie Inria et la Direction Générale de l'Armement de m'avoir accordé un financement pour cette thèse.

Je remercie également l'ensemble des membres de l'équipe MΘDAL d'Inria Lille qui m'ont accompagné durant toutes ces années et notamment Christophe Biernacki dont les cours en M2 ISN ont augmenté de manière significative (à $\alpha = 0.05$) mon intérêt pour les statistiques.

Comment ne pas remercier les membres du bureau A106 : Vincent, Samuel, Jérémie, Clément. On a partagé de bons moments dans et en dehors des bureaux successifs que nous avons occupés. Résumer leur compagnie à une simple aide statistique et informatique serait très réducteur. Rires, conversations futiles, jeux vidéo, jeux de société,... ont fait partie de la vie du bureau A106. Encore merci pour tous ces moments (plus ou moins) studieux passés dans la bonne humeur. Gaiement, finissons sur le cri de A106 : ouaaaaaaaaaaaaaaaaaaaaaaaaaaais !

Un petit mot pour la génération suivante de doctorants et plus particulièrement Maxime avec qui j'ai partagé un bureau durant la 2e partie de ma thèse. À ses côtés, j'ai connu beaucoup de défaites à Age of Empires 2 avant de connaître de belles victoires. J'ai pu également disputer des parties de badminton (je le battrai un jour) et fait quelques tours en vélo en sa compagnie. Quitter le bureau de Maxime soulagera mes oreilles de l'énoncé régulier du théorème de Bolzano-Weierstrass et de ses conséquences qui lui sont si chères.

Je remercie également mes amis : Jc, Dub, Olic, Tom, l'attoucheur, Faux Quentin, le roi du Togo pour les bons moments passés ensemble durant toutes ces années.

Je remercie ma famille et plus particulièrement ma mère qui m'a soutenu tout au long de mes études universitaires.

Notations

$\mathcal{M}_{n,p}(\mathbb{R})$ espace des matrices réelles de taille $n \times p$
I_n matrice identité de taille $n \times n$
0_n vecteur nul de longueur n
$0_{n,p}$ matrice nulle de taille $n \times p$
X^T transposée de la matrice X
x_i i^{e} élément du vecteur x
X_i i^{e} colonne de la matrice X
X_S matrice X restreinte aux colonnes dont l'indice appartient à l'ensemble S
$X_{i,.}$ i^{e} ligne de la matrice X
X_{ij} élément à la i^{e} ligne et j^{e} colonne de la matrice X
$\mathbb{S}(\text{x})$ support du vecteur x $\{i \mid x_i \neq 0\}$
n nombre d'individus
p nombre de variables
X matrice de design de taille $n \times p$
y vecteur réponse de longueur n
ϵ vecteur bruit de longueur n
β^* vecteur solution (inconnu) de longueur p
$\mathcal{G}$ partition de l'ensemble $\{1, \dots, p\}$
k nombre d'éléments non nuls de β^*
K nombre de groupes contenant au moins un élément non nuls de β^*
l taille des blocs au sein de matrice de la variance-covariance
ρ corrélation au sein des blocs de la matrice de variance-covariance
σ^2 variance du bruit
$\text{Card}(S)$ ou $|S|$ cardinal de l'ensemble S
$\|x\|_0$ norme ℓ_0 du vecteur x définie comme le nombre d'éléments non nuls de x, $\text{Card}\left(\left\{j \mid x_j \neq 0\right\}\right)$
$\|x\|_q$ norme ℓ_q du vecteur x définie par $\left(\sum_{i=1}^n |x_i|^q\right)^{\frac{1}{q}}$
$\|\!|X|\!\|_\infty$ norme matricielle définie par $\max_{i=1,\dots,n} \sum_{j=1}^p |X_{ij}|$
$\text{cor}(x, y)$ corrélation entre le vecteur x et y

$\text{sign}(x_i)$ le signe d'un réel x_i. Il vaut 1 si $x_i > 0$, -1 si $x_i < 0$ et 0 si $x_i = 0$

$\text{sign}(x)$ le vecteur contenant le signe des éléments du vecteur x

$f(n) = O(g(n))$ $\exists k > 0, \exists n_0 \; \forall n > n_0 \; |f(n)| \leq |g(n)| \cdot k$

$f(n) = \Omega(g(n))$ $\exists k > 0, \exists n_0 \; \forall n > n_0 \; |g(n)| \cdot k \leq |f(n)| \; \exists k > 0, \exists n_0 \; \forall n > n_0 \; |g(n)| \cdot k \leq |f(n)|$

Sommaire

Table des figures

Introduction

Depuis plusieurs années, la taille des données collectées a considérablement augmentée. On peut par exemple citer le domaine de la biologie où l'étude du génome humain amène à traiter des milliers de variables pour un nombre restreint d'individus. Le cas dit de la grande dimension où le nombre d'individus n est inférieur au nombre de variables p est alors devenu très répandu. Dans le domaine de l'analyse prédictive où le but est de prédire une réponse à l'aide d'un ensemble de variables explicatives, la méthode classique des moindres carrés souffre de problème en grande dimension ($n < p$). Dans ce cadre les méthodes de sélection de variables qui consistent à expliquer la réponse en fonction d'un nombre restreint de variables se sont développées. Un des buts est ainsi de gagner en interprétation car au vu de la quantité de variables disponibles, il est naturel de penser que seul un sous-ensemble de celles-ci ont effectivement un impact dans l'explication de la réponse. Dans ce but, une des méthodes les plus populaires est le lasso (Robert Tibshirani 1994) (et ses différentes variantes) qui est une méthode dite de régression pénalisée. Par l'introduction d'une pénalité ℓ_1 sur les coefficients à estimer, le lasso effectue sélection et prédiction.

La grande dimension amène également le problème de redondance au sein des données. Plusieurs variables portent la même information, cela se traduit par une corrélation élevée entre elles. Ces corrélations peuvent perturber les estimateurs de différente méthodes. L'estimation des coefficients peut être peu précise et avec une forte variance. La répétabilité de l'estimation lorsque de nouveaux individus sont ajoutés ou retirés peut aboutir à des estimateurs différents à cause de ces corrélations. Les variables sélectionnées entre les estimateurs peuvent varier et l'ensemble des variables corrélées à une variable d'intérêt ne sont pas forcément toutes sélectionnées. Dans ce contexte de corrélation, la sélection de groupes de variables permet d'éviter ces phénomènes en regroupant au préalable les variables en fonction de leurs corrélations. Ainsi, si le regroupement de variables a été fait de manière efficace, une variable d'importance et les variables qui lui sont corrélées font partie d'un même groupe et sont alors sélectionnées simultanément. Des méthodes combinant regroupement de variables et sélection ont été développées dans la littérature. Ces méthodes se basent généralement sur une unique partition des variables fournie a priori. Le choix d'une mauvaise partition peut alors entraîner de mauvaises performances de la méthode de sélection. Dans cette thèse, nous introduisons une méthode utilisant un ensemble de partitions des variables pour la sélection de groupes.

Ce manuscrit s'organise de la manière suivante :

Dans le chapitre 1, nous commençons par un état de l'art des méthodes de régression pénalisée dans le contexte de la sélection de variables. Puis nous énonçons les méthodes de classification non supervisée les plus couramment utilisées dans le but de regrouper les variables en fonction de leur dépendance. Enfin, les méthodes utilisant le regroupement de variables et la sélection en présence de corrélation sont présentées.

Une nouvelle méthode, appelée *Multi-Layer Group-Lasso*, combinant regroupement de va-

riables et sélection de groupes de variables est proposée dans le chapitre 2. Elle diffère des méthodes précédentes par l'utilisation de plusieurs partitions des variables et donc la possibilité de sélectionner des groupes de partitions distinctes. L'efficacité de la méthode est éprouvée sur différentes simulations et sur un jeu de données réelles.

Dans le chapitre 3, une procédure dont le but est de contrôler le nombre de faux positifs parmi les groupes sélectionnés par le *Multi-Layer Group-Lasso* est présentée. Cette procédure se base sur les tests statistiques et plus particulièrement les tests hiérarchiques, qui sont introduits à cette occasion. Ceci afin de prendre compte les particularités de notre méthode de sélection : la sélection de groupes issus de niveaux distincts de la hiérarchie et donc partageant potentiellement des variables en commun.

Le Chapitre 4 est dédié au choix du paramètre de régularisation λ du *Multi-Layer Group-Lasso*. La stratégie proposée sera comparée aux stratégies usuelles de choix de λ.

Les méthodes développées dans cette thèse ont été implémentées dans un package R (R Core Team 2016) `MLGL` disponible en ligne. Une description des principales fonctions et de son utilisation est présentée dans le chapitre 5.

Enfin, les conclusions et perspectives sont discutées en fin de manuscrit.

En annexe A, une réécriture du problème du fused-lasso est proposée afin d'utiliser un algorithme de résolution adapté pour le lasso.

Le travail ayant amené les questions de cette thèse a abouti à un article présenté en annexe C et un package présenté dans le chapitre 5.

Chapitre 1

État de l'art

Dans ce chapitre, nous nous intéressons aux méthodes de régression pénalisée dans le cadre de la sélection de variables en présence de corrélation. Les méthodes de régression pénalisée les plus courantes dans un cadre plus général que celui de la corrélation sont présentées. Nous nous intéressons ensuite à l'optimisation des paramètres de régularisation de ces méthodes.

En vue de faire de la sélection de groupes de variables, les méthodes de regroupement de variables sont introduites. Enfin, nous exposons quelques méthodes de la littérature combinant regroupement de variables et sélection.

1.1 Régression pénalisée pour la sélection de variables

Soit $X \in \mathcal{M}_{n,p}(\mathbb{R})$, la matrice décrivant la valeur de p variables mesurées sur n individus et représentées par les vecteurs colonnes $X_1, \dots, X_p$. Soit $y \in \mathbb{R}^n$, le vecteur réponse associé aux n individus. On se place dans le cadre d'une relation linéaire entre y et les colonnes de X :

$$y = \beta_0^* + X\beta^* + \epsilon$$

avec $\beta_0^* \in \mathbb{R}^n$ un vecteur constant contenant la constante du modèle, $\beta^* \in \mathbb{R}^p$ le vecteur (inconnu) des coefficients de la régression et $\epsilon \in \mathbb{R}^n$ un bruit centré et de matrice de variance-covariance $\sigma^2 I_n$ où I_n représente la matrice identité de taille n. Généralement, on suppose que ϵ est issu d'une loi $\mathcal{N}(0_n, \sigma^2 I_n)$. Nous imposons que le vecteur y soit centré ainsi que les colonnes de la matrice X afin d'éliminer la constante dans le modèle.

Le but est d'estimer les coefficients β^*. Dans notre cas, nous supposons que ce vecteur est creux, c'est-à-dire que seul un sous-ensemble d'éléments de β^* est non nul ($\|\beta^*\|_0 = k < p$). On appelle le support de β^*, les indices des éléments non nuls de β :

$$\mathbb{S}(\beta^*) = \left\{ j \in \{1, \dots, p\} \mid \beta_j^* \neq 0 \right\}.$$

On appelle une variable appartenant au support de β^* une *vraie variable* et une hors du support, une *fausse variable*.

Chaque coordonnée de β^* détermine l'influence de la variable dans l'explication de la réponse. Si $\beta^* = 0$ alors la variable X_i n'intervient pas pour l'expliquer. Trouver $\mathbb{S}(\beta^*)$, le support de β^*, est donc équivalent à effectuer de la sélection de variables. Dans la suite, nous nous intéressons à

différentes méthodes pour estimer le support de β^* par le biais de l'estimation de β^*.

1.1.1 Méthodes usuelles

Moindres carrés ordinaires

La méthode des moindres carrés ordinaires est la plus courante dans le cadre de l'estimation des coefficients d'un modèle linéaire. Elle consiste à minimiser la somme des carrés des résidus. L'estimateur $\hat{\beta}^{LS}$ de β^* est :

$$\hat{\beta}^{LS} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2} \|y - X\beta\|_2^2 \right\}, \tag{1.1}$$

où pour $x \in \mathbb{R}^n$, $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$.

L'estimateur des moindres carrés est défini de manière unique lorsque la matrice X est de plein rang. Dans ce cas, il est défini de manière explicite par :

$$\hat{\beta}^{LS} = (X^T X)^{-1} X^T y. \tag{1.2}$$

D'après le théorème de Gauss-Markov, l'estimateur des moindres carrés (1.2) est celui de variance minimale parmi tous les estimateurs sans biais de β^*, on dit qu'il est le BLUE (*Best Linear Unbiaised Estimator*). On a donc $\mathbb{E}[\hat{\beta}^{LS}] = \beta^*$, sa variance quant à elle est $\text{Var}(\hat{\beta}^{LS}) = (X^T X)^{-1}\sigma^2$.

Dans le cas de la grande dimension ($n < p$), cet estimateur pose problème : l'estimateur des moindres carrés n'est pas défini de manière unique et n'est pas identifiable.

Dans la suite, nous allons explorer différentes méthodes de régression pénalisée permettant de pallier à ce problème de l'estimateur des moindres carrés.

Régression Ridge

Une extension naturelle des moindres carrés à la grande dimension est la régression ridge (Hoerl et Kennard 1970). Elle consiste en l'ajout d'une contrainte sur la norme ℓ_2 du vecteur des coefficients à estimer. Ainsi, l'estimateur de la régression ridge est :

$$\hat{\beta}_\lambda^{\textbf{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2 \right\} \tag{1.3}$$

avec $\lambda > 0$ un paramètre de régularisation. Plus λ sera grand, plus les coefficients estimés seront contraints à prendre des petites valeurs en valeur absolue. L'optimiser en λ est une tâche nécessaire à l'obtention d'une solution optimale pour la prédiction. Parmi les méthodes les plus courantes, on citera la validation croisée V-fold, les critères d'informations (cf. Section 1.1.4 pour plus de détails).

Tout comme le problème des moindres carrés, la régression ridge possède une solution explicite :

$$\hat{\beta}_\lambda^{\textbf{ridge}} = (X^T X + \lambda I_n)^{-1} X^T y, \ \forall \lambda > 0. \tag{1.4}$$

On constate l'ajout d'une constante λ sur la diagonale de la matrice $X^T X$ dont les valeurs propres sont nécessairement positives ou nulles. La matrice $(X^T X + \lambda I_n)$ a alors toutes ses valeurs

propres strictement positives et est donc inversible. Elle diminue également le problème de mauvais conditionnement à l'origine d'une forte variance.

L'utilisation de l'estimateur de la régression ridge peut être vue comme un compromis biais-variance. En rajoutant une pénalité ℓ_2 sur le vecteur des coefficients estimés, on introduit un biais dans notre estimateur mais sa variance s'en trouve réduite. Plus λ augmente, plus le biais de l'estimateur ridge augmente. En contrepartie, plus λ augmente, plus la variance diminue.

Avec la régression ridge (1.3), la grande dimension ($n \geq p$) ne pose plus de problème mais l'estimateur (1.4) ne fournit pas de solution parcimonieuse, c'est-à-dire de solution ayant un support de taille strictement inférieure au nombre de variables.

Least Absolute Shrinkage and Selection Operator (Lasso)

L'idée du lasso (Robert Tibshirani 1994) est d'obtenir un estimateur parcimonieux de notre problème des moindres carrés. Ajouter une pénalité ℓ_0 sur le vecteur des coefficients à estimer peut sembler la façon la plus naturelle d'obtenir de la parcimonie, mais cela résulte en un problème NP-difficile (Natarajan 1995). Un autre moyen d'obtenir cette parcimonie est d'utiliser une pénalité ℓ_1 sur le vecteur des coefficients à estimer. Ainsi, l'estimateur lasso est défini par :

$$\hat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \tag{1.5}$$

avec $\lambda \geq 0$ un paramètre de régularisation, et $\|\beta\|_1 = \sum_{i=1}^{p} |\beta_i|$ définit la norme ℓ_1 du vecteur β. Plus la valeur de λ est faible, plus le nombre de coefficients non nuls est grand.

À l'inverse des moindres carrés et de la régression ridge, le lasso n'a pas de solution explicite (sauf dans le cas où $X^T X = I_n$). L'utilisation d'algorithmes tels que le lars (Efron et al. 2004) ou la descente de coordonnées (Friedman, Hastie et Robert Tibshirani 2010b) est nécessaire pour obtenir des solutions.

Pour comprendre pourquoi une pénalité ℓ_1 permet d'obtenir un estimateur parcimonieux et non une pénalité ℓ_2, on peut se reporter à la figure 1.1 représentant un cas en dimension 2. La zone bleue définit la zone de contrainte imposée par une valeur de λ fixée. Quant aux ellipses rouges, elles représentent les courbes de niveaux de la solution des moindres carrés ($\hat{\beta}$) où la norme croit progressivement jusqu'à atteindre la zone définie par la pénalité lasso ou ridge. La solution du lasso (respectivement ridge) est le premier point que rencontre ces lignes de niveau. La solution de la régression ridge et du lasso correspond au premier point de la zone de contrainte que les courbes de niveaux rencontrent la zone de pénalité. La forme de la pénalité lasso (due à la non différentiabilité de la fonction valeur absolue en 0) fournit plus d'opportunités d'obtenir des solutions parcimonieuses que la forme sphérique de la pénalité.

Ainsi, l'estimateur lasso sera parcimonieux, il contiendra au plus $\min(n, p)$ variables (Osborne, Presnell et Turlach 1999 ; R. J. Tibshirani 2013). La possibilité de faire de la régression et de la sélection en même temps en a fait une méthode très populaire ainsi que le développement d'algorithmes efficaces en temps de calcul et en stockage.

En résolvant le lasso, un chemin solution dépendant du paramètre de régularisation λ est obtenu. Ce chemin représente l'évolution des valeurs des coefficients estimés en fonction de λ. Ces différentes courbes sont linéaires par morceaux (cf. figure 1.2). Les ruptures de linéarités se font pour les valeurs du paramètre λ pour lesquelles le support de la solution change. Ces valeurs peuvent être trouvées à l'aide de l'algorithme lars. Le choix du paramètre λ est une étape importante et doit être choisi en fonction du but de l'utilisateur : prédiction et/ou sélection (cf. Section 1.1.4).

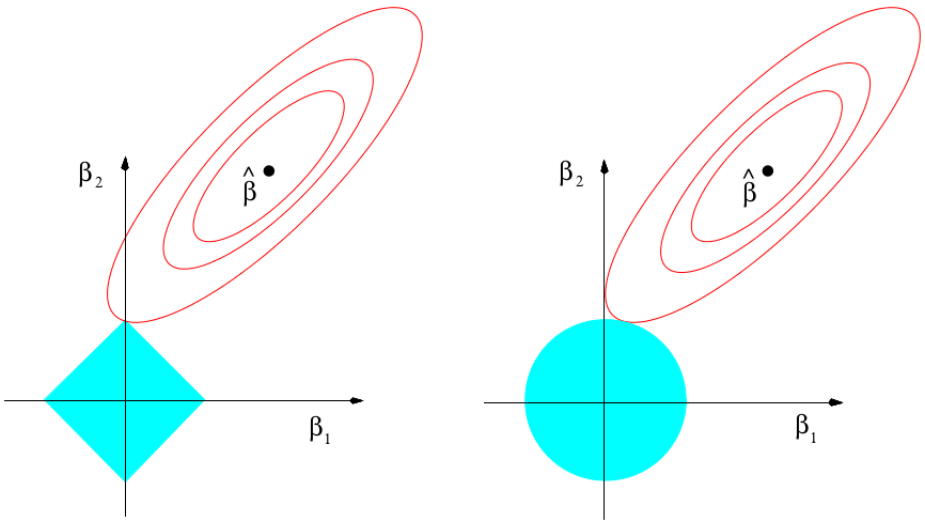


FIGURE 1.1 – À gauche (respectivement droite), en bleu, contour de la pénalité lasso (ridge) en dimension 2. $\hat{\beta}$ correspond à la solution des moindres carrés. En rouge, les courbes de niveaux de la solution des moindres carrés. Figure issue de (HASTIE, Robert TIBSHIRANI et FRIEDMAN 2003).

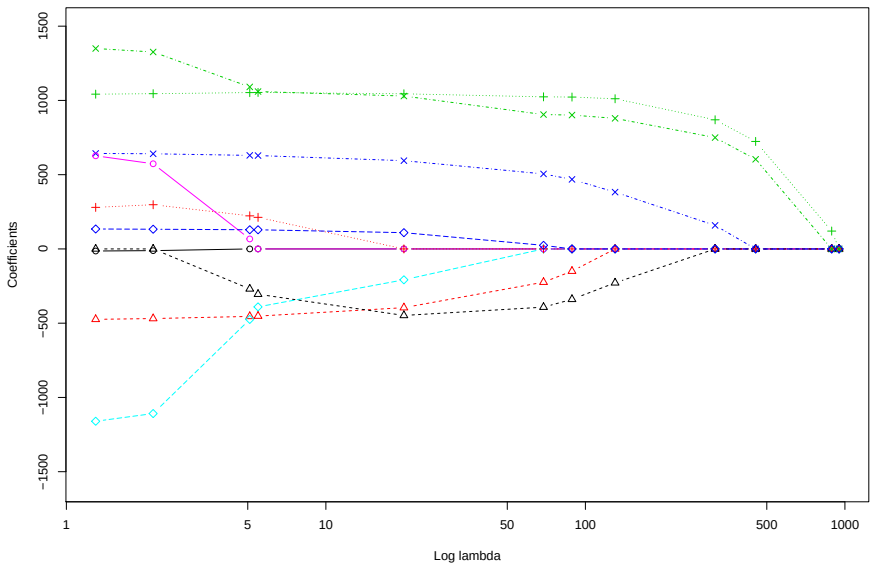


FIGURE 1.2 – Exemple de chemin solution du lasso. Les différentes valeurs de λ se trouvent en abscisse et la valeur des coefficients estimés en ordonnée. Une courbe représente l'évolution d'un coefficient estimé en fonction de λ.

Consistance de la sélection du lasso Dans la littérature, certains auteurs (Zhao et Yu 2006 ; Wainwright 2009) se sont intéressés à la consistance des variables sélectionnées du lasso, i.e. dans le cas où le support de β^* est creux, à l'existence d'une valeur de λ pour laquelle le lasso retrouve exactement le vrai support.

Une condition suffisante pour obtenir la consistance des variables sélectionnées est la *condition d'irreprésentabilité*, que l'on retrouve dans les deux articles cités ci-dessus.

Notons $S = \mathbb{S}(\beta^*)$ le support de la vraie solution β^* et S^c son complémentaire. On note X_S la matrice X restreinte aux colonnes dont l'indice appartient à l'ensemble S. Définissons $\text{sign}(x)$ le vecteur contenant le signe de chaque élément de x :

$$(\text{sign}(x))_i = \begin{cases} 1 & \text{si } x_i > 0 \\ 0 & \text{si } x_i = 0 \\ -1 & \text{si } x_i < 0 \end{cases}$$

La *condition d'irreprésentabilité* est satisfaite pour $\eta \in]0,1]$ si :

$$\left\| \frac{1}{n} X_{S^c}^T X_S \left(\frac{1}{n} X_S^T X_S \right)^{-1} \text{sign}(\beta_S^*) \right\|_\infty \leq 1 - \eta \tag{1.6}$$

où $\|x\|_\infty = \max_{i=1,\dots,n} |x_i|$.

Comme le signe de β_S^* est généralement inconnu, la condition se rencontre également sous la forme

$$\left\| \frac{1}{n} X_{S^c}^T X_S \left(\frac{1}{n} X_S^T X_S \right)^{-1} \right\|_\infty \leq 1 - \eta \tag{1.7}$$

où $|||M|||_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{p} |M_{ij}|$.

Cette condition nous fait penser à une contrainte sur les coefficients de régression des fausses variables (X_{S^c}) sur les vraies (X_S). Les fausses variables ne doivent avoir que peu de lien avec les vraies. En particulier, la condition (1.6) est d'autant plus violée que vraies et fausses variables sont fortement liées (par le biais de corrélation).

Pour mettre en lumière l'importance de cette hypothèse, nous exposons brièvement le théorème de Wainwright (2009). Le lecteur est invité à se référer à l'article pour plus de détails à propos des hypothèses et des grandeurs impliquées.

Nous nous plaçons dans le cadre d'une matrice de design X gaussienne de matrice de variance-covariance Σ. Supposons que la matrice Σ vérifie la condition d'irreprésentabilité pour une certaine valeur de $\eta \in]0,1]$:

$$\left\| \Sigma_{S^c S} \left(\Sigma_{SS} \right)^{-1} \right\|_\infty \leq 1 - \eta \tag{1.8}$$

avec $\Sigma_{S^c S}$ la matrice Σ restreinte aux lignes de S^c et aux colonnes de S.

On considère également une famille particulière de valeur de λ_n.

S'il existe $\delta > 0$ tel que

$$\frac{n}{2k \log(p-k)} > (1+\delta)\, \text{f}(\sigma^2, k, \Sigma, \lambda_n).$$

Alors, on a avec probabilité convergeant vers 1 quand n tend vers $+\infty$:

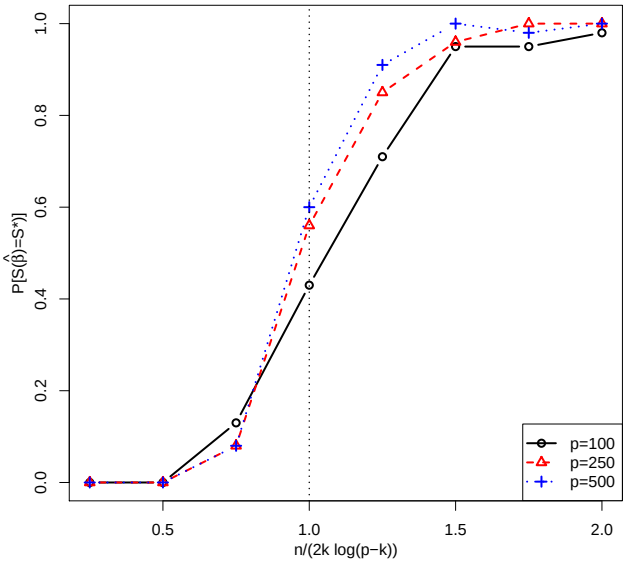


FIGURE 1.3 – Probabilité de retrouver exactement le vrai support ($\mathbb{P}[\mathbb{S}(\hat{\beta}) = \mathbb{S}(\beta^*)]$) en fonction du ratio $\frac{n}{2k\log(p-k)}$ (qui est la borne fournie par le théorème de WAINWRIGHT (2009)). Le schéma de simulation suit celui du premier exemple de WAINWRIGHT : les données sont simulées suivant une loi normale multivariée $\mathcal{N}(0_p, I_p)$, β^* contient $k = 0.1p$ éléments non nuls de valeur 0.5.

1. Le lasso a une unique solution $\hat{\beta}$ avec $\mathbb{S}(\hat{\beta}) \subseteq \mathbb{S}(\beta^*)$.
2. Si $\min_{i \in S} |\beta_i^*| > g(\lambda_n)$, alors $\text{sign}(\hat{\beta}) = \text{sign}(\beta^*)$ et $\|\hat{\beta}_S - \beta_S^*\|_\infty \leq g(\lambda_n)$.

Ainsi sous certaines conditions, le lasso est capable, avec probabilité tendant vers 1, de trouver une solution dont le support est inclus dans celui de β^*. Le vrai support est exactement retrouvé si les coefficients de la vraie solution β^* ne sont pas trop petits.

Ce théorème est un résultat asymptotique en n mais le nombre de variables p et la taille K du vrai support S sont autorisés à croître avec n.

La relation entre n, p et k du théorème de WAINWRIGHT peut être décrite de manière générale par $n = \Omega(k\log(p-k))$ (n est minorée par $k\log(p-k)$ à un facteur près). Ainsi, le nombre d'individus nécessaire à la consistance des variables sélectionnées par le lasso dépend du nombre de variables total et du nombre de vraies variables. Dans le cas où les variables sont indépendantes ($\Sigma = I_p$), la relation devient $n = 2k\log(p-k)$ pour que le lasso retrouve avec grande probabilité les vraies variables. Cette borne doit être gardée à l'esprit quand l'on simule ou traite des cas réels avec le lasso. Nous illustrons cette borne avec la figure 1.3.

De la condition d'irreprésentabilité découle le fait qu'en pratique si une vraie variable est fortement corrélée avec d'autres, généralement, une seule de cet ensemble sera sélectionnée par le lasso. La même variable ne sera pas forcément sélectionnée lors de réexécutions du lasso sur un nouvel échantillon issu d'une même population. Ces variables non sélectionnées le sont, non pas parce qu'elles ne contribuent pas de manière significative à l'explication de la réponse, mais parce qu'une autre variable déjà sélectionnée apporte la même information. Cela pose

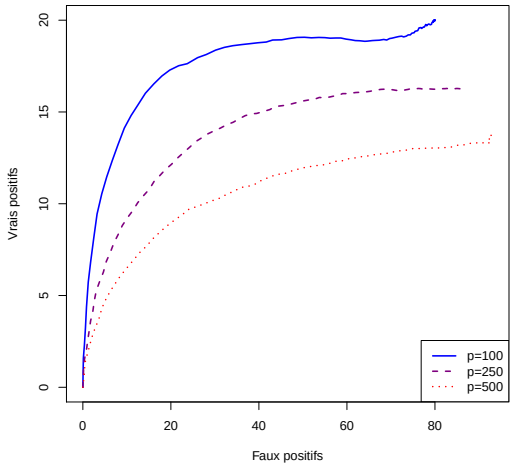


FIGURE 1.4 – Qualité du chemin solution du lasso en matière de vrais et faux positifs pour différentes valeurs du nombre de variables p dans un cas indépendant. Une courbe représente le nombre de vrais positifs en fonction du nombre de faux positifs pour l'ensemble des valeurs de λ testées.

des problèmes de répétabilité et d'interprétation notamment, puisque l'on souhaiterait obtenir l'ensemble de ces variables corrélées.

1.1.2 Dépendance entre vecteurs de variables

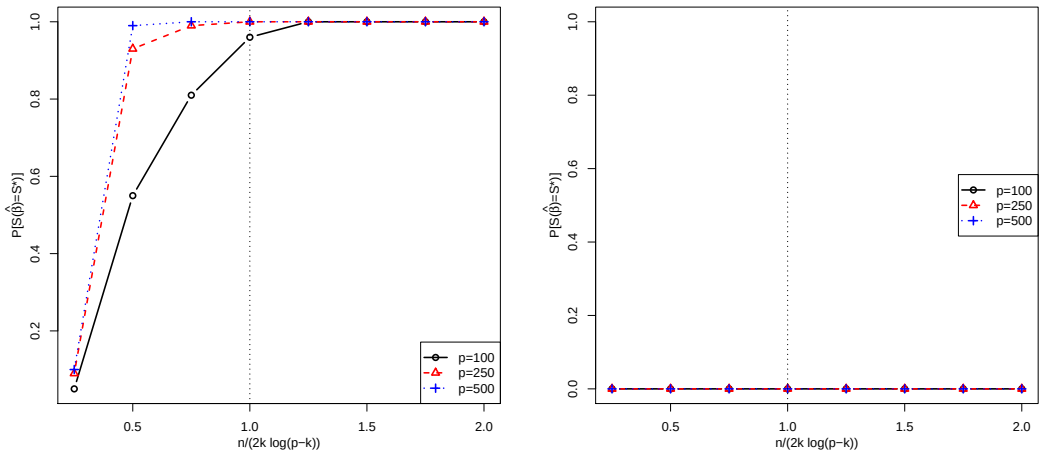
On distingue deux types de dépendances entre variables.

Dépendance due à la dimension

La première est une conséquence de la grande dimension ($n \ll p$). La grande dimension induit une dépendance linéaire entre les variables. Plus le nombre de variables p est grand par rapport au nombre de variables n, plus cette dépendance est présente au sein des données. Cela entraîne une dégradation des résultats dans le cas du lasso.

Pour illustrer cela, on simule une matrice X dont les individus suivent une loi gaussienne multivariée indépendante pour différentes valeurs du nombre de variables p. Une réponse y est simulée en suivant le modèle linéaire : $y = X\beta^* + \epsilon$ avec β^* un vecteur contenant 20 éléments non nuls et ϵ un vecteur gaussien centré.

Sur la figure 1.4, le chemin de solution est représenté en matière de vrais positifs et de faux positifs pour différentes valeurs de p. On constate qu'augmenter le nombre de variables, en gardant un nombre d'individus constant et un même nombre de vraies variables, diminue les performances du lasso. On retrouve moins de vrais positifs pour un même nombre de faux positifs.



(a) Cas de corrélation entre vraies variables

(b) Cas de corrélation entre vraies et fausses variables

FIGURE 1.5 – Probabilité de retrouver exactement le vrai support ($\mathbb{P}[\mathbb{S}(\hat{\beta}) = \mathbb{S}(\beta^*)]$) en fonction du ratio $\frac{n}{2k\log(p-k)}$. Un trait vertical indique la borne dans le cas indépendant.

Dépendance structurelle

Le second type de dépendance est une dépendance structurelle. Une structure de corrélation existe au sein des variables. Deux cas se produisent. Premièrement, les variables du support de la vraie solution β^* sont corrélées avec d'autres variables du support. Deuxièmement, les variables du support de β^* sont corrélées avec des variables non incluses dans le support.

Nous allons illustrer cette dépendance dans le cadre de simulation suivant :

— nombre de variables $p \in \{100, 250, 500\}$;
— $X_{1,.}, \dots, X_{n,.} \sim \mathcal{N}(0_p, \Sigma_\rho)$;
— Σ_ρ, une matrice diagonale par blocs de taille 5×5 où chaque bloc contient des 1 sur la diagonale et ρ partout ailleurs ;
— $\epsilon \sim \mathcal{N}(0_n, \sigma^2 I_n)$;
— $y = X\beta^* + \epsilon$ avec $\text{Card}(\mathbb{S}(\beta^*)) = 0.1 * p$.

Sur la figure 1.5, la probabilité de retrouver exactement le vrai support au sein du chemin solution est représentée dans le cas de corrélation entre les variables. Sur la figure 1.5a, les résultats pour le cas de dépendance entre vraies variables sont présentées. En comparant les courbes avec celles de la figure 1.3 (p. 8), on constate que les corrélations entre vraies variables facilitent le travail de sélection. En effet, moins d'individus sont nécessaires que dans le cas indépendant pour obtenir une probabilité de sélection de 1. À l'inverse, sur la figure 1.5b, la dépendance entre vraies et fausses variables complique la tâche de sélection pour le lasso. Pour les mêmes valeurs de n testées, les K vraies variables n'ont pas été retrouvées une seule fois.

Dans la suite, nous allons nous intéresser aux différentes méthodes de régression pénalisée traitant de la sélection de groupes de variables dans le cadre de corrélation.

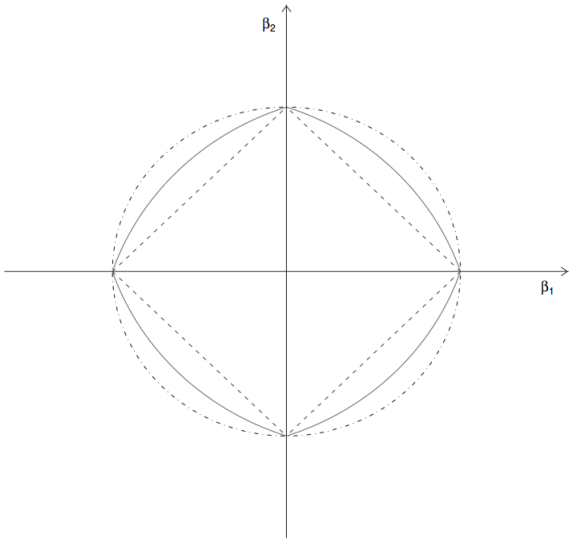


FIGURE 1.6 – Contour de la pénalité de l'elastic-net avec $\alpha = 0.5$ (———), du lasso (− − −) et de la régression ridge (− · − ·) en dimension 2. Figure issue de (ZOU et HASTIE 2005).

1.1.3 Stratégie de sélection sous dépendance

Elastic-Net

L'elastic-net (ZOU et HASTIE 2005) est un compromis entre le lasso et la régression ridge. Les deux pénalités de ces méthodes y sont présentes.

On définit ainsi l'estimateur de l'elastic-net :

$$\hat{\beta}^{\text{enet}}_{\lambda,\alpha} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\Big(\alpha\|\beta\|_1 + (1-\alpha)\|\beta\|_2^2\Big)\right\} \tag{1.9}$$

avec $\lambda \geq 0$ le paramètre de régularisation et $0 \leq \alpha \leq 1$ celui contrôlant le compromis entre la pénalité lasso et ridge. $\alpha = 1$ correspond au lasso, $\alpha = 0$ à la régression ridge. Les contours de la pénalité sont visible sur la figure 1.6.

Ce compromis entre ridge et lasso permet de sélectionner plus de $\min(n, p)$ variables tout en gardant un estimateur parcimonieux. La pénalité ridge va entraîner un effet de regroupement des variables corrélées que n'a pas la pénalité lasso grâce à la stricte convexité de la pénalité de l'elastic-net lorsque $\lambda_2 > 0$ (ZOU et HASTIE 2005). Donc, plusieurs variables fortement corrélées pourront être sélectionnées simultanément. Mais l'elastic-net ne définit pas à proprement parler de groupes et une analyse supplémentaire est nécessaire pour déterminer d'éventuels groupes. L'étude du chemin solution pour trouver les variables dont le coefficient estimé devient non nul à partir de la même valeur du paramètre de régularisation λ et à α fixé peut aider à déterminer des groupes a posteriori.

Fused-lasso

L'idée du fused-lasso (Robert TIBSHIRANI, SAUNDERS et al. 2005) est de prendre en compte la spatialité des variables. On veut que des variables voisines aient un effet similaire, c'est-à-dire

que leurs coefficients estimés aient une valeur proche. Le fused-lasso a d'abord été introduit pour encourager les coefficients associés à des variables consécutives d'un signal à être égaux puis a été généralisé à une structure de voisinage quelconque entre variables (Höfling, Binder et Schumacher 2010). De la même manière que pénaliser la valeur absolue des coefficients β_j encourage sa parcimonie, pénaliser la différence (en valeur absolue) de deux coefficients voisins va encourager l'égalité de ces deux coefficients.

On définit E une structure de voisinage contenant un ensemble d'arêtes. Si une arête (i,j) appartient E alors on dit que la variable d'indice i est voisine de celle d'indice j. L'estimateur du fused-lasso généralisé est :

$$\hat{\beta}^{\text{FL}}_{\lambda_1,\lambda_2,E} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2 \sum_{(i,j)\in E} |\beta_i - \beta_j| \right\} \tag{1.10}$$

avec $\lambda_1 \geq 0$ un paramètre de régularisation contrôlant la parcimonie des coefficients et $\lambda_2 \geq 0$ contrôlant l'égalité des coefficients voisins. Le cas classique du fused-lasso se retrouve en utilisant la structure de voisinage $E = \Big\{(i,i-1) \mid i = 2,\dots,p\Big\}$.

Le fused-lasso s'utilise quand le voisinage des données doit être pris en compte dans l'estimation. Optimiser deux paramètres peut être délicat, notamment dans la définition des paramètres à tester. On propose un moyen d'approximer le fused-lasso en annexe A afin de pouvoir utiliser l'algorithme Lars pour le résoudre.

Le fused-lasso est souvent utilisé dans le cas particulier où la matrice $X = I_n$ et avec un voisinage définit par les variables consécutives, ce qui revient à effectuer une segmentation d'un signal y quand $\lambda_1 = 0$. On nomme cette version le *fused-lasso signal approximator*. Dans ce cas, on dispose d'algorithmes efficaces pour sa résolution (Hoefling 2010).

La présence de deux paramètres de régularisation complexifie la tâche de leur optimisation.

Group-Lasso

Le group-lasso (Yuan et al. 2006) est l'équivalent du lasso pour la sélection de groupes de variables fournis a priori.

Soit $\mathcal{G} = \{G_1,\dots,G_K\}$ une partition en $K = \text{Card}(\mathcal{G})$ de l'ensemble des p variables. Pour $G_i \in \mathcal{G}$, on notera X_{G_i} (respectivement β_{G_i}) la matrice (respectivement le vecteur) restreinte aux variables du groupe G_i. Ainsi, on définit l'estimateur $\hat{\beta}^{\mathcal{G}}_{\lambda}$ du group-lasso par :

$$\hat{\beta}^{\mathcal{G}}_{\lambda} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{K} w_i \|\beta_{G_i}\|_2 \right\} \tag{1.11}$$

avec $w_i > 0$ un poids associé au groupe G_i. Généralement, on utilise $w_i = \sqrt{\text{Card}(G_i)}$. Ce poids permet de pénaliser plus fortement les grands groupes qui ont tendances à être sélectionnés en priorité.

Notons que si l'on restreint les groupes à une seule variable et que l'on pose $w_i = 1$ pour tout $i = 1,\dots,K$, on retrouve l'estimateur lasso.

La somme de pénalités ℓ_2 va encourager la parcimonie sur les groupes et non sur les variables isolées (cf. figure 1.1). Une somme de pénalités ℓ_1 ne peut aboutir à de la sélection de groupes mais à une sélection de variables (car $\sum_{i=1}^{K} \|\beta_{G_i}\|_1 = \|\beta\|_1$ comme les groupes $G_1,\dots,G_p$ forment une partition). Entre les groupes, la somme de pénalités ℓ_2 agit comme la pénalité ℓ_1 avec des

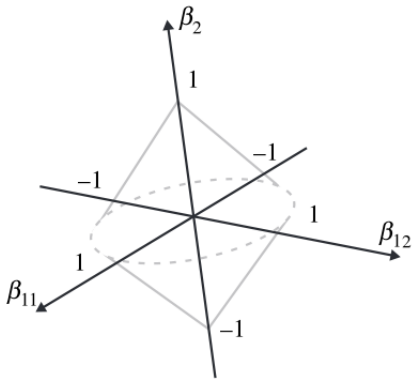


FIGURE 1.7 – Contour de la pénalité du group-lasso en dimension 3 pour un groupe de 2 éléments (β_{11} et β_{12}) et un contenant un seul élément (β_2). Figure issue de YUAN et al. (2006).

points de non-différentiabilité qui favorise la parcimonie au niveau des groupes. Au sein d'un groupe, la pénalité est l'équivalent d'un ridge et donc le group-lasso ne peut pas sélectionner un sous-ensemble des variables du groupe (cf. figure 1.7). Si en plus, on souhaite faire une sélection parcimonieuse au sein des groupes, on peut utiliser le *sparse group-lasso* (FRIEDMAN, HASTIE et Robert TIBSHIRANI 2010a) ou la pénalité bi-niveau de SZAFRANSKI, GRANDVALET et MORIZET-MAHOUDEAUX (2007).

Regrouper en amont les variables à l'aide d'une méthode basée sur la corrélation va permettre de faire de la sélection de groupes de variables corrélées, corrélations qui étaient problématiques dans le cadre du lasso.

Tout comme le lasso, la consistance de la sélection a été étudiée dans le cas du group-lasso (F. R. BACH 2008). Elle se base sur une version groupée de la *condition d'irreprésentabilité* du lasso (1.6). Notons $J = \left\{G_j \in \mathcal{G} \mid \beta^*_{G_j} \neq 0_{|G_j|}\right\}$ l'ensemble des vrais groupes.

$$\max_{i \in J^c} \frac{1}{w_i} \left\| X_{G_i}^T X_J (X_J^T X_J)^{-1} \text{Diag}\left(\frac{w_j}{\|\beta^*_{G_j}\|_2} \right) \beta^* \right\|_2 < 1 \tag{1.12}$$

avec $\text{Diag}\left(\frac{w_j}{\|\beta^*_{G_j}\|_2}\right)$ une matrice diagonale par blocs où le bloc associé au groupe G_j est de longueur $\text{Card}(G_j)$ et contient sur sa diagonale $\frac{w_j}{\|\beta^*_{G_j}\|_2}$.

Cette condition impose de faibles corrélations entre les variables du support et les autres. Lorsque tous les groupes sont de taille 1, elle est identique à la *condition d'irreprésentabilité* du lasso.

Sous cette condition et pour une certaine suite λ_n, le support estimé tend vers le vrai support J avec probabilité 1 lorsque n tend vers $+\infty$. Mais le nombre de variables p et de groupes K sont fixes. Cela en fait un théorème moins puissant que celui obtenu pour le lasso.

Le group-lasso se veut l'extension naturelle du lasso pour la sélection de groupes. Tout comme le lasso, il possède une propriété de consistance de la sélection. Cela en fait une méthode assez présente dans la littérature pour la sélection de groupes de variables corrélées (cf. Section

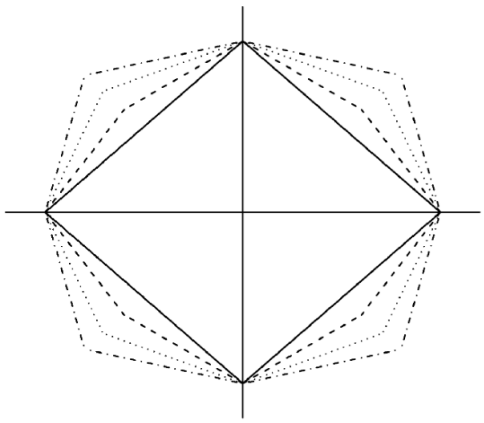


FIGURE 1.8 – Forme de la région de contrainte d'OSCAR pour différentes valeurs de c. Figure issue de BONDELL et REICH (2008).

1.2.2). Le group-lasso a été défini dans le cas où les groupes forment une partition des variables. Il est possible de l'utiliser dans un cas plus général où une même variable peut appartenir à plusieurs groupes. Nous présentons ce cas dans le chapitre 2.

***Octogonal Shrinkage and Clustering Algorithm for Regression* (OSCAR)**

OSCAR (BONDELL et REICH 2008) est une méthode de régression pénalisée dont le but est de sélectionner des variables tout en les regroupant. À l'inverse du group-lasso, OSCAR ne nécessite pas une partition a priori des variables en un nombre de groupes K donné. La pénalité d'OSCAR force les variables à avoir le même coefficient estimé mais sans prendre en compte a priori la structure de voisinage des données (ce qui est le cas du fused-lasso dont la pénalité force les variables voisines à avoir le même coefficient estimé).

L'estimateur est défini par :

$$\hat{\beta}^{\text{oscar}}_{\lambda,c} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\left(\|\beta\|_1 + c\sum_{j<k}\max\{|\beta_j|,|\beta_k|\}\right)\right\} \tag{1.13}$$

avec $c \geq 0$ et $\lambda \geq 0$. La norme ℓ_1 encourage la parcimonie et la norme ℓ_∞ par paires encourage l'égalité de coefficients. Poser $c = 0$ revient au lasso.

La forme de la pénalité en dimension 2 est octogonale, le paramètre c faisant varier les angles de l'octogone (figure 1.8).

Le pénalité OSCAR peut être réécrite comme une pénalité ℓ_1 ordonnée :

$$\hat{\beta}^{\text{oscar}}_{\lambda,c} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\sum_{j=1}^{p}(c(j-1)+1)|\beta|_{(j)}\right\} \tag{1.14}$$

avec $\lambda \geq 0$ et $|\beta|_{(1)} \leq |\beta|_{(2)} \leq \cdots \leq |\beta|_{(p)}$.

Les groupes sont déterminés a posteriori à l'aide des coefficients estimés. Les variables ayant un même coefficient estimés sont considérées comme appartenant au même groupe. Des variables

n'ayant aucune relation de corrélation peuvent donc se retrouver dans le même groupe. Plus la valeur de $c\lambda$ est grande, plus les coefficients associés à une paire de variables sont susceptibles d'être égaux. Bondell et Reich montrent que les coefficients associés à des variables corrélées sont plus à même d'être égaux pour une valeur plus faible de $c\lambda$ que des variables non corrélées. Cela ne peut donc se substituer à une méthode de regroupement de variables sur la base de corrélations. De plus, la présence de deux paramètres à optimiser et l'absence de méthodes efficaces pour en déterminer les valeurs en font une méthode peu pratique.

Intérêt du regroupement de variables en présence de corrélation

Nous allons monter l'intérêt du regroupement de variables en présence de corrélation en comparant les méthodes lasso et group-lasso dans un cadre où les variables sont corrélées entre elles.

Le cadre de simulation est le suivant :

— nombre d'individus $n = 100$;
— nombre de variables $p = 500$;
— $X_{1,.}, \dots, X_{n,.} \sim \mathcal{N}(0_p, \Sigma_\rho)$;
— Σ_ρ, une matrice diagonale par blocs de taille 5×5 où chaque bloc contient des 1 sur la diagonale et ρ partout ailleurs ;
— La partition $\mathcal{G}$ correspond aux différents blocs de variables ;
— $\epsilon \sim \mathcal{N}(0_n, \sigma^2 I_n)$;
— $y = X\beta^* + \epsilon$.

On appelle un groupe contenant au moins une variable appartenant au support de β^* (appelée aussi *vraie variable*) un *vrai groupe*. Une variable n'appartenant pas au support de β^* mais faisant partie d'un vrai groupe est appelé une *pseudo vrai variable*. On dit que c'est un *faux groupe* s'il ne contient que des variables en dehors du support de β^*.

On considère deux cas :

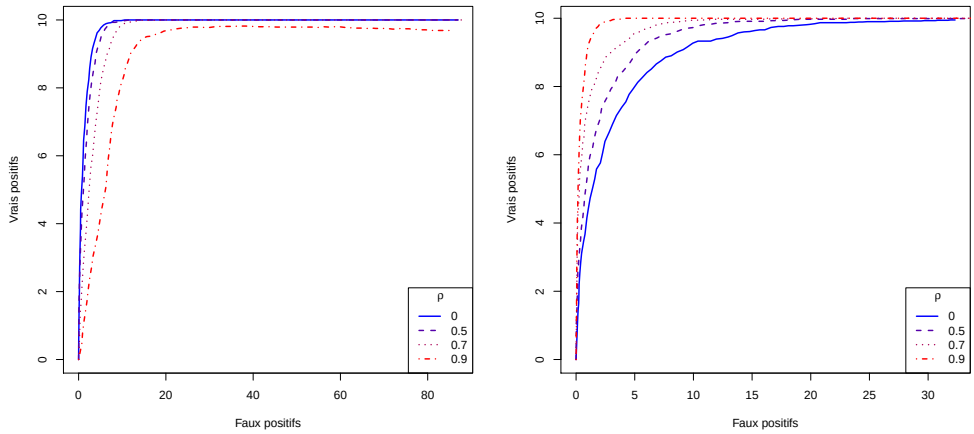
Cas 1 La taille du support S^* de β^* est $k = 10$. Les k vraies variables sont réparties dans des groupes séparés. On a donc une vraie variable par vrai groupe (qui sont donc au nombre de 10) ;

Cas 2 La taille du support S^* de β^* est $k = 50$. Il y a 10 vrais groupes de taille 5 contenant tous 5 vraies variables.

Rappelons que dans le théorème de Wainwright, dans le cas d'indépendance entre variables ($\Sigma_\rho = I_p$), le lasso pourra retrouver exactement les vraies variables pour $n = 2k\log(p-k)$ avec probabilité tendant vers 1 quand n augmente (cf. Section 1.1.1). Notons qu'ici, dans les deux cas, cette condition n'est pas vérifiée (on a $k\log(p-k) \simeq 62$ dans le cas 1 et $k\log(p-k) \simeq 305$ dans le cas 2). Nous nous plaçons donc dans un cas où le lasso sélectionnera certainement des fausses variables avant éventuellement de sélectionner la totalité des vraies variables.

Nous appliquons d'une part le lasso et d'autre part la group-lasso avec comme partition celle définie par la structure de corrélation de la matrice Σ_ρ. Pour le lasso, les faux et vrais positifs sont exprimés en matière de variables alors que pour le group-lasso, ils le sont en matière de groupes.

En présence de corrélations entre les variables, le lasso éprouve plus de difficultés pour retrouver l'ensemble des vraies variables (figure 1.9), mais il s'en tire relativement bien. Pour un même nombre de vraies variables sélectionnées, plus de fausses sont présentes quand la corrélation augmente. En ayant agrégées au préalable les variables en groupes (définis par les blocs de la matrice de variance-covariance), la qualité de la sélection du group-lasso s'améliore lorsque la corrélation augmente. Donc dans le cas où un groupe se restreint à une variable, le



(a) Qualité du chemin solution du lasso (Cas 1) (b) Qualité du chemin solution du group-lasso (Cas 1)

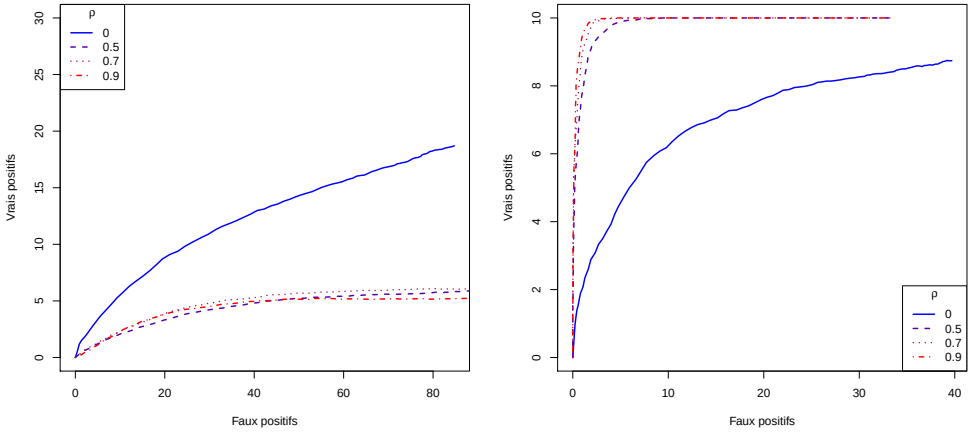
FIGURE 1.9 – Nombre de vraies variables (respectivement vrais groupes) sélectionné(e)s en fonction du nombre de fausses (respectivement faux) pour le lasso (respectivement group-lasso). ρ représente la corrélation au sein des groupes. Une courbe représente la moyenne de 100 chemins solution.

group-lasso devient avantageux par rapport au lasso en présence de corrélation. Pour le cas 2 (figure 1.10), le lasso ne peut retrouver l'ensemble des vraies variables même dans le cadre d'indépendance ($\rho = 0$). Le nombre d'individus énoncé dans la condition de WAINWRIGHT est loin d'être atteint. Le group-lasso garde une capacité de sélection importante en présence de corrélation avec un nombre élevé de vrais positifs et faible de faux positifs. La multiplicité des vraies variables au sein d'un groupe rendent le group-lasso beaucoup plus adapté à la sélection que le lasso.

En ayant choisi une (bonne) partition des variables, utiliser le group-lasso à la place du lasso donne donc de meilleurs résultats. Nous présentons dans la Section 1.2.1, les deux méthodes classiques de regroupement de variables. La qualité de la sélection dépend aussi fortement du paramètre de régularisation de la méthode utilisée. Dans la suite, nous présentons les méthodes les plus courantes pour optimiser le (ou les) paramètre(s) de régularisation des méthodes présentées.

Conclusion

Dans le cadre de variables corrélées, le group-lasso semble la méthode la plus adéquate pour faire de la sélection. En effet, les moindres carrés et la régression ridge ne permettent pas de faire de la sélection. Le fused-lasso ne peut quant à lui être utile qu'en présence de corrélation entre variables voisines. Les groupes sont définis a posteriori par les coefficients voisins de même valeur. De plus l'optimisation de deux paramètres de régularisation et l'absence d'algorithmes efficaces pour le résoudre dans le cas général limite son utilisation dans notre cas. OSCAR définit de la même manière les groupes mais sans la contrainte spatiale. Cependant, les corrélations ne sont pas explicitement prises en compte pour définir les groupes et il ne semble pas y avoir de méthodes faciles pour définir les valeurs des paramètres de régularisation à tester. Le group-lasso



(a) Qualité du chemin solution du lasso (Cas 2) (b) Qualité du chemin solution du group-lasso (Cas 2)

FIGURE 1.10 – Nombre de vraies variables (respectivement vrais groupes) sélectionné(e)s en fonction du nombre de fausses (respectivement faux) pour le lasso (respectivement group-lasso). ρ représente la corrélation au sein des groupes. Une courbe représente la moyenne de 100 chemins solution.

semble la méthode la plus adaptée pour notre problème. Les corrélations peuvent être prises en compte pour définir les groupes par une méthode de regroupement de variables.

1.1.4 Méthodes de choix du paramètre de régularisation

Dans le cadre de la régression pénalisée, le but des méthodes présentées ci-dessous est de choisir la meilleure valeur pour le (ou les) paramètre(s) de régularisation. On peut les classer en deux catégories : celles qui choisissent le paramètre

— dans un but de prédiction ;
— dans un but de sélection de variables.

Critères d'information

Les critères d'information sont des critères probabilistes utilisés pour faire de la sélection de modèle. Ils font un compromis entre qualité de l'ajustement et complexité du modèle.

Les critères d'information peuvent être écrit sous la forme générale :

$$GIC_\gamma(\beta) = -2\log(L(\beta)) + \gamma \mathrm{df}(\beta) \tag{1.15}$$

avec $L(\beta)$ la vraisemblance du modèle, $\mathrm{df}(\beta)$ le nombre de paramètres libres du modèle et $\gamma > 0$.

Pour un modèle de régression linéaire dans le cas gaussien, la formule est, à une constante près :

$$GIC_\gamma(\beta) = \frac{1}{\sigma^2}\|y - X\beta\|_2^2 + \gamma \mathrm{df}(\beta) \tag{1.16}$$

où σ^2 est la variance du bruit du modèle de régression linéaire.

Parmi ces critères, on note plus particulièrement :

Akaike Information Criterion (AIC) $\gamma = 2$ (Akaike 1974) ;

Bayesian Information Criterion (BIC) $\gamma = \log(n)$ (Schwarz 1978) ;

Risk Inflation Criterion (RIC) $\gamma = \log(p)$ (Foster et George 1994).

L'AIC et le BIC sont des critères asymptotiques, les utiliser pour de faibles tailles d'échantillons n'est pas forcément optimal. De plus, il a été montré que le BIC n'est pas consistant lorsque le nombre d'individus est petit par rapport au nombre de modèles possible (J. Chen et Z. Chen 2008). Dans le cadre non asymptotique, plusieurs corrections existent pour l'AIC (Hurvich et Tsai 1989).

Dans le cadre du lasso pour la sélection de variables, le biais de l'estimateur $\hat{\beta}_\lambda$ pose problème pour évaluer la qualité d'ajustement dans la formule du critère. De part le biais introduit par le lasso sur les coefficients estimés, la qualité d'ajustement des modèles parcimonieux est sous-évaluée et ces modèles sont donc sur-pénalisés. Les critères tendent alors à sélectionner des modèles moins parcimonieux. Une correction pour ce problème a été proposée dans Y. Zhang et Shen (2010). Elle consiste à remplacer le nombre de degrés de liberté du modèle par $\frac{2n}{\lambda}\|\hat{\beta}_\lambda\|_1$ dans une version corrigée du critère RIC.

Le nombre de paramètres libres (ou de degrés de liberté) $\mathrm{df}(\beta)$ est à estimer, ainsi que la variance du bruit du modèle de régression σ^2 quand celle-ci n'est pas connue (Giraud, Baraud et Huet 2007).

L'estimation du nombre de degrés de liberté a été étudiée dans la littérature pour les différentes méthodes présentées. Il est estimé par le nombre de coefficients non nuls de la solution pour le lasso (Efron et al. 2004), le nombre de blocs non nuls pour le fused-lasso (Robert Tibshirani, Saunders et al. 2005). Dans le cas d'OSCAR, les auteurs conjecturent que le nombre de degrés de liberté est le nombre de coefficients distincts (Bondell et Reich 2008). Dans le cas du group-lasso, une approximation est proposée dans Vaiter et al. (2012).

L'estimation de σ^2 ne pose pas de problème lorsque le nombre d'individus n est supérieur au nombre de variables p. La variance du bruit peut être estimée par l'estimateur classique $\|y - X\hat{\beta}^{LS}\|_2^2/(n-p-1)$ où $\hat{\beta}^{LS}$ est l'estimateur des moindres carrés de y sur X. Dans les cas de grande dimension, obtenir un estimateur consistant de σ^2 est problématique (Yongdai, Sunghoon et Hosik 2012 ; Lockhart et al. 2014). Une procédure usuelle se décompose en deux étapes : obtenir un estimateur parcimonieux de β^* puis estimer σ^2 par l'intermédiaire des moindres carrés sur le support estimé (Fan, Guo et Hao 2012). Cette procédure nécessite que l'estimateur de β obtenu vérifie la propriété de *sure screening*, c'est-à-dire que l'ensemble des variables du support de β^* soient sélectionnées. Cette procédure a tendance à sous-estimer σ^2 (Fan, Guo et Hao 2012).

Validation croisée *V-fold*

Les méthodes de validations croisées consistent à séparer les données en deux échantillons : un échantillon d'apprentissage qui servira à estimer les paramètres du modèle, et un échantillon test à tester le modèle estimé. Le choix du paramètre se fait sur la base d'un critère calculé sur l'échantillon test.

Parmi les méthodes de validations croisées (Arlot et Celisse 2010), la plus courante est la validation croisée *V-fold*. Elle consiste à séparer les données en V blocs disjoints, à apprendre le modèle sur $V-1$ blocs, puis à tester sur le bloc restant et à itérer pour que chaque bloc serve d'échantillon test. Nous la présentons dans le cas où un paramètre est à optimiser (cf. Algorithme 1), le principe ne change pas dans le cas où plusieurs sont à optimiser simultanément.

Algorithme 1 validation croisée *V-fold*

Soit Λ un ensemble de valeurs de paramètres à tester.
Diviser l'ensemble des individus $\mathcal{I} = \{1, \dots, n\}$ en K sous-ensembles disjoints $\mathcal{I}_v$, $v = 1, \dots, V$ de taille équivalente.
Pour $v = 1, .., V$ **Faire**
Appliquer la méthode de régression pénalisée pour les individus $\mathcal{I} \setminus \mathcal{I}_v$ pour $\lambda \in \Lambda$.
Calculer le critère $M^{(v)}(\lambda)$ pour les individus $\mathcal{I}_v$.
Fin Pour
Pour chaque valeur de λ, calculer le critère moyen

$$M(\lambda) = \frac{1}{V} \sum_{k=1}^{K} M^{(v)}(\lambda).$$

Choisir $\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}}\, M(\lambda)$.

Le critère $M^{(v)}(\lambda)$ classiquement utilisé est le calcul de l'erreur de prédiction à partir de l'estimateur $\hat{\beta}_\lambda^{(v)}$ fourni par la méthode de régularisation :

$$M^{(v)}(\lambda) = \sum_{i \in \mathcal{I}_v} (y_i - X_{i,\cdot} \hat{\beta}_\lambda^{(v)})^2$$

où $X_{i,\cdot}$ représente la i^e ligne de la matrice X.

La validation croisée *V-fold* est utilisée avec un objectif de prédiction et non de sélection. En particulier, il a été montré pour le lasso que la validation croisée *V-fold* basée sur l'erreur de prédiction ne permet pas de choisir les variables de manière efficace et a tendance à trop en sélectionner (Leng, Lin et Wahba 2006).

La validation croisée *V-fold* pose des problèmes en temps de calcul. Appliquer plusieurs fois la méthode de résolution peut s'avérer algorithmiquement coûteux en grande dimension. Cela dépend de la valeur de V qui est souvent fixée à 5 ou 10 en pratique.

Kappa

Les méthodes précédentes sont basées sur l'optimisation de l'erreur de prédiction pour choisir le paramètre de régularisation. Elles visent à optimiser l'erreur de prédiction et non à sélectionner des variables.

Le critère Kappa se base sur le κ de Cohen (Cohen 1960) qui mesure l'accord entre deux classifieurs répartissant des variables en deux groupes.

Soient $\mathcal{A}_1$ et $\mathcal{A}_2$ (respectivement $\mathcal{B}_1$ et $\mathcal{B}_2$) la répartition en deux groupes des p variables par la méthode $\mathcal{A}$ (respectivement $\mathcal{B}$). On définit les quantités suivantes :

	$\mathcal{B}_1$	$\mathcal{B}_2$
$\mathcal{A}_1$	n_{11}	n_{12}
$\mathcal{A}_2$	n_{21}	n_{22}

où n_{11} est le nombre de variables communes à $\mathcal{A}_1$ et $\mathcal{B}_1$, ...

On définit le critère κ pour quantifier la ressemblance entre les deux répartitions :

$$\kappa(\mathcal{A}, \mathcal{B}) = \frac{\mathbb{P}(a) - \mathbb{P}(e)}{1 - \mathbb{P}(e)}$$

avec $\mathbb{P}(a) = (n_{11} + n_{22})/p$ la similitude entre $\mathcal{A}$ et $\mathcal{B}$ et $\mathbb{P}(e) = (n_{11} + n_{12})(n_{11} + n_{21})/p^2 + (n_{22} + n_{21})(n_{22} + n_{12})/p^2$ la probabilité d'être en accord par hasard. Le κ de Cohen varie entre -1 (désaccord total entre les classifieurs) et 1 (accord total).

Dans la suite, on appelle *variable active*, une variable dont le coefficient estimé est non nul. Le critère κ peut être utilisé pour quantifier la stabilité de sélection de méthodes de régression pénalisée (Sun, Junhui Wang et Fang 2013). En appliquant une méthode de régression pénalisée sur deux sous-échantillons des données, la stabilité de la sélection est définie pour chaque valeur de λ comme la valeur du κ de Cohen entre les deux répartitions en variables actives et non-actives obtenues. L'idée est que les variables d'importances doivent se retrouver dans les deux ensembles de variables actives. La présence des fausses variables dans l'ensemble actif se voudra quant à elle plus dépendantes des sous-échantillons. Une faible valeur du critère de stabilité dénote alors une trop forte présence de fausses variables dans l'ensemble des variables actives.

Soient $\mathcal{I}_1$ et $\mathcal{I}_2$ deux ensembles d'individus indépendants et identiquement distribués (i.i.d.). On note $\Psi(\mathcal{I}_1, \lambda)$ la règle de répartition en variables actives et non actives obtenue par la méthode de régression pénalisée sur la matrice X restreinte aux individus de l'échantillon $\mathcal{I}_1$ pour λ. La stabilité de la sélection pour une valeur de λ donnée est définie par :

$$s(\lambda) = \mathbb{E}\left[\kappa(\Psi(\mathcal{I}_1, \lambda), \Psi(\mathcal{I}_2, \lambda))\right].$$

Cette quantité va être estimée à l'aide d'une méthode de type *bootstrap* (cf. Algorithme 2).

Algorithme 2 Kappa

Soit Λ un ensemble de valeurs de paramètre à tester.
Pour $b = 1, .., B$ **Faire**
Répartir aléatoirement les n individus en deux sous-ensembles $\mathcal{I}_1^b$ et $\mathcal{I}_2^b$ de taille $n/2$.
Lancer la méthode de régression pénalisée sur X restreinte aux individus $\mathcal{I}_1^b$ puis restreinte à $\mathcal{I}_2^b$. On obtient ainsi $\Psi(\mathcal{I}_1^b, \lambda)$ et $\Psi(\mathcal{I}_2^b, \lambda)$.
Calculer pour chaque $\lambda \in \Lambda$, $\kappa\left(\Psi(\mathcal{I}_1^b, \lambda), \Psi(\mathcal{I}_2^b, \lambda)\right)$.
Fin Pour
Calculer l'estimateur de la stabilité

$$\hat{s}(\lambda) = \frac{1}{B} \sum_{b=1}^{B} \kappa\left(\Psi(\mathcal{I}_1^b, \lambda), \Psi(\mathcal{I}_2^b, \lambda)\right)$$

Choisir $\hat{\lambda}$:

$$\hat{\lambda} = \min\left\{ \lambda \;\middle|\; \frac{\hat{s}(\lambda)}{\max_{\lambda'} \hat{s}(\lambda')} \geq 1 - \alpha \right\}$$

La dernière étape de l'Algorithme 2 est mise en place afin de ne pas obtenir un λ trop grand qui pourrait faire manquer des variables moyennement informatives et sous-estimer le modèle.

En pratique, un seuil $\alpha = 0.1$ est conseillé.

L'exécution multiple de la méthode de résolution rend la méthode Kappa coûteuse en temps de calcul. Le point intéressant est qu'elle se base sur la stabilité de la sélection en matière de variables actives et non-actives et non l'erreur de prédiction, la rendant potentiellement plus à même de trouver un λ optimal pour l'estimation du vrai support.

Stability selection

La *stability selection* (Meinshausen et Bühlmann 2010) ne cherche pas à estimer le paramètre de régularisation optimal mais la probabilité de sélection des variables. Elle n'est donc pas directement comparable aux méthodes présentées ci-dessus mais nous la présentons car elle permet de sélectionner les variables d'intérêts.

La probabilité de sélection est estimée à l'aide de B exécutions de la méthode de régression pénalisée sur différents échantillons des données (cf. Algorithme 3). L'idée est la même que pour le critère Kappa à la différence qu'elle se concentre sur les variables individuellement et non un ensemble. Les vraies variables vont se retrouver sélectionnées régulièrement pour les divers échantillons alors qu'une fausse variable sera sélectionnée plus épisodiquement car plus sensibles au rééchantillonnage qu'une vraie variable.

Algorithme 3 *Stability selection*

Soit Λ un ensemble de valeurs de paramètre à tester.
Pour $b = 1,..,B$ **Faire**
Tirer sans remise $\mathcal{I}_b$ un échantillon de taille $\frac{n}{2}$ de $\mathcal{I} = \{1,\dots,n\}$.
Appliquer la méthode de régression pénalisée pour les individus $\mathcal{I}_b$ pour $\lambda \in \Lambda$.
Pour chaque variable j et pour $\lambda \in \Lambda$, on pose $M_k^\lambda(b) = \mathbb{1}(k \in \hat{S}_\lambda^{(b)})$ où $\hat{S}_\lambda^{(b)}$ est le support de l'estimateur obtenu pour λ et l'échantillon $\mathcal{I}_b$.
Fin Pour
Pour $\lambda \in \Lambda$, calculer la probabilité de sélection estimée

$$\hat{\Pi}_j^\lambda = \frac{1}{B}\sum_{b=1}^{B} M_j^\lambda(b).$$

On définit l'ensemble des variables stables comme celles dont la probabilité de sélection estimée dépasse un seuil π_{thr} fixé

$$\hat{S}^{\text{stable}} = \{j \mid \max_{\lambda\in\Lambda} \hat{\Pi}_j^\lambda \geq \pi_{\text{thr}}\}.$$

On présente le résultat sous la forme d'un *stability path* (figure 1.11) représentant la probabilité de sélection en fonction de λ. Chaque courbe est associée à une variable distincte.

L'avantage de cette méthode est qu'elle se base sur la sélection et non l'estimation. Par contre le rééchantillonnage est coûteux en temps de calcul (en pratique $B = 100$).

Comparaison

Les méthodes présentées ci-dessus sont comparées dans le cas du lasso.
Le cadre de simulation est le suivant :

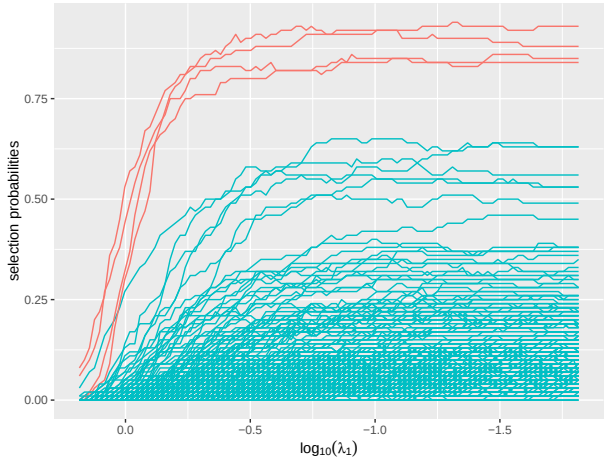


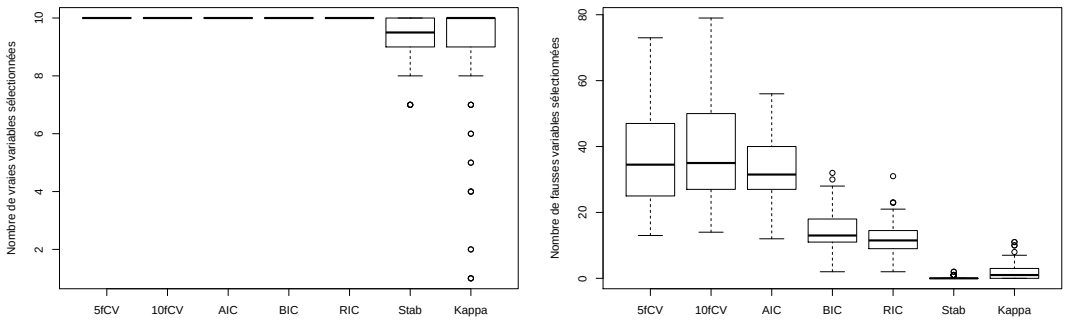
FIGURE 1.11 – Exemple de *stability path* pour le lasso obtenu à l'aide du package R quadrupen (GRANDVALET, CHIQUET et AMBROISE 2012). Probabilité de sélection d'une variable en fonction de λ. Chaque courbe est associée à une variable. Seules les variables dont la probabilité dépassent un certain seuil sont conservées (seuil de 0.75 ici).

— nombre d'individus $n = 100$;
— nombre de variables $p = 250$;
— taille du support S^* de β^* $k = 10$;
— $\beta^*_{S^*} = 0.5$;
— $X_{1,.}, \ldots, X_{n,.} \sim \mathcal{N}(0_p, I_p)$;
— $\epsilon \sim \mathcal{N}(0_n, \sigma^2 I_n)$ avec $\sigma = 0.5$;
— $y = X\beta^* + \epsilon$.

Rappelons que dans le théorème de WAINWRIGHT, dans le cas d'indépendance entre variables ($\Sigma_\rho = I_p$), le lasso pourra retrouver exactement les vraies variables pour $n = 2k\log(p-k)$ avec probabilité tendant vers 1 quand n augmente (cf. Section 1.1.1). Notons qu'ici, dans les deux cas, cette condition n'est pas vérifiée (on a $k\log(p-k) \simeq 55$ pour $p = 250$). Nous nous plaçons donc dans un cas où le lasso sélectionnera certainement des fausses variables avant éventuellement de sélectionner la totalité des vraies variables.

Les méthodes suivantes sont comparées : validation croisée *5-fold* (5fCV), validation croisée *10-fold* (10fCV), AIC, BIC, RIC, la *stability selection* avec un seuil $\pi_{\text{thr}} = 0.75$ et $B = 100$ rééchantillonnages (Stab) et le critère Kappa avec $\alpha = 0.1$ et $B = 20$ rééchantillonnages. Pour les critères d'information, la vraie valeur de la variance du bruit est utilisée.

Les résultats sont visibles sur la figure 1.12. Dans la totalité des cas, l'ensemble des variables du support de β^* sont retrouvées par les critères d'information et la validation croisée *V-fold*. Mais, un nombre important de fausses variables est sélectionné. Parmi les critères d'information, le RIC présente de meilleurs résultats, c'est celui qui pénalise le plus fortement la complexité du modèle. L'ensemble des critères d'information sélectionne la totalité des vraies variables mais le RIC sélectionne moins de fausses variables mais ce nombre reste élevé (une dizaine en moyenne). En matière de compromis entre le nombre de fausses et vraies variables sélectionnées, la *stability selection* et le critère Kappa sont les plus efficaces. Bien que le critère Kappa ne sélectionne pas toujours la totalité des vraies variables, un nombre restreint de fausses variables est sélectionné. On constate donc que les méthodes basées sur l'erreur de prédiction ne sont pas



(a) Nombre de vraies variables sélectionnées (b) Nombre de fausses variables sélectionnées

FIGURE 1.12 – Variables sélectionnées par les différentes méthodes pour le lasso pour $n = 250$ sur 100 répétitions.

efficaces pour choisir une valeur de λ adaptée à la sélection de variables. Les méthodes voulant estimer la stabilité sont néanmoins très coûteuse car elles nécessitent un nombre important de rééchantillonnages (au minimum 50), ce qui est problématique lorsque l'algorithme de résolution est coûteux en temps de calcul.

1.2 Regrouper les variables pour la sélection

1.2.1 Regroupement de variables non supervisé

Le group-lasso nécessite de fournir une partition des variables en un certain nombre de groupes. Nous allons présenter deux grandes familles de méthodes de classification de variables non supervisée : les algorithmes de partitionnement et la classification hiérarchique (voir (JAIN, MURTY et FLYNN 1999) pour une revue à ce sujet).

Algorithme de partitionnement

Les algorithmes de partitionnement fournissent une unique partition des variables en un nombre de groupes défini a priori. Chaque variable appartient à un unique groupe (dans le cas contraire, on parle de *fuzzy clustering*).

Le plus utilisé d'entre eux est l'algorithme des *k-means* (Algorithme 4). Il minimise la distance de chaque variable à son centre de groupe :

$$\sum_{i=1}^{K} \sum_{j \in G_i} \|X_j - g_i\|_2^2 \tag{1.17}$$

où $g_1, \ldots, g_K$ sont les centres respectifs des groupes $G_1, \ldots, G_K$.

Cette grandeur est appelée l'inertie intra-groupe et se note I_W. Plus cette grandeur est faible, plus les groupes obtenus sont séparés. L'inertie est une grandeur courante en classification. Elle

décrit la dispersion d'un nuage de points $(X_1,\dots,X_p)$ de $\mathbb{R}^n$ par rapport à son centre :

$$I = \sum_{i=1}^{p} \|X_i - g\|_2^2 \tag{1.18}$$

où g est le centre de gravité du nuage.

Lorsque les variables sont classées en différents groupes $G_1,\dots,G_K$ de centres respectifs $g_1,\dots,g_K$, l'inertie se décompose comme la somme de l'inertie inter-groupe (I_B) et de l'inertie intra-groupe (I_W) :

$$I = \underbrace{\sum_{k=1}^{K} \text{Card}(G_k)\, \|g_k - g\|_2^2}_{I_B} + \underbrace{\sum_{k=1}^{K} \sum_{j\in G_k} \|X_j - g_k\|_2^2}_{I_W}. \tag{1.19}$$

Plus l'inertie inter-groupe I_B est grande, plus l'inter intra-groupe I_W est faible et plus les groupes seront des groupes homogènes et bien séparés.

L'algorithme est assez sensible à son initialisation, sa convergence vers un minimum global n'est ainsi pas assurée. L'algorithme a une complexité de $O(iknp)$ où i est le nombre d'itérations et converge en pratique en un faible nombre d'itérations. Il est exécuté à plusieurs reprises et la partition minimisant le critère (1.17) est gardée.

Algorithme 4 *k-means*

Choisir aléatoirement K centres de groupes $g_1,\dots,g_K$ parmi l'ensemble des variables.
Répéter
  Calculer la distance de chaque variable aux différents centres :
    $\text{d}(X_j, g_i) = \|X_j - g_i\|_2^2$, $j = 1,\dots,p$, $i = 1,\dots,K$.
  Assigner chaque variable au groupe dont le centre est le plus proche :
    $G_i = \{j \,|\, \underset{i=1,\dots,K}{\text{argmin}}\ \text{d}(X_j, g_i)\}$.
  Recalculer les centres des groupes
    $g_i = \frac{1}{\text{Card}(G_i)} \sum_{j\in G_i} X_j$.
Jusqu'à convergence de la partition ou un nombre maximal d'itérations.
Retourne la partition estimée $G_1,\dots,G_K$.

Le choix du nombre de groupes est une étape cruciale dans les algorithmes de classification. On peut se référer à (Milligan et Cooper 1985 ; Halkidi, Batistakis et Vazirgiannis 2002) pour une revue des méthodes de choix du nombre de groupes. Ces méthodes se basent principalement sur un compromis entre une mesure inter-groupes et intra-groupes. La mesure intra-groupes quantifie la compacité des groupes tandis que la la mesure inter-groupes quantifie la séparation entre les groupes.

Classification Hiérarchique

Les algorithmes hiérarchiques fournissent un ensemble de stratégies de partitionnement. Plus précisément, l'algorithme fournit p partitions des p variables en un nombre de groupes allant de 1 à p. On note $\mathcal{G}_s$, la partition en s groupes obtenue par l'algorithme de classification

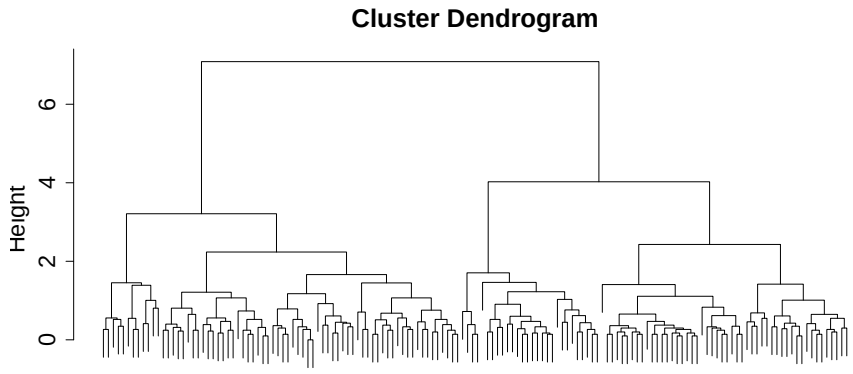


FIGURE 1.13 – Dendrogramme obtenu après une classification ascendante hiérarchique avec distance euclidienne et average linkage sur les données iris (FISHER 1936).

hiérarchique, on parle également de partition du niveau s de la hiérarchie. Chaque groupe d'une partition à un niveau donné est la réunion de groupes du niveau précédent. Le résultat d'une classification hiérarchique est représenté par un dendrogramme (figure 1.13).

Il existe des algorithmes dit ascendants ou descendants. Dans le cas ascendant, les variables sont placées dans des groupes séparés puis à chaque étape deux groupes fusionnent, les itérations sont répétées jusqu'à former un seul groupe. Dans le cas descendant, le phénomène inverse se produit. Nous détaillons dans l'Algorithme 5 le cas ascendant qui est le plus utilisé.

Algorithme 5 Classification Ascendante Hiérarchique (CAH)

Chaque variable est placée dans un groupe.
Calculer la dissimilarité entre chaque paire de variables.
Répéter
  Calculer un critère d'agrégation entre chaque paire de groupes.
  Agréger les 2 groupes les plus proches au sens du critère.
Jusqu'à ce que l'ensemble des variables forme un unique groupe.
Retourne Un ensemble de partition $\{\mathcal{G}_1, \ldots, \mathcal{G}_p\}$.

Contrairement aux *k-means*, l'algorithme hiérarchique est stable, i.e. des exécutions multiples donnent toujours la même solution. En matière de complexité algorithmique, la Classification Ascendante Hiérarchique (CAH) a un coût de $O(p^3)$ en utilisant une méthode de résolution naïve. La complexité est en général de $O(p^2 \log(p))$ mais elle dépend du choix du critère d'agrégation. Dans les cas des critères *single linkage* et du *complete linkage* (voir ci-dessous), la complexité est en $O(p^2)$. Dans notre cas ($p \gg n$), elle reste supérieur à celle des *k-means*. Le coût de stockage est également important avec le stockage d'une matrice de taille $p \times p$. Ces deux aspects sont problématiques quand la dimension p augmente fortement. La classification hiérarchique reste cependant très utilisée notamment pour l'imbrication des partitions qui fournit un lien entre celles-ci. Cela permet d'interpréter plus facilement le passage d'une partition à une autre.

Dissimilarité

Une dissimilarité est une application $\mathrm{d} : \Omega \times \Omega \to \mathbb{R}$ vérifiant notamment les conditions de symétrie, de positivité.

Symétrie : $\mathrm{d}(i,j) = \mathrm{d}(j,i)$;

Positivité : $\mathrm{d}(i,j) \geq 0$;

$\mathrm{d}(i,i) = 0$.

Toutes les distances sont des dissimilarités. En effet, une distance est une dissimilarité vérifiant en plus la propriété de séparation et l'inégalité triangulaire.

Généralement, la distance euclidienne est utilisée comme dissimilarité ou une dissimilarité basée sur les corrélations entre variables, par exemple $\mathrm{d}(X_i, X_j) = 1 - |\mathrm{cor}(X_i, X_j)|$ où $\mathrm{cor}(X_i, X_j)$ est le coefficient de corrélation entre les variables X_i et X_j.

Critères d'agrégation

Les critères d'agrégation mesurent la dissimilarité entre deux groupes de variables. Les plus courants sont :

Single linkage La dissimilarité entre deux groupes est mesurée par la dissimilarité minimum entre les variables des groupes :

$$\mathrm{d}(G_i, G_j) = \min_{x \in G_i, y \in G_j} \mathrm{d}(x,y)$$

Cette dissimilarité engendre parfois un effet de chaînage, les variables s'agrègent une à une à une en un même groupe.

Complete Linkage À l'inverse du single linkage, la dissimilarité maximum est prise comme dissimilarité entre les groupes :

$$\mathrm{d}(G_i, G_j) = \max_{x \in G_i, y \in G_j} \mathrm{d}(x,y).$$

Average linkage Elle correspond à la dissimilarité moyenne entre les paires de points des deux groupes :

$$\mathrm{d}(G_i, G_j) = \frac{1}{\mathrm{Card}(G_i) \times \mathrm{Card}(G_j)} \sum_{x \in G_i, y \in G_j} \mathrm{d}(x,y).$$

Méthode des centroïdes La dissimilarité entre les groupes est représentée par la dissimilarité entre leur centres :

$$\mathrm{d}(G_i, G_j) = \mathrm{d}(g_i, g_j)$$

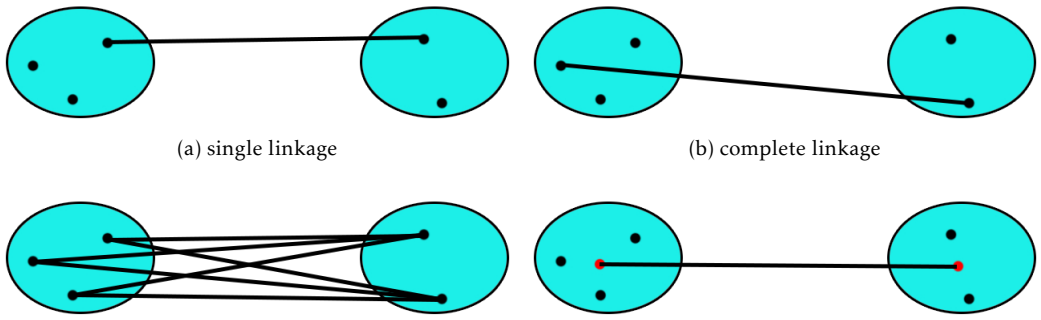
avec g_i (respectivement g_j) le centre de gravité du groupe G_i (respectivement G_j).

Méthode de Ward (Ward 1963) Elle permet de maximiser le gain d'inertie inter-groupes à chaque niveau de la CAH :

$$\mathrm{d}(G_i, G_j) = \frac{\mathrm{Card}(G_i) \times \mathrm{Card}(G_j)}{\mathrm{Card}(G_i) + \mathrm{Card}(G_j)} \mathrm{d}(g_i, g_j).$$

Les valeurs de ces critères sont représentées en ordonnée du dendrogramme (figure 1.13).

Pour le choix du niveau optimal de la CAH, on peut se référer au même critère que pour les *k-means* (Milligan et Cooper 1985 ; Halkidi, Batistakis et Vazirgiannis 2002). On notera également la règle basique du saut maximal. À chaque niveau s de la CAH est associé une partition $\mathcal{G}_s$ en s groupes et la valeur de dissimilarité des deux groupes joints h_s. La règle du saut maximal s'intéresse à la différence entre deux niveaux successifs : $l_s = h_{s-1} - h_s$. Une forte



(a) single linkage (b) complete linkage

(c) average linkage (d) centroïdes

FIGURE 1.14 – Illustration de différents critères d'agrégation pour la classification ascendante hiérarchique entre un groupe de 3 variables et un de deux variables. Les points rouges représentent les centres des groupes et les traits noirs, les grandeurs considérées dans le calcul du critère considéré.

augmentation implique une fusion de deux groupes éloignés au sens de la dissimilarité. Le niveau $\hat{s}$ optimal pour ce critère est donc

$$\hat{s} = \underset{s}{\operatorname{argmax}}\{h_{s-1} - h_s\}. \tag{1.20}$$

Dans le cas des données iris (figure 1.15), la valeur du saut maximal est l_2, deux groupes sont ainsi choisis.

1.2.2 Regroupement de variables et sélection de groupes

Nous avons vu précédemment l'intérêt d'utiliser le group-lasso par rapport au lasso en présence de dépendances entre variables. Le group-lasso nécessite une partition des variables a priori qui peut être déterminée par le biais d'une méthode de regroupement de variables. Nous présentons les méthodes de la littérature juxtaposant une méthode de classification et une méthode de régression pénalisée dans le cadre de dépendance entre variables.

Hierarchical Clustering and Averaging for Regression

La méthode a été proposée dans (PARK, HASTIE et Robert TIBSHIRANI 2007) et utilise classification ascendante hiérarchique et lasso. Elle se place dans le contexte de la sélection de variables en présence de corrélation. L'idée est, pour chaque partition de la CAH, de représenter chaque groupe par une méta-variable qui correspond au vecteur moyen des variables du groupe. Ensuite, pour chaque niveau de la CAH, un lasso est appliqué sur les méta-variables de la partition associée. Une fois, cette procédure appliquée à chaque niveau, une procédure de validation croisée permet de choisir le niveau de la CAH ainsi que le paramètre de régularisation λ (cf. algorithme 6).

Utiliser la moyenne des variables d'un groupe pour le représenter consiste à imposer une contrainte d'égalité entre les coefficients estimés qui leur sont associées. Cela dans le but de réduire la variance de l'estimateur.

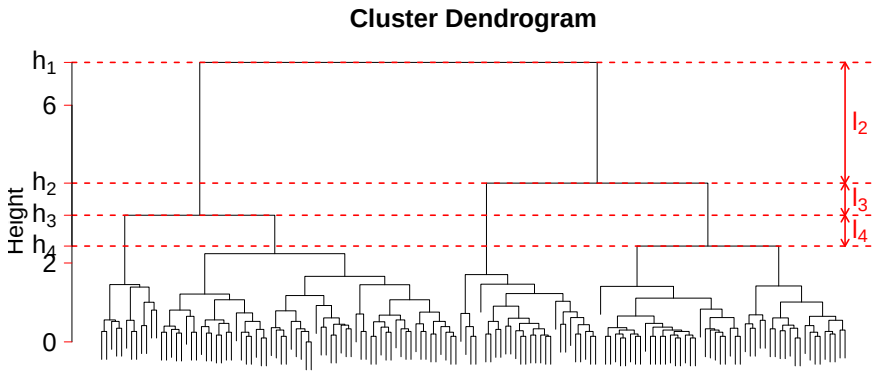


FIGURE 1.15 – Dendrogramme obtenu après une classification ascendante hiérarchique avec distance euclidienne et average linkage sur les données iris (FISHER 1936). h_s correspond à la valeur de la dissimilarité des deux groupes fusionnés pour former la partition du niveau s. l_s représente le saut effectué entre les niveaux s et $s-1$.

Algorithme 6 *Hierarchical Clustering and Averaging for Regression*

Appliquer une CAH sur l'ensemble des gènes.
Pour le niveau de la hiérarchie $s = 1,\dots,p$ **Faire**
Calculer le *supergène* $\tilde{X}_i^{(s)}$ en moyennant les variables de chaque groupe $G_i^{(s)}$:

$$\tilde{X}_i^{(s)} = \frac{1}{\text{Card}(G_i^{(s)})} \sum_{j \in G_i^{(s)}} X_j.$$

Appliquer un lasso sur la matrice des *supergènes* $\tilde{X}^{(s)}$.
Fin Pour
Appliquer une validation croisée pour choisir le niveau de la CAH et le paramètre de régularisation λ du lasso.
Retourne le paramètre de régularisation estimé $\hat{\lambda}$, le niveau de la hiérarchie estimée $\hat{s}$ et les groupes sélectionnés.

Supervised Group-Lasso

Le *Supervised Group-Lasso* (Ma, Song et Huang 2007) est une méthode en trois étapes : regroupement de variables, réduction de la dimension et sélection de groupes (cf. algorithme 7). Elle se différencie de la méthode précédente par le choix de la partition avant la phase de sélection. Comme précédemment, une étape de réduction de la dimension est présente par le biais d'un lasso sur chaque groupe de la partition choisie. Enfin la sélection des groupes est effectuée par un group-lasso.

Algorithme 7 *Supervised Group-Lasso* (SGL)

1. Classification
Appliquer les *k-means* pour un nombre de groupes allant de 1 à M.
Choisir le nombre de groupes m maximisant la *Gap Statistic* (Robert Tibshirani, Hastie et al. 1999).
Retourne une partition $\mathcal{G} = \{G_1, \dots, G_m\}$ en m groupes.

2. Réduction de la dimension
Pour $i = 1, \dots, m$ **Faire**
Appliquer un lasso sur les données restreintes au groupe G_i : X_{G_i}.
Appliquer une validation croisée *V-fold* pour choisir le paramètre du lasso.
Former $\tilde{G}_i$ le groupe contenant les variables de G_i dont les coefficients estimés sont non nuls.
Fin Pour
Retourne un ensemble de groupe $\tilde{\mathcal{G}} = \{\tilde{G}_1, \dots, \tilde{G}_m\}$.

3. Sélection de groupes
Appliquer un group-lasso sur $X_{\tilde{\mathcal{G}}}$ avec $\tilde{\mathcal{G}}$ comme partition.
Appliquer une validation croisée *V-fold* pour choisir le paramètre du group-lasso.
Retourne le paramètre de régularisation estimé $\hat{\lambda}$, les groupes sélectionnés.

Les auteurs privilégient les *k-means* à la classification ascendante hiérarchique pour des raisons de complexité algorithmique. Le choix de la *gap statistic* peut être discuté car elle nécessite d'appliquer les *k-means* sur un nombre important d'échantillons *bootstrap*, ce qui est coûteux en temps de calcul.

Cluster Representative Lasso & Cluster Group-Lasso

Les deux méthodes présentées dans (Bühlmann, Rütimann et al. 2013) se décomposent d'une manière similaire à la précédente : une étape de regroupement de variables et une de sélection.

La différence notable par rapport aux approches précédentes est la partie classification. Les auteurs proposent d'utiliser une distance basée sur les corrélations canoniques (Anderson 1984) au sein d'une classification ascendante hiérarchique. À chaque étape, les deux groupes ayant la plus forte corrélation canonique sont regroupés. La partition retournée est la première dont le coefficient de corrélation canonique maximal entre deux groupes passe sous un seuil τ fixé par l'utilisateur.

Les corrélations canoniques étudient la relation entre deux groupes de variables. Chaque groupe est représenté par une combinaison linéaire de ses variables puis la corrélation entre ces deux nouvelles variables est calculée. Le but étant de trouver les représentants maximisant la corrélation. Le coefficient de corrélation canonique est le coefficient maximal obtenu.

Le *standardized group-lasso* (Bühlmann et Geer 2011 ; Simon et Robert Tibshirani 2011) utilise la pénalité $\sum_{i=1}^{K} w_i \|X_{G_i}\beta_{G_i}\|_2$ au lieu de la pénalité classique du group-lasso : $\sum_{i=1}^{K} w_i \|\beta_{G_i}\|_2$. Ce choix est justifié par un meilleur comportement en présence de corrélation au sein des groupes, une forte corrélation au sein d'un vrai groupe va faciliter sa sélection par rapport au group-lasso classique. Cette pénalité est équivalente à faire un group-lasso classique avec X_{G_i} orthonormal pour tout $G_i \in \mathcal{G}$. On notera que l'utilisation du *standardized group-lasso* ajoute une restriction sur la taille des groupes (que n'a pas le group-lasso classique) : celle-ci doit être inférieure au nombre d'individus. Cette restriction peut être problématique dans les cas de très grande dimension.

Algorithme 8 *Cluster Representative Lasso* & Cluster Group-Lasso

1. Classification
Méthode 1 Appliquer une classification ascendante hiérarchique avec une dissimilarité basée sur les corrélations : $\mathrm{d}(X_i, X_j) = 1 - |\mathrm{cor}(X_i, X_j)|$, et l'*average linkage* comme critère de regroupement. Choisir la partition via la règle du saut maximal.
Méthode 2 Appliquer une classification ascendante hiérarchique avec une dissimilarité basée sur les corrélations canoniques.
Retourne une partition $\mathcal{G} = \{G_1, \ldots, G_m\}$ en m groupes.

2. Sélection de groupes
Cluster Group-Lasso Appliquer un *standardized group-lasso* sur X avec la partition $\mathcal{G}$.
Cluster Representative Lasso Appliquer un lasso sur les *supergènes* de $\mathcal{G}$ (cf. algorithme 6).
Appliquer une validation croisée *V-fold* pour choisir le paramètre de régularisation.
Retourne le paramètre de régularisation estimé $\hat{\lambda}$, les groupes sélectionnés.

Cluster Elastic-Net

À l'inverse des méthodes présentées ci-dessus, le Cluster Elastic-Net (Witten, Shojaie et F. Zhang 2014) intègre l'optimisation des groupes au sein du problème d'optimisation.

Les auteurs se placent dans le cadre où des groupes inconnus de variables corrélées existent dans les données. Au sein d'un même groupe, les variables ont une association similaire avec la réponse, c'est-à-dire, si les variables X_j et X_l sont dans le même groupe alors $X_j\beta_j$ et $X_l\beta_l$ prennent des valeurs similaires.

Pour cela, la *cluster penalty* est introduite :

$$\frac{1}{2}\sum_{i=1}^{K}\sum_{j\in G_i}\left\|X_j\beta_j - \frac{1}{\mathrm{Card}(G_i)}\sum_{l\in G_i} X_l\beta_l\right\|_2^2. \tag{1.21}$$

avec $\{G_1, \ldots, G_k\}$ une partition en K groupes. Elle pénalise la distance entre $X_l\beta_l$ et la contribution moyenne du groupe. Ce qui encourage $X_j\beta_j \approx X_l\beta_l$ pour j et l appartenant au même groupe.

Le problème d'optimisation est :

$$\hat{\beta}^{CEN}_{\lambda_1,\lambda_2,K} = \underset{\beta\in\mathbb{R}^p,\ \mathcal{C}\in\mathcal{P}_K^p}{\mathrm{argmin}} \left\{\|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \frac{\lambda_2}{2}\sum_{i=1}^{K}\sum_{j\in G_i}\left\|X_j\beta_j - \frac{1}{\mathrm{Card}(G_i)}\sum_{l\in G_i} X_l\beta_l\right\|_2^2\right\} \tag{1.22}$$

avec $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, K un nombre de groupes fixé, $\mathcal{G} = \{G_1, \ldots G_K\}$ une partition en K groupes

et $\mathcal{P}_K^p$ l'ensemble des partitions de p variables en K groupes. Notons que contrairement au group-lasso, les groupes ne sont pas définis a priori et sont à optimiser.

Dans le cas $K = p$, on retrouve le problème du lasso. Dans le cas $K = 1$, le Cluster Elastic-Net est équivalent à l'elastic-net sur des versions modifiées de X et y.

En présence de fortes corrélations, la pénalité va encourager les coefficients d'un même groupe à être proches les uns des autres. Dans le cas de faibles corrélations, les coefficients sont encouragés à être proche de 0. En effet, en supposant $\|X_i\|_2^2 = 1$, la *cluster penalty* peut être réécrite $\frac{1}{2}\sum_{g\in\mathcal{G}}\frac{1}{|g|}\sum_{j,l\in g}\left[(1-r_{jl})(\beta_j^2+\beta_l^2)+r_{jl}(\beta_j-\beta_l)^2\right]$ avec $r_{jl} = \text{cor}(X_j, X_l)$. Dans le cas de fortes corrélations positives entre deux variables au sein d'un même groupe, on a r_{jl} proche de 1, le terme $(\beta_j - \beta_l)^2$ est donc dominant et favorise la proximité des deux coefficients. Dans le cas d'absence de corrélations, $(\beta_j^2 + \beta_l^2)$ domine et la pénalité va encourager les variables à être proches de 0. Dans le cas de corrélations négatives, la pénalité encourage les coefficients à prendre des valeurs opposées. Les variables importantes sont donc moins seuillées que les autres, ce qui résulte en une meilleure prédiction.

Afin de trouver une solution (non optimale) à ce problème, les auteurs proposent d'optimiser alternativement la répartition des variables en K groupes par les *k-means* et l'estimation de β par une descente cyclique de coordonnées. Les paramètres λ_1, λ_2 et K sont ensuite optimisés par validation croisée.

L'optimisation de trois paramètres peut être problématique. Définir une plage de valeurs intéressantes pour les paramètres est un premier problème. L'exécution d'une procédure de validation croisée peut être particulièrement coûteuse pour l'optimisation de trois paramètres.

1.3 Conclusion

Les méthodes de régression pénalisée pour la sélection de variables sont couramment utilisées dans les cas de grande dimension ($p > n$). En présence de corrélation, le lasso et le group-lasso semblent les méthodes les plus populaires couplées à une méthode de regroupement de variables. Toutes ces méthodes fonctionnent avec une unique partition des variables à la fois.

Le choix du ou des paramètres de régularisation se fait généralement par validation croisée *V-fold* malgré les réexécutions multiples de la méthode de régression qu'elle nécessite. Cette méthode classique n'est pas la plus appropriée dans un but de sélection de variables, surestimant généralement la taille du support au profit de la prédiction. Des méthodes comme la *stability selection* ou le critère Kappa sont plus efficaces pour la sélection mais souffrent du même problème de réexécutions.

Chapitre 2

Classification de variables et group-lasso

Dans le chapitre précédent, nous avons vu que la présence de corrélations entre variables pose des problèmes pour les méthodes usuelles de sélection de variables. La prise en compte de ces dépendances peut se faire en considérant des groupes de variables. Le group-lasso à l'aide d'une partition des variables définie a priori va permettre d'améliorer la tâche de sélection de variables. La nécessité d'une partition a priori pour le group-lasso requiert l'utilisation d'une méthode de regroupement de variables comme la classification ascendante hiérarchique (CAH). Dans la littérature, les approches classiques juxtaposent CAH et group-lasso (cf. Section 1.2.2). Ces approches utilisent généralement une unique partition des variables correspondant à un niveau de la classification choisi au préalable. Faire un mauvais choix de niveau réduit alors la performance finale de la procédure. Notre objectif est de combiner CAH et group-lasso pour choisir de façon automatique les groupes de variables influentes à un niveau de régularisation fixé. Dans la suite, nous présentons tout d'abord cette nouvelle approche, puis nous la comparons aux approches existantes présentées en Section 1.2.2, et la testons sur un jeu de données réelles.

2.1 Combiner CAH et group-lasso

2.1.1 Notations

Classiquement, le group-lasso est défini pour une partition donnée des variables en groupes (cf. Section 1.1.3). On rappelle que l'ensemble de groupes $\mathcal{G} = \{G_1, \dots, G_m\}$ de $\{X_1, \dots, X_p\}$ est une partition si $\cup_{i=1}^{m} G_i = \{X_1, \dots, X_p\}$ et $\forall i, j \in \{1, \dots, m\}$, $i \neq j$, $G_i \cap G_j = \emptyset$.

Notre but est d'utiliser plusieurs partitions des variables à la fois, ce qui implique que plusieurs groupes fournis au group-lasso auront une intersection deux à deux non vide. Par exemple, entre deux partitions issues d'un même arbre de classification ascendante hiérarchique (CAH), certains groupes peuvent être communs aux deux partitions ou alors inclus l'un dans l'autre du fait de la structure hiérarchique (cf. Section 1.2.1 pour plus de détails sur la CAH). Dans ce cas, nous disons que $\mathcal{G}$ est un ensemble de groupes *chevauchant*, c'est-à-dire que $\cup_{i=1}^{m} = \{1, \dots, p\}$ et $\exists i, j \in \{1, \dots, m\}$, $i \neq j$, $G_i \cap G_j \neq \emptyset$. Les partitions obtenues sont ensuite utilisées simultanément au sein d'un group-lasso. La procédure peut alors sélectionner des groupes issus de partitions différentes en adéquation avec la variable à expliquer. La procédure est résumée

dans l'Algorithme 9.

Algorithme 9 *Multi-Layer Group-Lasso*

1. Regroupement de variables
Appliquer une CAH sur l'ensemble des variables.
Retourne Un ensemble de partitions des variables $\mathcal{G}_1,\dots,\mathcal{G}_p$ correspondant aux différents niveaux.

2. Sélection de groupes
Appliquer un group-lasso avec l'ensemble des partitions issues des différents niveaux de la CAH.
Retourne Un chemin solution représentant l'ensemble des groupes sélectionnés (potentiellement issus de différents niveaux) pour différentes valeurs du paramètre de régularisation.

Afin de mettre en œuvre cette procédure, nous nous appuyons sur le *group-lasso with overlap* (Jacob, Obozinski et Vert 2009) qui étend le group-lasso au-delà de l'utilisation d'une seule partition.

D'autres articles de la littérature définissent une pénalité pour de la sélection hiérarchique. On notera la *Composite Absolute Penalty* (CAP) (Zhao, Rocha et Yu 2009) qui définit une famille générale de pénalité pour prendre en compte les structures entre variables. Une des applications est la sélection hiérarchique mais dans une version différente de celle que nous adoptons. La hiérarchie utilisée dans la *Composite Absolute Penalty* définit un ordre de sélection des groupes de variables via la pénalité suivante :

$$T(\beta) = \sum_{m=1}^{M} w_m \|(\beta_{G_m}, \beta_{\text{groupes descendants de } G_m})\|_{\gamma_m} \tag{2.1}$$

où $w_m > 0$, $\gamma_m > 1$, et les groupes descendants de G_m sont les groupes qui doivent être inclus après G_m. Par exemple, supposons l'existence de deux variables X_1 et X_2 et de la hiérarchie de 3 groupes associée $\mathcal{G} = \{G_1 = \{X_1, X_2\}, G_2 = \{X_1\}, G_3 = \{X_2\}\}$. Dans ce cas, la CAP va imposer que les groupes G_2 et G_3 soient sélectionnés avant le groupe G_1. Tandis que nous souhaitons pouvoir sélectionner le groupe G_1 sans forcément sélectionner les groupes G_2 et G_3.

L'utilisation d'une hiérarchie de groupes au sein de la pénalité du group-lasso a été abordée dans le cas de la régression multi-tâches (Kim et Xing 2012). Une CAH est effectuée pour obtenir une hiérarchie entre les différentes variables à expliquer. Le but est que les estimateurs associés à différentes variables à expliquer fortement corrélées partagent un même ensemble de coefficients non nuls. Cette approche diffère également de la nôtre puisque nous utilisons une hiérarchie sur les variables explicatives.

2.1.2 *Group-lasso with overlap*

Le group-lasso est étendu aux groupes chevauchant via l'introduction d'une nouvelle pénalité : l'*overlap penalty*. Soit $\mathcal{G} = \{G_1,\dots,G_K\}$ un ensemble de K groupes tel que $\forall i = 1,\dots,K$, $G_i \subset \{1,\dots,p\}$. L'*overlap penalty* est définie par :

$$\Omega^{\mathcal{G}}_{\text{overlap}}(\beta) = \inf_{v \in \mathcal{V}_{\mathcal{G}},\ \sum_{i=1}^{K} v^{(G_i)} = \beta} \sum_{i=1}^{K} w_i \left\| v^{(G_i)} \right\|_2 \tag{2.2}$$

avec $w_i > 0$ un poids associé au groupe G_i et $\mathcal{V}_\mathcal{G}$ l'ensemble des matrices de la forme $\mathbf{v} = [v^{(G_1)}, \dots, v^{(G_K)}]$ représentant la concaténation des vecteurs $v^{(G_i)}$ avec $v^{(G_i)} \in \mathbb{R}^p$ de support $\mathbb{S}\left(v^{(G_i)}\right) \subset G_i$. Le vecteur $\beta \in \mathbb{R}^p$ à estimer est ainsi décomposé en une somme de K vecteurs de $\mathbb{R}^p$ dont chacun a son support inclus dans un groupe de $\mathcal{G}$.

Exemple (Illustration de l'ensemble $\mathcal{V}_\mathcal{G}$) *Notons* $\mathcal{G} = \left\{G_1 = \{1,2\}, G_2 = \{2,3,4\}, G_3 = \{3,4\}\right\}$, *un ensemble de trois groupes.*

L'ensemble $\mathcal{V}_\mathcal{G}$ *contient les matrices* $\mathbf{v}$ *de taille* 4×3 *de la forme :*

$$\mathbf{v} = \begin{pmatrix} v_1^{(G_1)} & 0 & 0 \\ v_2^{(G_1)} & v_2^{(G_2)} & 0 \\ 0 & v_3^{(G_2)} & v_3^{(G_3)} \\ 0 & v_4^{(G_2)} & v_4^{(G_3)} \end{pmatrix}.$$

□

L'estimateur du group-lasso avec groupes chevauchant est :

$$\hat{\beta}_\lambda^{\mathcal{G},\text{overlap}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\, \Omega_{\text{overlap}}^{\mathcal{G}}(\beta) \right\} \tag{2.3}$$

avec $\lambda \geq 0$ le paramètre de régularisation.

Le résultat de consistance du group-lasso présenté en Section 1.1.3 est étendu pour le *group-lasso with overlap* dans (Jacob, Obozinski et Vert 2009).

La solution (2.3) peut être trouvée par le biais des algorithmes de résolution du group-lasso classique. Pour cela, il faut réécrire le problème sous la forme d'un group-lasso classique avec une version modifiée de la matrice de design X.

Rappelons que pour $G_i \in \mathcal{G}$, on note X_{G_i} (respectivement β_{G_i}) la matrice (respectivement le vecteur) restreinte aux variables du groupe G_i. Définissons $\tilde{X}$ la matrice de taille $n \times (\sum_{i=1}^K \text{Card}(G_i))$ comme la concaténation des sous-matrices de X restreintes aux groupes de $\mathcal{G}$:

$$\tilde{X} = [X_{G_1}, \dots, X_{G_K}].$$

Le group-lasso avec groupes chevauchant avec la matrice X et l'ensemble de groupes $\mathcal{G}$ peut être réécrit comme un group-lasso avec la matrice $\tilde{X}$ et la partition $\tilde{\mathcal{G}} = \{\tilde{G}_1, \dots, \tilde{G}_K\}$ de l'ensemble $\left\{1, \dots, \sum_{i=1}^K \text{Card}(G_i)\right\}$ induite par $\mathcal{G}$. Le nouvel estimateur est :

$$\hat{\theta}_\lambda^{\tilde{\mathcal{G}}} = \underset{\theta \in \mathbb{R}^{\sum_{i=1}^K \text{Card}(\tilde{G}_i)}}{\operatorname{argmin}} \left\{ \frac{1}{2}\|y - \tilde{X}\theta\|_2^2 + \lambda \sum_{i=1}^K w_i \|\theta_{\tilde{G}_i}\|_2 \right\} \tag{2.4}$$

avec $\lambda \geq 0$ le paramètre de régularisation et w_i le poids associé au groupe $\tilde{G}_i$ (qui reste le même que celui de G_i).

Une fois l'estimateur $\hat{\theta}_\lambda^{\tilde{\mathcal{G}}}$ obtenu, l'estimateur $\hat{\beta}_\lambda^{\mathcal{G},\text{overlap}}$ de l'équation (2.3) est obtenu par $\hat{\beta}_\lambda^{\mathcal{G},\text{overlap}} = \sum_{i=1}^K \tilde{v}^{(G_i)}$ avec $\tilde{v}^{(G_i)} \in \mathbb{R}^p$ tel que son support $\mathbb{S}\left(\tilde{v}^{(G_i)}\right) = G_i$ et $\tilde{v}_{G_i}^{(G_i)} = \left(\hat{\theta}_\lambda^{\tilde{\mathcal{G}}}\right)_{\tilde{G}_i}$.

Exemple (Illustration de la construction de $\tilde{X}$) *Reprenons l'ensemble de groupes de l'exemple 1,* $\mathcal{G} = \left\{G_1 = \{1,2\}, G_2 = \{2,3,4\}, G_3 = \{3,4\}\right\}$. *La matrice X associée possède donc 4 colonnes.*

On crée la matrice $\tilde{X}$ à sept colonnes :

$$\tilde{X} = [\underbrace{X_1, X_2}_{X_{G_1}}, \underbrace{X_2, X_3, X_4}_{X_{G_2}}, \underbrace{X_3, X_4}_{X_{G_3}}].$$

La partition associée à cette matrice est $\tilde{\mathcal{G}} = \left\{\tilde{G}_1 = \{1,2\}, \tilde{G}_2 = \{3,4,5\}, \tilde{G}_3 = \{6,7\}\right\}$.

Pour plus de simplicité, on note $\hat{\theta}$ l'estimateur $\hat{\theta}_\lambda^{\tilde{\mathcal{G}}}$ (cf. équation (2.4)) et $\hat{\beta}$ l'estimateur $\hat{\beta}_\lambda^{\mathcal{G},overlap}$ (cf. équation (2.3)).

L'estimateur $\hat{\beta}$ est obtenu à partir de $\hat{\theta}$ de la manière suivante :

$$\hat{\beta} = \underbrace{\begin{pmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ 0 \\ 0 \end{pmatrix}}_{\tilde{v}^{(G_1)}} + \underbrace{\begin{pmatrix} 0 \\ \hat{\theta}_3 \\ \hat{\theta}_4 \\ \hat{\theta}_5 \end{pmatrix}}_{\tilde{v}^{(G_2)}} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \hat{\theta}_6 \\ \hat{\theta}_7 \end{pmatrix}}_{\tilde{v}^{(G_3)}}.$$

□

Utiliser plusieurs partitions différentes (et donc avoir des groupes chevauchant) afin de faire de la sélection de groupes ne pose donc pas de problème théoriquement. De plus, la méthode peut être résolue avec les mêmes algorithmes que pour le group-lasso. En se basant sur le *Group-lasso with overlap*, nous pouvons mettre en œuvre le *Multi-Layer Group-Lasso* introduit dans l'algorithme 9.

2.1.3 *Multi-Layer Group-Lasso*

Multi-Layer Group-Lasso

Soit $\mathcal{G}_* = \{\mathcal{G}_1, \ldots, \mathcal{G}_p\}$ l'ensemble des partitions des variables $\{X_1, \ldots, X_p\}$ correspondant aux p niveaux de la CAH. Notons $\tilde{X} = [X_{\mathcal{G}_1}, \ldots, X_{\mathcal{G}_p}]$ la concaténation des matrices $X_{\mathcal{G}_s}$, $s = 1, \ldots, p$ où $X_{\mathcal{G}_s} = [X_{G_1^s}, \ldots, X_{G_s^s}]$ et $G_i^s \in \mathcal{G}_s$. La matrice $\tilde{X}$ correspond à la duplication p fois (=nombre de partitions dans $\mathcal{G}_*$) de la matrice X. L'estimateur fourni par notre approche est défini par :

$$\hat{\beta}_\lambda^{\mathcal{G}_*} = \underset{\beta \in \mathbb{R}^{pS}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|y - \tilde{X}\beta\|_2^2 + \lambda \sum_{s=1}^{S} \rho_s \sum_{i=1}^{g_s} w_i^s \|\beta_{G_i^s}\|_2 \right\} \tag{2.5}$$

avec $\lambda \geq 0$ le paramètre de régularisation, G_i^s le i^e groupe de la partition $\mathcal{G}_s$ des variables issues du niveau s de la CAH et $w_i^s > 0$ le poids associé au groupe G_i^s. Comme pour le group-lasso, on utilisera le poids $w_i^s = \sqrt{|G_i^s|}$. Le paramètre $\rho_s \geq 0$ est un poids attribué à la partition $\mathcal{G}_s$, son but est de quantifier la confiance en le niveau correspondant. Une faible valeur de ρ_s favorise la sélection de groupes appartenant à la partition $\mathcal{G}_s$ mais la méthode reste cependant libre de sélectionner des groupes issus de plusieurs niveaux.

L'écriture du *Multi-Layer Group-Lasso* (cf. équation 2.5) est équivalent à celle du *Tree stuctured Group-Lasso* (Jie Wang et Ye 2015). Dans cet article est présentée une méthode de *screening* nommée Multi-Layer Feature Reduction pour le *Tree stuctured Group-Lasso*. Le but est d'éliminer certains groupes de l'arbre hiérarchique qui ne jouent pas un rôle dans l'explication de la réponse. C'est une méthode de pré-traitement à utiliser avant un algorithme pour estimer les coefficients des groupes d'intérêts. Nous allons nous différencier du *Tree stuctured Group-Lasso*

par un choix de poids particulier et une réécriture du problème dans le cas où la totalité des niveaux d'une CAH est utilisée. Dans ce cas, l'ensemble $\mathcal{G}_*$ contient p partitions des p variables contenant au total $p(p+1)/2$ groupes. La matrice $\tilde{X}$ est alors de dimension $n \times p^2$, ce qui est prohibitif. Pour réduire la taille de cette matrice, nous allons exploiter la structure hiérarchique d'arbre de la CAH.

Le *Multi-Layer Group-Lasso* s'inscrit dans la pénalité définie dans (Jenatton, Audibert et F. Bach 2011) qui inclut le lasso et le group-lasso classique et avec groupes chevauchant. Cette pénalité générale permet de définir les groupes en fonction des formes désirées du support de la solution.

Réduction de la taille de la matrice $\tilde{X}$

Premièrement, remarquons que dans un arbre hiérarchique obtenu par CAH, deux partitions successives notées $\mathcal{G}_s$ la partition en s groupes et $\mathcal{G}_{s-1}$ celle en $s-1$ groupes, ont $s-2$ groupes en commun. Cela vient du principe même de la CAH (cf. Section 1.2.1) qui à chaque étape agrège deux groupes pour n'en former plus qu'un, les autres restant inchangés. Donc, dans notre estimateur (2.5), certains groupes apparaissent dans différents niveaux et seuls les poids ρ_s associés aux niveaux auxquels ils appartiennent diffèrent. Les groupes communs à $\mathcal{G}_s$ et $\mathcal{G}_{s-1}$ sont pénalisés avec un poids de ρ_s et de ρ_{s-1}. Nous allons montrer que dans ce cas, seul le groupe avec le poids le plus faible peut être sélectionné.

Lemme 1 *Soit $\mathcal{G} = \{G_1, \dots, G_K\}$ un ensemble de groupes (et non une partition) de $\{1, \dots, p\}$ en K groupes. Supposons que $G_1 = G_2$ et $w_2 > w_1 > 0$. Soit $\hat{\theta}^{\mathcal{G}}_\lambda$ la solution de (2.4).*

Alors $(\hat{\theta}^{\mathcal{G}}_\lambda)_{G_2} = 0_{|G_2|}$.

La démonstration de ce lemme est disponible dans l'Annexe B.

Le lemme 1 montre que lorsque deux groupes de variables sont identiques mais pénalisés avec deux poids différents, les coefficients estimés associés au groupe avec le poids le plus fort sont forcément nuls. En utilisant ce lemme, on peut réduire la dimension de la matrice de design transformée $\tilde{X}$ dans (2.5) quand l'ensemble des partitions de la CAH est utilisé. Pour cela, un groupe appartenant à différentes partitions sera présent une unique fois dans la matrice $\tilde{X}$ et le poids représentant la qualité de la meilleure partition (le poids le plus faible) à laquelle il appartient lui sera associé dans la pénalité.

Définissons $\mathcal{G}^{\mathrm{u}}_*$, l'ensemble des groupes différents de la classification ascendante hiérarchique. $\mathcal{G}^{\mathrm{u}}_*$ est tel que pour tout groupe G^{u}_i et G^{u}_j, $i \neq j$, de $\mathcal{G}^{\mathrm{u}}_*$, on a $G^{\mathrm{u}}_i \neq G^{\mathrm{u}}_j$. Ceci n'est pas le cas de $\mathcal{G}_*$. Ce nouvel ensemble contient $2p-1$ groupes tous différents. Ils correspondent aux p groupes de une variable du niveau en p groupes puis un groupe supplémentaire par niveau. Ce nouvel ensemble va être utilisé à la place de $\mathcal{G}_*$ qui contenait $p(p+1)/2$ groupes.

Ainsi, pour un groupe G^{u}_j de l'ensemble $\mathcal{G}^{\mathrm{u}}_*$, le poids minimal des partitions auxquelles il appartient est noté :

$$\rho^{\mathrm{u}}_j = \min\left\{\rho_s \mid s \in 1, \dots, p \text{ tel que } G^{\mathrm{u}}_j \in \mathcal{G}_s\right\}.$$

Exemple (Ensemble $\mathcal{G}^{\mathrm{u}}_*$) *Supposons une hiérarchie de 3 variables définie par les 3 partitions $\mathcal{G}_1 = \left\{G^1_1 = \{1,2,3\}\right\}$, $\mathcal{G}_2 = \left\{G^2_1 = \{1,2\}, G^2_2 = \{3\}\right\}$ et $\mathcal{G}_3 = \left\{G^3_1 = \{1\}, G^3_2 = \{2\}, G^3_3 = \{3\}\right\}$. À chaque niveau est associé un poids ρ_i, $i = 1, \dots, 3$.*

Le groupe $\{3\}$ se trouve dans les partitions $\mathcal{G}_3$ et $\mathcal{G}_2$. Il ne se retrouve qu'une seule fois au sein de $\mathcal{G}^u_*$ qui est :

$$\mathcal{G}^u_* = \left\{G^u_1 = \{1\}, G^u_2 = \{2\}, G^u_3 = \{3\}, G^u_4 = \{1,2\}, G^u_5 = \{1,2,3\}\right\}.$$

Tableau 2.1 – Comparaison du nombre de variables des matrices X ,$\tilde{X}$ et $\check{X}$ noté respectivement p, $\tilde{p}$ et $\check{p}$. Pour $\check{p}$, le résultat est une moyenne sur 100 répétitions et entre parenthèses se trouve l'écart-type. Les données sont simulées par une loi gaussienne multivariée de moyenne nulle et de matrice de variance Σ, diagonale par blocs de taille 10 remplis de 1 sur la diagonale et 0.7 ailleurs. La CAH a été effectuée avec la méthode de Ward et la distance euclidienne.

p	$\tilde{p}$	$\check{p}$
100	10 000	821.99 (13.64)
250	62 500	2 404.35 (39.65)
500	250 000	5 321.71 (78.55)
750	562 500	8 483.52 (131.98)
1 000	1 000 000	11 698.15 (153.38)
1 500	2 250 000	18 481.74 (232.85)
2 000	4 000 000	25 475.53 (347.84)
2 500	6 250 000	32 619.04 (360.34)

De plus la qualité de la hiérarchie associée au groupe $G_3^u = \{3\}$ *est* $\rho_3^u = \min\{\rho_2, \rho_3\}$. □

Posons $\check{X} = X_{\mathcal{G}_*^{\mathrm{u}}}$. Alors, résoudre (2.5) est équivalent à résoudre

$$\hat{\beta}_\lambda^{\mathcal{G}_*^{\mathrm{u}}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2}\|y - \check{X}\beta\|_2^2 + \lambda \sum_{j=1}^{2p-1} \rho_j^{\mathrm{u}} w_j^{\mathrm{u}} \|\beta_{G_j^{\mathrm{u}}}\|_2 \right\} \tag{2.6}$$

avec $\lambda \geq 0$ le paramètre de régularisation, w_j^{u} le poids associé au groupe G_j^{u} (on utilisera $w_j^{\mathrm{u}} = \sqrt{|G_j^{\mathrm{u}}|}$) et ρ_j^{u} le plus petit poids des niveaux auxquels G_j^{u} appartient.

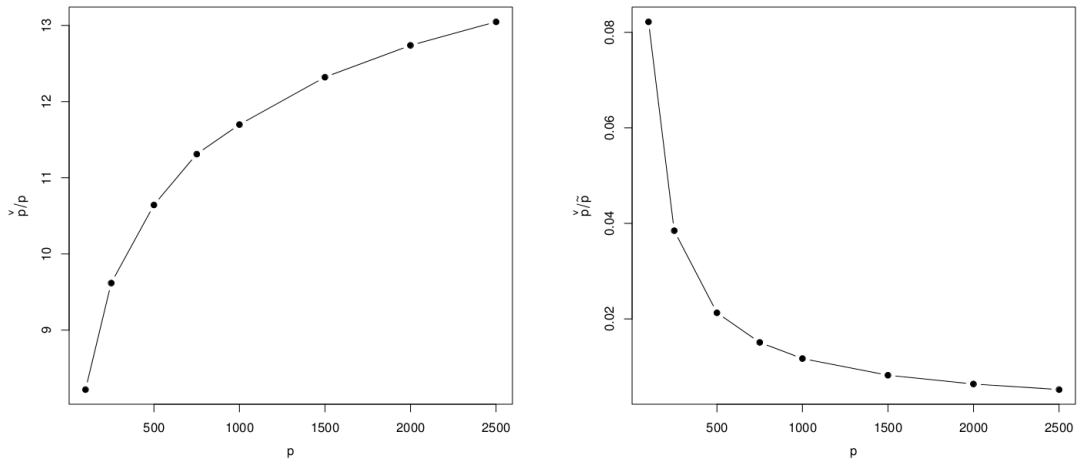
Dans la précédente formulation (équation (2.5)), en prenant l'ensemble des partitions de la CAH, la matrice $\tilde{X}$ était de taille $n \times p^2$ et contenait $p(p+1)/2$ groupes. Dans la nouvelle formulation (équation (2.5)), la matrice $\check{X}$ contient $2p-1$ groupes et est de taille aléatoire inconnue (car on ne connaît pas à l'avance la taille des groupes gardés à chaque niveau).

Dans le Tableau 2.1 et sur la figure 2.1, la taille de la matrice $\check{X}$ a été calculée dans un cadre de simulation détaillé dans la légende du tableau. On constate une très forte diminution entre le nombre de variables de $\tilde{X}$ et $\check{X}$, par exemple, on passe de 1 000 000 à environ 11 700 pour une matrice d'origine de 1000 variables. Par rapport à p, $\check{p}$ augmente de manière sur-linéaire (ratio $\frac{\check{p}}{p}$ strictement croissant). Pour les valeurs testées, la taille d'origine est multipliée par un facteur compris entre 8 et 13, ce qui est important mais reste raisonnable par rapport au nombre de variables de $\tilde{X}$.

2.1.4 Influence des poids associés aux partitions

Dans la méthode proposée (équation (2.6)), un poids ρ_s associé à la qualité de la partition est introduit. Ce poids est un a priori sur les partitions fournies i.e. les niveaux de la CAH. Il doit pénaliser faiblement les partitions intéressantes.

Comme choix de ρ_s, nous proposons d'utiliser la règle du saut maximal définie en Section 1.2.1. Pour rappel, la longueur de branche associée au niveau s de la hiérarchie est $l_s = h_{s-1} - h_s$



(a) Ratio $\frac{\check{p}}{p}$ en fonction de p

(b) Ratio $\frac{\check{p}}{\tilde{p}}$ en fonction de p

FIGURE 2.1 – Comparaison des tailles des différentes matrices de design (cf. Tableau 2.1). En abscisse, on trouve p le nombre de variables de la matrice X. En ordonnée, le ratio $\frac{\check{p}}{p}$ entre la taille de la matrice $\check{X}$ et X (à gauche) et le ratio $\frac{\check{p}}{\tilde{p}}$ entre la taille de la matrice $\check{X}$ et $\tilde{X}$ (à droite).

(cf. figure 2.2). Le niveau optimal est celui maximisant ce critère. On définit alors le poids ρ_s suivant :

$$\rho_s = \frac{1}{\sqrt{l_s}}. \tag{2.7}$$

Ce poids est minimal quand la longueur de branche est maximale.

Nous allons nous intéresser à la capacité de ce critère à décrire la qualité de la partition. Pour cela, nous nous plaçons dans le cadre de simulation suivant :

- nombre d'individus $n = 25$ à 200 par pas de 25 ;
- nombre de variables $p = 500$;
- $X_{1,.}, \dots, X_{n,.} \sim \mathcal{N}(0_p, \Sigma_\rho)$;
- Σ_ρ une matrice diagonale par blocs de taille 10×10 où chaque bloc contient des 1 sur la diagonale et ρ partout ailleurs.

La matrice X ainsi générée présente une partition claire dans sa structure de corrélation. Une classification ascendante hiérarchique avec la distance euclidienne et le critère de Ward est alors effectuée sur ces données. Sur la figure 2.3a, la hauteur de branche h_s est représentée et sur la figure 2.3b, le poids décrivant la qualité de la partition $\rho_s = \frac{1}{\sqrt{l_s}}$. On constate que la hauteur de branche augmente significativement en passant du niveau 50 au 49. Le niveau 50 correspond à la partition définie par les blocs de la matrice Σ_ρ. Plus il y a d'individus, plus le saut est marqué. Les poids représentant la qualité de la partition générée par ces hauteurs décrivent un minimum au niveau 50. Les poids restent dans la même gamme de valeur quand le nombre d'individus n varie malgré la variance de la hauteur. Le minimum de ρ_s demeure plus petit en présence

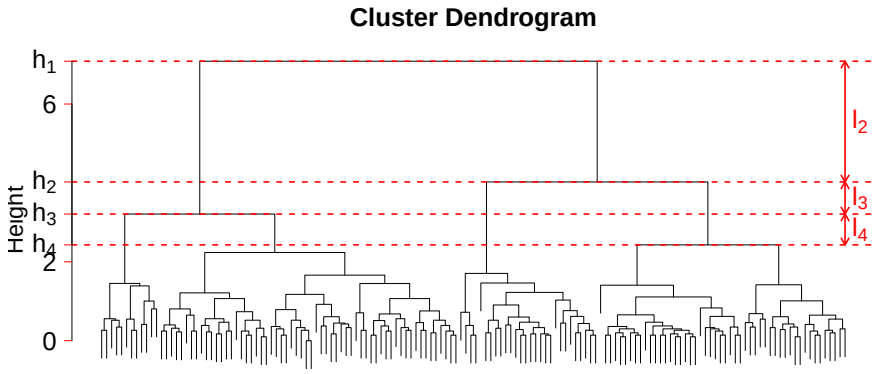


FIGURE 2.2 – Dendrogramme obtenu après une classification ascendante hiérarchique avec distance euclidienne et average linkage sur les données iris (FISHER 1936). h_s correspond à la valeur de la dissimilarité des deux groupes fusionnés pour former la partition du niveau s. l_s représente le saut effectué entre les niveaux s et $s-1$.

d'un plus grand nombre d'individus. On remarque que les partitions avec un faible nombre de groupes sont assez peu pénalisées, ce qui peut être un problème, car cela peut amener à sélectionner des groupes contenant beaucoup de variables, ce qui n'est pas forcément des plus intéressant.

Sur la figure 2.4a, on constate que les poids ρ_i^{u} et w_i^{u} $(= \sqrt{|G_i^{\mathrm{u}}|})$ semblent du même ordre de grandeur. Sur la figure 2.4b, le poids $\rho_i^{\mathrm{u}} w_i^{\mathrm{u}}$ est tracé pour chaque groupe de l'ensemble $\mathcal{G}_i^{\mathrm{u}}$. On voit que les valeurs minimales se trouvent aux alentours de l'indice 950. Ces valeurs minimales correspondent aux groupes formés aux niveaux précédant le niveau contenant 50 groupes et sont donc des groupes présents au niveau contenant 50 groupes. De plus, le poids sur la taille des groupes a permis de contrebalancer le faible poids de la qualité des grands groupes.

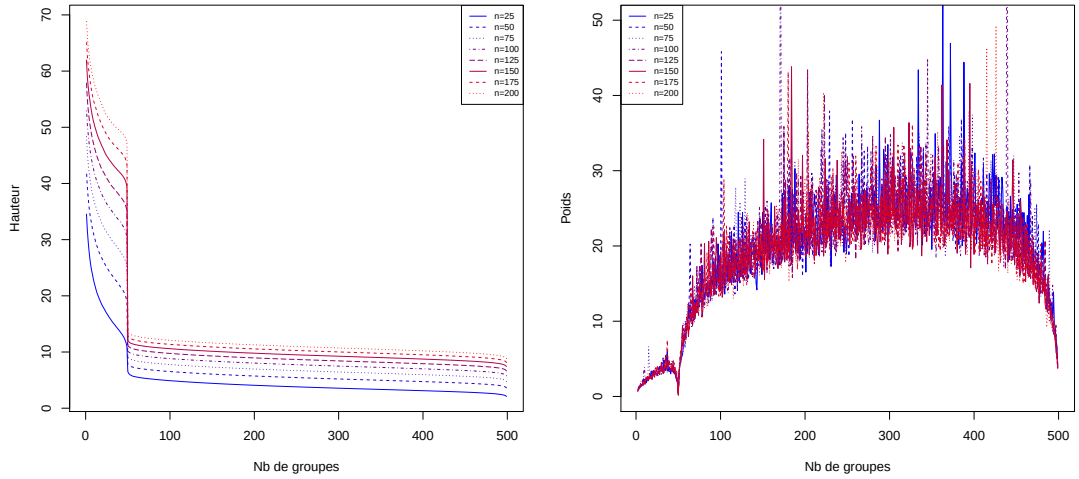
Le poids défini en (2.7), bien qu'assez basique, retranscrit bien la qualité de la partition avec un minimum en la partition définie par la structure de blocs. Les faibles valeurs pour des partitions avec peu de groupes (donc de grandes tailles) peut être gênant mais sont corrigées par les poids associés à la taille des groupes.

2.2 Simulations

Dans cette section, nous allons nous intéresser à l'étude de notre méthode dans différents cadres de simulation.

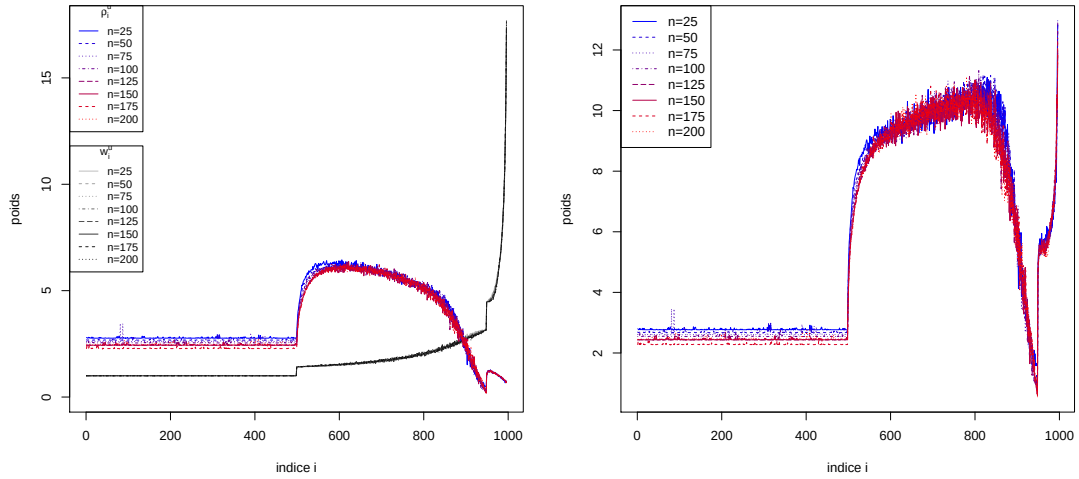
2.2.1 Présentation des cadres de simulations

La matrice de design X de taille $n \times p$ est générée à l'aide d'une loi normale multivariée de matrice de variance-covariance Σ $(X_{1,.}, \ldots, X_{n,.} \sim \mathcal{N}(0_p, \Sigma))$. Les différentes matrices Σ utilisées sont présentées ci-dessous et sur la figure 2.5.



(a) Hauteur des branches h_s (b) Poids ρ_s

FIGURE 2.3 – En abscisse, le nombre de groupes (également le niveau s) des différentes partitions de la hiérarchie. À gauche, la hauteur des branches h_s de la CAH. À droite, le poids ρ_s définissant la qualité du niveau s de la hiérarchie. Moyenne sur 100 réalisations.



(a) Poids ρ_i^{u} et w_i^{u} (b) Poids $\rho_i^{\mathrm{u}} w_i^{\mathrm{u}}$

FIGURE 2.4 – Poids issus du *Multi-Layer Group-Lasso* (2.6) pour chaque groupe de $\mathcal{G}^{\mathrm{u}}$. Sur le graphe de gauche, les courbes en dégradé de gris à noir représentent le poids w_i^{u} correspondant à la racine carrée de la taille des groupes tandis que celles en dégradé de bleu à rouge représentent le poids ρ_i^{u}. En abscisse est représenté l'indice du groupe au sein de l'ensemble $\mathcal{G}^{\mathrm{u}}$. Les p premiers indices correspondent aux variables seules et les $p-1$ suivants aux groupes des niveaux suivants.

Cadre 1 : structure de blocs de taille uniforme

Dans ce 1er cadre, la matrice de variance-covariance de X est une matrice diagonale par blocs. Tous les blocs ont même taille l et chaque bloc contient des 1 sur la diagonale et ρ partout ailleurs.

Cadre 2 : structure de blocs de taille non uniforme

Dans le cadre 2, la matrice Σ a la même forme que précédemment mais la taille des groupes n'est plus uniforme. Les groupes sont de trois tailles différentes l, $\frac{l}{2}$ et $\frac{3l}{2}$.

Cadre 3 : présence de variables indépendantes

La matrice Σ est décomposée en deux parties : la 1re partie a la même forme que le cadre 1, tandis que la 2e est diagonale. On a donc une partie des variables qui forme une structure de groupes et l'autre qui contient des variables "isolées". La proportion de variables isolées est notée π.

Cadre 4 : corrélation inter-blocs

La structure de blocs du cadre 1 est conservée mais une corrélation ρ_2 entre les blocs est rajoutée.

2.2.2 Stabilité de la classification ascendante hiérarchique

Dans cette section, la stabilité de la classification ascendante hiérarchique (CAH) lorsque le nombre d'individus varie est étudiée. Nous allons nous intéresser aux hauteurs de branches et aux partitions.

Pour comparer les partitions issues de deux différents arbres hiérarchiques, nous allons utiliser le coefficient de corrélation cophénétique (Sokal et Rohlf 1962) et l'indice de Rand (Rand 1971).

Mesures de comparaison

Indice de Rand

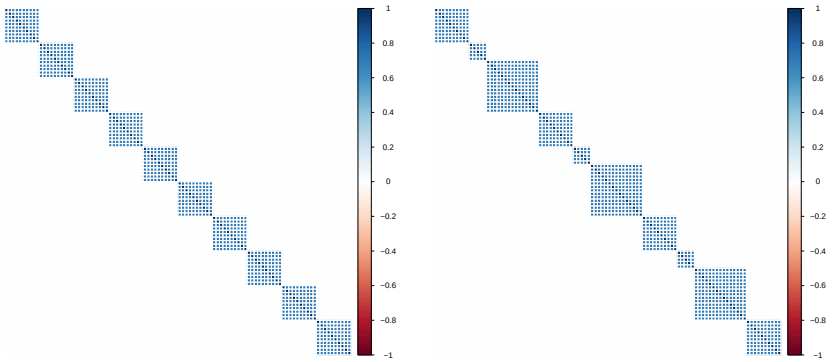
L'indice de Rand (Rand 1971) permet de comparer deux partitions d'un même ensemble en un même nombre ou un nombre différents de groupes. Pour chaque paire de points, on regarde s'ils sont classés de la même manière dans les deux partitions.

Soit $\Omega = \{1,\dots,n\}$ un ensemble de n éléments et $\mathcal{G} = \{G_1,\dots,G_r\}$ et $\mathcal{H} = \{H_1,\dots,H_s\}$ deux partitions de Ω en respectivement r et s groupes.

On définit les grandeurs suivantes :

— a, le nombre de paires d'éléments de Ω qui sont dans le même groupe dans $\mathcal{G}$ et dans le même groupe dans $\mathcal{H}$. $a = \text{Card}(A)$ avec $A = \left\{(i,j) \in \Omega^2 \mid \exists k,l \text{ tels que } i,j \in G_k,\ i,j \in H_l\right\}$;

— b, le nombre de paires d'éléments de Ω qui sont dans des groupes différents dans $\mathcal{G}$ et dans des groupes différents dans $\mathcal{H}$. $b = \text{Card}(B)$ avec $B = \left\{(i,j) \in \Omega^2 \mid \exists k_1 \neq k_2, \exists l_1 \neq l_2 \text{tels que } i \in G_{k_1}, j \in G_{k_2} \text{ et } i \in H_{l_1}, j \in H_{l_2}\right\}$;



(a) Cadre 1 (b) Cadre 2

(c) Cadre 3 (d) Cadre 4

FIGURE 2.5 – Matrices de variance-covariance des cadres de simulations.

— c, le nombre de paires d'éléments de Ω qui sont dans le même groupe dans $\mathcal{G}$ et dans des groupes différents dans $\mathcal{H}$. $c = \text{Card}(C)$ avec $C = \left\{(i,j) \in \Omega^2 \mid \exists k, \exists l_1 \neq l_2 \text{ tels que } i,j \in G_k \text{ et } i \in H_{l_1}, j \in H_{l_2}\right\}$;
— d, le nombre de paires d'éléments de Ω qui sont dans des groupes différents dans $\mathcal{G}$ et dans le même groupe dans $\mathcal{H}$. $d = \text{Card}(D)$ avec $D = \left\{(i,j) \in \Omega^2 \mid \exists k_1 \neq k_2, \exists l \text{ tels que } i \in G_{k_1}, j \in G_{k_2} \text{ et } i,j \in H_l\right\}$.

L'indice de Rand représente le taux d'accord entre les deux partitions :

$$\text{R} = \frac{a+b}{a+b+c+d}. \tag{2.8}$$

Cet indice est compris entre 0 (partitions totalement différentes) et 1 (partitions identiques).

L'indice de Rand ajusté (Hubert et Arabie 1985) est une version corrigée pour prendre en compte l'aléatoire dans la répartition en groupe. Il correspond à $\text{R}_{\text{adj}} = \frac{\text{indice - indice espéré}}{\text{indice maximal - indice espéré}}$.

Notons $n_{ij} = \text{Card}(G_i \cap H_j)$ et $\binom{a}{b} = \frac{a!}{b!(a-b)!}$.

	H_1	H_2	$\dots$	H_s	Total
G_1	n_{11}	n_{12}	$\dots$	n_{1s}	g_1
G_2	n_{21}	n_{22}	$\dots$	n_{2s}	g_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
G_r	n_{r1}	n_{r2}	$\dots$	n_{rs}	g_r
Total	h_1	h_2	$\dots$	h_s	

L'indice de Rand ajusté vaut alors :

$$\text{R}_{\text{adj}} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{g_i}{2}\sum_j \binom{h_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{g_i}{2} + \sum_j \binom{h_j}{2}\right] - \left[\sum_i \binom{g_i}{2}\sum_j \binom{h_j}{2}\right]/\binom{n}{2}}. \tag{2.9}$$

Il vaut au plus 1 et peut prendre des valeurs négatives lorsque les partitions sont peut liées. L'indice de Rand est nul lorsque les accords entre les partitions sont dus au hasard.

Distance Cophénétique

La distance cophénétique de deux éléments est la valeur de la dissimilarité (obtenue par le critère de Ward par exemple) à laquelle ces éléments se sont retrouvés dans le même groupe (Sokal et Rohlf 1962). Pour comparer deux arbres hiérarchiques, nous calculons la corrélation entre leurs matrices de distance cophénétique respectives. Une valeur proche de 1 indique une forte ressemblance entre les deux arbres hiérarchiques. Une valeur proche de 0 indique quant à elle une faible ressemblance.

Résultats

Pour cela, nous faisons varier le nombre d'individus n de 25 à 200 par pas de 25 et regardons différents critères. Chaque CAH a été faite avec la distance euclidienne et la méthode de Ward.

Comparaison des partitions

Pour calculer l'indice de Rand ajusté, les partitions obtenues pour la CAH pour $n = 200$ individus sont comparées à celles obtenues pour un nombre d'individus plus faible. Seules

les partitions en un même nombre de groupes sont comparées. Ces résultats sont visibles sur la figure 2.6. Sur la figure 2.7, le coefficient de corrélation cophénétique entre deux arbres hiérarchiques obtenus avec un nombre différent d'individus est représenté. Nous présentons les résultats pour le cadre 3 de simulation qui en plus d'une structure de groupes contient des variables isolées. Cela afin d'étudier l'impact de ces variables lorsque le nombre d'individus varie.

Sur la figure 2.6, on remarque deux points atteignant un indice de Rand ajusté de 1 pour n suffisamment grand. Ces deux points correspondent à deux partitions bien précises. Dans la 1re avec un nombre de groupes plus élevé, chaque bloc défini a son propre groupe et chaque variable isolée forme un groupe. Dans la 2e, les groupes sont identiques pour les blocs mais les variables isolées sont toutes dans le même groupe. La présence de trop peu d'individus entraîne des différences dans les partitions en ces deux points. Les variables isolées jouent le rôle de bruit et se retrouvent dans des groupes correspondant à des blocs. Plus le nombre d'individus augmente, plus ces variables isolées sont séparées facilement des blocs d'intérêts.

Nombre de variables p Sur les figures 2.6a et 2.6b, on observe également que lorsque le nombre de variables p augmente de 250 à 500, le nombre d'individus nécessaire pour obtenir un indice de Rand ajusté de 1 aux deux niveaux d'intérêts augmente. Sur les figures 2.7a et 2.7b, la corrélation cophénétique entre les arbres hiérarchiques obtenus pour différentes valeurs de n a globalement baissée lorsque p a augmenté.

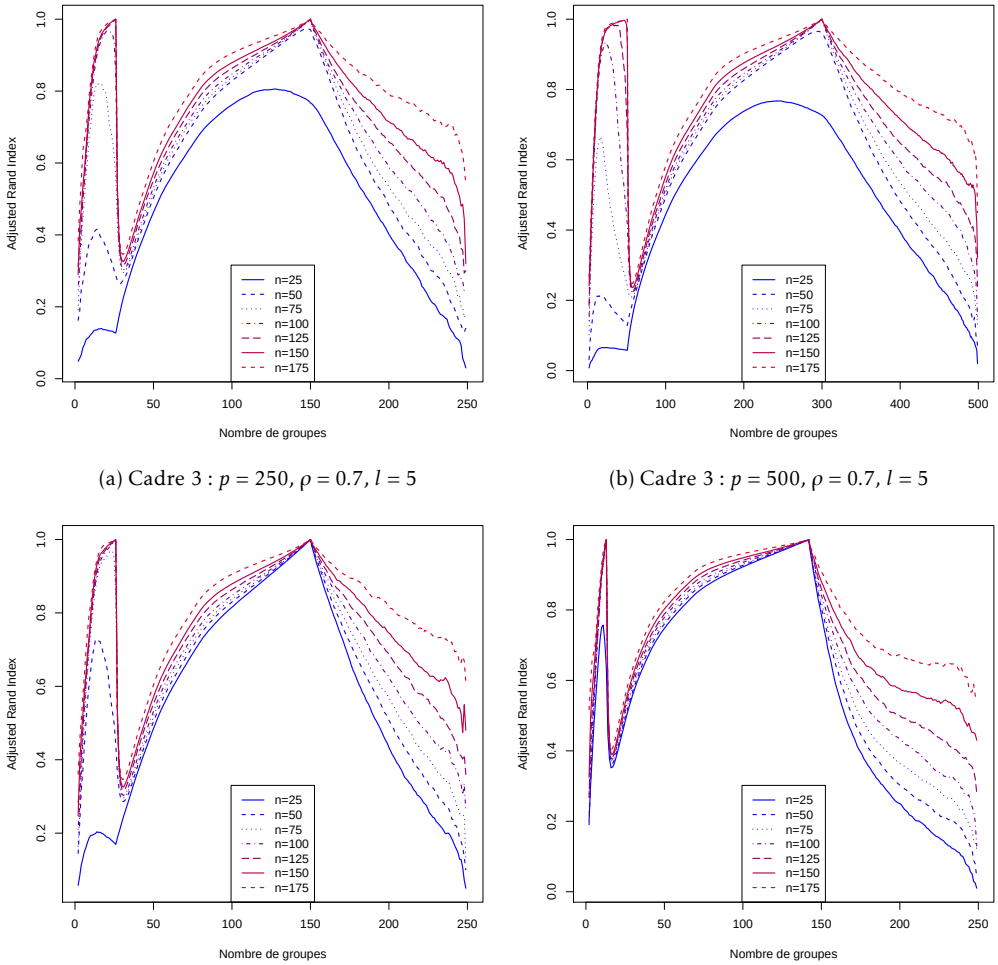
Corrélation intra-blocs ρ Entre les figures 2.6a et 2.6c, le niveau de corrélation intra-blocs a augmenté de 0.7 à 0.9. On observe dans ce cas que moins d'individus sont nécessaires pour obtenir une stabilité au niveau des deux niveaux d'intérêts. La corrélation augmente de la même manière entre les figures 2.7a et 2.7c. Les arbres obtenus pour un faible nombre d'individus sont plus corrélés avec les autres lorsque la corrélation intra-blocs augmente.

Taille des blocs l Entre les figures 2.6c et 2.6d, la taille des blocs l de la matrice de variance-covariance a augmentée de 5 à 10, réduisant ainsi le nombre total de groupes. Dans ce cas, le nombre d'individus nécessaire pour une partition plus stable s'en trouve réduit. De la même manière pour le coefficient de corrélation cophénétique (figures 2.7c et 2.7d), l'augmentation de la taille des blocs (et donc la réduction de leur nombre) augmente l'adéquation entre des arbres obtenus avec un nombre d'individus différent.

Comparaison des poids

Les poids obtenus dans le cadre 1 ont été montrés sur la figure 2.3. Nous nous intéressons au nombre de groupes du niveau minimum dans ce même cadre. Sur la figure 2.8, on peut noter qu'avec suffisamment d'individus, le poids représentant la qualité du niveau retrouve bien le bon nombre de groupes défini par la structure en blocs de la matrice de variance-covariance. Ce nombre minimal est d'autant plus grand que la corrélation au sein des blocs est faible. Avec des groupes plus grands, le nombre d'individus nécessaire à l'obtention du bon niveau diminue.

En conclusion, quand le nombre d'individus varie, les poids restent sensiblement les mêmes. Trop peu d'individus peut entraîner le choix d'un niveau non optimal en matière de nombre de groupes. Plus la corrélation est forte, moins il est nécessaire d'avoir d'individus pour que les poids retranscrivent la vérité. Plus les groupes sont grands, moins il est nécessaire d'avoir d'individus.



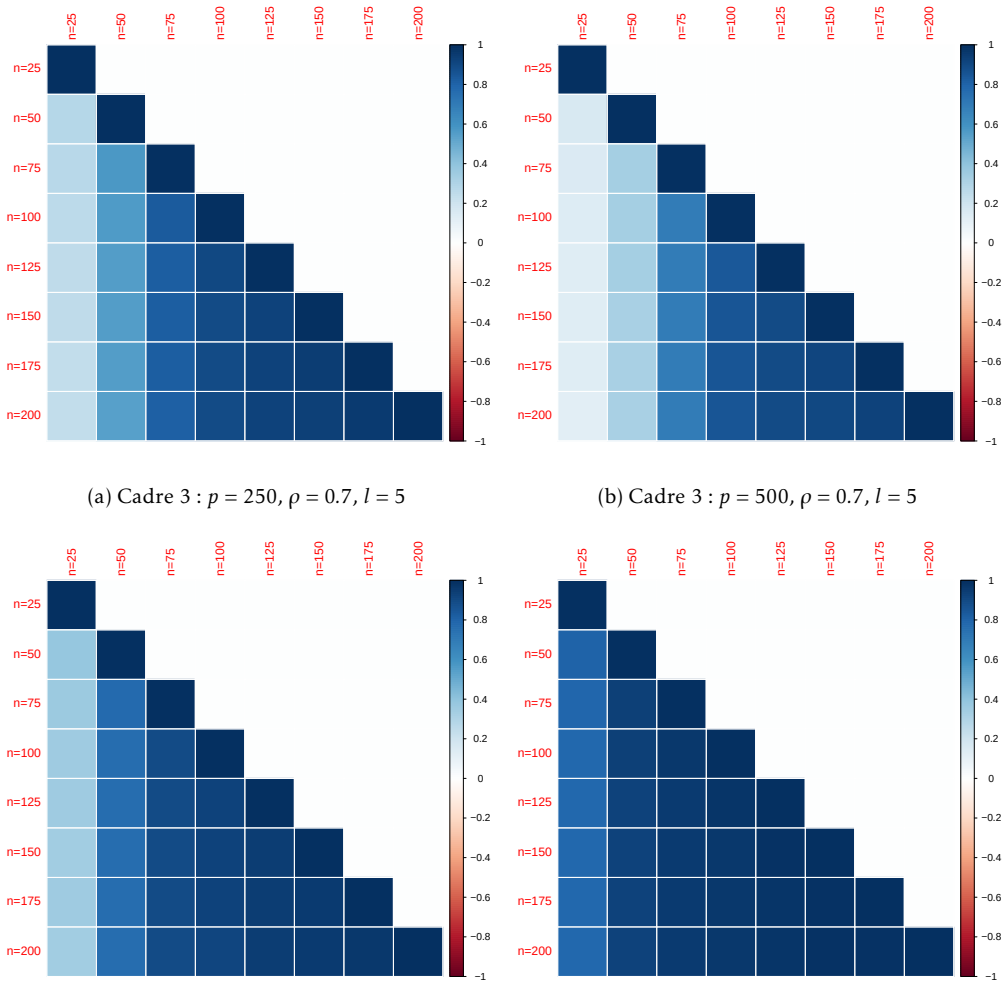
(a) Cadre 3 : $p = 250$, $\rho = 0.7$, $l = 5$

(b) Cadre 3 : $p = 500$, $\rho = 0.7$, $l = 5$

(c) Cadre 3 : $p = 250$, $\rho = 0.9$, $l = 5$

(d) Cadre 3 : $p = 250$, $\rho = 0.9$, $l = 10$

FIGURE 2.6 – Indice de Rand ajusté entre les partitions obtenues par CAH pour un nombre différent d'individus. Entre la figure 2.6a et 2.6b, le nombre de variables a augmenté de 250 à 500. Entre la figure 2.6a et 2.6c, la corrélation intra-blocs a augmentée de 0.7 à 0.9. Entre la figure 2.6c et 2.6d, la taille des blocs a augmentée de 5 à 10.



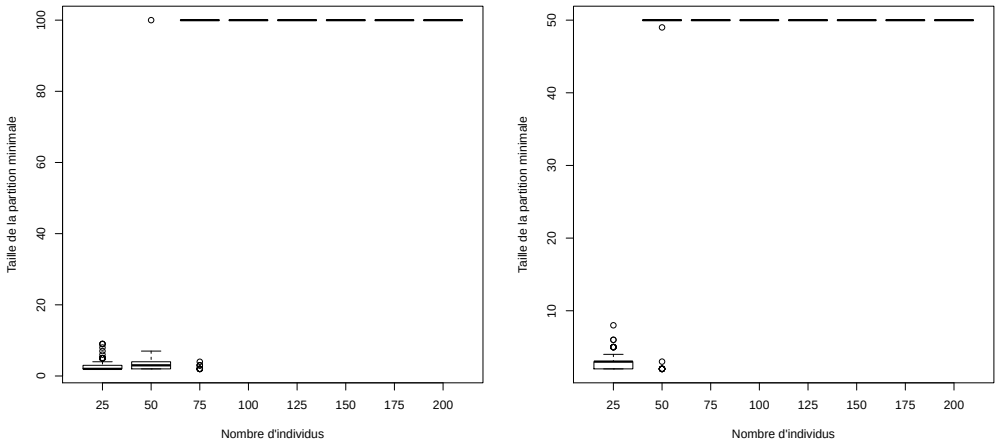
(a) Cadre 3 : $p = 250$, $\rho = 0.7$, $l = 5$

(b) Cadre 3 : $p = 500$, $\rho = 0.7$, $l = 5$

(c) Cadre 3 : $p = 250$, $\rho = 0.9$, $l = 5$

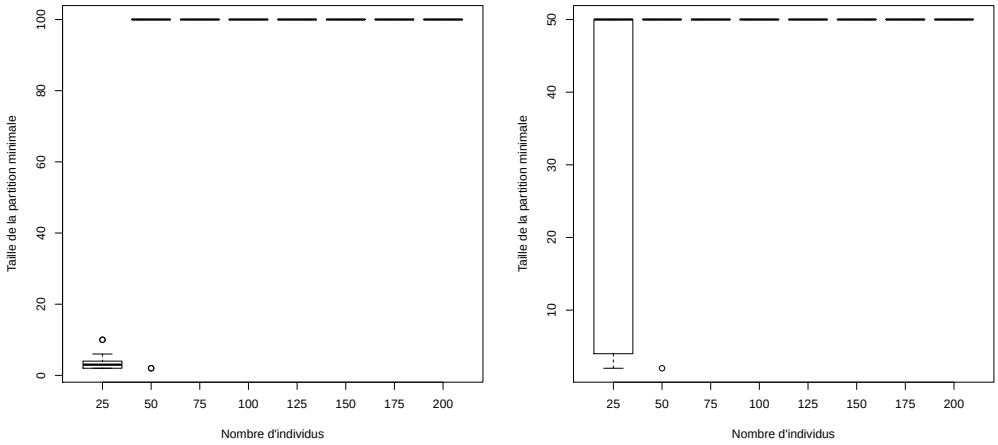
(d) Cadre 3 : $p = 250$, $\rho = 0.9$, $l = 10$

FIGURE 2.7 – Coefficient de corrélation entre les matrices de distance cophénétique de deux arbres de classification en fonction du nombre d'individus n dans le cadre 3 de simulation. Entre la figure 2.7a et 2.7b, le nombre de variables a augmenté de 250 à 500. Entre la figure 2.7a et 2.7c, la corrélation intra-blocs a augmentée de 0.7 à 0.9. Entre la figure 2.7c et 2.7d, la taille des blocs a augmentée de 5 à 10.



(a) Cadre 1 : $p = 500$, $\rho = 0.5$, $l = 5$

(b) Cadre 1 : $p = 500$, $\rho = 0.5$, $l = 10$



(c) Cadre 1 : $p = 500$, $\rho = 0.7$, $l = 5$

(d) Cadre 1 : $p = 500$, $\rho = 0.7$, $l = 10$

FIGURE 2.8 – Taille de la partition de poids ρ_s minimum dans le cadre de simulation 1. De gauche à droite, la taille des blocs augmente de 5 à 10. De haut en bas, la corrélation intra-blocs augmente de 0.5 à 0.7. Les boîtes à moustaches sont réalisées avec 100 répétitions.

2.2.3 Sensibilité du *Multi-Layer Group-Lasso* aux paramètres des cadres de simulations

Le comportement de notre méthode est étudié lorsque les différents paramètres des cadres de simulations varient. Dans les différents cadres de simulations, on prend la valeur $\beta_j^* = 1$ pour une variable d'indice j appartenant au support de β^*. Plusieurs variables du support de β^* ne peuvent se trouver dans un même bloc de corrélation défini par la structure de la matrice de variance-covariance de X. La variance du bruit est quant à elle fixée pour avoir un ratio signal sur bruit ($(\beta^*)^T X^T X \beta^* / \|\epsilon\|_2^2$) de 2. Le critère utilisé est le nombre de vrais positifs en fonction du nombre de faux positifs sur l'ensemble du chemin solution du *Multi-Layer Group-Lasso*. Nous considérons comme vrai positif un groupe sélectionné contenant une variable du support de la vraie solution β^* et des variables corrélées avec celle-ci. Si un groupe sélectionné contient deux variables du support n'ayant aucun lien de corrélation, il est considéré comme un faux positif car ces variables auraient dû être sélectionnées dans deux groupes distincts. Les résultats sont montrés sur les figures 2.9 et 2.10 dans le cadre 1 et 3 de simulation. Les résultats sont similaires dans les autres cadres de simulations.

Nombre d'individus n Sur la figure 2.9, on constate que plus le nombre d'individus n augmente, plus la qualité du chemin s'améliore. En effet, le nombre maximal de vrais positifs augmente jusqu'à atteindre le maximum K. Dans le même temps, le nombre de faux positifs nécessaire pour atteindre ce nombre de vrais positifs diminue.

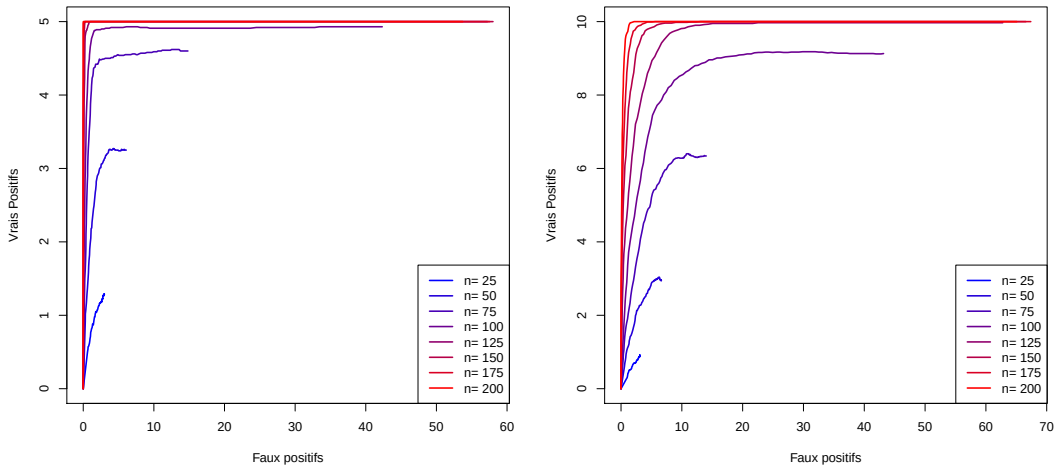
Nombre de vrais groupes K Entre les figures 2.9a et 2.9b le nombre de vrais groupes K augmente de 5 à 10. On voit alors qu'il est nécessaire d'avoir plus d'individus pour retrouver l'ensemble des vrais groupes. Pour un nombre d'individus faible ($n = 25$ ou 50), le fait d'avoir plus de vrais groupes ne permet pas d'améliorer le chemin solution qui semble équivalent. Ce constat fait penser au théorème de Wainwright pour le lasso (cf. Section 1.1.1) qui dit que sous de bonnes conditions et avec suffisamment d'individus, le lasso retrouve la totalité des vraies variables pour une certaine valeur de λ ou également que plus il y a de vraies variables, plus il est nécessaire d'avoir d'individus pour les retrouver toutes.

Taille des blocs l Sur la figure 2.10b, en augmentant la taille des blocs de 5 à 10 et donc en réduisant le nombre de groupes de 100 à 50, la qualité du chemin solution s'améliore. On peut conjecturer que le rôle joué par le nombre de variables dans le théorème de Wainwright est ici occupé par le nombre de groupes et donc réduire leur nombre facilite la tâche de sélection.

Corrélation intra-blocs ρ Sur la figure 2.10, on peut voir que plus la corrélation intra-blocs ρ est élevée plus la qualité du chemin solution est élevée avec un nombre de vrais positifs atteint pour un nombre plus faible de faux positifs.

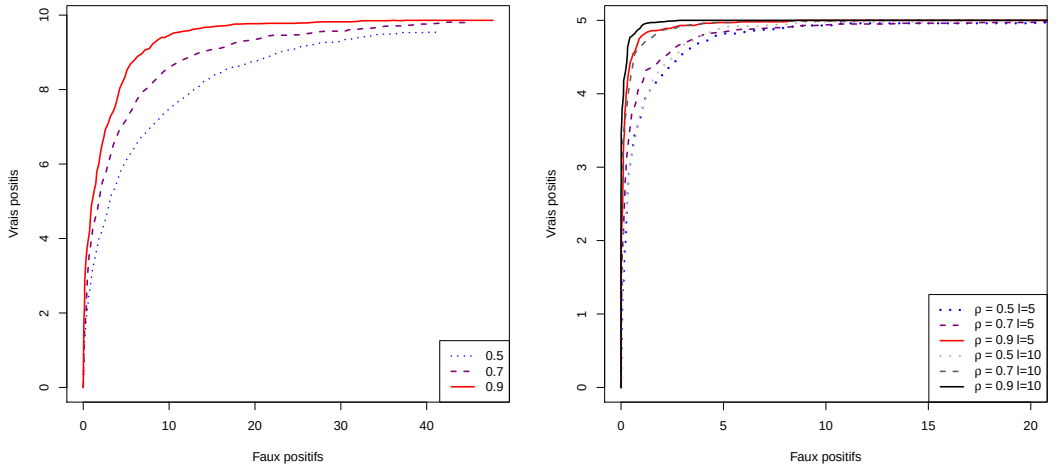
2.2.4 Comparaison du *Multi-Layer Group-Lasso* avec le group-lasso

Nous allons comparer le *Multi-Layer Group-Lasso* avec le group-lasso sur un seul niveau. La partition est choisie en utilisant la règle du saut maximal sur laquelle est basée le calcul des poids ρ_s (cf. Section 2.1.4). Comme précédemment, le critère utilisé est le nombre de vrais positifs en fonction du nombre de faux positifs sur l'ensemble du chemin solution du *Multi-Layer Group-Lasso*. Nous considérons comme vrai positif un groupe sélectionné contenant une variable du support de la vraie solution β^* et des variables corrélées avec celle-ci. Si un groupe sélectionné



(a) $p = 500$, $\rho = 0.7$, $l = 5$, $K = 5$ (b) $p = 500$, $\rho = 0.7$, $l = 5$, $K = 10$

FIGURE 2.9 – Nombre de vrais positifs en fonction du nombre de faux positifs pour le chemin solution du *Multi-Layer Group-Lasso* en fonction du nombre d'individus n dans le cadre 3 de simulation. De gauche à droite, le nombre de vrais groupes augmente de 5 à 10. Les courbes sont la moyenne de 100 répétitions.



(a) $n = 100$, $p = 500$, $l = 5$, $K = 10$ (b) $n = 100$, $p = 500$, $K = 10$

FIGURE 2.10 – Nombre de vrais positifs en fonction du nombre de faux positifs pour le chemin solution du *Multi-Layer Group-Lasso* en fonction de la corrélation ρ au sein des blocs de la matrice de variance-covariance dans le cadre 1 de simulation. À gauche, le cas $l = 5$, à droite le cas $l = 10$ y est superposé pour mieux comparer. Les courbes sont la moyenne de 100 répétitions.

contient deux variables du support n'ayant aucun lien de corrélation, il est considéré comme un faux positif car ces variables auraient dû être sélectionnées dans deux groupes distincts.

Sur la figure 2.11 est représenté le nombre de vrais positifs en fonction du nombre de faux positifs pour l'ensemble du chemin solution pour le *Multi-Layer Group-Lasso* et le group-lasso sur le meilleur niveau selon la règle du saut maximal dans le cadre 2 de simulation. Sur les graphes présentés, notre méthode obtient de meilleures performances que le group-lasso sur un seul niveau. En effet, le nombre de vrais positifs pour le *Multi-Layer Group-Lasso* sur l'ensemble du chemin est supérieur à celui du group-lasso sur une unique partition, le tout avec un nombre plus faible de faux positifs. L'écart entre les deux courbes pour les deux méthodes varie beaucoup selon les cas. Pour comprendre cet écart, intéressons-nous à la figure 2.12 qui représente l'histogramme du niveau de la partition sélectionnée par le critère du saut maximal. C'est la partition associée à ce niveau qui est utilisée dans le group-lasso. Dans le cas où $p = 250$, le nombre de groupes de la partition optimale définie par la structure de blocs de la matrice de variance-covariance est de 50, alors que pour $p = 500$ il est de 100. On constate que le niveau choisi par le critère du saut maximal n'est pas toujours celui associé à la partition optimale en matière de nombre de groupes. Des niveaux associés à des partitions en un nombre très faible de groupes sont parfois sélectionnés. Ce sont ces différences de niveaux qui créent l'écart plus ou moins important entre les deux méthodes. On note par exemple que dans le cas $\rho = 0.7$ (figures 2.12b et 2.12d), peu de partitions avec un nombre de groupes non optimal sont sélectionnées, cela résulte en un faible écart entre les deux méthodes en matière de qualité de la sélection (figures 2.11b et 2.11d). Pour les figures 2.12a et 2.12c, le phénomène inverse se produit et résulte en un écart important de la qualité du chemin solution entre les deux méthodes (figures 2.11a et 2.11c). Dans le cas où une partition est dominante, notre méthode et le group-lasso sur une seule partition fournissent les mêmes résultats (figures 2.13 et 2.14).

La méthode proposée obtient de meilleurs résultats que le group-lasso quand le critère de choix du nombre de groupes de la partition ne permet pas de distinguer clairement une partition dominante. Dans ce cas, même si la partition avec le meilleur critère a très peu de groupes (figure 2.11), notre méthode réussit à sélectionner les groupes d'intérêts de taille raisonnable.

2.2.5 Comparaison du *Multi-Layer Group-Lasso* avec les méthodes présentées en section 1.2.2

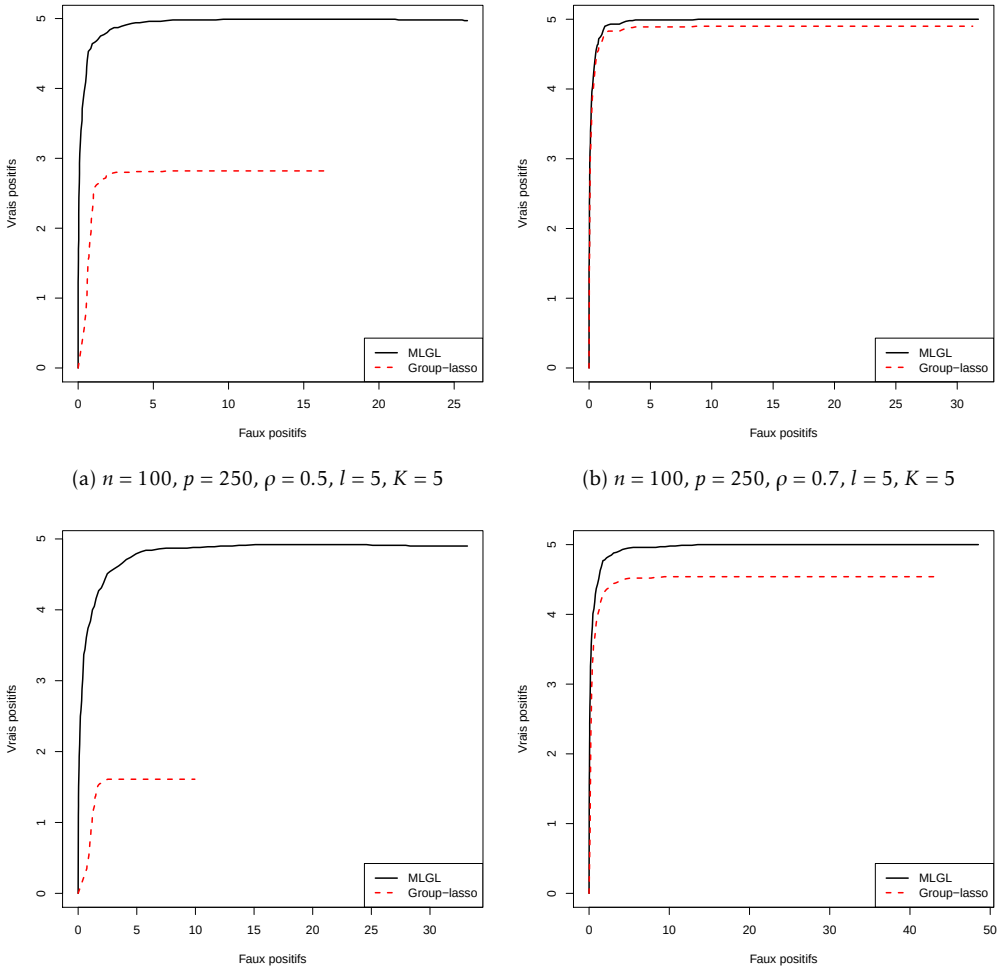
Le *Multi-Layer Group-Lasso* va maintenant être comparé aux méthodes juxtaposant classification et sélection présentées dans la Section 1.2.2. Les méthodes sont comparées par la qualité du chemin solution, c'est-à-dire le nombre de vrais positifs en fonction du nombre de faux positifs sur l'ensemble du chemin solution des différentes méthodes.

Méthodes comparées

Pour la méthode *Hierarchical Clustering and Averaging for Regression* (HCAR) (cf. Algorithme 6), la classification ascendante hiérarchique (CAH) est faite avec la distance euclidienne et le critère de Ward. Comme nous souhaitons comparer les méthodes par la qualité du chemin solution, le chemin solution du niveau du $\hat{\lambda}$ choisi par validation croisée est retourné.

Pour le *Supervised Group-Lasso* (SGL) (cf. Algorithme 7), la méthode *k-means* est remplacée par une CAH avec la distance euclidienne et le critère de Ward. La partition optimale est sélectionnée par la règle du saut maximal et non par la gap statistic.

Pour le *Cluster Representative Lasso* (CRL) et le Cluster Group-Lasso (CGL) (cf. Algorithme 8), la CAH est faite avec la distance euclidienne et le critère de Ward. La partition optimale est sélectionnée par la règle du saut maximal.



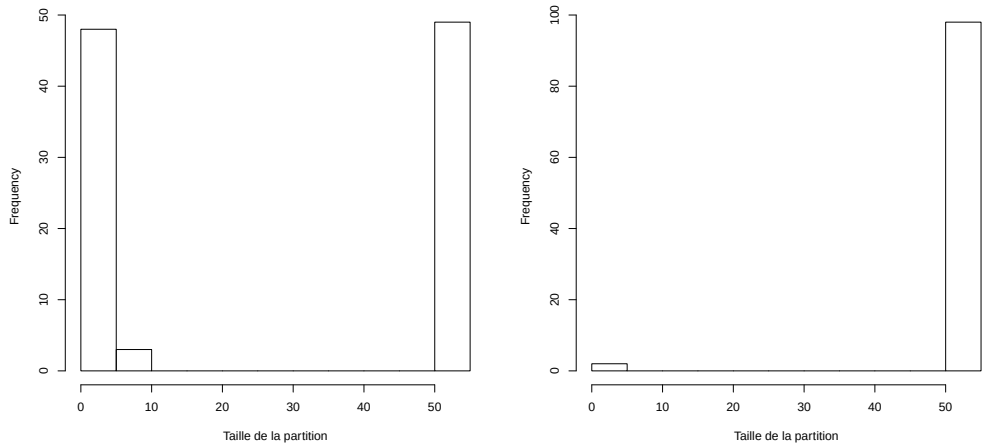
(a) $n = 100$, $p = 250$, $\rho = 0.5$, $l = 5$, $K = 5$

(b) $n = 100$, $p = 250$, $\rho = 0.7$, $l = 5$, $K = 5$

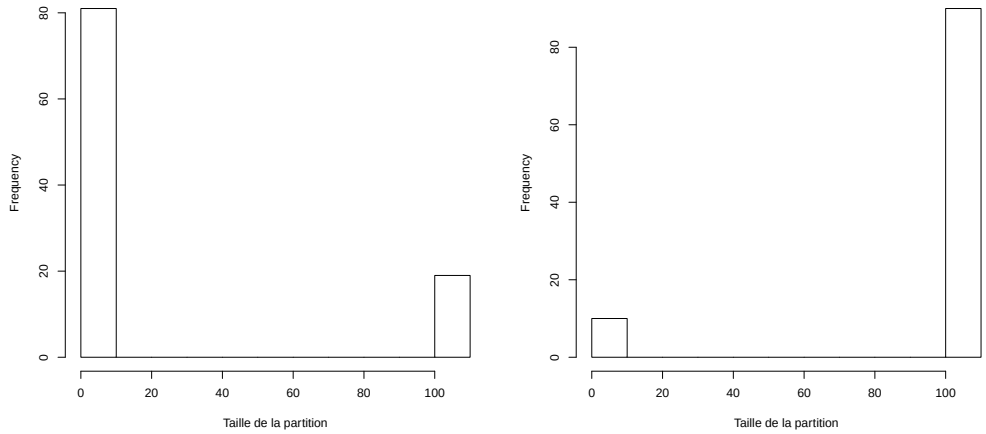
(c) $n = 100$, $p = 500$, $\rho = 0.5$, $l = 5$, $K = 5$

(d) $n = 100$, $p = 500$, $\rho = 0.7$, $l = 5$, $K = 5$

FIGURE 2.11 – Nombre de vrais positifs en fonction du nombre de faux positifs pour le chemin solution du *Multi-Layer Group-Lasso* (trait plein noir) et du group-lasso (pointillés rouges) dans le cadre 2 de simulation. De gauche à droite, la corrélation intra-blocs augmente de 0.5 à 0.7. De haut en bas, le nombre de variables augmente de 250 à 500. Les courbes sont la moyenne de 100 répétitions.

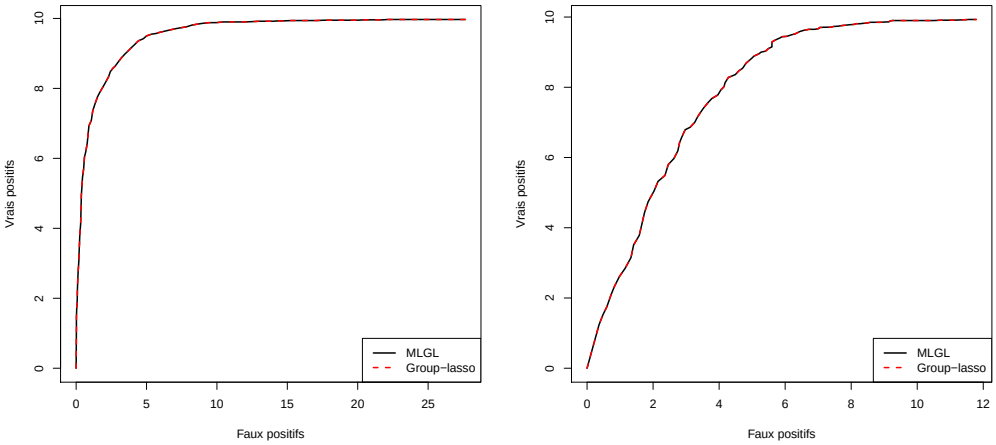


(a) $n = 100$, $p = 250$, $\rho = 0.5$, $l = 5$, $K = 5$ (b) $n = 100$, $p = 250$, $\rho = 0.7$, $l = 5$, $K = 5$



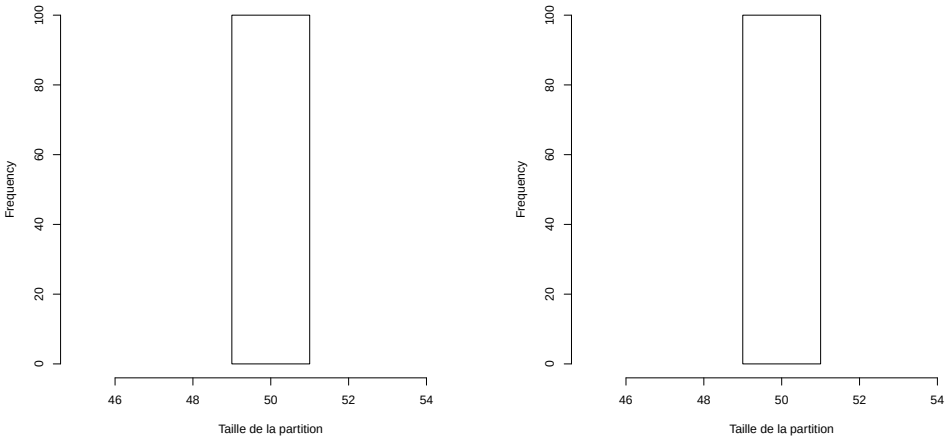
(c) $n = 100$, $p = 500$, $\rho = 0.5$, $l = 5$, $K = 5$ (d) $n = 100$, $p = 500$, $\rho = 0.7$, $l = 5$, $K = 5$

FIGURE 2.12 – Taille (en nombre de groupes) de la partition sélectionnée par le critère de saut maximal dans le cadre 2 de simulation. De gauche à droite, la corrélation intra-blocs augmente de 0.5 à 0.7. De haut en bas, le nombre de variables augmente de 250 à 500.



(a) Cadre 1 : $p = 500$, $\rho = 0.9$, $l = 10$, $K = 10$ (b) Cadre 4 : $p = 500$, $\rho = 0.9$, $\rho_2 = 0.3$, $l = 10$, $K = 10$

FIGURE 2.13 – Nombre de vrais positifs en fonction du nombre de faux positifs pour le chemin solution du *Multi-Layer Group-Lasso* (trait plein noir), du group-lasso (pointillés rouges). De gauche à droite, le cadre de simulation change, cela correspond à l'ajout d'une corrélation inter-blocs ρ_2. Les courbes sont la moyenne de 100 répétitions.



(a) Cadre 1 : $p = 500$, $\rho = 0.7$, $l = 10$, $K = 10$ (b) Cadre 4 : $p = 500$, $\rho = 0.9$, $\rho_2 = 0.3$, $l = 10$, $K = 10$

FIGURE 2.14 – Taille (en nombre de groupes) du niveau sélectionné par le critère de saut maximal. De gauche à droite, le cadre de simulation change, cela correspond à l'ajout d'une corrélation inter-blocs ρ_2.

Ce choix a été fait pour que les méthodes comparées utilisent la même CAH et la même règle pour former les partitions.

Résultats

Sur la figure 2.15, le chemin solution est représenté dans le cadre 1 de simulation pour différentes valeurs de paramètres. Dans l'ensemble des quatre graphes, le *Multi-Layer Group-Lasso* fait partie des meilleures méthodes avec un nombre de vrais positifs élevé pour un nombre de faux positifs parmi les plus faibles. Le *Cluster Representative Lasso* et le *Supervised Group-Lasso* ont des résultats similaires au *Multi-Layer Group-Lasso*.

Corrélation intra-blocs ρ Lorsque la corrélation intra-blocs ρ augmente de 0.5 à 0.9 entre les figures 2.15a à 2.15b, les performances de la méthode *HCAR* diminuent assez fortement. Le nombre de vrais positifs diminue par rapport au cas $\rho = 0.5$ et aux autres méthodes. Les performances du *Supervised Group-Lasso* diminuent également. Les trois autres méthodes voient leurs performance augmenter, le nombre de vrais positifs atteint la valeur de K pour un nombre moindre de faux positifs. Dans ce cas, le *Multi-Layer Group-Lasso* et le *Cluster Representative Lasso* sont les meilleurs.

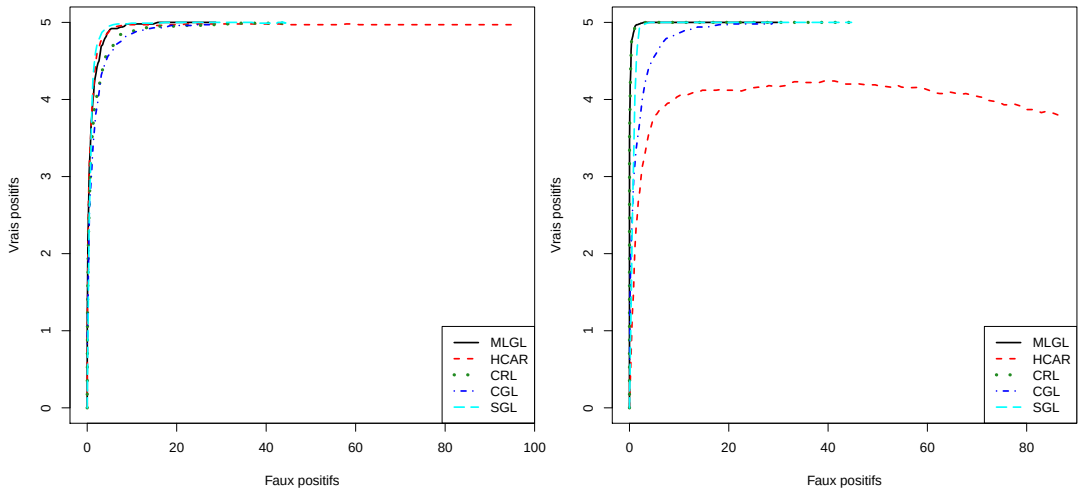
Nombre K de variables du support Entre les figures 2.15b et 2.15c, le nombre de vraies variables passe de 5 à 10. On constate que les mêmes trois méthodes obtiennent les meilleurs résultats (MLGL, CRL, SGL). Cependant le nombre maximal de vrais positifs est plus difficilement atteint (plus de faux positifs sont nécessaires).

Taille l des blocs Quand la taille des blocs diminue de 10 à 5 entre les figures 2.15b et 2.15d, les méthodes voient leurs résultat devenir très proches. Le *Supervised Group-Lasso* garde des performances similaires entre les deux cas.

Ajout de corrélation inter-blocs Entre les figures 2.16a (respectivement 2.16b) et 2.15b (respectivement 2.15d), de la corrélation inter-blocs a été rajoutée. Le *Multi-Layer Group-Lasso* et le *Cluster Representative Lasso* gardent de bonnes performances lorsque de l'ajout de corrélation intra-blocs est effectuée. À l'inverse, le *Supervised Group-Lasso* voit ses performances diminuer par rapport aux deux méthodes citées précédemment. Un plus grand nombre de faux positifs est nécessaire pour atteindre la valeur maximale K de vrais positifs.

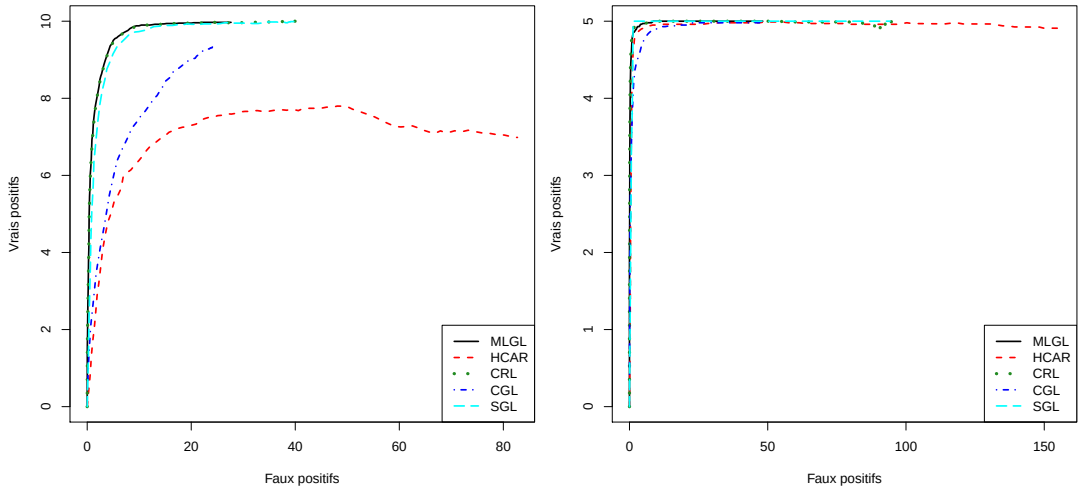
Présence de variables isolées Dans le cadre de simulation 3 (présence de variables isolées), les résultats dans des cas de faibles corrélations sont plus mitigés. Sur la figure 2.17, le *Multi-Layer Group-Lasso* a des performances plus faibles en présence de faibles corrélations intra-blocs et de groupes de plus faibles tailles. Cependant, les autres méthodes (à l'exception de HCAR) voient leurs performance se dégrader de manière plus marquée que le *Multi-Layer Group-Lasso* qui conserve très peu de faux positifs.

Groupes de taille non uniforme Sur la figure 2.18, les résultats sont présentés en présence de groupes de taille non uniforme (cadre 2 de simulation). On retrouve les mêmes conclusions que dans le cadre 1 de simulation mais le *Multi-Layer Group-Lasso* est généralement meilleur avec un nombre de vrais positifs maximal atteint plus rapidement.



(a) Cadre 1 : $n = 50$, $p = 500$, $K = 5$, $\rho = 0.5$, $l = 10$

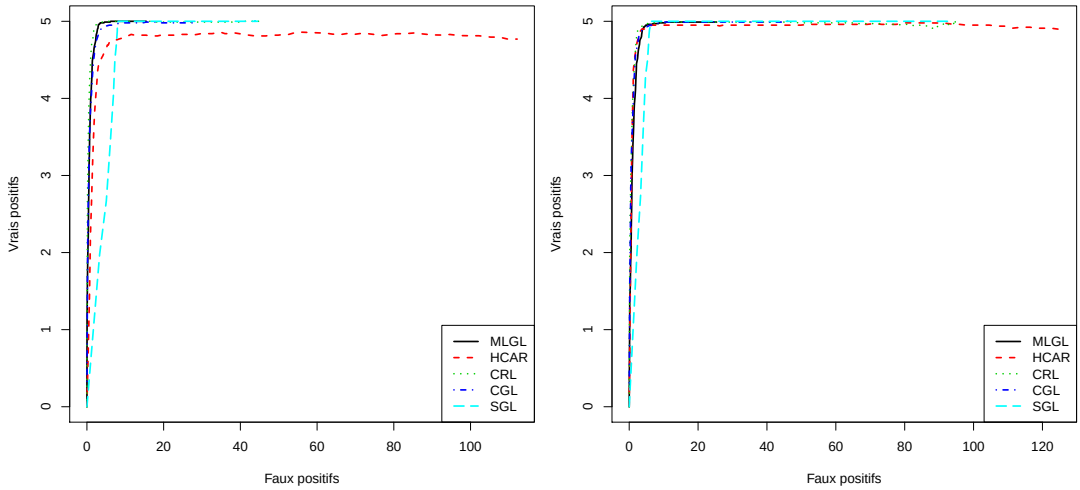
(b) Cadre 1 : $n = 50$, $p = 500$, $K = 5$, $\rho = 0.9$, $l = 10$



(c) Cadre 1 : $n = 50$, $p = 500$, $K = 10$, $\rho = 0.9$, $l = 10$

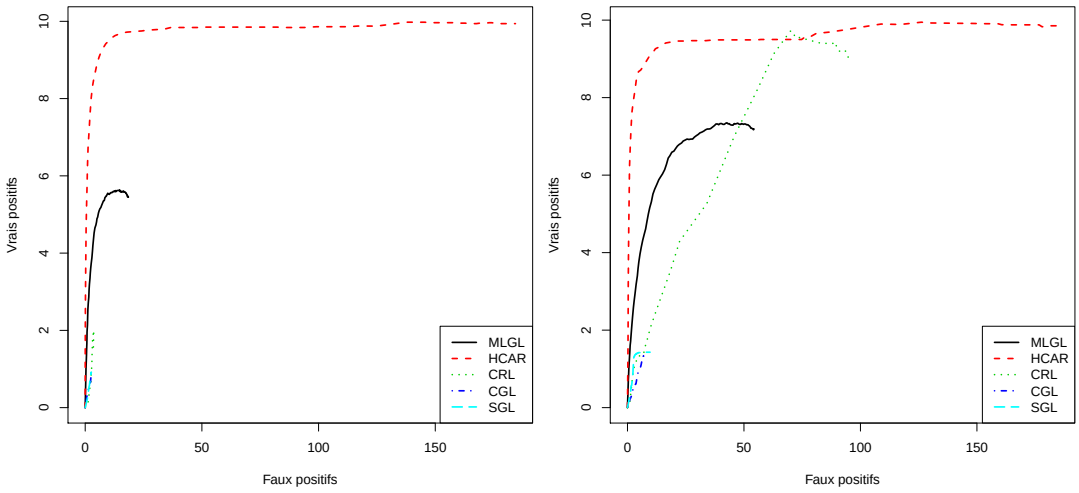
(d) Cadre 1 : $n = 50$, $p = 500$, $K = 5$, $\rho = 0.9$, $l = 5$

FIGURE 2.15 – Nombre de vrais positifs en fonction du nombre de faux positifs pour le chemin solution du *Multi-Layer Group-Lasso* (MLGL, noir), du *Hierarchical Clustering and Averaging for Regression* (HCAR, rouge), du *Cluster Representative Lasso* (CRL, vert), du Cluster Group-Lasso (CGL, bleu foncé) et *Supervised Group-Lasso* (SGL, bleu clair) dans le cadre 1 de simulation. Les courbes sont la moyenne de 100 répétitions. Entre les figures 2.15a et 2.15b, la corrélation intra-blocs augmente de 0.5 à 0.9. Entre les figures 2.15b et 2.15c, le nombre de vraie variables K augmente de 5 à 10. Entre les figures 2.15b et 2.15d, la taille des blocs diminue de 10 à 5 et donc le nombre total de blocs augmente.



(a) Cadre 4 : $n = 50$, $p = 500$, $K = 5$, $\rho = 0.9$, $l = 10$ (b) Cadre 4 : $n = 50$, $p = 500$, $K = 5$, $\rho = 0.9$, $l = 5$

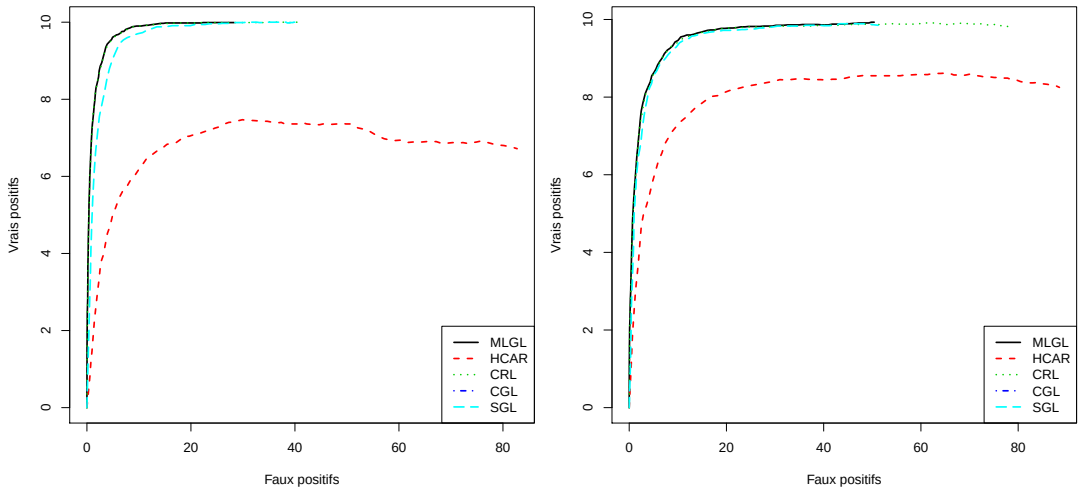
FIGURE 2.16 – Nombre de vrais positifs en fonction du nombre de faux positifs pour le chemin solution du *Multi-Layer Group-Lasso* (MLGL, noir), du *Hierarchical Clustering and Averaging for Regression* (HCAR, rouge), du *Cluster Representative Lasso* (CRL, vert), du Cluster Group-Lasso (CGL, bleu foncé) et *Supervised Group-Lasso* (SGL, bleu clair) dans le cadre 1 de simulation. Les courbes sont la moyenne de 100 répétitions. Entre les figures 2.16a et 2.16b, la taille des blocs diminue de 10 à 5 et donc le nombre total de blocs augmente. Entre les figures 2.16a (respectivement 2.16b) et 2.15b (respectivement 2.15d), de la corrélation inter-blocs a été rajoutée.



(a) Cadre 3 : $n = 50$, $p = 500$, $K = 5$, $\rho = 0.5$, $l = 5$

(b) Cadre 3 : $n = 50$, $p = 500$, $K = 5$, $\rho = 0.7$, $l = 5$

FIGURE 2.17 – Nombre de vrais positifs en fonction du nombre de faux positifs pour le chemin solution du *Multi-Layer Group-Lasso* (MLGL, noir), du *Hierarchical Clustering and Averaging for Regression* (HCAR, rouge), du *Cluster Representative Lasso* (CRL, vert), du Cluster Group-Lasso (CGL, bleu foncé) et *Supervised Group-Lasso* (SGL, bleu clair) dans le cadre 1 de simulation. Les courbes sont la moyenne de 100 répétitions. De gauche à droite, la corrélation intra-blocs ρ augmente de 0.5 à 0.7.



(a) Cadre 2 : $n = 50$, $p = 500$, $K = 10$, $\rho = 0.9$, $l = 10$

(b) Cadre 2 : $n = 50$, $p = 500$, $K = 10$, $\rho = 0.9$, $l = 5$

FIGURE 2.18 – Nombre de vrais positifs en fonction du nombre de faux positifs pour le chemin solution du *Multi-Layer Group-Lasso* (MLGL, noir), du *Hierarchical Clustering and Averaging for Regression* (HCAR, rouge), du *Cluster Representative Lasso* (CRL, vert), du Cluster Group-Lasso (CGL, bleu foncé) et *Supervised Group-Lasso* (SGL, bleu clair) dans le cadre 1 de simulation. Les courbes sont la moyenne de 100 répétitions. De gauche à droite, la taille moyenne des groupes l passe de 10 à 5.

2.3 Données réelles

Les données concernent la production de riboflavine par la bactérie *Bacillus subtilis*. Elles comportent une matrice contenant le niveau d'expression normalisé de 4088 gènes pour 71 échantillons et un vecteur réponse contenant la log-transformée du taux de production de riboflavine pour chaque individu. Ces 71 individus sont issus de 28 souches de bactéries mesurées à différents temps (2 à 6 mesures par souche). Dans la suite, on appelle ce jeu de données *riboflavin*.

Le contexte expérimental est présenté dans (Lee et al. 2001 ; Zamboni et al. 2005). Les données ont été utilisées comme application dans (Bühlmann, Kalisch et Meier 2014), mais les résultats n'y sont pas analysés. Dans (Bühlmann, Rütimann et al. 2013), la matrice d'expression a servi dans un cadre d'application pseudo-réel : une variable réponse est générée à partir de la matrice d'expression des gènes et un vecteur de coefficients β^*.

Un deuxième jeu est présenté (que l'on appelle *riboflavingrouped*) contenant l'expression des 4088 gènes pour 111 individus, chacun représentant la mesure d'une souche à un temps donné. 66 échantillons de ce jeu de données font partie du jeu de données précédent.

Parmi les 4088 gènes, une liste de 8 gènes sont connus comme ayant un lien biologique avec la production de riboflavine.

2.3.1 Comparaison du *Multi-Layer Group-Lasso* dans le cadre de l'étude de Bühlmann, Kalisch et Meier (2014)

Nous allons appliquer notre méthode sur les deux jeux de données *riboflavin* et *riboflavingrouped* et comparer les résultats obtenus avec les méthodes utilisées dans (Bühlmann, Kalisch et Meier 2014) : le lasso (avec validation croisée *10-fold*), la *stability selection* (avec un seuil de 0.5) et des régressions linéaires simples avec test de nullité et correction des tests multiples de Bonferonni-Holm (Holm 1979). Le seuil de 0.5 a été choisi car la probabilité de sélection maximale obtenue pour le jeu de données *riboflavin* est de 0.58.

Comparaison des résultats pour *riboflavin* et *riboflavingrouped* Sur la figure 2.19 est représentée l'intersection des sélections du lasso et de la *stability selection*. On note d'assez grandes différences entre les résultats des deux jeux de données. Pour le lasso, on dénote 4 variables communes sur une sélection de 16 et 30 variables pour *riboflavin* et *riboflavingrouped*. Pour la *stability selection*, deux sont en communs. Pour les régressions linéaires simples, 53 variables sont sélectionnées pour le jeu de données *riboflavin* et plus de 1000 pour le second jeu de données mais uniquement 35 en communs.

En utilisant la CAH avec average linkage, le *Multi-Layer Group-Lasso* sélectionne au plus 20 groupes contenant une seule variable dans le cas du jeu de données *riboflavin*. Pour le jeu *riboflavingrouped*, un groupe de taille 4 est sélectionné suivi de 11 variables. 4 variables sont communes aux deux sélections.

Quelles que soient les méthodes utilisées, de grandes différences existent entre les variables sélectionnées sur le jeu de données *riboflavin* et sur *riboflavingrouped*.

Gènes liés à la production de riboflavine Les gènes associés biologiquement à la production de riboflavine ne sont curieusement que très peu corrélés au sein des jeux de données avec la réponse (cf. Tableau 2.2). Ces gènes ne se retrouvent pas parmi les gènes sélectionnés par notre méthode ni par les autres sur les deux jeux de données.

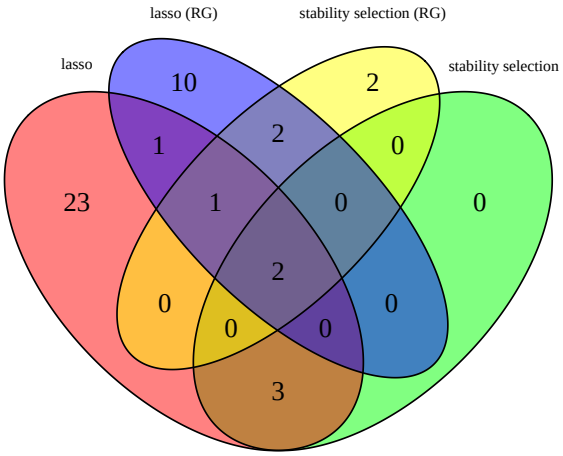
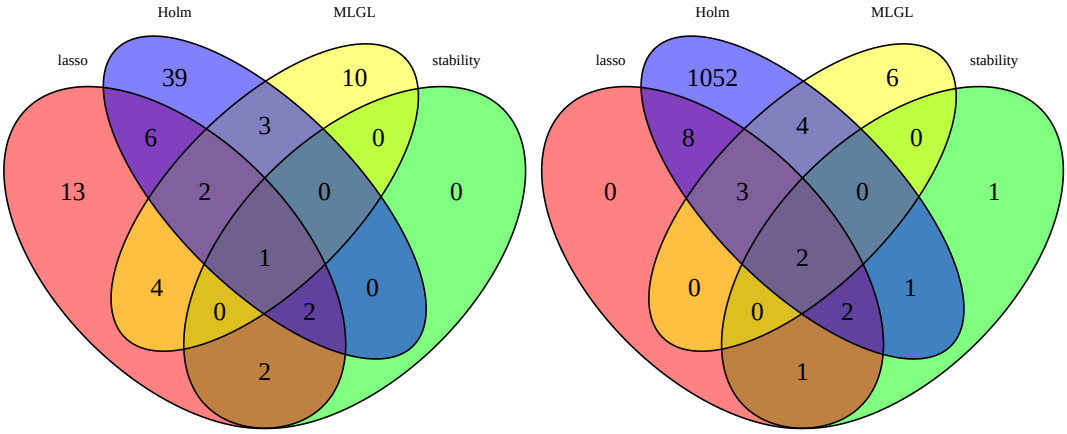


FIGURE 2.19 – Diagramme de Venn pour comparer les variables sélectionnées par lasso et *stability selection* et validation croisée pour les jeux de données *riboflavin* et *riboflavingrouped*. Sur le diagramme, *RG* indique les résultats pour le jeu *riboflavingrouped*.

TABLEAU 2.2 – Corrélation des gènes liés à la production de riboflavine avec la réponse dans les jeux de données *riboflavin* et *riboflavingrouped*.

Gène	ribA	ribB	ribC	ribG	ribH	ribR	ribT	ypaA
Corr. *riboflavin*	0.26	0.25	0.24	0.28	0.20	0.11	-0.25	0.22
Corr. *riboflavingrouped*	-0.23	-0.29	-0.27	-0.34	-0.31	0.41	-0.55	0.52



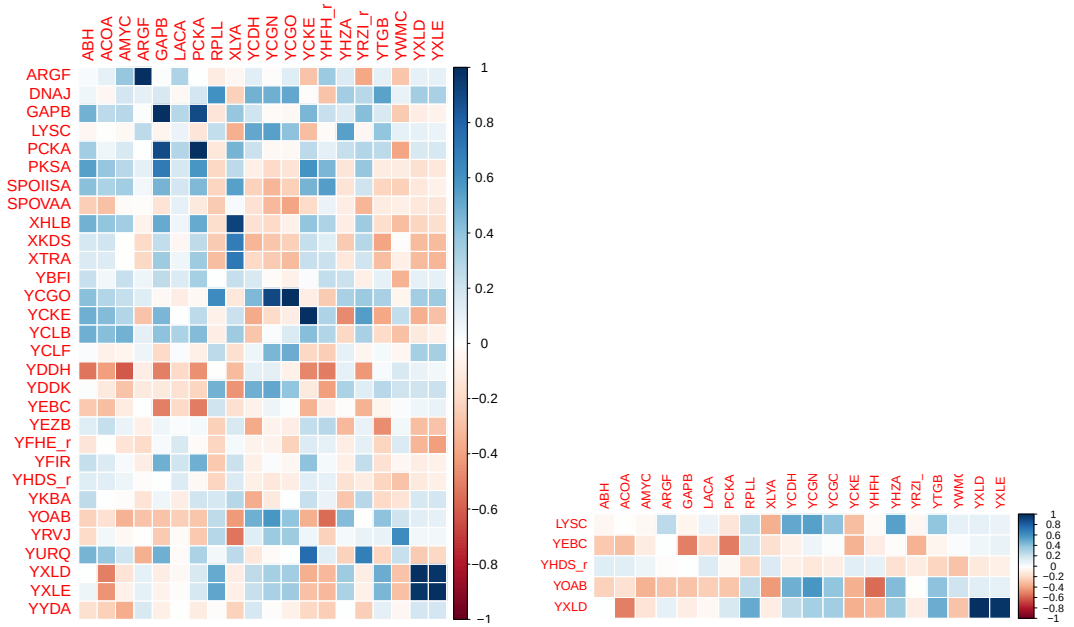
(a) jeu de données *riboflavin* (b) jeu de données *riboflavingrouped*

FIGURE 2.20 – Comparaison de la sélection du *Multi-Layer Group-Lasso* (MLGL) avec le lasso, la *stability selection* et la régression linéaire simple avec test de nullité (Holm) pour le jeu de données *riboflavin* (à gauche) et *riboflavingrouped* (à droite).

Comparaison du *Multi-Layer Group-Lasso* avec les autres méthodes Sept gènes sélectionnés par le lasso le sont aussi par notre méthode pour *riboflavin* dont un seul est présent dans la *stability selection* (cf. figure 2.20). Dans le cas de *riboflavingrouped*, on retrouve 5 éléments communs avec le lasso et 2 avec la *stability selection*. Un certain nombre de gènes ne sont sélectionnés que par notre méthode. La régression linéaire avec test de nullité et correction de Bonferonni-Holm sélectionne un nombre très important de gènes dans le cas de *riboflavingrouped*. Cette méthode simple n'est pas adaptée dans ce cas.

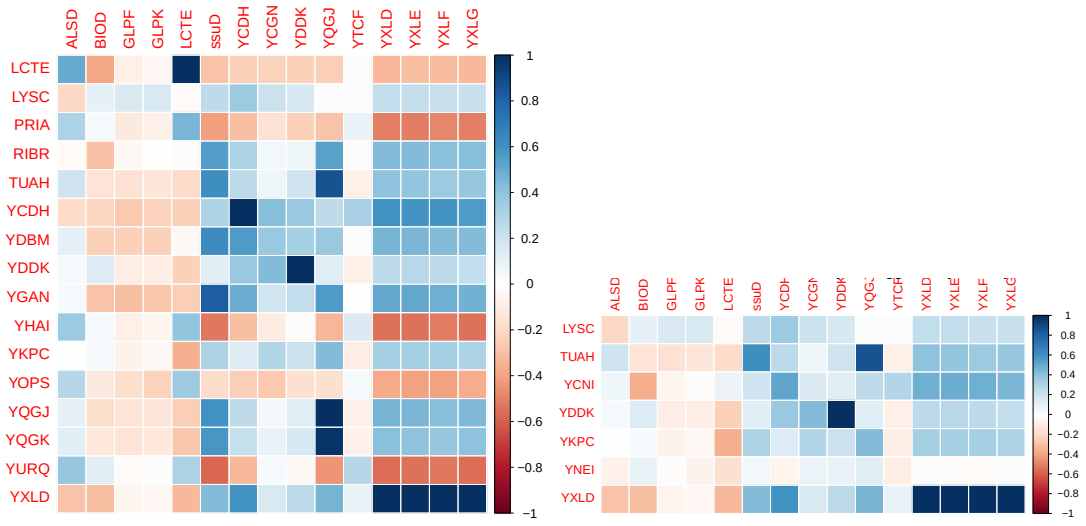
Sur les figures 2.21 et 2.22 est représentée la corrélation entre les gènes sélectionnés par le *Multi-Layer Group-Lasso* et par d'autres méthodes. Pour le jeu de données *riboflavin*, le gène commun à la *stability selection* et à notre méthode (gène YXLD) est corrélé avec un autre gène de la sélection. Hormis celui-ci, aucune autre forte corrélation est à noter entre notre sélection et celle de la *stability selection*. Un seul gène de notre sélection (YCGN) est corrélée avec un gène de la sélection lasso. Pour *riboflavingrouped*, on voit qu'un gène sélectionné par la *stability selection* (YXLD) est corrélé avec trois autres gènes sélectionnés par notre méthode (ces gènes sont sélectionnés en tant que groupes). On retrouve ce même effet avec la sélection lasso ainsi que le gène YQGK corrélé à un autre gène sélectionné par le lasso et également par notre méthode.

Conclusion Dans cet exemple, notre méthode sélectionne donc principalement des variables seules. Entre les deux jeux de données, on retrouve un ensemble commun de 4 variables. Les gènes liés à la production de riboflavine ne sont pas retrouvés dans la sélection des différentes méthodes et ne présentent pas de corrélation avec les gènes de la sélection. Le gène YXLD est sélectionné par l'ensemble des méthodes sur les deux jeux de données. Avec notre méthode, différents gènes corrélés à celui-ci sont également sélectionnés. Notre méthode sélectionne des gènes également sélectionnés par les autres méthodes. L'appartenance des gènes sélectionnés à des mêmes catégories fonctionnelles a été étudiée mais aucune catégorie ne ressort significativement. Les résultats sont difficiles à analyser car les gènes importants d'un point de vue



(a) lasso versus *Multi-Layer Group-Lasso* (b) *stability selection* versus *Multi-Layer Group-Lasso*

FIGURE 2.21 – Corrélation des variables sélectionnées par le *Multi-Layer Group-Lasso* avec celles sélectionnées par le lasso (à gauche) et la *stability selection* (à droite) pour le jeu de données *riboflavin*. Les variables en colonnes représentent les variables sélectionnées par notre méthode.



(a) lasso versus *Multi-Layer Group-Lasso* (b) *stability selection* versus *Multi-Layer Group-Lasso*

FIGURE 2.22 – Corrélation des variables sélectionnées par le *Multi-Layer Group-Lasso* avec celles sélectionnées par le lasso (à gauche) et la *stability selection* (à droite) pour le jeu de données *riboflavingrouped*. Les variables en colonnes représentent les variables sélectionnées par notre méthode.

biologique ne le sont pas mathématiquement. Dans la section suivante, nous allons utiliser ce jeu de données dans un cadre pseudo-réel en générant une réponse.

2.3.2 Exemple pseudo-réel

Nous allons appliquer l'exemple présenté dans (Bühlmann, Rütimann et al. 2013) avec les données *riboflavin*. Dans cet article, seules les 1000 variables avec la plus forte variance sont conservées. La vraie solution β^* et la réponse y sont générées.

$S = \mathbb{S}(\beta^*)$ le support de β^* est composé de 10 variables choisies comme suit : une variable est tirée au hasard parmi l'ensemble des variables. Cette variable et les 9 qui lui sont le plus corrélées sont conservées. On pose $\beta^*_S = 1$. La réponse y est générée par $y = X\beta^* + \epsilon$, avec $\epsilon \sim \mathcal{N}(0_n, \sigma^2 I_n)$ avec $\sigma \in \{1, 3, 12\}$.

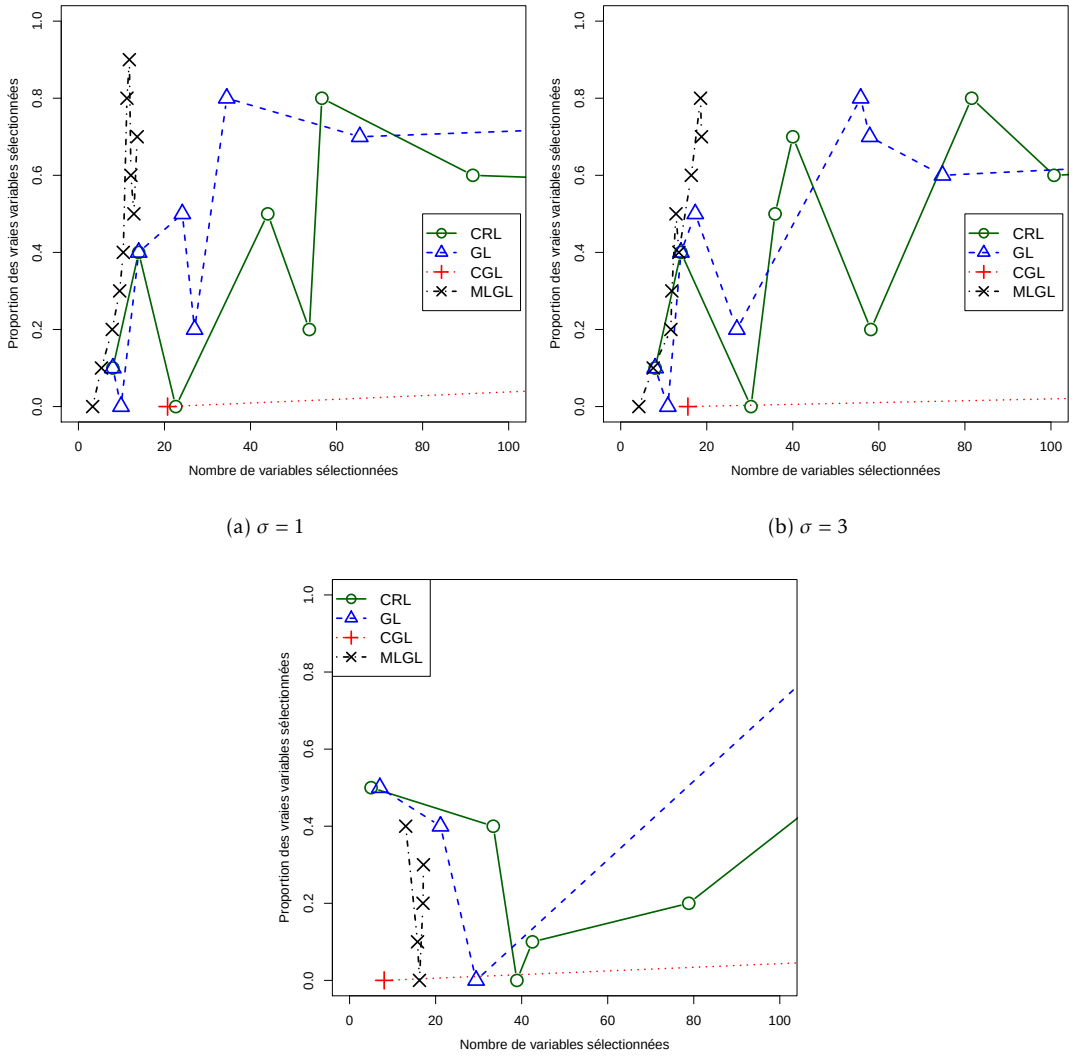
La classification ascendante hiérarchique (CAH) des variables est faite en utilisant une dissimilarité basée sur les corrélations : $\mathrm{d}(X_i, X_j) = \sqrt{2n}\sqrt{1 - |\mathrm{cor}(X_i, X_j)|}$ et le critère average linkage.

Le *Multi-Layer Group-Lasso* est comparé au Cluster Group-Lasso, *Cluster Representative Lasso* et au Group-Lasso. La partition utilisée est celle choisie par règle du saut maximal. Les résultats sont représentés par la proportion de vraies variables sélectionnées en fonction du nombre de variables sélectionnées.

Sur la figure 2.23, on remarque que, dans les cas $\sigma = 1$ et $\sigma = 3$, notre méthode sélectionne une forte proportion des vraies variables (0.8-0.9) tout en sélectionnant un nombre restreint de variables. Les autres méthodes arrivent à sélectionner l'ensemble des vraies variables mais en sélectionnant la plupart des variables. Quand les méthodes sélectionnent un faible nombre de variables, le *Multi-Layer Group-Lasso* est la meilleure en matière de vraies variables sélectionnées.

2.4 Conclusion

Notre objectif dans ce chapitre était de sélectionner des variables en présence de corrélation. Nous avons proposé une méthode, le *Multi-Layer Group-Lasso*, combinant classification ascendante hiérarchique et group-lasso. Elle prend en compte l'ensemble des niveaux de la CAH, alors que les méthodes usuelles (group-lasso, cluster group-lasso,...) n'en utilisent qu'un seul. Dans ces méthodes, un mauvais choix de niveau peut résulter en de mauvaises performances en sélection. Les résultats obtenus par simulation montrent l'efficacité de notre méthode par rapport à l'utilisation d'une seule partition pour le group-lasso. Le *Multi-Layer Group-Lasso* fournit un chemin solution avec des performances équivalentes aux autres méthodes et supérieures dans certains cas. Toutefois toutes ces méthodes sélectionnent parfois un nombre important de faux groupes avant de sélectionner l'ensemble des vrais groupes. Séparer les faux positifs des vrais positifs dans le chemin solution du *Multi-Layer Group-Lasso* est donc nécessaire pour optimiser les performances en sélection.



(a) $\sigma = 1$

(b) $\sigma = 3$

(c) $\sigma = 12$

FIGURE 2.23 – Proportion des vraies variables sélectionnées en fonction de la taille de la sélection de 4 méthodes : *Cluster Representative Lasso* (CRL, en vert), Group-Lasso (GL, en bleu), Cluster Group-Lasso (CGL, en rouge) et *Multi-Layer Group-Lasso* (MLGL, en noir).

Chapitre 3

Sélection de groupes de variables à taux d'erreurs contrôlé

Le Chapitre 2 introduit une procédure de régularisation basée sur des groupes de variables obtenus par classification ascendante hiérarchique (CAH). Cette procédure, appelée *Multi-Layer Group-Lasso*, fournit un chemin solution dépendant d'un paramètre de régularisation λ. Dans ce chemin, il convient de séparer les groupes de variables influentes des autres groupes. Dans le cas du lasso, différentes procédures ont été proposées dans ce sens. On notera notamment (Wasserman et Roeder 2009) qui propose une procédure testant l'importance des variables sélectionnées et (Meinshausen et Yu 2009) qui propose une étape de seuillage sur les coefficients estimés. De telles procédures ne sont pas applicables en l'état pour le *Multi-Layer Group-Lasso*, notamment car plusieurs variables représentent un groupe. De plus, le chemin solution du *Multi-Layer Group-Lasso* affiche une particularité par rapport au group-lasso classique : pour une même valeur de λ, plusieurs groupes sélectionnés peuvent contenir des variables en commun. L'objectif de ce chapitre est de développer une procédure prenant en compte les particularités du *Multi-Layer Group-Lasso* et permettant de contrôler un taux d'erreur de type Family-Wise Error Rate (FWER) ou False Discovery Rate (FDR).

3.1 Test statistique

3.1.1 Test d'importance de variables

Dans la suite nous énonçons différents tests pour tester l'importance d'un ensemble de variables dans le cadre de la régression linéaire.

Test des modèles emboîtés

Le test des modèles emboîtés permet de tester si un ensemble de variables peut être éliminé du modèle de régression sans perdre trop d'information. Ce test se place dans un cadre gaussien et ne peut être applicable qu'en faible dimension ($p < n$).

Pour cela, on suppose que l'on dispose d'un modèle complet :

$$\text{complet} : y = \beta_0 + X_Q\beta_Q + X_P\beta_P + \epsilon$$

et d'un modèle réduit :

$$\text{réduit} : y = \beta_0 + X_Q\beta_Q + \epsilon$$

avec P et Q deux ensembles de variables.

L'hypothèse nulle H_0 et alternative H_1 du test sont :

$$\begin{aligned} H_0 &: \beta_P = 0_{|P|} \\ H_1 &: \exists k \in P,\ \beta_k \neq 0. \end{aligned}$$

On note SCR(réduit) (respectivement SCR(complet)), la somme des carrés des résidus définie par $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ avec $\hat{y}_i$ la valeur prédite de y_i à partir du modèle réduit (respectivement complet). La statistique de test est :

$$F = \frac{\frac{\text{SCR(réduit)}-\text{SCR(complet)}}{\text{df(réduit)}-\text{df(complet)}}}{\frac{\text{SCR(complet)}}{\text{df(complet)}}}$$

avec df(réduit) (respectivement df(complet)) le nombre de degrés de liberté du modèle réduit (respectivement complet).

Sous H_0, la statistique de test F suit une loi de Fisher à df(réduit) − df(complet) et df(complet) degrés de liberté.

Score test

Tout comme le test des modèles emboîtés, le score test (Goeman, S. A. Van De Geer et Van Houwelingen 2006) permet de tester l'importance des variables au sein d'un modèle de régression linéaire.

L'hypothèse nulle H_0 et alternative H_1 du test sont :

$$\begin{aligned} H_0 &: \beta = 0 \\ H_1 &: \exists k,\ \beta_k \neq 0. \end{aligned}$$

Le test consiste en une approche Bayésienne. Une loi a priori de β est choisie telle que son espérance soit nulle et sa matrice de variance-covariance de la forme $\tau^2\Sigma$ avec Σ une matrice semi-définie positive et τ^2 un hyperparamètre.

La statistique de test S est égale à $\frac{y^T X\Sigma X^T y}{y^T y}$. La loi de S sous l'hypothèse nulle n'est pas connue mais elle peut être approximée (Goeman, S. A. Van De Geer et Van Houwelingen 2006).

Tout comme le test des modèles emboîtés, il est possible de prendre en compte un modèle réduit et un modèle complet afin de tester la nullité d'un sous-ensemble des coefficients (Goeman, Houwelingen et Finos 2011). Le modèle complet est :

$$\text{complet} : y = \beta_0 + X_Q\beta_Q + X_P\beta_P + \epsilon$$

et le modèle réduit :

$$\text{réduit} : y = \beta_0 + X_Q\beta_Q + \epsilon$$

avec P et Q deux ensembles de variables. Le cardinal de Q doit être strictement inférieur au nombre d'individus n.

L'hypothèse nulle H_0 et alternative H_1 du test sont :

$$\begin{aligned} H_0 &: \beta_P = 0_{|P|} \\ H_1 &: \exists k \in P,\ \beta_k \neq 0. \end{aligned}$$

Dans ce cas, la statistique de test est :

$$S = \frac{(y - X_Q\beta_Q)X_P X_P^T (y - X_Q\beta_Q)}{n\sigma^2}$$

où σ^2 est la variance du bruit.

Une fois de plus la distribution sous H_0 n'est pas connue et doit être approximée (Goeman, Houwelingen et Finos 2011). Un des avantages du score test par rapport au test des modèles emboîtées est que la dimension total $|P| + |Q|$ peut excéder le nombre d'individus n.

3.1.2 Correction des tests multiples

On considère maintenant qu'un nombre m de tests est effectué. Dans le tableau de contingence suivant, on définit différentes grandeurs résultant de la réalisation de ces m tests.

	H_0 **est vraie**	H_1 **est vraie**	**Total**
rejet de H_0	V	S	R
non-rejet de H_0	U	T	$m - R$
Total	m_0	$m - m_0$	m

Lorsque m tests sont effectués et chacun de ces tests est contrôlé à un niveau α, la probabilité de rejeter au moins une hypothèse nulle devient assez forte. Avec un niveau de contrôle α, on a sous hypothèse d'indépendance :

$$\begin{aligned} \mathbb{P}(V \geq 1) &= 1 - \mathbb{P}(V = 0) \\ &= 1 - (1 - \alpha)^m. \end{aligned}$$

Par exemple, dans le cas où $\alpha = 0.05$ et $m = 10$, la probabilité de rejeter au moins une fois l'hypothèse nulle est d'environ 0.40, assez éloignée du niveau α appliqué à chaque test. Une correction des p-valeurs calculées devient nécessaire pour garder un contrôle global à un niveau raisonnable.

Correction de Bonferroni

La correction de Bonferroni (Dunn 1959) permet de contrôler le Family-Wise Error Rate (FWER) qui est la probabilité de rejeter à tort au moins une hypothèse nulle parmi l'ensemble des tests effectués : $\mathbb{P}(V \geq 1)$.

Pour avoir un contrôle du FWER au niveau α, chaque test doit être contrôlé au niveau α/m, ou de manière équivalente chaque p-valeur est multipliée par m puis contrôlée au niveau α. Ainsi, chaque p-valeur p_i, $i = 1, \ldots, m$ est ajustée à :

$$p_i^{\text{adj}} = p_i \times m$$

puis comparée à α pour déterminer si l'hypothèse nulle $H_{0,i}$ associée est rejetée ou non.

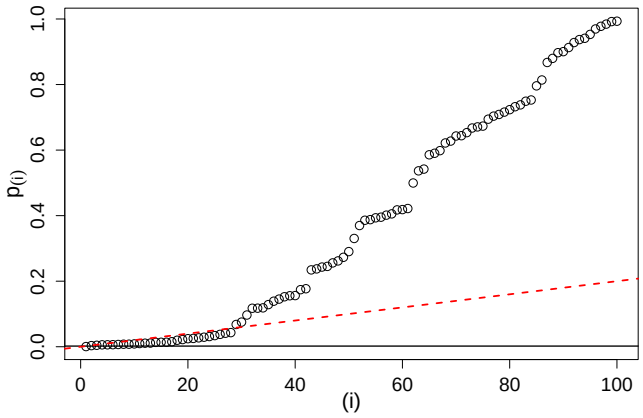


FIGURE 3.1 – Comparaison des seuils des corrections de Bonferroni (trait plein noir) et Benjamini-Hochberg (pointillés rouges) pour $\alpha = 0.2$.

La correction de Bonferroni est réputée conservative lorsque le nombre de tests effectués est grand et en présence de dépendance dans les données (BLAND et ALTMAN 1995).

Correction de Benjamini-Hochberg

La correction de Benjamini-Hochberg (BENJAMINI et HOCHBERG 1995) contrôle le False Discovery Rate (FDR) qui est l'espérance de la proportion de rejet à tort parmi l'ensemble des rejets : $\mathbb{E}[\frac{V}{\max(1,R)}]$.

Soient $p_{(1)},\dots,p_{(m)}$ les m p-valeurs ordonnées dans l'ordre croissant associées aux hypothèses nulles $H_{0,(1)},\dots,H_{0,(m)}$. Pour un contrôle du FDR au niveau q, le principe est de trouver le plus grand indice i tel que $p_{(i)} \leq \frac{i}{m}q$. Les hypothèses nulles $H_{0,(1)},\dots,H_{0,(i)}$ sont ensuite rejetées.

Sur la figure 3.1, 100 tests sont effectués (dont 80 vraies hypothèses nulles). On peut voir qu'en appliquant la correction de Bonferroni aucune hypothèse n'est rejetée alors qu'avec la correction de Benjamini-Hochberg 28 hypothèses nulles sont rejetées (dont 8 vraies). Le seuil après correction de Bonferroni est ici trop restrictif (5×10^{-4}) et ne permet pas de rejeter les fausses hypothèses nulles.

3.1.3 Test hiérarchique

Dans la procédure de test développée, nous souhaitons prendre en compte la spécificité du chemin solution du *Multi-Layer Group-Lasso* : des groupes issus de différents niveaux de la CAH et donc potentiellement chevauchant peuvent être sélectionnés. Cependant, nous souhaitons que les groupes dont l'hypothèse nulle est rejetée soient tous d'intersections deux à deux vides. Nous présentons dans la suite, deux méthodes de test hiérarchique, une qui contrôle le FWER et une le FDR, permettant de traiter ces particularités.

Contrôle du FWER

Dans (Meinshausen 2008), une méthode de test hiérarchique est proposée afin de tester l'importance des variables pour la prédiction dans le cadre de la régression linéaire. Elle permet un contrôle du FWER. Le but est de tester la nullité de certains coefficients en les considérant en groupes pour capter des phénomènes de corrélation entre variables associées notamment. L'idée est alors de tester successivement l'importance de groupes de variables imbriqués les uns dans les autres obtenus par le biais d'une classification ascendante hiérarchique (CAH). Le nombre total de tests effectués est défini par l'arbre hiérarchique.

Modèle

On rappelle le modèle de régression linéaire :

$$y = X\beta + \epsilon$$

où X est une matrice de taille $n \times p$. On se place dans le cadre gaussien, $\epsilon \sim \mathcal{N}(0_n, \sigma^2 I_n)$.

L'hypothèse nulle du test de l'importance d'un groupe $G \subset \{1, \dots, p\}$ est

$$H_{0,G} \ : \ \forall k \in G, \ \beta_k = 0$$

contre l'hypothèse alternative

$$H_{1,G} \ : \ \exists k \in G, \ \beta_k \neq 0.$$

Ces hypothèses seront testées à l'aide des tests présentés en Section 3.1.1.

Vocabulaire des arbres hiérarchiques

Dans un premier temps, définissons le vocabulaire des arbres hiérarchiques décrivant une structure hiérarchique entre les variables. On note $\mathcal{T}$ un arbre hiérarchique. On appelle *racine* de l'arbre, le groupe contenant l'ensemble des variables de l'arbre. Une *feuille* est un groupe ne contenant aucun autre groupe, l'ensemble des feuilles est noté $\mathcal{L}$. Le cardinal de l'arbre $\mathcal{T}$ est défini comme le nombre de feuilles de l'arbre. Le *parent* d'un groupe G noté $\text{pa}(G)$ est le plus petit groupe contenant G, les *ancêtres* de G sont l'ensemble des groupes contenant G, $\text{an}(G) = \{C \in \mathcal{T} \mid G \subsetneq C\}$, on a donc $\text{pa}(G) \in \text{an}(G)$. Les *enfants* (*children*) de G sont les groupes descendants directement de G, $\text{ch}(G) = \{C \in \mathcal{T} \mid C \subsetneq G, \nexists D \in \mathcal{T}, C \subsetneq D \subsetneq G\}$. Les *descendants* (*offsprings*) de G sont les groupes inclus dans G, $\text{off}(G) = \{C \in \mathcal{T} \mid C \subsetneq G\}$, on a clairement $\text{ch}(G) \subset \text{off}(G)$. Enfin, les frères et sœurs (*siblings*) de G sont les autres enfants du parent de G, $\text{si}(G) = \{C \in \mathcal{T} \mid C \in \text{ch}(\text{pa}(G)), C \neq G\}$.

Procédure de test

Une CAH sur l'ensemble des variables est effectuée, un arbre $\mathcal{T}$ est ainsi obtenu. Cet arbre va définir la succession des différents tests à effectuer. Certains tests ne seront effectués qu'à condition que l'hypothèse nulle d'un test précédent soit rejetée.

La procédure commence par tester la racine de l'arbre $\mathcal{T}$ que l'on note G_1. Si l'hypothèse nulle H_{0,G_1} est rejetée au niveau α, ses enfants sont testés et ainsi de suite. Les p-valeurs des tests sont ajustées pour prendre en compte la succession et l'imbrication des tests. La procédure s'arrête lorsque plus aucune hypothèse nulle ne peut être rejetée. Le but est ici de trouver les plus petits groupes de variables ayant un effet significatif sur la réponse.

Ajustement des p-valeurs

Afin d'assurer le contrôle du FWER à un niveau global α, deux ajustements des p-valeurs sont ainsi faits (Meinshausen 2008).

Soit G le groupe dont l'hypothèse $H_{0,G}$ est testée et p_G la p-valeur du test en résultant.

Ajustement pour la multiplicité des tests Le 1er ajustement pour prendre en compte la multiplicité des tests est :

$$p_G^{\text{adj}} = p_G \frac{p}{|G|}. \tag{3.1}$$

Ce 1er ajustement se fait donc par le ratio entre la taille totale de l'arbre et la taille du groupe associé à l'hypothèse nulle testée. Avec cet ajustement, le groupe à la racine n'est pas ajusté et les feuilles d'un arbre issu de la CAH reçoivent un ajustement de type Bonferroni.

Ajustement pour la hiérarchie Le 2e ajustement consiste à prendre en compte le lien de parenté entre les tests, en imposant que la p-valeur associée à un sous-groupe ne puisse être plus petite que celles de ses ancêtres. L'ajustement est donc défini comme :

$$p_G^{h,\text{adj}} = \max_{D \in \mathcal{T}: G \subseteq D} p_D^{\text{adj}}. \tag{3.2}$$

Ainsi, pour un niveau de contrôle α, aucune hypothèse nulle ne peut être rejetée si l'hypothèse nulle associée au groupe parent n'a pas été rejetée.

Avec ces ajustements, en testant chaque p-valeur à un niveau α, le FWER est contrôlé à ce même niveau sur l'ensemble de l'arbre.

Contrôle du FDR

Une méthode pour contrôler le FDR dans les tests hiérarchiques est proposée dans (Yekutieli 2008).

Soit $\mathcal{T} = \{G_1, \ldots, G_T\}$ un arbre hiérarchique. On note $\mathcal{F}_0$ l'ensemble des groupes n'ayant pas de parents et $\mathcal{F}_t = \{D \in \mathcal{F} \mid \text{pa}(D) = G_t\}$ l'ensemble des groupes dont le parent est G_t.

Pour chaque famille de groupes $\mathcal{F}_t$ ainsi définie, un test est appliqué pour tester l'importance des variables de chaque groupe de la famille. Une correction de Benjamini-Hochberg est appliquée sur les p-valeurs obtenues afin de contrôler le FDR à un niveau q. Si une hypothèse nulle est rejetée, la famille contenant les enfants de cette hypothèse est testée.

Exemple (Illustration du test hiérarchique pour contrôler le FDR) *Sur la figure 3.2 est présentée la hiérarchie et les familles d'hypothèses associées. Un losange représente une hypothèse nulle rejetée à un niveau de contrôle q.*

La procédure de test hiérarchique commence par la famille $\mathcal{F}_0$, elle contient un seul groupe. L'importance de ce groupe est testée à l'aide d'un test des modèles emboîtés par exemple. Une correction des tests multiples est effectuée. Supposons que l'hypothèse nulle associée à $\mathcal{G}_1$ est rejetée, la famille $\mathcal{F}_1$ contenant les enfants de $\mathcal{G}_1$ va donc être testée dans la suite de la même manière que $\mathcal{F}_0$ et ainsi de suite.

Au final, supposons que les hypothèses nulles associées aux groupes $\mathcal{G}_1$, $\mathcal{G}_2$ $\mathcal{G}_3$, $\mathcal{G}_4$, $\mathcal{G}_6$, $\mathcal{G}_{11}$ et $\mathcal{G}_{13}$ ont été rejetées.

Seuls les groupes $\mathcal{G}_4$, $\mathcal{G}_{11}$ et $\mathcal{G}_{13}$ sont des outer nodes, c'est-à-dire qu'ils n'ont pas d'enfants dont l'hypothèse nulle a été rejetée. La famille $\mathcal{F}_5$ n'a pas été testée durant le processus car l'hypothèse nulle de son parent (le groupe $\mathcal{G}_5$) n'a pas été rejetée. □

En utilisant cette stratégie, le FDR est contrôlé au niveau $2q\delta^*$ avec $\delta^* = 1.44$ sur l'ensemble de l'arbre. Si l'on souhaite un contrôle uniquement sur les groupes qui ne sont pas parents d'autres groupes dont l'hypothèse nulle associée est rejetée (appelés *outer nodes*), le FDR est alors contrôlé à un niveau $2Lq\delta^*$ où L est la profondeur de l'arbre hiérarchique (le nombre de niveaux

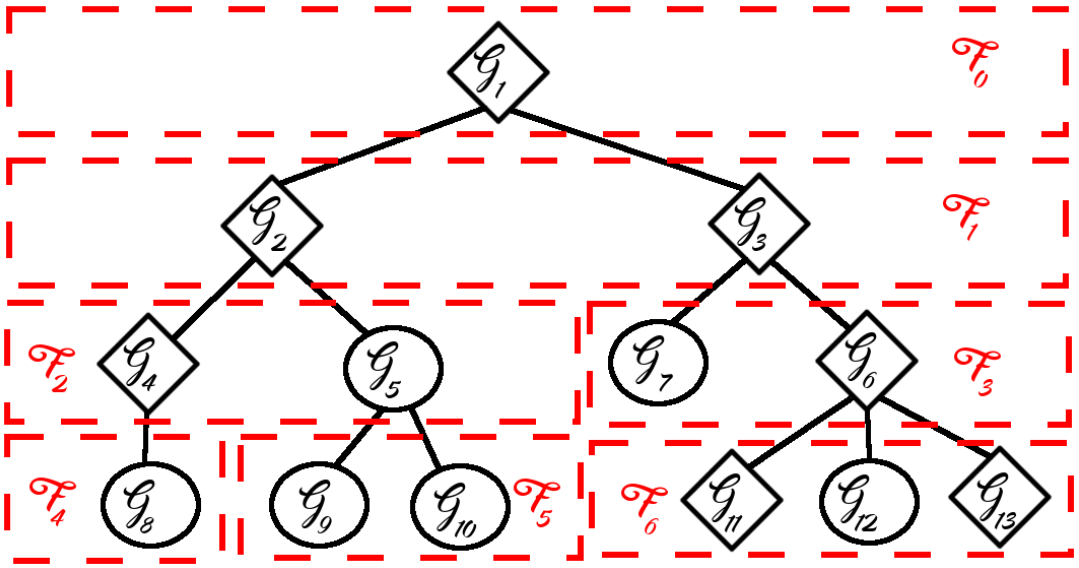


FIGURE 3.2 – Illustration du test hiérarchique contrôlant le FDR. Les losanges correspondent aux groupes dont les hypothèses nulles sont rejetées tandis que les cercles sont celles non rejetées. Les cadres rouges correspondent aux différentes familles de groupes.

de l'arbre). Cette borne paraît grossière. Vouloir contrôler le FDR à un niveau global de 0.05 ou 0.1 va produire une borne d'autant plus conservative que l'arbre est profond (L grand), ce qui se produit pour des arbres hiérarchiques complets avec un grand nombre de variables.

Comparaison des procédures de test hiérarchique contrôlant le FWER et le FDR

Ces deux procédures de test hiérarchique vont être utilisées avec le test des modèles emboîtés et le score test. Le test des modèles emboîtés nous oblige à travailler en faible dimension ($p < n$). Le score test impose quand à lui que le nombre de variables du modèle réduit soit plus petit que n, ce qui nous oblige également à travailler en faible dimension.

Nous comparons les deux approches proposées dans différents cadres de simulation. Les cadres de simulation sont issus de (ZOU et HASTIE 2005 ; MEINSHAUSEN 2008). Dans les cas présentés, l'arbre est obtenu par une CAH avec le critère de Ward et la distance euclidienne.

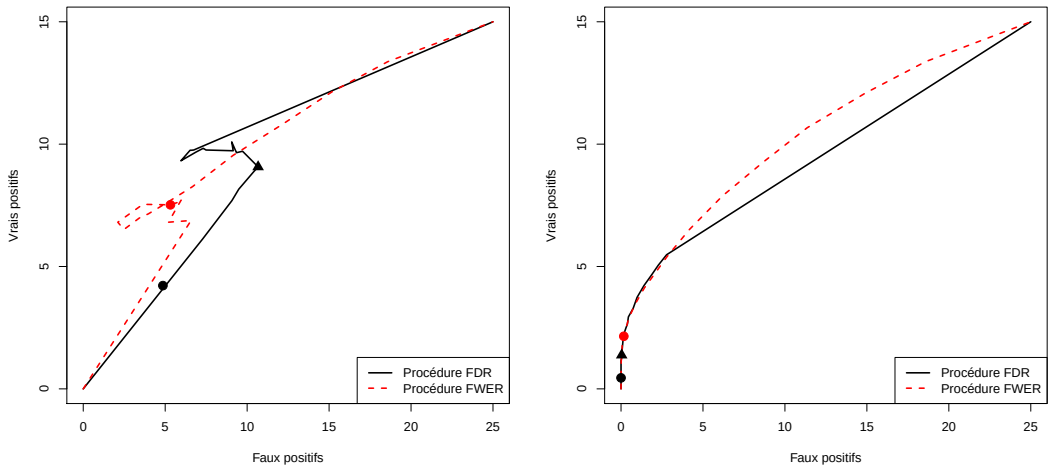
Cadres de simulation

Cadre 1 :

— $n = 50$, $p = 40$;
— $\beta^* = (3,\dots,3,0,\dots,0)$ où le 1er bloc a une longueur de 15 et le second de 25 ;
— $X_k = Z_{j(k)} + \varepsilon$ avec $\varepsilon \sim \mathcal{N}(0_n, I_n)$

$$j(k) = \begin{cases} 1 & \text{si } k \leq 5 \\ 2 & \text{si } 6 \leq k \leq 10 \\ 3 & \text{si } 11 \leq k \leq 15 \\ 4 & \text{si } 15 < k \end{cases}$$

— $Z_j \sim \mathcal{N}(0_n, I_n)$ pour $j = 1, 2, 3$ et $Z_4 = 0$;



(a) Courbe ROC sur les variables sélectionnées (b) Courbe ROC sur les groupes sélectionnés

FIGURE 3.3 – Résultats du cadre 1 de simulation. Courbe ROC entre le test hiérarchique contrôlant le FWER (trait noir plein) et le FDR (pointillés rouges). Les disques et les triangles représentent le seuil global $\alpha = 0.05$ pour le FWER et $q = 0.05$ pour le FDR. Le triangle correspond au contrôle sur l'ensemble de l'arbre tandis que le disque à celui sur les *outer nodes*.

— $y = X\beta^* + \epsilon$ avec $\epsilon \sim \mathcal{N}(0_n, 15^2 I_n)$.

Dans ce 1er cadre, les variables sont séparées en quatre groupes. Chaque variable d'un même groupe étant la somme d'un vecteur référence avec du bruit créant ainsi une corrélation par blocs.

Cadre 2 :

— $n = 20$, $p = 8$;
— $\beta^* = (3.1, 5, 0, 0, 2, 0, 0, 0)$;
— $X_{1,.}, \ldots, X_{n,.} \sim \mathcal{N}(0_p, \Sigma)$ avec $\Sigma_{i,j} = 0.5^{|i-j|}$;
— $y = X\beta^* + \epsilon$ avec $\epsilon \sim \mathcal{N}(0_n, 3^2 I_n)$.

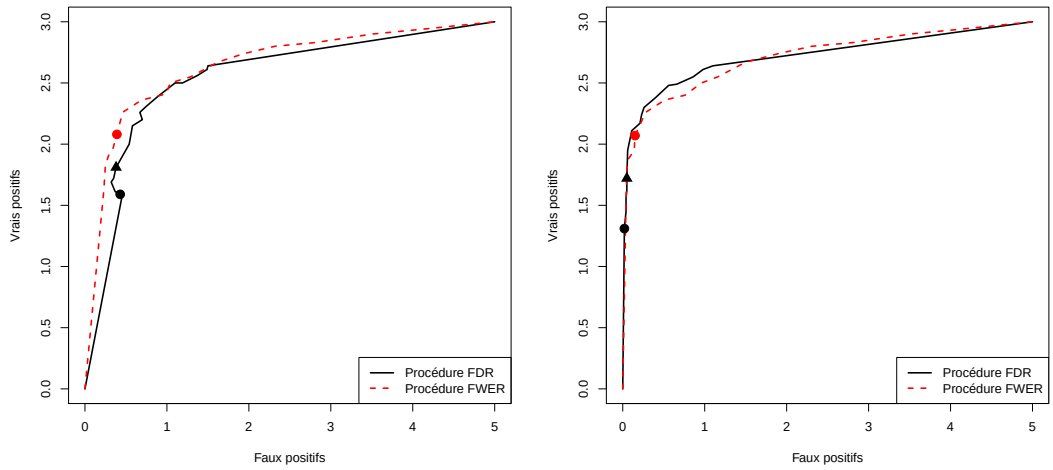
Dans ce 2e cadre, les variables présentent une corrélation (faible) dépendant de la proximité.

Critère de comparaison

Les deux procédures de contrôle du FWER et du FDR sont comparées à l'aide d'une courbe ROC (*Receiver Operating Characteristic*). Elle représente le nombre de vrais positifs en fonction du nombre de faux positifs lorsque que le niveau de contrôle α ou q varie. Dans le cas des variables, une variable sélectionnée est considérée comme un vrai positif si elle appartient au support de la vraie solution β^* sinon c'est un faux positif. De même pour les groupes, un groupe sélectionné est considéré comme un vrai positif s'il contient au moins une variable du support de β^*.

Résultats

Sur la figure 3.3a, la forme de la courbe ROC qui fait un retour en arrière est due au principe du test hiérarchique. Un groupe contenant des vraies et fausses variables peut avoir son hypothèse nulle rejetée à un certain niveau α mais pas celles de ses enfants. Lorsque ce niveau augmente, son enfant contenant majoritairement des vraies variables a son hypothèse nulle



(a) Courbe ROC sur les variables sélectionnées (b) Courbe ROC sur les groupes sélectionnés

FIGURE 3.4 – Résultats du cadre 2 de simulation. Courbe ROC entre le test hiérarchique contrôlant le FWER (trait noir plein) et le FDR (pointillés rouges). Les disques et les triangles représentent le seuil global $\alpha = 0.05$ pour le FWER et $q = 0.05$ pour le FDR. Le triangle correspond au contrôle sur l'ensemble de l'arbre tandis que le disque à celui sur les *outer nodes*.

rejeté mais pas celui contenant des fausses variables. Et comme seuls les enfants sont conservés, cela peut entraîner une diminution du nombre de faux positifs alors que le niveau α a augmenté (cela n'est pas le cas dans les courbes ROC "classiques").

Comparaison de l'ensemble de la courbe ROC Sur la 1re partie des courbes ROC pour les groupes (figures 3.3b et 3.4b), on constate que les courbes sont proches. Par contre, pour les variables (figures 3.3a et 3.4a), on remarque que la courbe ROC associée au FWER est au-dessus de celle associée au FDR avant de revenir à un niveau équivalent voire de s'inverser. Pour de faibles valeurs de α et q, la méthode avec FWER semble meilleure, tandis que pour des plus grandes valeurs, cela ne semble pas aussi tranché.

Contrôle au niveau $\alpha = q = 0.05$ Au niveau global usuel $\alpha = q = 0.05$, la méthode contrôlant le FWER sélectionne un plus grand nombre de groupes (figures 3.3b et 3.4b). Et en matière de variables (figures 3.3a et 3.4a), la qualité semble meilleure en général, les groupes sélectionnés contiennent plus de vraies variables et moins de fausses. Sans surprise, à cause du facteur L dans la borne de contrôle, le contrôle du FDR pour l'ensemble de l'arbre sélectionne un plus grand nombre de variables que le contrôle sur les *outer nodes* rejetés grâce à la correction moindre de celle-ci.

En pratique, de faibles valeurs de α et q sont utilisées (0.05 en général). Il semble que la méthode FWER soit la meilleure sur les simulations proposées. Ces simulations sont d'assez faible dimension mais se rapprochent en taille du cadre d'utilisation de la procédure de test que nous proposons dans la suite. En effet, notre procédure de test va être appliquée sur les groupes

sélectionnés par le *Multi-Layer Group-Lasso* pour une valeur du paramètre de régularisation fixée. Ces groupes seront en nombre limité en pratique.

3.2 Test hiérarchique multiple

Dans cette section, nous proposons une méthode de test pour le chemin solution du *Multi-Layer Group-Lasso*. Son but est pour chaque valeur du paramètre de régularisation λ de conserver un ensemble de groupes influents tout en contrôlant le FWER ou le FDR. Nous commençons par introduire cette procédure dans le cas du group-lasso classique puis l'étendons au *Multi-Layer Group-Lasso*.

3.2.1 Procédure de test pour le group-lasso

L'idée est de tester la nullité des coefficients estimés. On retrouve une stratégie de ce type dans (Wasserman et Roeder 2009) où après avoir effectué un lasso et choisi la valeur de λ par validation croisée, les coefficients du support sont réestimés par les moindres carrés classiques et leurs nullités testées. Ce processus de réestimation par moindres carrés est possible car le support a forcément une taille inférieure au nombre d'individus. Réestimer les coefficients afin d'obtenir un meilleur R^2 avait été proposé dans (Efron et al. 2004).

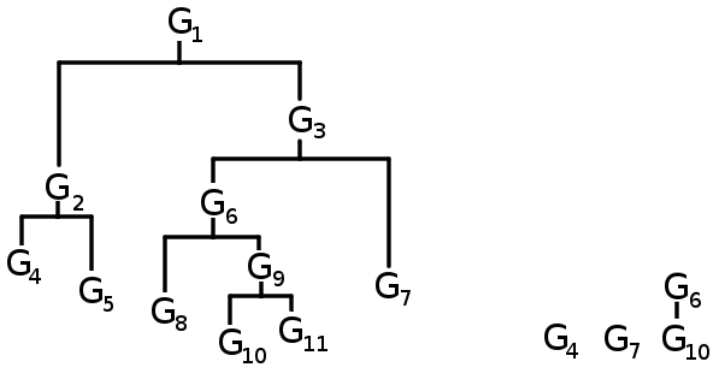
Dans le cas du group-lasso, la réestimation immédiate par moindres carrés est impossible. En effet, pour une valeur de λ, rien n'oblige le support à contenir moins de variables que le nombre d'individus. Par contre, le support contient moins de groupes que d'individus (Liu et J. Zhang 2009). En représentant chaque groupe par une variable, il est possible d'effectuer une réestimation. Nous proposons donc de représenter chaque groupe par sa 1re composante principale.

Notons $\mathcal{I} = \{1,\dots,n\}$, l'ensemble des indices associés aux individus. Cet ensemble est divisé aléatoirement en deux sous-ensembles $\mathcal{I}_1$ et $\mathcal{I}_2$ de même taille. Dans un premier temps, un group-lasso est appliqué sur les données restreintes aux individus $\mathcal{I}_1$. Dans un second temps, la procédure de test est appliquée sur les données restreintes aux individus de l'ensemble $\mathcal{I}_2$. La procédure de test est décrite dans l'algorithme 10.

Algorithme 10 Procédure de test pour le group-lasso

Soit Λ l'ensemble des paramètres de régularisation testé pour le group-lasso.
Pour $\lambda \in \Lambda$ **Faire**
  Soient $G_1,\dots,G_m$ les groupes sélectionnés par group-lasso pour λ.
  Calculer la composante principale $\tilde{X}_i$ de X_{G_i} pour $i = 1,\dots,m$. On note $\tilde{X} = [\tilde{X}_1,\dots,\tilde{X}_m]$.
  Calculer $\tilde{\beta}$ l'estimateur des moindres carrés de $\tilde{X}$ et y.
  Tester la nullité des coefficients de $\tilde{\beta}$ à l'aide du test adéquat (cf. Section 3.1.1).
  Appliquer une correction des tests multiples (Benjamini-Hochberg ou Bonferroni (cf. Section 3.1.2)) à un niveau α.
Fin Pour

La procédure de test proposée sera évaluée dans la Section 3.3.



(a) Hiérarchie de l'exemple 5 (b) Sélection de l'exemple 5

FIGURE 3.5 – Illustration de l'exemple 5 (Sélection du *Multi-Layer Group-Lasso*).

3.2.2 Adaptation pour la sélection de groupes chevauchant

Particularité du *Multi-Layer Group-Lasso*

Dans le cas du *Multi-Layer Group-Lasso*, des groupes inclus les uns dans les autres peuvent être sélectionnés pour une même valeur de λ. Le principe est alors de séparer les groupes sélectionnés : d'une part, ceux contenus ou contenant d'autres groupes sont regroupés dans des arbres $\mathcal{T}_1, \ldots, \mathcal{T}_j$, d'autre part ceux n'ayant aucune intersection avec les autres groupes forment l'ensemble $\mathcal{S}$ (cf. exemple 5 (Sélection du *Multi-Layer Group-Lasso*) et figure 3.5). Sur les premiers une stratégie de test hiérarchique va être appliquée et sur les autres une stratégie de test classique.

Exemple (Sélection du *Multi-Layer Group-Lasso*) *Soient 6 variables dont la hiérarchie forme les groupes suivants :* $G_1 = \{1,2,3,4,5,6\}$, $G_2 = \{1,2\}$, $G_3 = \{3,4,5,6\}$, $G_4 = \{1\}$, $G_5 = \{2\}$, $G_6 = \{3,4,5\}$, $G_7 = \{6\}$, $G_8 = \{3\}$, $G_9 = \{4,5\}$, $G_{10} = \{4\}$, $G_{11} = \{5\}$. *La hiérarchie est visible sur la figure 3.5.*

Supposons que pour une valeur de λ, *les groupes* G_4, G_6, G_7 *et* G_{10} *sont sélectionnés. On a alors* $\mathcal{S} = \{G_4, G_7\}$ *et* $\mathcal{T}_1 = \{G_6, G_{10}\}$ *avec* G_6 *qui contient* G_{10}. □

Complétion des hiérarchies

Afin d'appliquer la stratégie de test hiérarchique, il est nécessaire que les groupes forment une hiérarchie *complète*, c'est-à-dire que l'union des variables des enfants d'un groupe soit égale à ce groupe. Cela est nécessaire car la nullité des coefficients associés à un groupe est testée au travers de sa décomposition en feuilles, si un groupe ne possède qu'un sous-groupe, il ne peut être testé par la méthode de MEINSHAUSEN, le groupe n'étant pas décrit entièrement par ses enfants. Dans ce cas, on ajoute des groupes afin de former de telles hiérarchies que l'on note $\mathcal{T}_1', \ldots, \mathcal{T}_j'$ (cf. exemple 6 (Complétion d'une hiérarchie)). Ces nouveaux groupes peuvent ne pas faire partie des groupes originaux.

Exemple (Complétion d'une hiérarchie) *Les groupes* $G_6 = \{3,4,5\}$ *et* $G_{10} = \{4\}$ *de l'arbre* $\mathcal{T}_1$ *ne forment pas une hiérarchie complète (*G_6 *ne peut être exprimé comme l'union parfaite de ses sous-groupes). On crée alors le groupe* $G' = \{3,5\}$ *qui est le complémentaire de* G_{10} *dans* G_6. *On obtient alors une nouvelle hiérarchie* $\mathcal{T}_1' = \{G_6, G_{10}, G'\}$ *qui est complète.* □

Application de la procédure de test

Une fois les hiérarchies complétées, la 1re composante principale des feuilles de chaque hiérarchie est calculée. Les feuilles jouent le rôle des variables dans la stratégie de test hiérarchique de Meinshausen (cf. Section 3.1.3). On obtient de nouveaux arbres hiérarchiques $\tilde{\mathcal{T}}_1', \dots, \tilde{\mathcal{T}}_j$ où les groupes sont décrits par les composantes principales des feuilles. La stratégie de test hiérarchique contrôlant le FWER peut alors être effectuée sur chaque hiérarchie.

Exemple (Représentation des groupes par la composante principale) *Les feuilles de $\mathcal{T}_1'$, $X_{G_{10}}$ et $X_{G'}$ sont représentées par leur 1res composantes principales $\tilde{X}_1$ et $\tilde{X}_2$ définissant ainsi la matrice $\tilde{X}$. Un arbre $\tilde{\mathcal{T}}_1$ associé à $\tilde{X}$ est créé. Il contient les groupes $\tilde{G}_1 = \{1, 2\}$, $\tilde{G}_2 = \{1\}$ et $\tilde{G}_3 = \{2\}$. Le test hiérarchique est effectué sur la matrice $\tilde{X}$ et l'arbre $\tilde{\mathcal{T}}_1$.* □

Donc, pour chaque valeur de λ, la stratégie de test hiérarchique est appliquée sur les hiérarchies $\mathcal{T}_1, \dots, \mathcal{T}_j$ formées à partir des groupes sélectionnés. La stratégie est résumée pour une hiérarchie dans l'algorithme 11.

Algorithme 11 Gestion des groupes chevauchant dans la sélection du *Multi-Layer Group-Lasso*

Complétion des hiérarchies Compléter les groupes de la hiérarchie $\mathcal{T}$ afin de former une hiérarchie complète $\mathcal{T}'$.

Représentation des groupes Calculer la 1re composante principale pour chaque feuille de $\mathcal{T}'$. On obtient un nouvel arbre $\tilde{\mathcal{T}}$ décrivant la hiérarchie entre les composantes des feuilles de $\mathcal{T}'$.

Appliquer la stratégie de test hiérarchique Appliquer la stratégie de test hiérarchique avec $\tilde{\mathcal{T}}$ avec le niveau de contrôle α.

Niveau de contrôle du FWER

Pour un arbre $\tilde{\mathcal{T}}_i$, le test est effectué avec un niveau de contrôle $\frac{\alpha \times |\tilde{\mathcal{T}}_i|}{m}$ où $m = |\mathcal{S}| + \sum_{i=1}^{j} |\tilde{\mathcal{T}}_i|$. Pour chaque groupe appartenant à $\mathcal{S}$, le niveau de contrôle est de $\frac{\alpha}{m}$. Ainsi le FWER est contrôlé à un niveau global α :

$$
\begin{aligned}
\mathbb{P}\left(\exists G : p^G_{\text{adj}} \leq \alpha\right) &= \mathbb{P}\left(\left\{\exists G \in \mathcal{S} : p^G_{\text{adj}} \leq \alpha\right\} \cup \bigcup_{i=1}^{j} \left\{\exists G \in \tilde{\mathcal{T}}_i : p^G_{\text{adj}} \leq \alpha\right\}\right) \\
&= \mathbb{P}\left(\left\{\exists G \in \mathcal{S} : p^G \leq \frac{\alpha}{m}\right\} \cup \bigcup_{i=1}^{j} \left\{\exists G \in \tilde{\mathcal{T}}_i : p^G \leq \frac{\alpha \times |\tilde{\mathcal{T}}_i|}{m}\right\}\right) \\
&\leq \mathbb{P}\left(\left\{\exists G \in \mathcal{S} : p^G \leq \frac{\alpha}{m}\right\}\right) + \sum_{i=1}^{j} \mathbb{P}\left(\exists G \in \tilde{\mathcal{T}}_i : p^G \leq \frac{\alpha \times |\tilde{\mathcal{T}}_i|}{m}\right) \\
&\leq \alpha \frac{|\mathcal{S}|}{m} + \sum_{i=1}^{j} \alpha \frac{|\mathcal{T}_i|}{m} \\
&\leq \alpha
\end{aligned}
$$

La procédure de test hiérarchique multiple pour le *Multi-Layer Group-Lasso* est présentée dans l'algorithme 12. Dans la partie suivante, son utilisation avec le *Multi-Layer Group-Lasso* est

présentée.

Algorithme 12 Test hiérarchique multiple pour le *Multi-Layer Group-Lasso*

Soit Λ l'ensemble des paramètres de régularisation testé pour le group-lasso.
Pour $\lambda \in \Lambda$ **Faire**
Définition des hiérarchies Séparer les groupes sélectionnés en hiérarchie $\mathcal{T}_1, \dots, \mathcal{T}_j$ et en $\mathcal{S}$ l'ensemble des groupes non contenus et ne contenant pas d'autres groupes.
Groupes appartenant à des hiérarchies Pour chaque hiérarchie $\mathcal{T}_i$, appliquer l'algorithme 11 avec un niveau de contrôle $\frac{\alpha \times |\mathcal{T}_i|}{m}$.
Groupes hors hiérarchies Pour l'ensemble $\mathcal{S}$, appliquer la stratégie de l'algorithme 10 pour une valeur de λ avec un niveau de contrôle $\frac{\alpha \times |\mathcal{S}|}{m}$.
Fin Pour

3.2.3 *Multi-Layer Group-Lasso* et test hiérarchique multiple

Afin d'utiliser la procédure de test hiérarchique multiple proposée précédemment avec le *Multi-Layer Group-Lasso*, nous proposons d'utiliser la stratégie définie dans l'algorithme 13.

Les individus sont divisés en deux parties afin de ne pas utiliser les mêmes données pour la création des modèles et la méthode de choix de modèle, ce qui est le principe général de la validation croisée. Cette stratégie d'appliquer la phase de test sur des données différentes a également été utilisée dans (Wasserman et Roeder 2009) pour son *multi-stage lasso*. Ainsi sur le 1er ensemble d'individus, la classification ascendante hiérarchique est effectuée. Sur le 2e ensemble, notre group-lasso est appliqué aux données avec l'arbre et les poids calculés sur le 1er ensemble. La méthode de test hiérarchique multiple (algorithme 12) est quant à elle appliquée sur le 1er ensemble, les données y sont utilisées pour calculer les 1res composantes principales des différents groupes et effectuer les tests associés.

Algorithme 13 *Multi-Layer Group-Lasso* et test hiérarchique multiple

1. Diviser l'ensemble des n individus $\mathcal{I} = \{1, \dots, n\}$ de X en 2 sous-ensembles disjoints $\mathcal{I}_1$ et $\mathcal{I}_2$ de taille $\frac{n}{2}$.
2. Effectuer la CAH à partir des individus $\mathcal{I}_1$.
3. Appliquer le *Multi-Layer Group-Lasso* (2.6) sur les individus $\mathcal{I}_2$.
4. Appliquer la méthode de test hiérarchique multiple (algorithme 12) sur les individus $\mathcal{I}_1$.

3.3 Simulations

Les cadres de simulations sont les mêmes que ceux présentés dans la Section 2.2.1. Dans les différents cadres de simulations, on prend la valeur $\beta_j^* = 1$ pour une variable d'indice j appartenant au support de β^*. Plusieurs variables du support de β^* ne peuvent se trouver dans un même bloc de corrélation défini par la structure de la matrice de variance-covariance de X. La variance du bruit est quant à elle fixée pour avoir un ratio signal sur bruit $((\beta^*)^T X^T X \beta^* / \|\epsilon\|_2^2)$ de 2. Le comportement de la procédure de test hiérarchique appliquée au *Multi-Layer Group-Lasso* est étudié. L'ensemble du chemin solution du *Multi-Layer Group-Lasso* avant et après application de la procédure de test est comparé par le biais du nombre de vrais positifs et faux positifs. Nous

considérons comme vrai positif un groupe sélectionné contenant une variable du support de la vraie solution β^* et des variables corrélées avec celle-ci. Si un groupe sélectionné contient deux variables du support n'ayant aucun lien de corrélation, il est considéré comme un faux positif car ces variables auraient dû être sélectionnées dans deux groupes distincts.

3.3.1 Comparaison de différents tests

La procédure complète décrite dans l'algorithme 13 a été utilisée pour tester la procédure de test hiérarchique multiple dans les simulations de cette section. Pour tester l'importance des variables au sein des procédures de test hiérarchique, différents tests statistiques ont été utilisés : le test des modèles emboîtés et le score test (cf. Section 3.1.1). Pour le test des modèles emboîtés, deux cas ont été étudiés. Dans le premier, le modèle réduit correspond au modèle ne contenant qu'une constante et le complet celui contenant les variables dont l'importance est à tester (le test est dénoté par *F-test* dans les graphes). Dans le deuxième, le modèle complet comprend la totalité des variables et le modèle réduit contient toutes les variables exceptées celles dont l'importance est testée (dénoté par *partial F-test*). Toutes les courbes sont la moyenne de 100 répétitions.

Contrôle à un niveau $\alpha = 0.05$ et $q = 0.05$

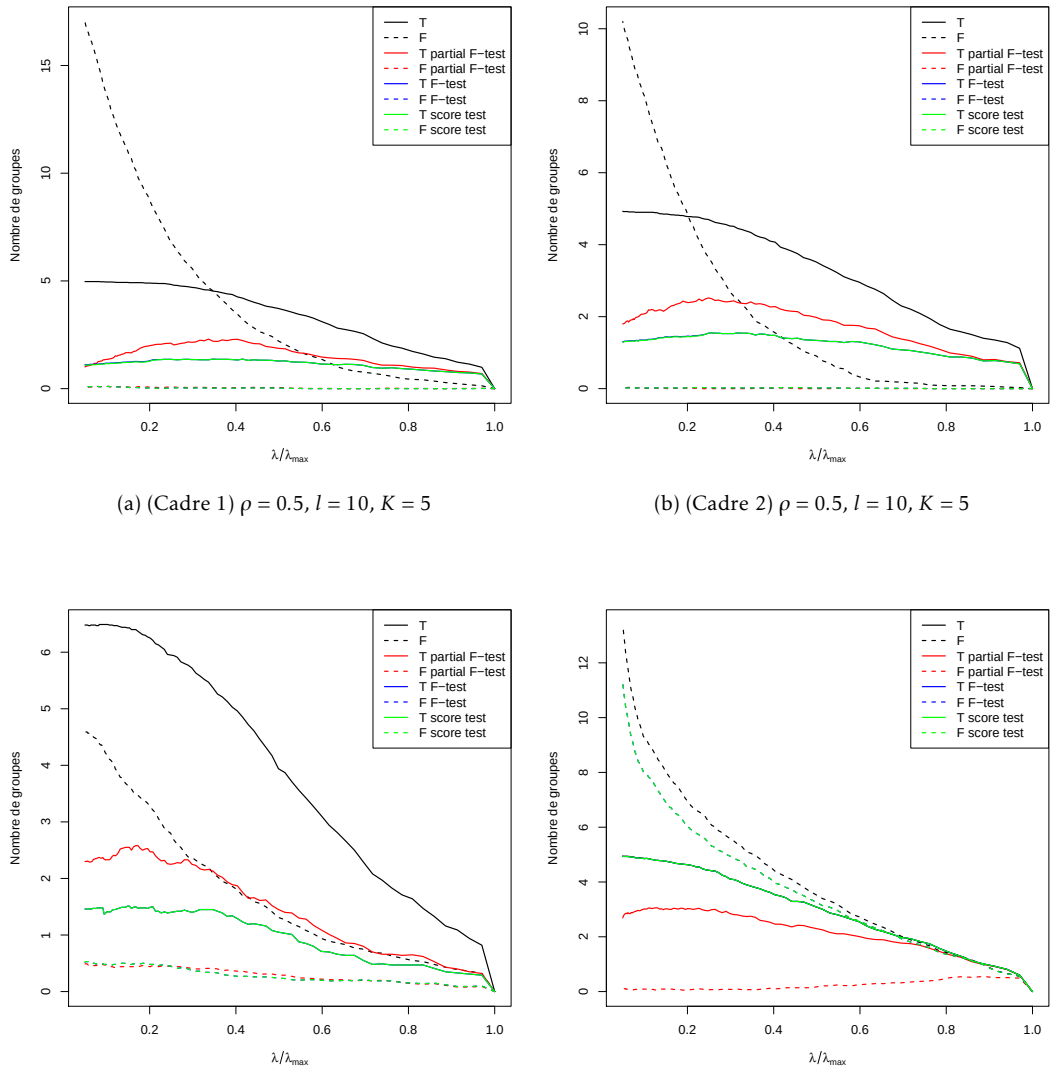
Sur la figure 3.6, les résultats obtenus par l'utilisation de la procédure de test hiérarchique avec contrôle du FWER sont présentés. On constate que le partial F-test donne de meilleurs résultats pour le contrôle du FWER à un niveau α. Le F-test et le score test fournissent des résultats similaires (les courbes sont confondues). Dans le cadre avec de la corrélation entre les blocs de variables (figure 3.6d), peu de groupes sont sélectionnés quand le score test et le F-test sont utilisés, peu de groupes ont leurs hypothèses nulles non rejetées. Cela aboutit à la découverte de tous les vrais positifs mais à un nombre de faux positifs très important. Dans ce cadre, le partial F-test obtient de meilleurs résultats avec une sélection très faible de faux positifs. On constate un très faible nombre de faux groupes sélectionnés mais un certain nombre de vrais groupes sont manqués.

Sur la figure 3.7, la procédure a été appliquée avec le contrôle du FDR. On fait ici les mêmes constatations que pour le contrôle du FWER : le partial F-test fournit de meilleurs résultats. En comparant les figures 3.7a à 3.6b et 3.7b à 3.6c, on ne peut conclure sur la supériorité du contrôle du FDR ou du FWER, les résultats varient d'un cas à l'autre. La méthode avec contrôle du FDR domine dans le cadre 2 avec en moyenne 3.5 vrais positifs pour 2.5 pour le contrôle du FWER avec un même nombre de faux positifs. Dans le cadre 3, l'inverse se produit.

Comparaison de l'ensemble de la courbe ROC

Les procédures de test hiérarchique ont été comparées à un niveau de contrôle α fixé. Nous allons comparer ces méthodes plus globalement pour divers valeurs de α en traçant des courbes ROC pour différentes valeurs de $\frac{\lambda}{\lambda_{\max}} \in \{0.75, 0.5, 0.25, 0.1\}$.

Sur la figure 3.8, le partial F-test semble globalement meilleur pour le contrôle du FWER pour les différentes valeurs de $\frac{\lambda}{\lambda_{\max}}$. Les courbes sont assez proches pour un faible nombre de faux positifs mais quand ce nombre augmente le partial F-test devient meilleur. Il obtient généralement un meilleur compromis entre le nombre de vrais et faux positifs pour $\alpha = 0.05$. On retrouve les mêmes comportements pour les autres cadres de simulation dans le cas du contrôle du FWER. Pour le contrôle du FDR, nous arrivons aux mêmes conclusions.



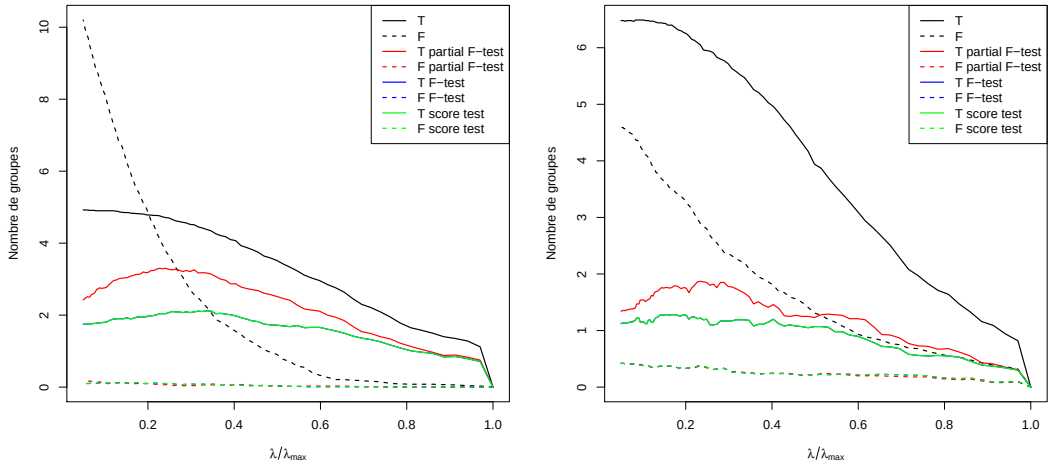
(a) (Cadre 1) $\rho = 0.5$, $l = 10$, $K = 5$

(b) (Cadre 2) $\rho = 0.5$, $l = 10$, $K = 5$

(c) (Cadre 3) $\rho = 0.7$, $l = 10$, $K = 10$, $\pi = \frac{1}{2}$.

(d) (Cadre 4) $\rho = 0.7$, $\rho_2 = 0.3$, $l = 5$, $K = 5$

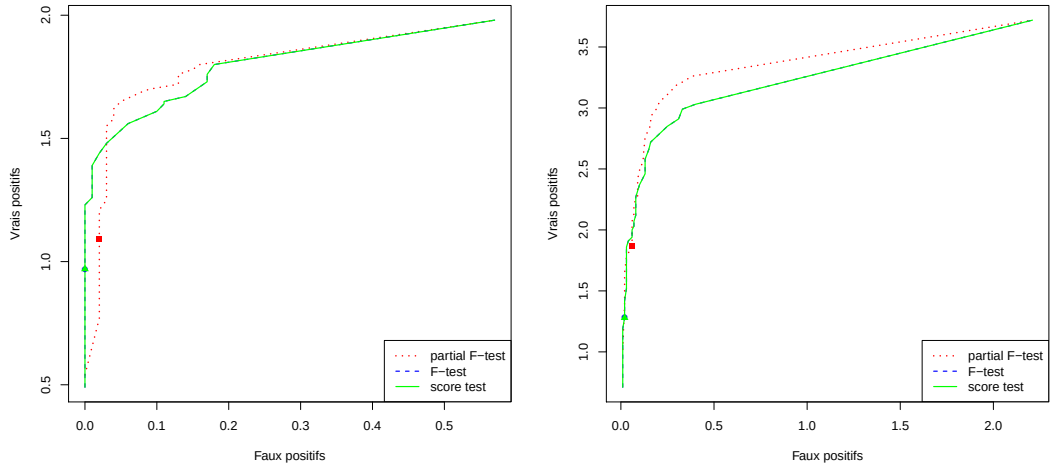
FIGURE 3.6 – Chemins solutions pour la procédure de test hiérarchique multiple (algorithme 13) pour $n = 100$ et $p = 500$. Les tests ont été effectués pour $\alpha = 0.05$ avec contrôle du FWER. En abscisse, les valeurs du paramètre de régularisation sont représentées. En trait plein, on trouve le nombre de vrais groupes sélectionnés et en pointillés celui de faux groupes. Le noir correspond au chemin solution du group-lasso sur l'ensemble de la CAH (cf. équation (2.6)). Les autres couleurs correspondent aux 3 tests utilisés.



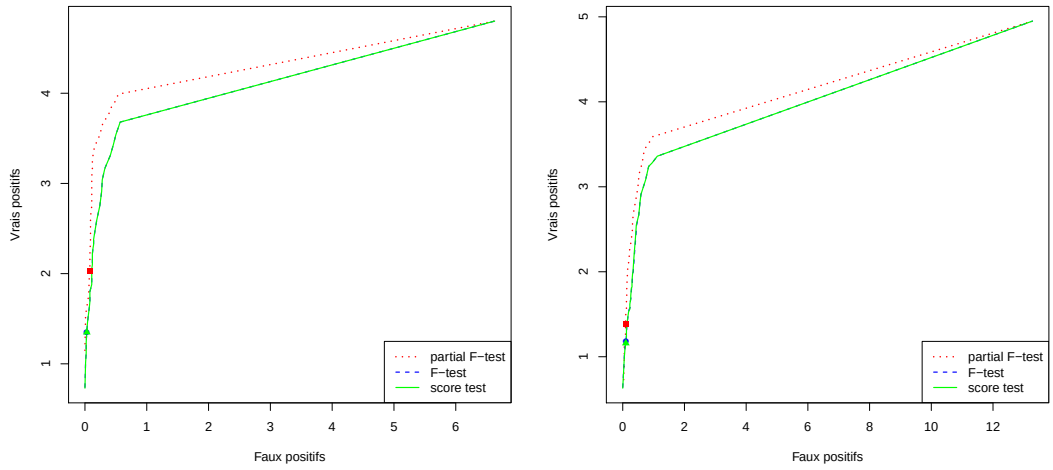
(a) (Cadre 2) $\rho = 0.5$, $l = 10$, $K = 5$

(b) (Cadre 3) $\rho = 0.7$, $l = 10$, $K = 10$, $\pi = \frac{1}{2}$

FIGURE 3.7 – Chemins solutions pour la procédure de test hiérarchique multiple (algorithme 13) pour $n = 100$ et $p = 500$. Les tests ont été effectués pour $\alpha = 0.05$ avec contrôle du FDR. En abscisse, les valeurs du paramètres de régularisation sont représentées. En trait plein, on trouve le nombre de vrais groupes sélectionnés et en pointillés celui de faux groupes. Le noir correspond au chemin solution du group-lasso sur l'ensemble de la CAH (cf. équation (2.6)). Les autres couleurs correspondent aux 3 tests utilisés.

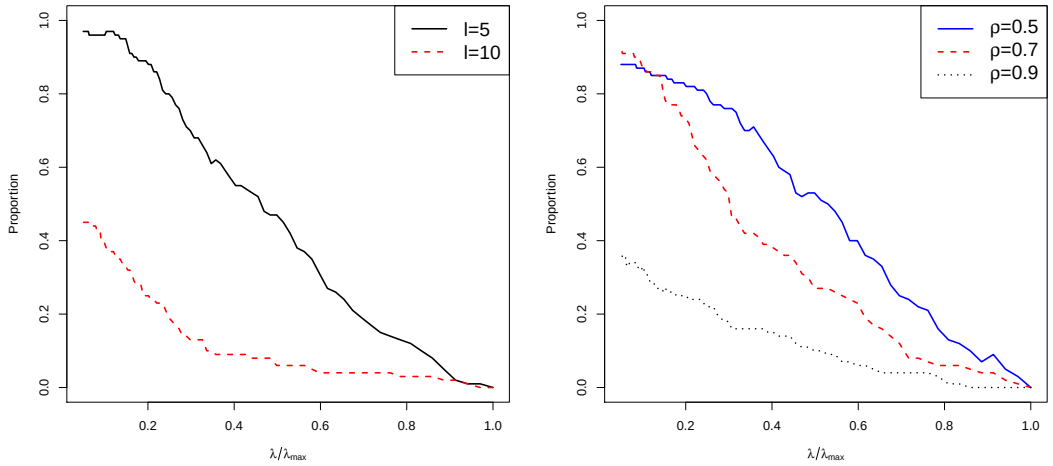


(a) $\frac{\lambda}{\lambda_{\max}} = 0.75$ (b) $\frac{\lambda}{\lambda_{\max}} = 0.5$



(c) $\frac{\lambda}{\lambda_{\max}} = 0.25$ (d) $\frac{\lambda}{\lambda_{\max}} = 0.1$

FIGURE 3.8 – Courbes ROC obtenues pour le cadre 1 de simulation pour $n = 100$, $p = 500$, $K = 5$, $\rho = 05$, $l = 10$ et différentes valeurs de $\frac{\lambda}{\lambda_{\max}}$. Les points de couleurs représentent la valeur $\alpha = 0.05$ pour les différents tests.



(a) $p = 250$, $K = 5$, $\rho = 0.7$ (b) $p = 500$, $K = 5$, $l = 10$

FIGURE 3.9 – Proportion du nombre d'échantillons contenant au moins une hiérarchie en fonction de $\frac{\lambda}{\lambda_{\max}}$ dans le cas du cadre 3 de simulation. À gauche, la taille des groupes varie. Àdroite, la corrélation intra-blocs varie.

Dans la suite, toutes les simulations sont donc effectuées avec le partial F-test qui a montré de meilleurs résultats.

3.3.2 Comparaison des procédures de test hiérarchique pour contrôler le FDR et le FWER

Dans la section précédente, nous avons cherché à déterminer le meilleur test pour notre procédure de test hiérarchique multiple que ce soit pour contrôler le FDR ou le FWER. Dans cette section, nous allons comparer les deux procédures de test hiérarchique.

Présence de hiérarchies au sein de la sélection

Tout d'abord, afin de comparer au mieux les tests hiérarchiques, regardons la présence de hiérarchie au sein des 100 échantillons, c'est-à-dire la présence de groupes sélectionnés par le *Multi-Layer Group-Lasso* contenus dans ou contenant d'autres groupes sélectionnés pour une même valeur de λ. Sur la figure 3.9, on constate que la proportion d'échantillons contenant au moins une hiérarchie augmente lorsque λ diminue (plus de groupes sont sélectionnés). Plus le paramètre de régularisation λ est faible, plus la méthode est susceptible de sélectionner des groupes contenus ou contenant d'autres groupes déjà sélectionnés. En présence de très fortes corrélations ($\rho = 0.9$), la présence de hiérarchie diminue. De manière générale, plus le contexte est défavorable pour des approches basées sur les groupes (faible corrélation, augmentation de la taille des groupes, augmentation du nombre de vrais groupes), plus il y a de chance que la sélection contienne des groupes imbriqués.

Dans la suite de cette sous-section, les courbes sont la moyenne des échantillons contenant au moins une hiérarchie dans leur chemin solution.

Contrôle à un niveau $\alpha = 0.05$ et $q = 0.05$

Sur la figure 3.10, on constate que globalement la procédure utilisant le test hiérarchique avec contrôle du FWER est meilleure que celle contrôlant le FDR. Pour un ratio $\frac{\lambda}{\lambda_{\max}}$ élevé, le contrôle du FDR et FWER donnent des résultats équivalents pour $\alpha = 0.05$ voire meilleurs pour le FDR. Cela s'explique par une faible proportion de hiérarchie pour ces valeurs de λ (cf. figure 3.9). Dans ce cas, un ajustement classique (correction de Bonferroni ou de Benjamini-Hochberg) est utilisé pour lequel le contrôle du FDR fournit de meilleurs résultats. Lorsque la proportion d'échantillons contenant des hiérarchies augmente, le test hiérarchique avec contrôle du FWER devient meilleur en matière de vrais positifs, le nombre de faux positifs est quant à lui légèrement supérieur à celui obtenu par contrôle du FDR pour $q = 0.05$. Cela peut s'expliquer par la correction à apporter pour obtenir un niveau de contrôle q du FDR (Section 3.1.3) qui peut être assez forte.

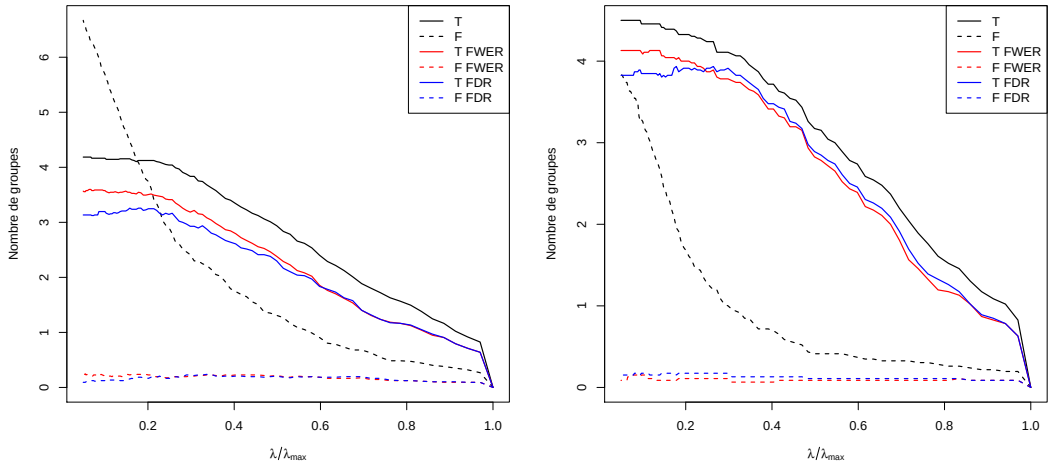
Comparaison de l'ensemble de la courbe ROC

De manière plus globale, la tendance pour $\alpha = 0.05$ sur la figure 3.10*c* semble se confirmer sur les courbes ROC de la figure 3.11. Pour une forte valeur de $\frac{\lambda}{\lambda_{\max}}$, la courbe ROC associée au test hiérarchique contrôlant le FDR est meilleure que celle du FWER, puis la tendance semble s'inverser. Les courbes restent cependant proches et au vu des valeurs pour $\alpha = 0.05$ (représentées par des points sur les courbes ROC), la borne de la procédure de contrôle du FDR (Section 3.1.3) semble trop conservative.

En conclusion, la présence de groupes formant une hiérarchie au sein de la sélection dépend de la valeur de λ. Il y a plus de chances d'en avoir pour de faibles valeurs de λ (moins de pénalisation et donc plus de groupes sélectionnés). En présence de hiérarchie, le test hiérarchique avec contrôle du FWER semble plus performant notamment à cause de la trop forte correction imposée par le test hiérarchique contrôlant le FDR. En cas d'absence de hiérarchie dans la sélection, une correction de Benjamini-Hochberg pour le contrôle du FDR est à privilégier.

3.4 Conclusion

Dans ce chapitre, nous avons proposé une procédure de test hiérarchique multiple pour dissocier les groupes influents parmi les groupes sélectionnés par le *Multi-Layer Group-Lasso*. Cette procédure prend en compte la présence potentielle de hiérarchies parmi les groupes sélectionnés. L'intérêt de cette méthode est de contrôler le nombre de faux positifs. Pour chaque valeur du paramètre de régularisation λ du *Multi-Layer Group-Lasso*, les groupes conservés après la procédure de test hiérarchique multiple ne contiennent plus de groupes chevauchant, ils sont donc plus facilement interprétable. Il est nécessaire de trouver le paramètre λ optimal du chemin solution pour la procédure de test hiérarchique multiple.



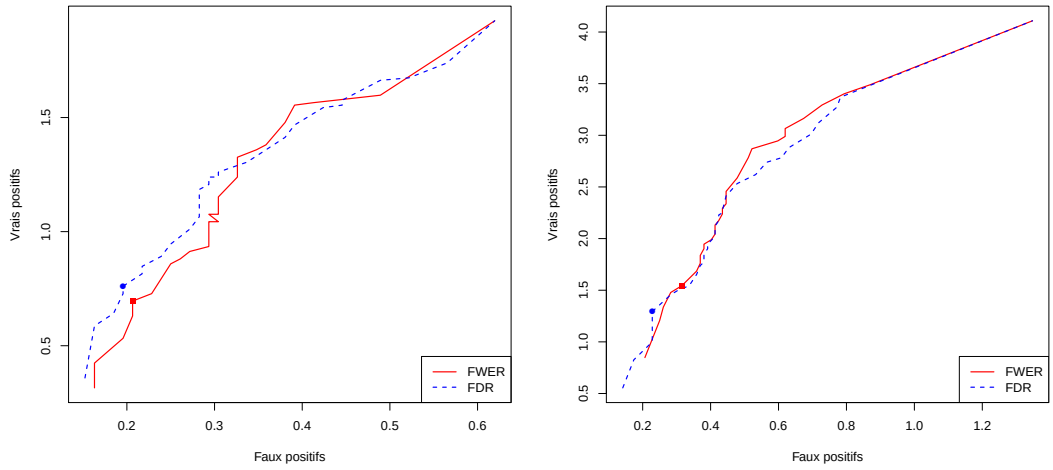
(a) $p = 250$, $K = 5$, $\rho = 0.7$, $l = 5$

(b) $p = 250$, $K = 5$, $\rho = 0.7$, $l = 10$

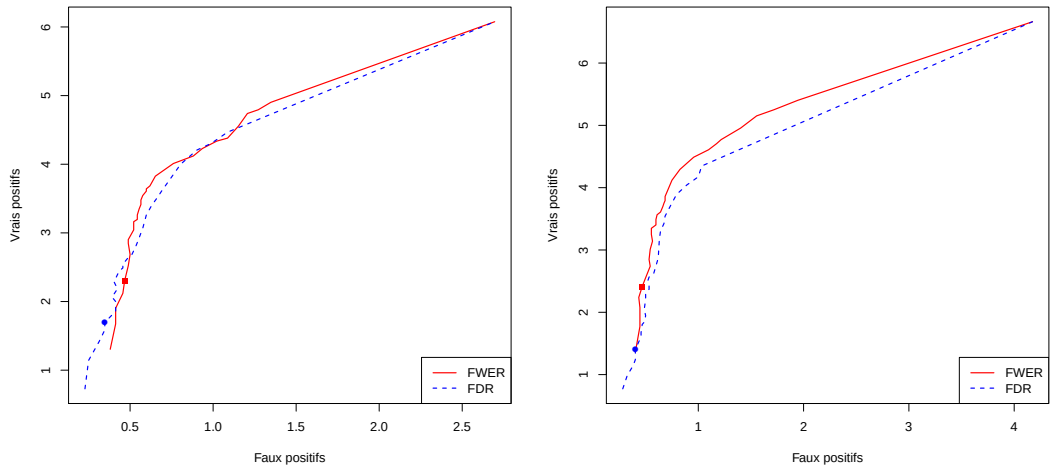
(c) $p = 500$, $K = 5$, $\rho = 0.7$, $l = 10$

(d) $p = 500$, $K = 5$, $\rho = 0.9$, $l = 10$

FIGURE 3.10 – Groupes sélectionnés par la procédure de test hiérarchique multiple contrôlant le FDR (en bleu) et celle contrôlant le FWER (en rouge) à un niveau $\alpha = 0.05$ pour le cadre 3 de simulation. Le noir correspond au chemin solution du group-lasso sur l'ensemble de la CAH (cf. équation (2.6)). Entre les graphes 3.10a et 3.10b, la taille des blocs de la matrice de variance-covariance passe de 5 à 10. Entre les graphes 3.10b et 3.10c, le nombre de variables passe de 250 à 500. Entre les graphes 3.10c et 3.10d, la corrélation intra-blocs augmente de 0.7 à 0.9.



(a) $\frac{\lambda}{\lambda_{\max}} = 0.75$ (b) $\frac{\lambda}{\lambda_{\max}} = 0.50$



(c) $\frac{\lambda}{\lambda_{\max}} = 0.25$ (d) $\frac{\lambda}{\lambda_{\max}} = 0.10$

FIGURE 3.11 – Courbes ROC obtenues pour le cadre 3 de simulation pour $p = 500$, $K = 5$, $\rho = 0.7$, $l = 10$ et différentes valeurs de $\frac{\lambda}{\lambda_{\max}}$. En rouge, la courbe ROC associée au test hiérarchique contrôlant le FWER, en bleu, celle associée à celui contrôlant le FDR. Les points de couleurs représentent la valeur $\alpha = 0.05$.

Chapitre 4

Choix du paramètre de régularisation

Dans le chapitre précédent, une procédure de test hiérarchique multiple pour le *Multi-Layer Group-Lasso* a été présentée. Elle s'applique pour chaque valeur du paramètre de régularisation λ aux groupes sélectionnés par le *Multi-Layer Group-Lasso*. Elle fournit un contrôle du nombre de faux positifs pour chaque valeur de λ. La valeur de λ reste cependant à optimiser. Classiquement une approche de type validation croisée *V-fold* est utilisée mais une telle méthode s'avère coûteuse en temps de calcul à cause des réexécutions multiples qu'elle nécessite et est classiquement utilisée dans une optique de prédiction (en minimisant l'erreur de prédiction) et non de sélection. L'objectif est ici de trouver une valeur optimale du paramètre λ avec un coût minimum en temps de calcul.

4.1 Choix de la valeur du paramètre de régularisation

Le comportement de la procédure de test hiérarchique multiple a été étudié Section 3.3. Au vu des différentes simulations effectuées, on constate que le nombre de faux positifs reste relativement faible et croît légèrement sur l'ensemble du chemin solution quand la valeur de λ diminue. Tandis que le nombre de vrais positifs croît pour se stabiliser voire décroître pour de faibles valeurs de λ. La méthode est très conservative en ce sens. Ces observations faîtes sur la plupart des simulations effectuées, nous suggérons de sélectionner la valeur de λ maximisant le nombre de rejets. Ainsi, nous espérons maximiser le nombre de vrais positifs, tout en gardant un nombre de faux positifs faible. Les groupes sélectionnés pour cette valeur de λ sont les groupes considérés comme d'importance pour expliquer y.

Ainsi, la procédure complète que nous proposons est décrite dans l'algorithme 14. Elle consiste à utiliser le *Multi-Layer Group-Lasso* sur la moitié des individus puis la procédure de test hiérarchique sur la seconde moitié puis de choisir la valeur du paramètre de régularisation λ maximisant le nombre de rejets dans la phase de test hiérarchique.

Dans la suite, nous testons l'efficacité de ce choix de λ en le comparant à d'autres méthodes classiques de choix de λ.

Algorithme 14 *Multi-Layer Group-Lasso* et test hiérarchique multiple avec choix de λ

1. Diviser l'ensemble des n individus $\mathcal{I} = \{1, \dots, n\}$ de X en 2 sous-ensembles disjoints $\mathcal{I}_1$ et $\mathcal{I}_2$ de taille $\frac{n}{2}$.
2. Effectuer la CAH à partir des individus $\mathcal{I}_1$.
3. Appliquer le *Multi-Layer Group-Lasso* (2.6) sur les individus $\mathcal{I}_2$.
4. Appliquer la méthode de test hiérarchique multiple (algorithme 12) sur les individus $\mathcal{I}_1$.
5. Choisir la valeur de λ maximisant le nombre de rejets.

4.2 Simulations

Dans cette section, nous allons tester l'efficacité du choix de λ au sein de la procédure. Puis nous la comparons avec d'autres méthodes de choix de λ. Les cadres de simulations sont les mêmes que ceux présentés dans la Section 2.2.1. Dans les différents cadres de simulations, on prend la valeur $\beta_j^* = 1$ pour une variable d'indice j appartenant au support de β^*. Plusieurs variables du support de β^* ne peuvent se trouver dans un même bloc de corrélation défini par la structure de la matrice de variance-covariance de X. La variance du bruit est quant à elle fixée pour avoir un ratio signal sur bruit ($(\beta^*)^T X^T X \beta^* / \|\epsilon\|_2^2$) de 2. Les différentes procédures de choix de λ sont évaluées par le biais du nombre de vrais positifs et faux positifs. Nous considérons comme vrai positif un groupe sélectionné contenant une variable du support de la vraie solution β^* et des variables corrélées avec celle-ci. Si un groupe sélectionné contient deux variables du support n'ayant aucun lien de corrélation, il est considéré comme un faux positif car ces variables auraient dû être sélectionnées dans deux groupes distincts.

4.2.1 Choix de la valeur du paramètre de régularisation

Nous comparons le choix de la valeur de λ que nous proposons (celle maximisant le nombre de rejets, notée $\hat{\lambda}_{RM}$) avec la valeur de λ proposant la meilleure solution en matière de vrais positifs (notée $\hat{\lambda}_{TPM}$). Cette solution est considérée comme la meilleure pour notre procédure.

Dans le Tableau 4.1, on présente la qualité des groupes sélectionnés en matière de vrais positifs et faux positifs pour la valeur de $\hat{\lambda}_{RM}$ choisie. Premièrement, on constate que le nombre de faux positifs est assez faible après application de la méthode de post-traitement. On voit que pour $\hat{\lambda}_{RM}$, le nombre de vrais positifs est très proche de celui associé à la meilleure valeur de λ ($\hat{\lambda}_{TPM}$). Cependant le nombre de faux positifs associés est plus élevé (plus du double généralement). Le choix de la valeur de λ maximisant le nombre de rejets semble donc cohérent.

4.2.2 Comparaison avec différentes méthodes de choix de paramètres

Dans cette section, nous comparons notre procédure complète (cf. algorithme 14) au *Multi-Layer Group-Lasso* utilisée avec la validation croisée *5-fold*, la *stability selection* et le critère Kappa (Section 1.1.4).

Présentation des méthodes

Pour l'ensemble des méthodes (validation croisée *5-fold*, la *stability selection* et le critère Kappa), la CAH est effectuée en utilisant l'ensemble des individus. Cette CAH est utilisée avec le *Multi-Layer Group-Lasso* pour les différents échantillons bootstraps.

Tableau 4.1 – Qualité de la sélection pour $\hat{\lambda}$ maximisant le nombre de rejets dans notre procédure dans le cadre 3 de simulation pour $n = 100$ et $p = 500$. Dans le tableau, *VP* et *FP* correspondent respectivement au nombre de vrais positifs et faux positifs. $\hat{\lambda}_{\text{RM}}$ (respectivement $\hat{\lambda}_{\text{TPM}}$) correspond à la valeur de λ maximisant le nombre de rejets (respectivement vrais positifs). *MLGL* correspond au *Multi-Layer Group-Lasso* et *THM* à la procédure de test hiérarchique multiple. Ainsi *MLGL* + $\hat{\lambda}_{\text{RM}}$ correspond à la composition du chemin solution pour $\lambda = \hat{\lambda}_{\text{RM}}$. *MLGL* + *THM* + $\hat{\lambda}_{\text{RM}}$ à celle du chemin solution pour $\lambda = \hat{\lambda}_{\text{RM}}$ après application de la procédure de test hiérarchique multiple. *MLGL* + *THM* + $\hat{\lambda}_{\text{TPM}}$ correspond à la meilleure solution en matière de vrais positifs après application du *Multi-Layer Group-Lasso* et de la procédure de test hiérarchique multiple. K, l et ρ désignent les paramètres du cadre 3 de simulation (Section 2.2.1). K désigne le nombre de groupes appartenant au support de la vraie solution β^*, l désigne la taille des blocs de la matrice de variance-covariance et ρ la corrélation intra-blocs.

		$K = 5$				$K = 10$			
		$l = 5$		$l = 10$		$l = 5$		$l = 10$	
		VP	FP	VP	FP	VP	FP	VP	FP
$\rho = 0.9$	MLGL + $\hat{\lambda}_{\text{RM}}$	4.95	7.38	4.89	9.08	6.09	9.44	6.61	11.61
	MLGL + THM + $\hat{\lambda}_{\text{RM}}$	4.91	0.28	4.78	0.8	4.15	0.44	4.75	1.28
	MLGL + THM + $\hat{\lambda}_{\text{TPM}}$	4.95	0	4.86	0.05	4.27	0.14	4.86	0.57
$\rho = 0.7$	MLGL + $\hat{\lambda}_{\text{RM}}$	3.35	3.29	4.13	2.02	2.49	2.76	5.43	2.22
	MLGL + THM + $\hat{\lambda}_{\text{RM}}$	2.95	0.51	3.95	0.27	1.59	0.64	3.39	0.54
	MLGL + THM + $\hat{\lambda}_{\text{TPM}}$	3.02	0.13	3.97	0.14	1.65	0.23	3.4	0.39
$\rho = 0.5$	MLGL + $\hat{\lambda}_{\text{RM}}$	3.1	2.33	3.03	2.96	2.64	4.05	2.75	3.18
	MLGL + THM + $\hat{\lambda}_{\text{RM}}$	2.89	0.32	2.6	0.53	1.68	0.41	1.53	0.63
	MLGL + THM + $\hat{\lambda}_{\text{TPM}}$	2.95	0.04	2.7	0.13	1.72	0.13	1.58	0.26

Pour le critère Kappa, la valeur estimée du paramètre de régularisation a été choisie comme celle maximisant $\hat{s}(\lambda)$ (cf. algorithme 2).

Pour la *stability selection*, les probabilités de sélection sont estimées pour chaque groupe de la CAH. Les groupes dont la probabilité de sélection dépasse le seuil 0.75 sont sélectionnés.

Comparaison du test hiérarchique multiple avec choix de λ à d'autres méthodes de sélection de paramètres de régularisation

Les résultats pour le cadre 3 de simulation sont présentés dans le Tableau 4.2.

De manière générale, la méthode proposée obtient moins de vrais positifs que les autres méthodes mais avec un nombre de faux positifs très faible comparé à celles-ci.

La *stability selection* obtient les meilleurs résultats sélectionnant un nombre plus important de vrais positifs tout en gardant un nombre de faux positifs raisonnable. Mais elle nécessite un nombre important de rééchantillonnages (50 ou 100 en général), et est donc coûteuse en temps de calcul. Quand la taille l des groupes augmente, on constate une augmentation du nombre de faux positifs pour la *stability selection*. Cela peut s'expliquer par le fait que les niveaux d'intérêts (ceux avec un poids minimal) contiennent moins de groupes et donc ces groupes sont plus souvent sélectionnés lors des rééchantillonnages ce qui rend leur probabilité de sélection plus élevée. Les bons résultats de la *stability selection* associée au *Multi-Layer Group-Lasso* prouvent l'efficacité de notre méthode pour la sélection de groupes.

La validation croisée *5-fold* sélectionne un nombre important de faux positifs, plus particulièrement dans le cas plus facile qu'est $\rho = 0.9$. On constate notamment que les valeurs de $\hat{\lambda}$ estimées par validation croisée sont très faibles (figure 4.1) comparées aux valeurs sélectionnées par les autres méthodes. Ces valeurs sont très proches de la valeur minimale de λ testée. La validation croisée a tendance à sélectionner le modèle le plus complexe dans un but d'optimiser le pouvoir prédictif et fait au final très peu de sélection. Comparé à la validation croisée 5-fold, le critère Kappa sélectionne nettement moins de faux positifs mais avec un nombre de vrais positifs inférieur.

Notre procédure avant application de la phase de test multiple (MLGL + $\hat{\lambda}_{\text{RM}}$) sélectionne moins de groupes que la validation croisée. Pour une forte valeur de corrélation ($\rho = 0.9$), elle sélectionne plus de groupes que le critère Kappa. Le phénomène s'inverse dans le cas $\rho = 0.5$. Après application du test hiérarchique multiple (MLGL + THM + $\hat{\lambda}_{\text{RM}}$), notre méthode sélectionne au global moins de faux positifs que les autres méthodes. Elle reste cependant inférieure en matière de vrais positifs.

Test hiérarchique multiple avec différentes méthodes de choix de λ

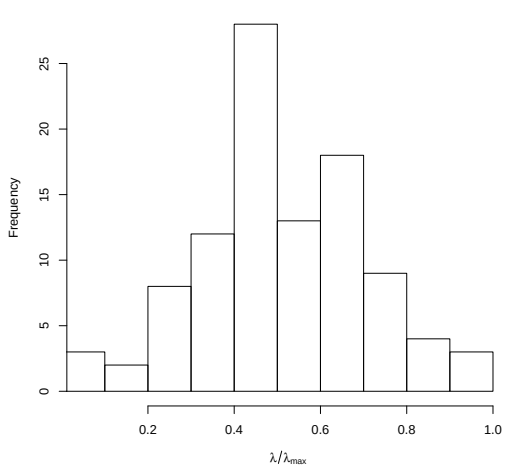
Après avoir estimé la valeur de $\hat{\lambda}$ avec des méthodes comme la validation croisée *V-fold* ou le critère Kappa, nous pouvons appliquer notre méthode de test hiérarchique multiple. Les résultats sont compilés dans le Tableau 4.3. Dans le cas $K = 5$, utiliser les méthodes Kappa ou validation croisée pour estimer $\hat{\lambda}$ produit de meilleurs résultats que choisir le $\hat{\lambda}$ maximisant le nombre de rejets. Dans le cas plus compliqué $K = 10$, les résultats sont plus contrastés mais restent assez proches de ceux de la méthode proposée. On en conclut que choisir la valeur de $\hat{\lambda}$ comme celle maximisant le nombre de rejets n'est pas le choix optimal mais fournit d'assez bons résultats de manière générale.

Seuillage

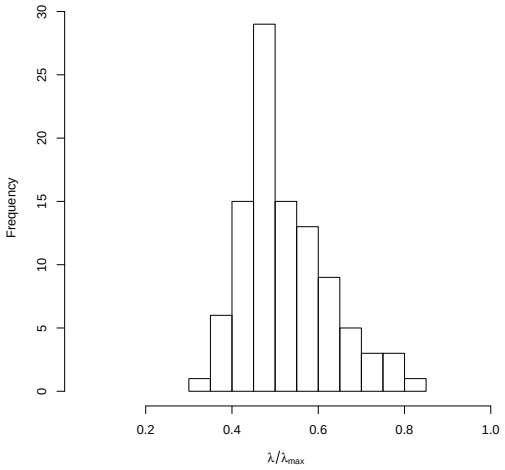
Il est également intéressant de comparer le choix de l'utilisation d'un test statistique à celui d'un seuillage dur des coefficients estimés, c'est-à-dire de mettre à 0 les coefficients dont la valeur

Tableau 4.2 – Qualité de la sélection de différentes méthodes de choix de λ pour le cadre 3 de simulation pour $n = 100$ et $p = 500$. Dans le tableau, *VP* et *FP* correspondent respectivement au nombre de vrais positifs et faux positifs. $\hat{\lambda}_{\text{RM}}$ correspond à la valeur de λ maximisant le nombre de rejets. *MLGL* correspond au *Multi-Layer Group-Lasso* et *THM* à la procédure de test hiérarchique multiple. Ainsi *MLGL* + $\hat{\lambda}_{\text{RM}}$ correspond à la composition du chemin solution pour $\lambda = \hat{\lambda}_{\text{RM}}$. *MLGL* + *THM* + $\hat{\lambda}_{\text{RM}}$ à celle du chemin solution pour $\lambda = \hat{\lambda}_{\text{RM}}$ après application de la procédure de test hiérarchique multiple. $\hat{\lambda}_{\text{vc 5-f}}$ correspond à la valeur de λ choisie par validation croisée *5-fold*, $\hat{\lambda}_{\text{Kappa}}$ à celle choisie par critère Kappa et *stability* à la *stability selection* (cf. Section 1.1.4). K, l et ρ désignent les paramètres du cadre 3 de simulation (Section 2.2.1). K désigne le nombre de groupes appartenant au support de la vraie solution β^*, l désigne la taille des blocs de la matrice de variance-covariance et ρ la corrélation intra-blocs.

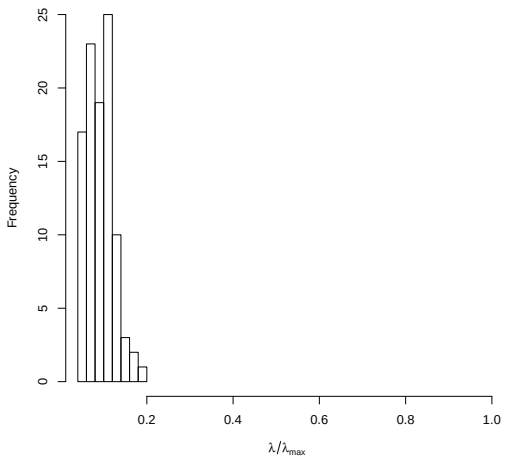
		$K = 5$				$K = 10$			
		$l = 5$		$l = 10$		$l = 5$		$l = 10$	
		VP	FP	VP	FP	VP	FP	VP	FP
$\rho = 0.9$	MLGL + $\hat{\lambda}_{\text{RM}}$	4.95	7.38	4.89	9.08	6.09	9.44	6.61	11.61
	MLGL + THM + $\hat{\lambda}_{\text{RM}}$	4.91	0.28	4.78	0.8	4.15	0.44	4.75	1.28
	MLGL + $\hat{\lambda}_{\text{Kappa}}$	3.66	2.14	4.3	2.64	5.78	13.25	5.84	8.06
	MLGL + $\hat{\lambda}_{\text{vc 5-f}}$	4.96	24.37	4.87	23.4	8.15	30.46	7.74	27.21
	MLGL + stability	4.99	0.15	5	0.4	7.4	0.22	9.92	0.22
$\rho = 0.7$	MLGL + $\hat{\lambda}_{\text{RM}}$	3.35	3.29	4.13	2.02	2.49	2.76	5.43	2.22
	MLGL + THM + $\hat{\lambda}_{\text{RM}}$	2.95	0.51	3.95	0.27	1.59	0.64	3.39	0.54
	MLGL + $\hat{\lambda}_{\text{Kappa}}$	2.59	1.68	3.86	1.21	2.76	4.36	6.22	3.29
	MLGL + $\hat{\lambda}_{\text{vc 5-f}}$	3.73	6.32	4.36	5.33	3.35	6.35	6.46	4.55
	MLGL + stability	4.52	0.61	5	1.79	3.63	0.84	9.8	1.58
$\rho = 0.5$	MLGL + $\hat{\lambda}_{\text{RM}}$	3.1	2.33	3.03	2.96	2.64	4.05	2.75	3.18
	MLGL + THM + $\hat{\lambda}_{\text{RM}}$	2.89	0.32	2.6	0.53	1.68	0.41	1.53	0.63
	MLGL + $\hat{\lambda}_{\text{Kappa}}$	3.19	3.83	3.08	3.32	2.93	5.50	3.56	5.34
	MLGL + $\hat{\lambda}_{\text{vc 5-f}}$	3.55	6.73	3.49	6.28	3.34	7.67	3.72	5.71
	MLGL + stability	3.17	1.17	4.85	1.61	2.54	1.79	8.01	1.51



(a) Test hiérarchique multiple + $\hat{\lambda}_{RM}$



(b) méthode Kappa



(c) validation croisée 5-fold

FIGURE 4.1 – Valeurs de $\hat{\lambda}$ sélectionnées par différentes méthodes dans le cadre 3 de simulation pour $p = 500$, $\rho = 0.9$, $K = 5$ et $l = 5$.

Tableau 4.3 – Qualité de la sélection de différentes méthodes de choix de $\hat{\lambda}$ pour le cadre 3 de simulation. Les méthodes Kappa, validation croisée *5-fold* sont utilisées pour estimer $\hat{\lambda}$. Pour cette valeur, la procédure de test hiérarchique multiple (THM) est appliquée. Dans le tableau, *VP* et *FP* correspondent respectivement au nombre de vrais positifs et faux positifs. K désigne le nombre de groupes appartenant au support de la vraie solution β^*, l désigne la taille des blocs de la matrice de variance-covariance et ρ la corrélation intra-blocs.

		$K=5$				$K=10$			
		$l=5$		$l=10$		$l=5$		$l=10$	
		VP	FP	VP	FP	VP	FP	VP	FP
$\rho=0.9$	MLGL + THM + $\hat{\lambda}_{\text{RM}}$	4.91	0.28	4.78	0.8	4.15	0.44	4.75	1.28
	MLGL + $\hat{\lambda}_{\text{Kappa}}$ + THM	3.46	0	4.11	0.01	1.94	0.02	2.87	0.21
	MLGL + $\hat{\lambda}_{\text{vc 5-f}}$ + THM	3.94	0.05	4.13	0.21	0.98	0.16	1.97	0.74
$\rho=0.7$	MLGL + THM + $\hat{\lambda}_{\text{RM}}$	2.95	0.51	3.95	0.27	1.59	0.64	3.39	0.54
	MLGL + $\hat{\lambda}_{\text{Kappa}}$ + THM	1.85	0.18	3.25	0.17	0.79	0.34	2.32	0.43
	MLGL + $\hat{\lambda}_{\text{vc 5-f}}$ + THM	2.73	0.27	3.57	0.21	0.94	0.41	2.36	0.47
$\rho=0.5$	MLGL + THM + $\hat{\lambda}_{\text{RM}}$	2.89	0.32	2.6	0.53	1.68	0.41	1.53	0.63
	MLGL + $\hat{\lambda}_{\text{Kappa}}$ + THM	2.56	0.17	2.07	0.24	1.09	0.25	1.09	0.39
	MLGL + $\hat{\lambda}_{\text{vc 5-f}}$ + THM	2.87	0.20	2.33	0.26	1.13	0.31	1.04	0.42

Tableau 4.4 – Nombre de vrais groupes pouvant être sélectionnés par une méthode de seuillage sur les coefficients sans sélectionner de faux groupes.

	$K=5$		$K=10$	
	$l=5$	$l=10$	$l=5$	$l=10$
$\rho=0.9$	4.91	4.66	4.3	3.3
$\rho=0.7$	3.29	4.09	2.31	4.72
$\rho=0.5$	3.33	3.14	2.48	2.56

est proche de 0. Un seuil positif est fixé par l'utilisateur, tout coefficient (en valeur absolue) inférieur à ce seuil est considéré comme nul. Le seuillage de coefficients pose cependant plusieurs problèmes : il est nécessaire de choisir une valeur de seuil et il n'est pas possible de gérer une hiérarchie de groupes. S'il y a présence de hiérarchie, seules les feuilles de l'arbre seront utilisées. Pour le choix du seuil, des méthodes de validation croisée peuvent être utilisées, mais nous allons comparer ici à un seuil théorique : le plus grand seuil permettant de ne sélectionner que des vrais positifs. Pour chaque valeur de λ, un seuil différent est calculé. Le nombre maximal de vrais positifs obtenu pour les différentes simulations est dans le Tableau 4.4. On peut comparer ce nombre au nombre de vrais positifs trouvés par notre procédure dans le Tableau 4.1. On remarque que le seuillage optimal permet de trouver un nombre plus important de vrais positifs que notre méthode avec une différence plus grande dans les cas de faible corrélation. L'utilisation d'un seuillage semble donc intéressante mais elle nécessité une procédure pour choisir le seuil de manière optimale.

4.2.3 Comparaison avec les méthodes présentées en section 1.2.2

L'algorithme 14 présente notre stratégie complète comportant *Multi-Layer Group-Lasso*, test hiérarchique multiple et choix du paramètre de régularisation λ. Dans cette section, nous nous intéressons aux résultats des procédures présentées dans la Section 1.2.2 : le *Cluster Representative Lasso*, le *Cluster Group-Lasso*, le *Supervised Group-Lasso* et *Hierarchical Clustering and Averaging for Regression*. Ces méthodes utilisent toutes une validation croisée *V-fold* avec comme critère l'erreur de prédiction pour choisir la valeur optimale de λ. Nous utilisons $V = 10$ pour la validation croisée et $\alpha = 0.05$ avec contrôle du FWER pour notre procédure de test hiérarchique multiple. Les données sont générées suivant le cadre 1 de simulation avec une corrélation intra-blocs $\rho = 0.7$, des blocs de taille $l = 5$ et $K = 10$ vraies variables, différentes valeurs de n et p sont testées.

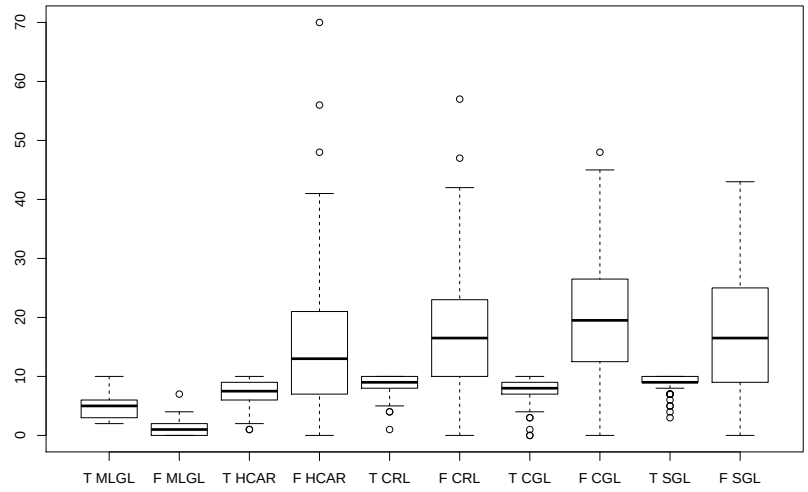
Résultats de la sélection

Sur la figure 4.2, les résultats sont présentés pour $p = 500$ et $n = 100$ et $= 200$. De manière générale, le *Multi-Layer Group-Lasso* avec test hiérarchique multiple propose un meilleur compromis entre le nombre de faux positifs et de vrais positifs. Les autres procédures sélectionnent en moyenne un nombre plus important de faux positifs que de vrais positifs. Quand le nombre d'individus augmente, on constate que toutes les procédures trouvent un plus grand nombre de vrais positifs. Pour le *Multi-Layer Group-Lasso*, le nombre de faux positifs diminue quand n augmente, on ne constate pas cet effet pour les autres procédures.

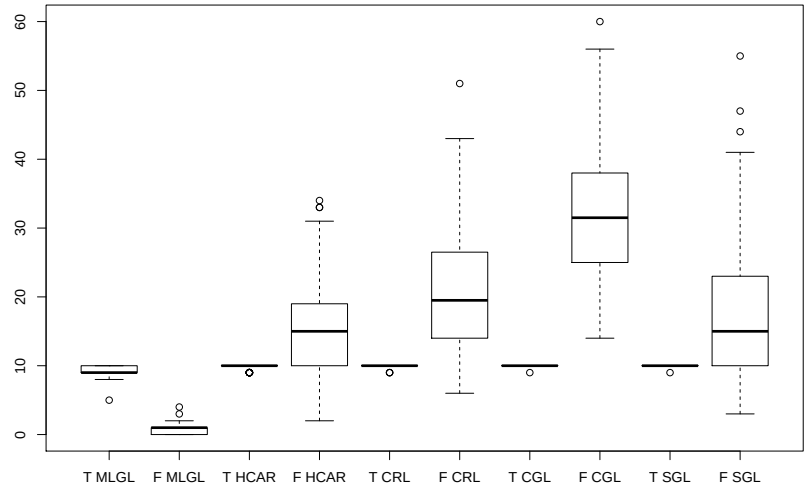
Temps de calcul Les temps de calcul des différentes procédures ont été calculé en utilisant les algorithmes de résolution du package R `glmnet` (Friedman, Hastie et Robert Tibshirani 2010b) pour résoudre le lasso, `ggl asso` (Yang et Zou 2015) pour le group-lasso et `standGL` (Simon 2011) pour le *Standardized Group-Lasso*. La validation croisée a été effectuée en utilisant les fonctions de ces packages. La même CAH a été utilisée pour ces méthodes.

Sur la figure 4.3 est représenté le temps moyen sur 100 répétitions pour les différentes méthodes. On constate que la méthode la plus rapide est le *Cluster Representative Lasso*. En effet, cette méthode n'effectue qu'un seul lasso sur une matrice de taille réduite. Malgré l'utilisation du lasso, la méthode *HCAR* est très lente. Elle comporte l'exécution de p algorithmes de résolution du lasso et de validation croisée. Le Cluster group-lasso est également très lent mais ne travaille qu'avec un unique niveau de la CAH. L'algorithme de résolution du *Standardized Group-Lasso* est codé uniquement en R (à l'inverse des autres algorithmes), ce qui n'est pas optimal en temps d'exécution pour un algorithme itératif. Le *Multi-Layer Group-Lasso* a un temps d'exécution assez important par rapport au *Cluster Representative Lasso* et assez proche de celui du *Supervised Group-Lasso*. Le temps assez élevé du *Supervised Group-Lasso* s'explique par l'exécution de validation croisée *V-fold* pour chaque groupe de la partition du niveau choisi de la CAH.

***Multi-Layer Group-Lasso* et test hiérarchique multiple** Observons plus en détail le temps de calcul de la procédure du *Multi-Layer Group-Lasso* avec test hiérarchique multiple. Sur la figure 4.4 est représenté le temps d'exécution des trois différentes parties : la CAH, le group-lasso sur l'ensemble des partitions et le test hiérarchique multiple. On remarque que la partie la plus coûteuse est la procédure de test hiérarchique multiple. Cette partie est codée uniquement en R. Ses performances pourrait être améliorées par une implémentation en C++ et la réutilisation des calculs. En effet, j'ai fait le choix de faire une fonction utilisable pour une unique valeur de λ. Par exemple, entre les valeurs de λ pour lesquelles les groupes sélectionnés par le *Multi-Layer Group-Lasso* changent, les calculs (ceux pour obtenir les composantes principales notamment) ne sont pas réutilisés. La mise en place de la fonction de test hiérarchique pour l'ensemble du

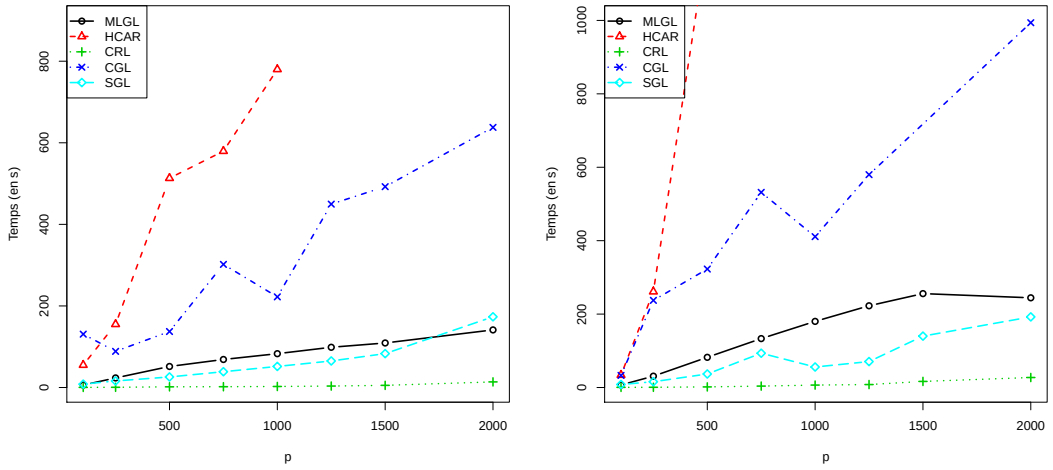


(a) $n = 100$, $p = 500$, $\rho = 0.7$, $l = 5$, $K = 10$



(b) $n = 200$, $p = 500$, $\rho = 0.7$, $l = 5$, $K = 10$

FIGURE 4.2 – Sélection de différentes procédures combinant classification et group-lasso : le *Multi-Layer Group-Lasso* avec procédure de test hiérarchique multiple (MLGL), *Hierarchical Clustering and Averaging for Regression* (HCAR), *Cluster Representative Lasso* (CRL), Cluster Group-Lasso (CGL) et *Supervised Group-Lasso* (SGL). *T* correspond au nombre de vrais positifs et *F* au nombre de faux positifs.



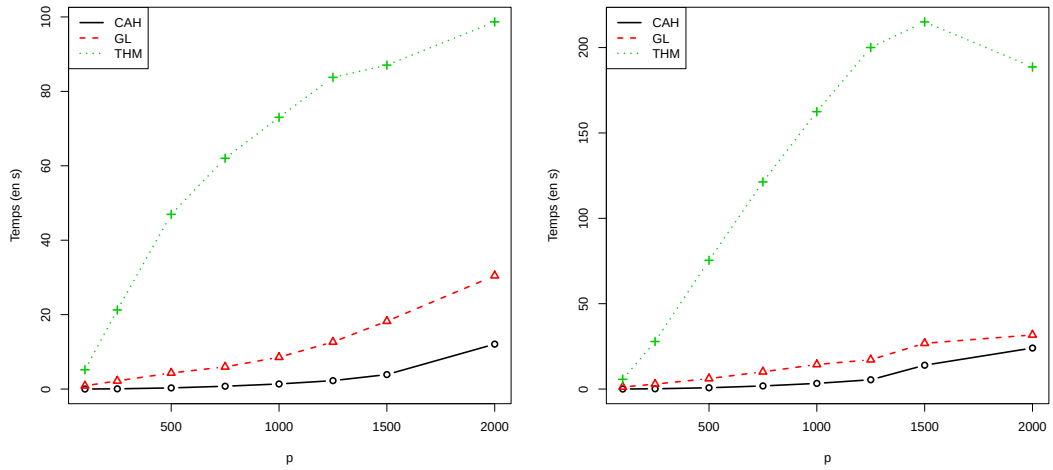
(a) $n = 100$, $\rho = 0.7$, $l = 5$, $K = 10$ (b) $n = 200$, $\rho = 0.7$, $l = 5$, $K = 10$

FIGURE 4.3 – Temps d'exécution de différentes procédures combinant classification et régression : le *Multi-Layer Group-Lasso* avec procédure de test hiérarchique multiple (MLGL, cercle noir), *Hierarchical Clustering and Averaging for Regression* (HCAR, triangle rouge), *Cluster Representative Lasso* (CRL, + vert), Cluster Group-Lasso (CGL, croix bleu foncé) et *Supervised Group-Lasso* (SGL, losange bleu clair). Les courbes sont la moyenne de 100 répétitions.

chemin solution afin de réutiliser est une perspective à court terme pour améliorer le temps de calcul.

4.3 Conclusion

L'objectif de ce chapitre était de proposer un choix du paramètre de λ optimal pour le *Multi-Layer Group-Lasso* après application de la procédure de Test Hiérarchique Multiple. La procédure de test se révélant assez conservative, le choix du paramètre λ s'est tourné vers celui maximisant le nombre de rejets. Nous avons montré sur simulations que ce choix de λ s'approchait du choix optimal en matière de vrais positifs. Les méthodes de choix de paramètres (validation croisée *V-fold*, critère Kappa) donnent de moins bons résultats en matière de compromis entre vrais et faux positifs. Couplées à la procédure de test hiérarchique multiple, ces méthodes produisent des résultats similaires à ceux de la valeur de λ maximisant le nombre de rejets, mais un en temps conséquent dû à la réexécution multiple de l'algorithme de résolution. Le *Multi-Layer Group-Lasso* avec Test Hiérarchique Multiple surclasse les méthodes combinant regroupement de variables et sélection mais est plus lente que certaines de ces méthodes notamment à cause de la procédure de test hiérarchique multiple. Toutefois, une optimisation de son implémentation est tout à fait possible.



(a) $n = 100$, $\rho = 0.7$, $l = 5$, $K = 10$ (b) $n = 200$, $\rho = 0.7$, $l = 5$, $K = 10$

FIGURE 4.4 – Temps d'exécution des différentes étapes de l'algorithme 14 : la Classification Ascendante Hiérarchique (CAH, cercle noir), le group-lasso sur l'ensemble des partitions (GL, triangle rouge) et le Test Hiérarchique Multiple (THM, + vert).

Chapitre 5

Packages implémentés

5.1 MLGL

Les méthodes présentées dans cette thèse ont été implémentées dans un package R nommé `MLGL` pour *Multi-Layer Group-Lasso*. Le package est disponible sur R-forge : `http://r-forge.r-project.org/R/?group_id=2223` et prochainement sur le CRAN.

Pour effectuer la classification ascendante hiérarchique (CAH), le package `fastcluster` (Müllner 2013) a été utilisé au lieu de la fonction `hclust` de R ou d'autres packages tels que `cluster` (Maechler et al. 2015) et `flashClust` (Langfelder et Horvath 2012). Ce package fournit une implémentation C++ interfacée avec R de la CAH avec une complexité moindre que celle de base ($O(p^2)$ au lieu de $O(p^3)$) pour obtenir le même résultat. Il surpasse également les autres packages (`cluster`, `flashClust`).

Différents packages sont disponibles pour la résolution du group-lasso : `gglasso` (Yang et Zou 2015), `grplasso` (Meier, S. Van De Geer et Bühlmann 2008) et `SGL` (Simon, Friedman et al. 2013). Le choix s'est porté sur le package `gglasso`. Il propose une implémentation en fortran d'un algorithme de descente pour résoudre le group-lasso. La rapidité et l'exactitude des résultats fournis comparés aux autres packages en font le meilleur pour la résolution du group-lasso.

La procédure de tests hiérarchiques pour le chemin solution du *Multi-Layer Group-Lasso* a quant à elle été implémentée en R. Elle utilise le package `FactoMineR` (Lê, Josse et Husson 2008) pour calculer la composante principale des groupes sélectionnés.

Dans la suite, nous détaillons les principales fonctions et proposons un exemple illustratif afin de faciliter l'application des méthodes présentées dans cette thèse.

5.1.1 Principales fonctions

La fonction principale est la fonction `MLGL`, elle implémente la méthode (2.6) proposée au chapitre 2. Elle effectue trois tâches :

1. CAH avec distance euclidienne et méthode de Ward sur la matrice X ;
2. Calcul des poids ρ^u correspondant aux différentes longueurs de branches ;
3. Résolution de group-lasso sur l'ensemble des partitions.

Ses principaux arguments sont :

X la matrice de design de taille $n \times p$;

y le vecteur réponse de longueur n ;

hc le résultat d'une CAH (objet de type `hclust` qui peut être obtenu à partir de la fonction `hclust` de R ou celle du package `fastcluster`). Si l'utilisateur ne fournit pas de CAH, une CAH avec distance euclidienne et méthode de Ward sur la matrice X est effectuée ;

lambda les différentes valeurs du paramètre de régularisation ;

weightLevel les poids ρ_s stockés dans un vecteur de taille p (= nombre de niveaux de la hiérarchie). Le 1er élément correspond au niveau contenant p groupes ;

weightGroup poids correspondant aux groupes (par défaut la racine carrée de la taille des groupes) ;

intercept un booléen spécifiant si une constante doit être ajoutée dans le modèle.

Seuls les deux premiers arguments sont obligatoires. Les autres tels que les poids ou la CAH peuvent être calculés par la fonction avec les valeurs présentées dans cette thèse. Par exemple, le paramètre `weightLevel` utilise par défaut le choix énoncé équation (2.7).

La sortie de la fonction est une liste dont les principaux éléments sont :

lambda vecteur contenant les différentes valeurs du paramètre de régularisation ;

var une liste dont le i^e élément correspond aux indices des variables sélectionnées pour la i^e valeur de λ ;

group une liste dont le i^e élément correspond aux indices du groupe des variables sélectionnées pour la i^e valeur de λ ;

beta une liste dont le i^e élément correspond aux valeurs des coefficients estimés pour la i^e valeur de λ ;

Par exemple, la sortie ci-dessous indique que pour la 10e valeur contenue dans le vecteur `lambda`, 3 groupes sont sélectionnés (les groupes d'indices 1, 2 et 3). Le 1er groupe se compose des variables d'indice 1 et 2 dans la matrice X, le 2e de celles d'indices 3, 4 et 5 et le dernier de celles d'indices 6, 7 et 8. Les coefficients associés à ces variables se trouvent dans `$beta[[10]]`.

```
$group[[10]]
[1] 1 1 2 2 2 3 3 3

$var[[10]]
[1] 1 2 3 4 5 6 7 8

$beta[[10]]
[1]  0.25  0.28  0.32  0.05 -0.1  0.84 -0.1  0.5
```

La sortie peut être mise sous forme matricielle en utilisant la fonction `listToMatrix`. Le chemin solution peut être représenté à l'aide de la fonction `plot`.

Les fonctions `hierarchicalFWER` et `hierarchicalFDR` permettent de réaliser le processus de test hiérarchique (cf. algorithme 12) pour une sélection associée à une valeur de λ. Ces deux fonctions partagent les mêmes arguments :

X la matrice de design de taille $n \times p$;

y le vecteur réponse de longueur n ;

group structure de groupes des variables sélectionnées (par exemple vecteur `$group[[i]]` de la sortie de la fonction `MLGL`) ;

var indice des variables sélectionnées (par exemple vecteur `$var[[i]]` de la sortie de la fonction `MLGL`) ;

test fonction pour tester la nullité des coefficients. Le package fournit deux fonctions : `partialFtest` et `Ftest`. La fonction comprend 3 arguments : X, y et un vecteur contenant les indices des variables dont la nullité est à tester. Elle retourne une p-valeur.

Les éléments de la sortie de ces fonctions sont à utiliser avec les fonctions `selFWER` et `selFDR` afin d'obtenir les groupes sélectionnés à un niveau de contrôle désiré. Pour la fonction `selFDR`, il est possible de choisir le contrôle du FDR sur l'ensemble de l'arbre (paramètre `outer = FALSE`) ou seulement sur les *outer nodes* (`outer = TRUE`) ainsi que de préciser si le niveau de contrôle α fourni est celui s'appliquant à chaque famille testée (option `global = FALSE`) ou celui souhaité au global (`global = TRUE`).

La fonction `fullProcess` implémente l'ensemble du processus présentée dans cette thèse, c'est-à-dire : la classification ascendante hiérarchique sur la moitié des individus, le calcul des poids, le group-lasso sur l'autre moitié des individus, le post-traitement à un niveau de contrôle spécifié sur la 1re moitié des individus et le choix de λ (cf. algorithme 13).

Ses arguments sont :

X la matrice de design de taille $n \times p$;

y le vecteur réponse de longueur n ;

control "FWER" ou "FDR" pour le contrôle désiré pour la méthode de post-traitement ;

alpha le niveau de contrôle désiré ;

test la fonction utilisée pour tester la nullité des coefficients ;

hc sortie de la CAH ;

plot si TRUE, un graphique montrant le nombre de groupes sélectionnés avant et après la phase de post-traitement est affiché.

Cette fonction retourne une liste contenant :

res sortie de la fonction `MLGL` ;

lambdaOpt valeur de λ maximisant le nombre de rejets ;

var indices des variables sélectionnées ;

group structure de groupes des variables sélectionnées.

D'autres fonctions sont disponibles dans le package. On notera la fonction `cv.MLGL` pour réaliser une validation croisée V-fold, la fonction `stability.MLGL` pour réaliser une stability selection.

5.1.2 Exemple illustratif

Tout d'abord, on simule 50 individus suivant une loi normale multivariée avec matrice de variance-covariance par blocs (comme dans le cadre 1 de simulation, cf. section 2.2.1) à l'aide de la fonction `simuBlockGaussian`. Cette matrice contient 12 blocs de taille 5 avec un corrélation intra-blocs de 0.7.

```
X <- simuBlockGaussian(50, 12, 5, 0.7)
```

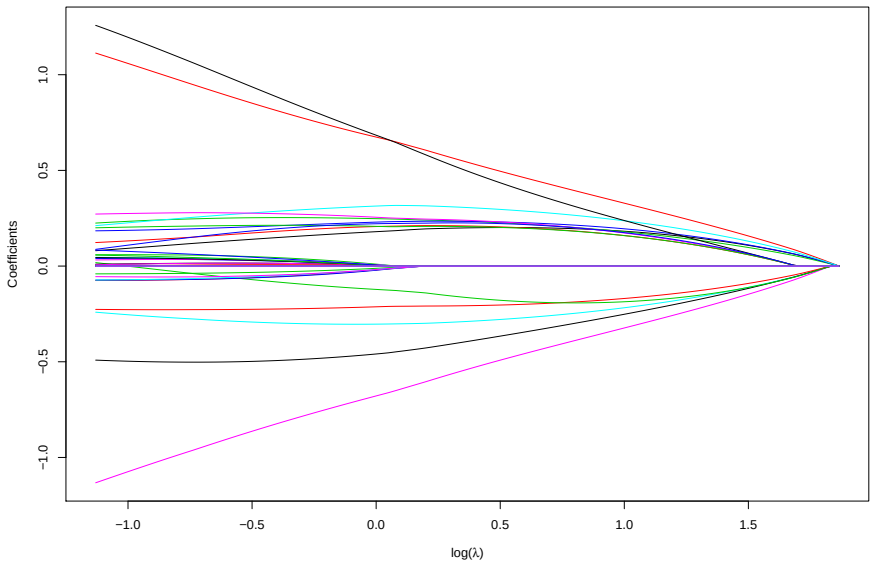


FIGURE 5.1 – Sortie graphique du chemin solution obtenue à l'aide de la fonction `plot` sur la sortie de la fonction `MLGL`.

On génère ensuite une réponse y de la forme $y = X\beta^* + \epsilon$ où ϵ suit une loi normale. Le support de la solution β^* est $\{2,7,12\}$.

```
y <- drop(X[,c(2,7,12)]%*%c(2,2,-2) + rnorm(50,0,0.5))
```

On effectue une CAH et le group-lasso sur l'ensemble de la CAH à l'aide de la fonction `MLGL`.

```
res <- MLGL(X, y)
```

On peut visualiser le chemin solution à l'aide la fonction `plot` (figure 5.1). Chaque courbe représente l'évolution de l'estimation d'un coefficient en fonction du paramètre de régularisation λ.

```
plot(res)
```

Si l'on souhaite appliquer la procédure de test hiérarchique avec contrôle du FWER pour la 30e valeur de λ avec un niveau de contrôle $\alpha = 0.05$, on exécute le code suivant :

```
outFWER <- hierarchicalFWER(X, y, res$group[[30]], res$var[[30]],
                            partialFtest)
sel <- selFWER(outFWER, alpha = 0.05)
```

On obtient :

```
> sel
$toSel
[1] TRUE TRUE TRUE

$groupId
[1] 85 93 94
```

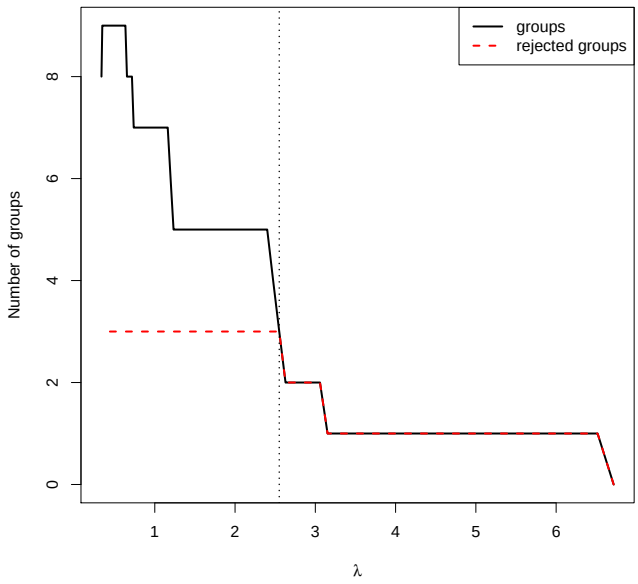


FIGURE 5.2 – Sortie graphique de la fonction `fullProcess` représentant le nombre de groupes sélectionnés avant (en noir) et après (en rouge) la phase de post-traitement. Le trait vertical indique la valeur de λ maximisant le nombre de rejets.

où `groupId` correspond aux numéros des groupes sélectionnés du vecteur `res$group[[30]]`.

Le processus complet (cf. algorithme 13) peut être exécuté de la manière suivante :

```
resFull <- fullProcess(X, y, plot = TRUE)
```

On obtient le graphe suivant (figure 5.2) représentant le nombre de groupes sélectionnés par la méthode avant (en noir) et après (en rouge) la phase de test hiérarchique. Un trait vertical indique la valeur de λ maximisant le nombre de rejets (valeur stockée dans `$lambdaOpt`).

On récupère les sorties suivantes détaillant les groupes sélectionnés et leurs structures en variables :

```
> resFull$lambdaOpt
[1] 2.550467

> resFull$selectedGroups
[1] 81 86 99

> resFull$group
[1] 81 81 81 81 81 86 86 86 86 86 99 99 99 99 99

> resFull$var
[1]  1  2  3  4  5 11 12 13 14 15  6  7  8  9 10
```

En conclusion, le package R `MLGL` fournit un moyen simple d'utiliser les méthodes proposées dans cette thèse tout en laissant la possibilité aux utilisateurs de changer certains paramètres importants comme la CAH, les poids, ...Il propose également d'autres outils comme la *stability selection*, la validation croisée *V-fold*, le *group-lasso with overlap* et une procédure de test pour group-lasso.

5.2 HDpenReg et MPAgenomics

Ces deux packages sont issus de l'application qui a générée les questions de la thèse. Initialement développés durant mon travail d'ingénieur au sein de l'équipe MΘDAL d'Inria dans le cadre de l'Action de Développement Technologique *MPAgenomics*. L'objectif était de développer une suite intégrée d'outils logiciels permettant d'analyser conjointement les données de puces de génotypage d'un grand nombre de patients.

Les packages sont disponibles sur le CRAN aux adresses `https://cran.r-project.org/package=HDPenReg` et `https://cran.r-project.org/package=MPAgenomics` et également sur R-forge : `https://r-forge.r-project.org/R/?group_id=1658`.

Description

`HDPenReg`

Le package contient une implémentation en C++ de l'algorithme lars pour résoudre le lasso. Cette implémentation se veut plus rapide que le package `lars` généralement utilisé en R. `HDPenReg` contient également un algorithme EM permettant de résoudre le fused-lasso et le lasso.

`MPAgenomics`

`MPAgenomics` est un package R permettant de traiter conjointement des données génomique d'un grand nombre de patients. Ces données sont obtenues à l'aide de puces de génotypage (type Affymetrix SNP 6.0). Le package permet, partant des données brutes, d'effectuer l'ensemble du traitement de ces données : extraction des signaux, normalisation (débruitage), segmentation des signaux, sélection de marqueurs. Il interface des packages existants. Le package a été conçu afin que des néophytes puissent l'utiliser en facilitant l'interaction entre les différentes fonctions des packages et en gérant l'optimisation de certains paramètres.

Le package a abouti à une publication dans *BMC Bioinformatics*. L'article est disponible en Annexe C.

Exemples

Des tutoriels sont disponibles sous la forme de vignette au sein de ces deux packages.

Le tutoriel de `HDPenReg` (9 pages) présente les principales fonctions pour utiliser l'algorithme Lars et un exemple d'utilisation. Il est accessible dans R via les commandes :

```
> library("HDPenReg")
> vignette("HDPenReg")
```

Le tutoriel de `MPAgenomics` (27 pages) présente la procédure d'installation des différents packages qu'il interface, les différentes fonctions de l'analyse et un exemple complet reproductible sur un jeu de données réelles disponible en ligne. Il est accessible dans R via les commandes :

```
> library("MPAgenomics")
> vignette("MPAgenomics")
```

5.3 Rankcluster

Le package R Rankcluster a été réalisé durant mon stage de fin d'études au sein de l'équipe MΘDAL d'Inria. Ce travail a débouché sur la publication d'un article :

Julien Jacques, Quentin Grimonprez et Christophe Biernacki (2014). « Rankcluster : An R Package for Clustering Multivariate Partial Rankings ». In : *The R Journal* 6.1.

Le package Rankcluster est disponible sur le CRAN : https://cran.r-project.org/package=Rankcluster et sur R-forge : https://r-forge.r-project.org/R/?group_id=1650.

Description

Rankcluster implémente en C++ une méthode de clustering de données de rang. Une donnée de rang est le résultat d'un classement de plusieurs objets selon un ordre. Par exemple, supposons qu'une personne doit classer selon son ordre de préférence les trois destinations de vacances suivantes : $\mathcal{O}_1$=Mer, $\mathcal{O}_2$=Montagne, $\mathcal{O}_3$=Campagne ; le résultat de ce classement est une donnée de rang. Le clustering se fait à l'aide d'un modèle de mélange dont les paramètres sont estimés par un algorithme SEM-Gibbs (Stochastic Expectation-Maximization). L'algorithme permet de prendre en compte les rangs partiels (seul un sous-ensemble des objets ont été ordonnés) et les données de rangs multivariées (différentes données de rang faites par un même individu).

Exemples

Des exemples avec code R sont fournis au sein de l'article qui est également disponible sous forme de vignette au sein du package.

```
> library("Rankcluster")
> vignette("Rankcluster")
```

Conclusion

Conclusion

Dans le cadre de cette thèse, nous nous sommes intéressés à la sélection de variables en présence de corrélation dans le but de sélectionner les variables influentes pour l'explication d'une variable donnée. Dans ce cadre, le regroupement de variables permet d'augmenter la qualité de la sélection. Un des problèmes du regroupement de variables est de choisir le nombre de groupes de la partition désirée. Les méthodes usuelles utilisent généralement une unique partition des variables pour l'étape de sélection. Un mauvais choix de cette partition peut alors engendrer de mauvaises performances en matière de sélection. Nous avons proposé d'utiliser un ensemble de partitions pour réaliser la sélection. La méthode proposée, le *Multi-Layer Group-Lasso*, utilise l'ensemble des partitions issues d'une classification ascendante hiérarchique (CAH) dans un group-lasso. Ainsi, le *Multi-Layer Group-Lasso* peut sélectionner des groupes issus de différentes partitions en adéquation avec la variable à expliquer. Chaque partition est associée à un poids au sein de la pénalité décrivant sa qualité au regard d'un critère choisi. Cela permet d'orienter la sélection vers les partitions d'intérêts. La méthode proposée a été comparée aux méthodes usuelles et son efficacité prouvée.

Le *Multi-Layer Group-Lasso* dépend d'un paramètre de régularisation à choisir. Plus ce paramètre est grand, plus le nombre de groupes sélectionnés est faible. Une procédure de test hiérarchique multiple a été développée afin de choisir au mieux ce paramètre et les groupes sélectionnés. La procédure proposée ne demande pas d'exécutions multiples de l'algorithme de résolution comme la validation croisée *V-fold* ou la *stability selection*. Elle se base sur des tests statistiques et permet de prendre en compte la spécificité de notre group-lasso qui est la possibilité de sélectionner des groupes chevauchant. Le test utilisé permet de contrôler le Family-Wise Error Rate (FWER) ou le False Discovery Rate (FDR). Cette procédure de test peut aussi être appliquée dans le cas du group-lasso classique. La procédure de test hiérarchique multiple sélectionne un faible nombre de faux positifs mais reste très conservative en matière de vrais positifs.

L'ensemble des méthodes proposées dans cette thèse a été implémenté dans le package R `MLGL` disponible en ligne. Il implémente ces méthodes en se basant sur des packages ayant fait leurs preuves pour leur rapidité d'exécution et leur exactitude. Des fonctions pour chaque méthode de cette thèse sont présentes ainsi qu'une fonction permettant d'exécuter le processus complet (classification ascendante hiérarchique, *Multi-Layer Group-Lasso* et test hiérarchique multiple). Des sorties graphiques agrémentent quelques fonctions.

Perspectives

Dans cette thèse, nous nous sommes intéressés au cadre de la régression linéaire, c'est-à-dire quand la variable à expliquer est quantitative. Le *Multi-Layer Group-Lasso* peut être utilisé dans le cas de la régression logistique. En effet, le group-lasso a été étendu au cas de la régression logistique et un algorithme de résolution développé (Meier, S. Van De Geer et Bühlmann 2008). Basé sur cet algorithme, le cas du group-lasso avec groupes chevauchant a été énoncé dans (Zeng et Breheny 2015). Il en va de même pour la méthode de test hiérarchique multiple qui n'a besoin que d'un test statistique et que le nombre de groupes sélectionnés reste inférieur au nombre d'individus. Une étude dans ce cas reste à faire, ainsi que de vérifier s'il est possible d'obtenir la même simplification de la taille de la matrice de design modifiée d'un point de vue théorique.

Le *Multi-Layer Group-Lasso* utilise un algorithme de classification non supervisée pour produire une hiérarchie des variables. Une autre perspective serait de prendre en compte la variable à expliquer une méthode de classification supervisée afin de produire cette hiérarchie. Cela permettrait de créer les partitions en prenant en compte la variable à expliquer et de potentiellement améliorer la sélection.

La qualité de la partition est ici mesurée à l'aide du critère du saut maximal. Ce critère bien que basique a prouvé son efficacité dans les simulations proposées. Mais dans des cas plus généraux, ce critère risque de ne pas être optimal. Par exemple, dans le cas où la matrice de variance-covariance est de type Toeplitz ($\text{cor}(X_i, X_j) = \rho^{|i-j|}$), ce critère ne permet pas de choisir une partition intéressante (généralement une partition en un seul groupe est choisie). Il est également possible d'utiliser une distance différente ou une classification ascendante hiérarchique à noyau pour obtenir des partitions intéressantes dans des cas comme celui-ci. Différentes CAH à noyau ont été testées sur nos simulations mais aucunes améliorations significatives n'ont été constatées. Le même constat est fait dans (Qin, Lewis et Noble 2003) sur des données réelles. Cependant, une étude plus approfondie dans des cas différents aurait un intérêt.

La procédure de test hiérarchique multiple proposée contrôle relativement bien le nombre de faux positifs mais reste conservative en matière de vrais positifs. Pour la rendre moins conservative, le contrôle du k-FWER au lieu de FWER peut être une solution. Dans le cas du FDR, nous avons énoncé que la borne proposée dans (Yekutieli 2008) paraissait grossière. Une étude théorique de celle-ci peut être une piste intéressante de recherche.

Bibliographie

Akaike, Hirotogu (1974). « A new look at the statistical model identification ». In : *IEEE Transactions on Automatic Control* 19.6, p. 716–723.

Anderson, Theodore (1984). *An Introduction to Multivariate Statistical Analysis*. 3rd Edition. Wiley. 752 p. isbn : 978-0-471-36091-9.

Arlot, Sylvain et Alain Celisse (2010). « A survey of cross-validation procedures for model selection ». In : *Statistics Surveys* 4, p. 40–79. issn : 1935-7516.

Arnold, Taylor et Ryan Tibshirani (2014). « Efficient Implementations of the Generalized Lasso Dual Path Algorithm ». In : arXiv : 1405.3222 [1]. url : http://arxiv.org/abs/1405.3222.

Bach, Francis R. (2008). « Consistency of the Group Lasso and Multiple Kernel Learning ». In : *J. Mach. Learn. Res.* 9, p. 1179–1225.

Benjamini, Yoav et Yosef Hochberg (1995). « Controlling the False Discovery Rate : A Practical and Powerful Approach to Multiple Testing ». In : *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1, p. 289–300.

Bland, J. Martin et Douglas G. Altman (1995). « Multiple significance tests : the Bonferroni method ». In : *BMJ* 310.6973, p. 170.

Bondell, Howard D. et Brian J. Reich (2008). « Simultaneous regression shrinkage, variable selection and clustering of predictors with OSCAR ». In : *Biometrics* 64.1, p. 115–123.

Bühlmann, Peter et Sara van de Geer (2011). *Statistics for High-Dimensional Data : Methods, Theory and Applications*. Springer Publishing Company, Incorporated.

Bühlmann, Peter, Markus Kalisch et Lukas Meier (2014). « High-Dimensional Statistics with a View Toward Applications in Biology ». In : *Annual Review of Statistics and Its Application* 1.1, p. 255–278.

Bühlmann, Peter, Philipp Rütimann et al. (2013). « Correlated variables in regression : clustering and sparse estimation ». In : *Journal of Statistical Planning and Inference* 143, p. 1835–3871.

Chen, Jiahua et Zehua Chen (2008). « Extended Bayesian information criteria for model selection with large model spaces ». In : *Biometrika* 95.3, p. 759–771.

Cohen, Jacob (1960). « A coefficient of agreement for nominal scales ». In : *Educational and Psychological Measurement* 20, p. 37–46.

Dunn, Olive Jean (1959). « Estimation of the Medians for Dependent Variables ». In : *Ann. Math. Statist.* 30.1, p. 192–197.

Efron, Bradley et al. (2004). « Least angle regression ». In : *Annals of Statistics* 32, p. 407–499.

Fan, Jianqing, Shaojun Guo et Ning Hao (2012). « Variance estimation using refitted cross-validation in ultrahigh dimensional regression ». In : *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 74.1, p. 37–65. issn : 1467-9868.

Fisher, Ronald A. (1936). « The use of multiple measurements in taxonomic problems ». In : *Annals of Eugenics* 7.2, p. 179–188. issn : 2050-1439.

1. http://arxiv.org/abs/1405.3222

Foster, Dean P. et Edward I. George (1994). « The Risk Inflation Criterion for Multiple Regression ». In : *The Annals of Statistics* 22.4, p. 1947–1975.

Friedman, Jerome, Trevor Hastie et Robert Tibshirani (2010a). « A note on the group lasso and a sparse group lasso ». In : arXiv : 1001.0736 [2]. url : http://arxiv.org/abs/1001.0736.

— (2010b). « Regularization Paths for Generalized Linear Models via Coordinate Descent ». In : *Journal of Statistical Software* 33.1.

Giraud, Christophe, Yannick Baraud et Sylvie Huet (2007). « Gaussian Model Selection with Unknown Variance ». url : https://hal.archives-ouvertes.fr/hal-00123420.

Goeman, Jelle J., Hans C. van Houwelingen et Livio Finos (2011). « Testing against a high-dimensional alternative in the generalized linear model : asymptotic type I error control ». In : *Biometrika* 98.2, p. 381–390.

Goeman, Jelle J., Sara A. Van De Geer et Hans C. Van Houwelingen (2006). « Testing against a high dimensional alternative ». In : *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 68.3, p. 477–493. issn : 1467-9868.

Grandvalet, Yves, Julien Chiquet et Christophe Ambroise (2012). *Sparsity by Worst-Case Quadratic Penalties*. Rapp. tech. arXiv preprint. url : http://arxiv.org/abs/1210.2077.

Grimonprez, Quentin et Serge Iovleff (2016). *HDPenReg : High-Dimensional Penalized Regression*. R package version 0.93.1.

Halkidi, Maria, Yannis Batistakis et Michalis Vazirgiannis (2002). « Clustering Validity Checking Methods : Part II ». In : *SIGMOD Rec.* 31.3, p. 19–27.

Hastie, Trevor, Robert Tibshirani et Jerome Friedman (2003). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. The Elements of Statistical Learning. Springer. isbn : 978-0-387-95284-0.

Hoefling, Holger (2010). « A Path Algorithm for the Fused Lasso Signal Approximator ». In : *Journal of Computational and Graphical Statistics* 19.4, p. 984–1006.

Hoerl, Arthur E. et Robert W. Kennard (1970). « Ridge Regression : Biased Estimation for Nonorthogonal Problems ». In : *Technometrics* 12.1, p. 55–67.

Höfling, Holger, Harald Binder et Martin Schumacher (2010). « A Coordinate-Wise Optimization Algorithm for the Fused Lasso ». In : arXiv : 1011.6409 [3]. url : http://arxiv.org/abs/1011.6409.

Holm, Sture (1979). « A Simple Sequentially Rejective Multiple Test Procedure ». In : *Scandinavian Journal of Statistics* 6.2, p. 65–70.

Hubert, Lawrence et Phipps Arabie (1985). « Comparing partitions ». In : *Journal of Classification* 2.1, p. 193–218.

Hurvich, Clifford et chin-ling Tsai (1989). « Regression and time series model selection in small samples ». In : *Biometrika* 76.2, p. 297–307.

Jacob, Laurent, Guillaume Obozinski et Jean-Philippe Vert (2009). « Group Lasso with Overlap and Graph Lasso ». In : *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. New York, NY, USA : ACM, p. 433–440.

Jain, Anil K., M. Narasimha Murty et Patrick J. Flynn (1999). « Data Clustering : A Review ». In : *ACM Comput. Surv.* 31.3, p. 264–323. issn : 0360-0300.

Jenatton, Rodolphe, Jean-Yves Audibert et Francis Bach (2011). « Structured Variable Selection with Sparsity-Inducing Norms ». In : *J. Mach. Learn. Res.* 12, p. 2777–2824. issn : 1532-4435.

Kim, Seyoung et Eric P. Xing (2012). « Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eQTL mapping ». In : *Ann. Appl. Stat.* 6.3, p. 1095–1117.

2. http://arxiv.org/abs/1001.0736
3. http://arxiv.org/abs/1011.6409

Langfelder, Peter et Steve Horvath (2012). « Fast R Functions for Robust Correlations and Hierarchical Clustering ». In : *Journal of Statistical Software* 46.11, p. 1–17.

Lê, Sébastien, Julie Josse et François Husson (2008). « FactoMineR : A Package for Multivariate Analysis ». In : *Journal of Statistical Software* 25.1, p. 1–18.

Lee, J. M. et al. (2001). « RNA expression analysis using an antisense Bacillus subtilis genome array ». In : *Journal of Bacteriology* 183.24, p. 7371–7380.

Leng, Chenlei, Yi Lin et Grace Wahba (2006). « A note on the lasso and related procedures in model selection ». In : *Statistica Sinica* 16.4, p. 1273–1284.

Liu, Han et Jian Zhang (2009). « Estimation Consistency of the Group Lasso and its Applications ». In : *JMLR*.

Lockhart, Richard et al. (2014). « A significance test for the lasso ». In : *The Annals of Statistics* 42.2, p. 413–468.

Ma, Shuangge, Xiao Song et Jian Huang (2007). « Supervised group Lasso with applications to microarray data analysis ». In : *BMC Bioinformatics* 8.1, p. 60.

Maechler, Martin et al. (2015). *cluster : Cluster Analysis Basics and Extensions*. R package version 2.0.3.

Meier, Lukas, Sara Van De Geer et Peter Bühlmann (2008). « The group lasso for logistic regression ». In : *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 70.1, p. 53–71.

Meinshausen, Nicolai (2008). « Hierarchical testing of variable importance ». In : *Biometrika* 95.2, p. 265–278.

Meinshausen, Nicolai et Peter Bühlmann (2010). « Stability selection ». In : *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 72.4, p. 417–473.

Meinshausen, Nicolai et Bin Yu (2009). « Lasso-type recovery of sparse representations for high-dimensional data ». In : *Ann. Statist.* 37.1, p. 246–270.

Milligan, Glenn W. et Martha C. Cooper (1985). « An examination of procedures for determining the number of clusters in a data set ». In : *Psychometrika* 50.2, p. 159–179.

Müllner, Daniel (2013). « fastcluster : Fast Hierarchical, Agglomerative Clustering Routines for R and Python ». In : *Journal of Statistical Software* 53.1, p. 1–18. issn : 1548-7660.

Natarajan, Balas K. (1995). « Sparse Approximate Solutions to Linear Systems ». In : *SIAM J. Comput.* 24.2, p. 227–234. issn : 0097-5397.

Osborne, Michael R., Brett Presnell et Berwin A. Turlach (1999). « On the LASSO and Its Dual ». In : *Journal of Computational and Graphical Statistics* 9, p. 319–337.

Park, Mee Young, Trevor Hastie et Robert Tibshirani (2007). « Averaged gene expressions for regression ». In : *Biostatistics* 8.2, p. 212–227.

Qin, Jie, Darrin P. Lewis et William Stafford Noble (2003). « Kernel hierarchical gene clustering from microarray expression data ». In : *Bioinformatics (Oxford, England)* 19.16, p. 2097–2104. issn : 1367-4803.

R Core Team (2016). *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. url : https://www.R-project.org/.

Rand, William M. (1971). « Objective Criteria for the Evaluation of Clustering Methods ». In : *Journal of the American Statistical Association* 66.336, p. 846–850.

Schwarz, Gideon (1978). « Estimating the Dimension of a Model ». In : *The Annals of Statistics* 6.2, p. 461–464.

Simon, Noah (2011). *standGL : Standardized Group Lasso*. R package version 1.1.1.

Simon, Noah, Jerome Friedman et al. (2013). « A sparse-group lasso ». In : *Journal of Computational and Graphical Statistics*.

Simon, Noah et Robert Tibshirani (2011). *Standardization and the group lasso penalty*. Rapp. tech.

Sokal, Robert R. et F. James Rohlf (1962). « The comparison of dendrograms by objective methods ». In : *Taxon* 11.2, p. 33–40.

Sun, Wei, Junhui Wang et Yixin Fang (2013). « Consistent Selection of Tuning Parameters via Variable Selection Stability ». In : *Journal of Machine Learning Research* 14, p. 3419–3440.

Szafranski, Marie, Yves Grandvalet et Pierre Morizet-Mahoudeaux (2007). « Hierarchical Penalization ». In : *Proceedings of the 20th International Conference on Neural Information Processing Systems*. NIPS'07, p. 1457–1464.

Tibshirani, Robert (1994). « Regression Shrinkage and Selection Via the Lasso ». In : *Journal of the Royal Statistical Society, Series B* 58, p. 267–288.

Tibshirani, Robert, Trevor Hastie et al. (1999). *Clustering methods for the analysis of DNA microarray data*. Rapp. tech.

Tibshirani, Robert, Michael Saunders et al. (2005). « Sparsity and smoothness via the fused lasso ». In : *Journal of the Royal Statistical Society Series B*, p. 91–108.

Tibshirani, Ryan J. (2013). « The lasso problem and uniqueness ». In : *Electronic Journal of Statistics* 7, p. 1456–1490.

Vaiter, Samuel et al. (2012). « The degrees of freedom of the Group Lasso for a General Design ».

Wainwright, Martin J. (2009). « Sharp Thresholds for High-dimensional and Noisy Sparsity Recovery Using L1-constrained Quadratic Programming (Lasso) ». In : *IEEE Trans. Inf. Theor.* 55.5, p. 2183–2202. issn : 0018-9448.

Wang, Jie et Jieping Ye (2015). « Multi-Layer Feature Reduction for Tree Structured Group Lasso via Hierarchical Projection ». In : *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., p. 1279–1287.

Ward, Joe H. (1963). « Hierarchical Grouping to Optimize an Objective Function ». In : *Journal of the American Statistical Association* 58.301, p. 236–244.

Wasserman, Larry et Kathryn Roeder (2009). « High-dimensional variable selection ». In : *Ann. Statist.* 37.5A, p. 2178–2201.

Witten, Daniela M., Ali Shojaie et Fan Zhang (2014). « The Cluster Elastic Net for High-Dimensional Regression With Unknown Variable Grouping ». In : *Technometrics* 56.1, p. 112–122.

Yang, Yi et Hui Zou (2015). « A fast unified algorithm for solving group-lasso penalize learning problems ». In : *Statistics and Computing* 25.6, p. 1129–1141. issn : 1573-1375.

Yekutieli, Daniel (2008). « Hierarchical False Discovery Rate–Controlling Methodology ». In : *Journal of the American Statistical Association* 103.481, p. 309–316.

Yongdai, Kim, Kwon Sunghoon et Choi Hosik (2012). « Consistent Model Selection Criteria on High Dimensions ». In : *J. Mach. Learn. Res.* 13, p. 1037–1057.

Yuan, Ming et al. (2006). « Model selection and estimation in regression with grouped variables ». In : *Journal of the Royal Statistical Society, Series B* 68, p. 49–67.

Zamboni, Nicola et al. (2005). « Transient expression and flux changes during a shift from high to low riboflavin production in continuous cultures of Bacillus subtilis ». In : *Biotechnology and Bioengineering* 89.2, p. 219–232.

Zeng, Yaohui et Patrick Breheny (2015). « Overlapping group logistic regression with applications to genetic pathway selection ». In : url : http://arxiv.org/abs/1510.05144.

Zhang, Yongli et Xiaotong Shen (2010). « Model selection procedure for high-dimensional data ». In : *Statistical Analysis and Data Mining* 3.5, p. 350–358.

Zhao, Peng, Guilherme Rocha et Bin Yu (2009). « The composite absolute penalties family for grouped and hierarchical variable selection ». In : *Ann. Statist* 37, p. 3468–3497.

Zhao, Peng et Bin Yu (2006). « On Model Selection Consistency of Lasso ». In : *J. Mach. Learn. Res.* 7, p. 2541–2563. issn : 1532-4435.

Zou, Hui et Trevor Hastie (2005). « Regularization and variable selection via the Elastic Net ». In : *Journal of the Royal Statistical Society, Series B* 67, p. 301–320.

Annexe A

Solution approchée du fused-lasso

A.1 Définition

Dans cette annexe, nous présentons une manière d'approximer les solutions du fused-lasso. Cette méthode permet d'utiliser l'algorithme lars sur une matrice de design transformée.

Tout d'abord, rappelons le problème du fused-lasso défini à la Section 1.1.3. Nous l'énonçons ici dans sa forme classique où deux points consécutifs sont voisins. Soit λ_1, $\lambda_2 > 0$, l'estimateur du fused-lasso est :

$$\hat{\beta}^{\text{FL}}_{\lambda_1,\lambda_2} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2 \sum_{j=2}^{p} |\beta_j - \beta_{j-1}| \right\}. \tag{A.1}$$

Il peut être réécrit sous la forme

$$\hat{\beta}^{\text{FL}}_{\lambda_1,\lambda_2} = \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda_1(\|\beta\|_1 + \alpha\|J\beta\|_1) \right\} \tag{A.2}$$

avec $\alpha = \frac{\lambda_2}{\lambda_1}$ et J une matrice de taille $p \times (p-1)$ définie par

$$J = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix}.$$

Procédons à une réécriture de la fonction objectif afin de se rapprocher de la forme d'un lasso.

$$\begin{aligned} &\frac{1}{2}\|y - X\beta\|_2^2 + \lambda_1(\|\beta\|_1 + \alpha\|J\beta\|_1) \\ =&\frac{1}{2}\|y - X\beta\|_2^2 + \lambda_1\|\gamma\|_1 \\ =&\frac{1}{2}\|y - \check{X}\gamma\|_2^2 + \lambda_1\|\gamma\|_1 \end{aligned}$$

avec $\gamma = \left(\beta_1, \ldots, \beta_p, \alpha(\beta_2 - \beta_1), \ldots, \alpha(\beta_p - \beta_{p-1})\right)^T$, $\check{X} = [X, 0_{n,p-1}]$.

On obtient alors une fonction objectif de la même forme que celle du lasso mais avec une matrice de design modifiée et une structure imposée sur les solutions.

On peut donc réécrire le problème A.2 comme :

$$\hat{\gamma}^{\text{FL}}_{\lambda_1,\alpha} = \underset{\gamma \in \Gamma}{\operatorname{argmin}} \left\{ \frac{1}{2} \|y - \check{X}\gamma\|_2^2 + \lambda_1 \|\gamma\|_1 \right\} \tag{A.3}$$

avec $\Gamma = \left\{ \gamma \in \mathbb{R}^{2p-1} \mid \gamma_{i+1} - \gamma_i = \frac{1}{\alpha}\gamma_{p+i}, \forall i = 1, \ldots, p \right\}$. Les p premiers éléments de $\hat{\gamma}^{\text{FL}}_{\lambda_1,\alpha}$ correspondent aux valeurs de $\hat{\beta}^{\text{FL}}_{\lambda_1,\alpha\lambda_1}$.

L'algorithme lars ne peut résoudre le fused-lasso sous cette forme car il ne permet pas de restreindre l'ensemble des solutions possibles. Nous sommes en présence d'un problème d'optimisation sous contraintes. Il est donc possible d'écrire la forme lagrangienne de l'équation (A.3). Les $p-1$ contraintes sont $\gamma_{i+1} - \gamma_i = \frac{1}{\alpha}\gamma_{p+i}$, $i = 1, \ldots, p-1$ qui peuvent être résumées en une seule :

$$\sum_{i=1}^{p-1} \left(\gamma_{i+1} - \gamma_i - \frac{1}{\alpha}\gamma_{p+i} \right)^2 = 0.$$

En effet, cette contrainte est équivalente à impliquer que chaque élément de la somme est nul, ce qui correspond à nos $p-1$ contraintes.

Le lagrangien de (A.3) est :

$$\hat{\gamma}^{\text{FL}}_{\lambda_1,\alpha,C} = \underset{\gamma \in \mathbb{R}^{2p-1}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|y - \check{X}\gamma\|_2^2 + \lambda_1 \|\gamma\|_1 + C \sum_{i=1}^{p-1} \left(\gamma_{i+1} - \gamma_i - \frac{1}{\alpha}\gamma_{p+i} \right)^2 \right\}. \tag{A.4}$$

avec $C > 0$, $\lambda_1 > 0$ et $\alpha > 0$.

Afin que la structure de la solution γ soit respectée, une valeur de C suffisamment grande sera imposée.

Afin de résoudre ce problème avec l'algorithme lars, nous allons utiliser la même astuce de réécriture que (Zou et Hastie 2005) qui ont réécrit l'elastic-net sous la forme d'un lasso pour le résoudre avec le lars.

Pour cela, définissons la matrice M de taille $p \times 2(p-1)$ par $M = [J, \frac{-1}{\alpha} I_{p-1}]$. Un simple calcul permet de vérifier que

$$\|M\gamma\|_2^2 = \sum_{i=1}^{p-1} \left(\gamma_{i+1} - \gamma_i - \frac{1}{\alpha}\gamma_{p+i} \right)^2.$$

On peut donc réécrire l'équation A.4 comme

$$\begin{aligned}
\hat{\gamma}^{\text{FL}}_{\lambda_1,\alpha,C} &= \underset{\gamma \in \mathbb{R}^{2p-1}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|y - \check{X}\gamma\|_2^2 + \lambda_1 \|\gamma\|_1 + C \sum_{i=1}^{p-1} \left(\gamma_{i+1} - \gamma_i - \frac{1}{\alpha}\gamma_{p+i} \right)^2 \right\} \\
&= \underset{\gamma \in \mathbb{R}^{2p-1}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|y - \check{X}\gamma\|_2^2 + \lambda_1 \|\gamma\|_1 + C\|M\gamma\|_2^2 \right\} \\
&= \underset{\gamma \in \mathbb{R}^{2p-1}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\tilde{y} - \tilde{X}\gamma\|_2^2 + \lambda_1 \|\gamma\|_1 \right\}
\end{aligned} \tag{A.5}$$

avec $C > 0$, $\lambda_1 > 0$, $\alpha > 0$, $\tilde{X}$ la matrice de taille $(n+p) \times (2p-1)$ définie par

$$\tilde{X} = \begin{pmatrix} X & 0_{n,p-1} \\ \sqrt{C}J & -\sqrt{C}\frac{1}{\alpha} I_{p-1} \end{pmatrix}$$

et $\tilde{y} \in \mathbb{R}^{n+p}$ tel que $\tilde{y} = \begin{pmatrix} y \\ 0_p \end{pmatrix}$.

L'écriture du problème se fait donc sous la forme d'un lasso et peut donc être résolu avec l'algorithme lars. Avec les paramètres α et C fixés par l'utilisateur, le lars retourne un chemin solution avec un ratio constant entre les paramètres λ_1 et λ_2.

Une limite de cette approche est l'augmentation de la taille de la matrice d'entrée qui passe de $n \times p$ à $(n+p) \times (2p-1)$ mais celle-ci est relativement creuse.

A.2 Simulations

Nous allons comparer l'estimateur de l'approche présentée à celui obtenu par un algorithme utilisé classiquement pour résoudre le fused-lasso et étudier l'impact du paramètre C. L'algorithme lars est utilisé pour trouver le chemin solution de l'équation A.5 pour différentes valeurs de α à l'aide du package `HDPenReg` (Grimonprez et Iovleff 2016). Dans le cas du *Fused-Lasso Signal Approximator* (c'est-à-dire $X = I_n$), notre méthode est comparée aux solutions fournies par le package R `flsa` (Hoefling 2010). Dans le cas du fused-lasso classique, le package `genlasso` (Arnold et Ryan Tibshirani 2014) est utilisée. Les estimateurs obtenus $\hat{\gamma}^{\text{FL}}_{\lambda_1,\alpha,C}$ par notre méthode seront comparés à la vraie valeur de β^* par l'erreur quadratique moyenne ($\frac{1}{p}\|\beta^* - \hat{\gamma}^{\text{FL}}_{\lambda_1,\alpha,C}\|_2^2$). L'erreur quadratique moyenne associée à l'estimateur obtenu par les packages `genlasso` ou `flsa` servira de référence.

Simulation 1 : *Fused-Lasso Signal Approximator*

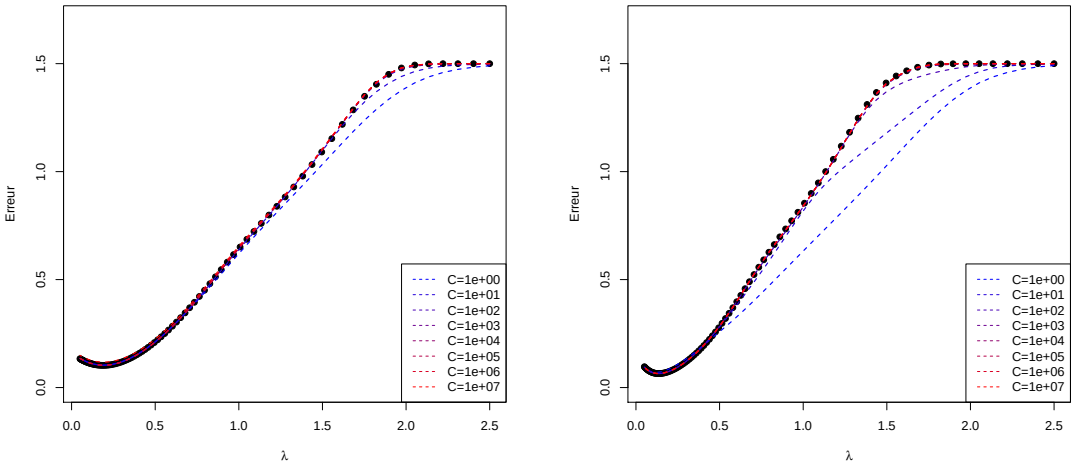
- $n = p = 100$
- β^* composé de 10 segments de longueurs 10 dont 6 non nuls dont les valeurs appartiennent à $\{-2, -1, 1, 2\}$
- $y = \beta + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, 0.4^2 I)$
- $C \in \{10^0, 10^1, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7\}$, $\alpha \in \{0.5, 1, 2\}$

Sur la figure A.1, on constate que, pour de faibles valeurs de λ, la valeur de C n'influe quasiment pas sur l'erreur quadratique moyenne. Par contre, pour des valeurs plus grandes, des différences importantes existent pour les valeurs de C inférieures à 10 000. On constate qu'avec une valeur de C suffisamment grande, notre méthode atteint la même erreur quadratique moyenne que l'algorithme du package `flsa`.

Simulation 2 : Fused-lasso

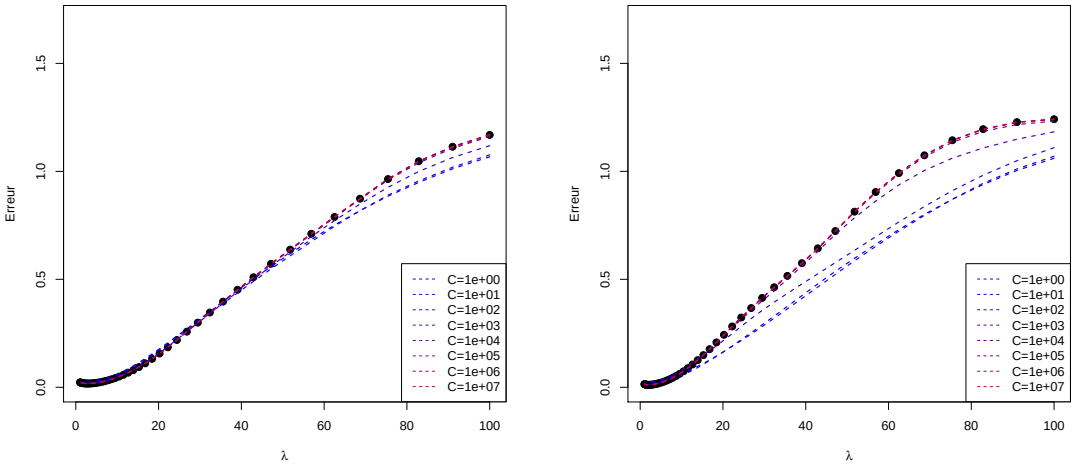
- $n = 50$, $p = 20$
- β^* composé de 5 segments de longueurs 4 dont 2 non nuls
- $y = \beta + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, I)$
- $C \in \{10^0, 10^1, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7\}$, $\alpha \in \{0.5, 1, 2\}$

Dans le cas du fused-lasso classique (figure A.2), on établit les mêmes constatations que précédemment. Il semble également que plus la valeur de α est grande, plus l'erreur pour de faibles valeurs de C est grande.



(a) $\alpha = 0.5$ (b) $\alpha = 2$

FIGURE A.1 – Erreur quadratique moyenne ($\frac{1}{p}\|\beta^* - \hat{\gamma}^{\text{FL}}_{\lambda_1,\alpha,C}\|_2^2$) en fonction de λ_1. Les points noirs représentent les valeurs de références obtenues à l'aide du package `flsa`, les courbes pointillées représentent les valeurs obtenues par l'algorithme lars pour différentes valeurs de C.



(a) $\alpha = 0.5$ (b) $\alpha = 2$

FIGURE A.2 – Erreur quadratique moyenne ($\frac{1}{p}\|\beta^* - \hat{\gamma}^{\text{FL}}_{\lambda_1,\alpha,C}\|_2^2$) en fonction de λ_1. Les points noirs représentent les valeurs de références obtenues à l'aide du package `gen1asso`, les courbes pointillées représentent les valeurs obtenues par l'algorithme lars pour différentes valeurs de C.

Annexe B

Démonstration du Lemme 1

Dans cette annexe, nous présentons la démonstration du Lemme 1 du chapitre 2. Nous rappelons son énoncé :

Lemme 1 *Soit $\mathcal{G} = \{G_1, \dots, G_K\}$ un ensemble de groupes (et non une partition) de $\{1, \dots, p\}$ en K groupes. Supposons que $G_1 = G_2$ et $w_2 > w_1 > 0$. Soit $\hat{\theta}^{\mathcal{G}}_{\lambda}$ la solution de (2.4).*

Alors $(\hat{\theta}^{\mathcal{G}}_{\lambda})_{G_2} = 0_{|G_2|}$.

Afin de montrer ce lemme, nous introduisons les conditions de Karush-Kuhn-Tucker pour les problèmes d'optimisation.

Condition du 1er ordre de Karush-Kuhn-Tucker Si θ est une solution du group-lasso (équation 2.4) alors θ vérifie $\forall i = 1, \dots, K$:

$$X_{G_i}^T(y - X\theta) = \lambda w_i s_{G_i} \tag{B.1}$$

avec s_{G_i} appartenant au sous-différentiel de la fonction $\|.\|_2$ en θ_{G_i},

$$s_{G_i} \in \begin{cases} \left\{ \frac{\theta_{G_i}}{\|\theta_{G_i}\|_2} \right\} & \text{si } \theta_{G_i} \neq 0_{|G_i|} \\ \left\{ z \in \mathbb{R}^{|G_i|} \;\middle|\; \|z\|_2 \leq 1 \right\} & \text{si } \theta_{G_i} = 0_{|G_i|} \end{cases}$$

Le sous-différentiel d'une fonction $f : U \to \mathbb{R}$ où U est un sous-ensemble convexe de $\mathbb{R}^p$ est l'ensemble de ses sous-gradients. Le sous-gradient est une généralisation du gradient pour les fonctions non différentiables. On dit que le vecteur $v \in U$ est un sous-gradient de f au point x_0 si $\forall x \in U : f(x) - f(x_0) \geq \langle v, x - x_0 \rangle$.

De cette condition, on peut déduire que si $\|X_{G_i}^T(y - X\theta)\|_2 < \lambda w_i$ alors $\theta_{G_i} = 0_{|G_i|}$.

Preuve (Lemme 1) Reprenons les hypothèses du Lemme 1 et supposons que $G_1 = G_2$ et $w_2 > w_1 > 0$. Soit θ une solution du group-lasso (équation (2.4)), on veut montrer qu'on ne peut pas avoir $\theta_{G_2} \neq 0_{|G_2|}$.

— Soit $\theta_{G_1} = 0_{|G_1|}$. Montrons que l'on a forcément $\theta_{G_2} = 0_{|G_2|}$.

Si $\theta_{G_1} = 0_{|G_1|}$, d'après les conditions de Karush-Kuhn-Tucker (KKT) on a :

$$X_{G_1}^T(y - X\theta) = \lambda w_i s_{G_i}$$

avec $s_{G_i} \in \left\{ z \in \mathbb{R}^{|G_i|} \;\middle|\; \|z\|_2 \leq 1 \right\}$.
En remarquant que $\|s_{G_i}\|_2 \leq 1$, on a :

$$\begin{aligned}
&\|X_{G_1}^T(y - X\theta)\|_2 \leq \lambda w_1 \\
&\|X_{G_2}^T(y - X\theta)\|_2 \leq \lambda w_1 \text{ car } X_{G_1} = X_{G_2} \\
&\|X_{G_2}^T(y - X\theta)\|_2 < \lambda w_2 \text{ car } w_1 < w_2
\end{aligned}$$

d'où $\theta_{G_2} = 0_{|G_2|}$.

— Soit $\theta_{G_1} \neq 0_{|G_1|}$. Montrons que l'on a forcément $\theta_{G_2} = 0_{|G_2|}$.
Si $\theta_{G_1} \neq 0_{|G_1|}$, d'après les conditions de Karush-Kuhn-Tucker (KKT) on a :

$$\begin{aligned}
X_{G_1}^T(y - X\theta) &= \lambda w_1 \frac{\theta_{G_1}}{\|\theta_{G_1}\|_2} \\
\|X_{G_1}^T(y - X\theta)\|_2 &= \left\| \lambda w_1 \frac{\theta_{G_1}}{\|\theta_{G_1}\|_2} \right\|_2 \\
\|X_{G_1}^T(y - X\theta)\|_2 &= \lambda w_1 \\
\|X_{G_2}^T(y - X\theta)\|_2 &= \lambda w_1 \text{ car } X_{G_1} = X_{G_2} \\
\|X_{G_2}^T(y - X\theta)\|_2 &< \lambda w_2 \text{ car } w_1 < w_2
\end{aligned}$$

d'où $\theta_{G_2} = 0_{|G_2|}$.
Ainsi, θ_{G_2} est toujours nul, ce qui prouve le lemme. □

Annexe C

Article associé au package R MPAgenomics

Cette annexe présente l'article *MPAgenomics : An R package for multi-patient analysis of genomic markers* publié dans le journal *BMC Bionformatics*. Il présente les différentes étapes du processus d'analyse présent dans le package R `MPAgenomics`.

Quentin Grimonprez, Alain Celisse, Samuel Blanck, Meyling Cheok, Martin Figeac et Guillemette Marot (2014). « MPAgenomics : an R package for multi-patient analysis of genomic markers ». In : *BMC Bioinformatics* 15.1, p.1471-2105.

Grimonprez *et al. BMC Bioinformatics* 2014, **15**:394
http://www.biomedcentral.com/1471-2105/15/394

BMC Bioinformatics

SOFTWARE Open Access

MPAgenomics: an R package for multi-patient analysis of genomic markers

Quentin Grimonprez[1*], Alain Celisse[1,2], Samuel Blanck[1], Meyling Cheok[3], Martin Figeac[4] and Guillemette Marot[1,5]

Abstract

Background: Last generations of Single Nucleotide Polymorphism (SNP) arrays allow to study copy-number variations in addition to genotyping measures.

Results: `MPAgenomics`, standing for multi-patient analysis (MPA) of genomic markers, is an R-package devoted to: (*i*) efficient segmentation and (*ii*) selection of genomic markers from multi-patient copy number and SNP data profiles. It provides wrappers from commonly used packages to streamline their repeated (sometimes difficult) manipulation, offering an easy-to-use pipeline for beginners in R.
The segmentation of successive multiple profiles (finding losses and gains) is performed with an automatic choice of parameters involved in the wrapped packages. Considering multiple profiles in the same time, `MPAgenomics` wraps efficient penalized regression methods to select relevant markers associated with a given outcome.

Conclusions: `MPAgenomics` provides an easy tool to analyze data from SNP arrays in R. The R-package `MPAgenomics` is available on CRAN.

Keywords: SNP arrays, Segmentation of genomic data, Marker selection, Multi-patient analysis, R package

Background

Genome-wide Single Nucleotide Polymorphism (SNP) arrays have been widely used over the past few years [1]. First generations were measuring only genetic variations of Single Nucleotide Polymorphisms, which are single base pair mutations at specific loci. Last generations (e.g. SNP5.0, SNP6.0) also include non-polymorphic probes in order to study copy-number variations along the genome in addition to genotyping measures. These arrays are especially used to study the impact of diseases, e.g. cancer, on the human genome.

Analyzing data from genome-wide SNP arrays within R requires several packages, e.g. `aroma` for normalization of Affymetrix® SNP arrays [2,3], `changepoint` or `cghseg` for segmentation of copy number profiles [4], `cghcall` for labelling segments [5], and `glmnet` for penalized regressions [6]. Each package performs a specific task along the whole analysis but none of them is related to the others. Output formats of given packages are often not compatible with input formats required by the other, making their use tricky for beginners in R. One main contribution of the `MPAgenomics` R package is to aggregate these commonly used packages, providing wrappers to inter-relate them automatically.

At each step of the analysis a large amount of packages are available to perform normalization, segmentation or marker selection. A careful choice of only a few methods is required to provide an easy-to-use and efficient tool.

In this software article, we describe two different pipelines implemented in the R package `MPAgenomics`. Both of them perform the whole analysis from raw data to normalization, and then either successive segmented profiles, or a list of genomic markers selected from all available profiles.

Implementation

`MPAgenomics` is implemented in R [7]. The package is divided in four main parts: data normalization, segmentation, calling and marker selection. Each part depends on different packages. `MPAgenomics` provides wrappers for some functions of these packages and facilitates the interaction between outputs and inputs of different functions.

*Correspondence: quentin.grimonprez@inria.fr
[1] MODAL team, Inria Lille-Nord Europe, Villeneuve-d'Ascq, France
Full list of author information is available at the end of the article

It remedies some problems with the wrapped packages such as confusing parameter names.

Data normalization

The normalization process in `MPAgenomics` contains *technical biases correction* and *copy number and allele B fraction estimation*. Following [8], *allele B fraction* refers to the proportion of the total signal coming from allele B. Normalization methods are available for Affymetrix® arrays (10K, 100K, 500K, GenomeWideSNP 5 & 6, and CytoScanHD). The estimation of the total copy number and allele B fraction is made by *CRMAv2* [9] originally implemented in the `aroma` packages. For studies with matched normal-tumor samples, a better estimation is suggested and implemented for the allele B fraction of the tumoral sample with the *TumorBoost* method [8].

The use of `aroma` packages is difficult for neophytes due to the strict folder architecture it requires and the documentation of the project which is mainly dedicated to experts able to criticize each method proposed and understand details of each procedure.

`MPAgenomics` provides documentation with a detailed example explaining how to quickly analyze data. The tutorial can be accessed in R by running the following commands:

```
library(MPAgenomics)
vignette("MPAgenomics")
```

More details on each step or wrapper are given to help advanced users to run each function separately.

Several features in the original `aroma` packages create new folders and files within the architecture. Matching files from different processes associated with a given sample can be tricky for neophytes. `MPAgenomics` implements a wrapper to build the folder architecture, check filenames automatically, process *CRMAv2* and *TumorBoost* normalization steps. Miscellaneous functions are also provided to ease some actions like signal extraction. Furthermore, different graphs such as the copy number profile can be saved in the working directory for further visualization.

The following steps (segmentation, calling and/or selection of genomic markers) are available in two settings. One is aroma-based and exploits the folder architecture and the files generated along the process. The second does not depend on aroma and allows advanced users to use their own normalized data.

Segmentation

Although the use of manual annotations provides the best segmentation results [10], it appears essential for multi-patient analysis to avoid relying on them since they are time-consuming.

Therefore, following simulation results of [10], `MPAgenomics` wraps the `CGHSEG` [11,12] and PELT (Pruned Exact Linear Time) [13] segmentation methods which appeared to be those with the best overall performance.

PELT and `CGHSEG` methods fit a Gaussian maximum likelihood model but they differ in the way they choose the number of segments. `CGHSEG` requires the maximal number of segments as input. In `MPAgenomics`, the optimal number of segments is chosen according to a penalty $C \times K \times \left(2.5 + \log\left(\frac{P}{K}\right)\right)$ with a profile of length P, K the number of segments and $C > 0$ a parameter to choose [14]. This choice is performed using slope heuristics [15]. The PELT method returns a segmentation with a number of segments automatically chosen by the algorithm according to a penalty $K\rho\log(P)$ with $\rho > 0$ a parameter to choose. The choice of the penalty parameter has been raised in [4]. `MPAgenomics` suggests an automatic sample-specific choice of ρ chromosome by chromosome (see package vignette for details on the method). In `MPAgenomics`, the two methods, `CGHSEG` with the slope heuristic and PELT with our calibration method, are proposed. By default, `CGHSEG` is used because it is quicker than PELT due to the multiple execution required by the ρ calibration method we propose.

The implemented segmentation methods are independently available for both copy number and allele B fraction profiles. In the case of allele B fraction segmentation, only heterozygous SNPs are kept. First, a naive genotype call [8] is performed on each normal sample in order to separate heterozygous SNPs from homozygous SNPs. Naive genotyping method assumes SNPs are bi-allelic and therefore is not recommended for tumor samples. Thus allele B fraction segmentation in `MPAgenomics` requires matched normal-tumor pairs. Then, following [16], the resulting signal is centered on 0.5 and symmetrized, which makes it similar to the usual copy number.

Calling

From each segmented profile, the *CGHcall* method [17] is run to label every copy-number segment in terms of *loss*, *normal*, and *gain*.

CGHcall depends on a parameter, named *cellularity*, corresponding to the contamination of a sample with healthy cells. In `MPAgenomics`, this parameter can be modified by users, by default its value is 1 meaning that tumor samples are pure.

In the aroma-dependent function, segmentation and calling are performed with the same wrapper. The calling is run for each profile separately. Results are saved in text format in the working folder architecture.

Selection of genomic markers

The goal is to select genomic markers (e.g. SNPs or CNV) associated with a given response from all patient profiles simultaneously. There is no need to perform segmentation

and calling before the multi-patient analysis, marker selection is made over all copy-number profiles. However, segmentation can be performed before marker selection if wanted, in order to reduce the noise and the dimensionality of the problem.

Assuming I individuals and P potential markers, then for each individual i, y_i denotes the response and $x_{i,p}$ the corresponding normalized value of copy number or allele B fraction signal at genomic position p.

Due to the huge number of markers ($P >> I$), `MPAgenomics` uses by default the *lasso* [18] regularization method to select very few ones. This method offers two advantages: (*i*) it selects only few variables, easing the interpretability of results, (*ii*) there exist some algorithms such as the lars [19] to solve quickly the *lasso* problem and support high-dimensional data.

The lasso regularization method consists in minimizing $g_\lambda : \beta \in \mathbb{R}^P \mapsto g_\lambda(\beta)$, where

$$g_\lambda(\beta) = \sum_{i=1}^{I} (y_i - (X\beta)_i)^2 + \lambda \sum_{p=1}^{P} |\beta_p| \quad ,$$

with $(X\beta)_i = \sum_p x_{i,p}\beta_p$ and $\lambda > 0$ controlling the number of non-zero coordinates of β. After minimization, non-zero coefficients β_p correspond to influential positions to predict the response.

`MPAgenomics` genomic marker selection drastically improves currently available packages in terms of computation time. With the linear regression model, it efficiently provides the exact solution by using the new R package `HDPenReg`, which is an optimized implementation of the *lars* algorithm [19] specially dedicated to a huge number of markers.

Since the theoretical grounding of Lasso when $P >> I$ relies on a theoretical condition (see [20]) that cannot be easily checked in practice, the spike and slab algorithm [21,22] – a three steps algorithm performing filtering, estimation and variable selection – is also provided in `MPAgenomics` as an alternative.

Logistic regression is also available for binary responses. In this case, `MPAgenomics` wraps the `glmnet` package [6] in the whole process. Unlike `HDPenReg` it does not provide the exact solution but is computationally very efficient. With `glmnet` and `HDPenReg`, the regularization parameter λ is chosen by k-fold cross-validation [23]. The selected variables are the most relevant ones regarding the response.

Discussion

`MPAgenomics` is mainly dedicated to beginners in SNP array analyses. It solves problems commonly encountered by neophytes such as interaction between different packages or specialized documentation dedicated to experts in the field. In addition, `MPAgenomics` suggests careful and automatic choices of crucial parameters at each part of the analysis.

To achieve simplicity of usage, `MPAgenomics` does not propose all options implemented in the wrapped packages, especially for normalization. However, outputs are generated in such a way that interaction between wrapped packages and `MPAgenomics` is facilitated. For example, the strict directory structure of `aroma` packages is built by `MPAgenomics`. Therefore, advanced users may directly use specific options of `aroma` to enhance their analysis without renormalizing data from scratch.

As specified in the data normalization section, segmentation, calling and marker selection steps can be performed without the use of aroma. This allows users to provide their own normalized data into matrices. This is useful for non-Affymetrix® SNP arrays, CGH arrays or high-throughput sequencing data. For the latter, count data might need a variance-stabilizing transformation into Gaussian data before using current segmentation, calling and marker selection. For example, the Anscombe transform [24] can be used in addition to appropriate normalization specific to the used technology (target sequencing, whole-genome sequencing).

Currently, copy number and allele B fraction are segmented independently from each other. Research is ongoing to propose joint segmentation methods allowing to detect uniparental disomies, fragments which present a normal copy number but a loss of heterozygosity in the corresponding allele B fraction.

Conclusions

`MPAgenomics` provides user-friendly wrappers for normalization and multi-patient analysis of high-throughput genomic data. It offers a guideline for beginners in copy-number variation analysis focusing on proven methods for their effectiveness. `MPAgenomics` also provides automatic choices of crucial parameters for segmentation and selection of markers.

Even though normalization is provided for Affymetrix® arrays, other steps (segmentation, calling, and marker selection) can be applied to normalized data from other DNA arrays and next-generation sequencing data.

Availability and requirements

Project name: MPAgenomics
Project home page: http://cran.at.r-project.org/package=MPAgenomics
Operating system(s): Platform independent
Programming language: R
Other requirements: none
License: GNU GPL (>=2)
Any restrictions to use by non-academics: None

Competing interests

The authors declare that they have no competing interests.

Authors' contributions
QG implemented the first versions of `MPAgenomics` and `HDPenReg`. He helped in their maintenance and drafted the manuscript and the vignette of the package. AC contributed for choices of crucial parameters in segmentation, and helped draft the manuscript. SB maintained `MPAgenomics` and its vignette. MC and MF participated in discussions on data analysis and results. GM conceived of the project and managed it, selected key packages for wrapping. She occasionally participated to the implementation and helped draft the manuscript and the vignette. All authors read and approved the final manuscript.

Acknowledgements
We thank Serge Iovleff for his help implementing `HDPenReg`. We also thank Claude Preudhomme and Olivier Nibourel for providing data presented in the vignette of the package, and for their helpful clinical competences to interpret the results. The development of this package was funded by the Inria Technological Development Action (ADT) named *MPAGenomics*.

Author details
[1]MODAL team, Inria Lille-Nord Europe, Villeneuve-d'Ascq, France. [2]Laboratoire Paul Painlevé, Université Lille 1, Villeneuve-d'Ascq, France. [3]Inserm, U837, Team 3, Cancer Research Institute of Lille, Lille, France. [4]Plate-forme de génomique fonctionnelle et structurale, IFR-114, Université Lille 2, Lille, France. [5]EA 2694, Université Lille 2,Lille, France.

Received: 23 July 2014 Accepted: 19 November 2014
Published online: 14 December 2014

References

1. LaFramboise T: **Single nucleotide polymorphism arrays: a decade of biological, computational and technological advances.** *Nucleic Acids Res* 2009, **37**(13):4181–4193.
2. Bengtsson H, Simpson K, Bullard J, Hansen K: **aroma.affymetrix: A generic framework in R for analyzing small to very large Affymetrix data sets in bounded memory.** Technical Report 745, Department of Statistics, University of California, Berkeley; 2008.
3. Bengtsson H, Bullard J, Hansen K, Neuvial P, Purdomand E, Robinson M, Simpson K: **Aroma project.** 2010. [http://www.aroma-project.org/]
4. Killick R, Eckley I: **Changepoint: An R Package for Changepoint Analysis.** 2013. R package version 1.1, [http://www.lancs.ac.uk/~killick/Pub/KillickEckley2011.pdf]
5. van de Wiel M, Vosse S: **CGHcall: Calling Aberrations for Array CGH Tumor Profiles.** 2012. R package version 2.20.0 [http://www.bioconductor.org/packages/release/bioc/vignettes/CGHcall/inst/doc/CGHcall.pdf]
6. Friedman JH, Hastie T, Tibshirani R: **Regularization paths for generalized linear models via coordinate descent.** *J Stat Softw* 2010, **33**(1):1–22.
7. R Core Team: *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing; 2014. [http://www.R-project.org/]
8. Bengtsson H, Neuvial P, Speed TP: **Tumorboost: Normalization of allele-specific tumor copy numbers from a single pair of tumor-normal genotyping microarrays.** *BMC Bioinformatics* 2010, **11:**245.
9. Bengtsson H, Wirapati P, Speed TP: **A single-array preprocessing method for estimating full-resolution raw copys from all Affymetrix genotyping arrays including GenomeWideSNP 5 & 6.** *Bioinformatics* 2009, **25**(17):2149–2156.
10. Hocking T, Schleiermacher G, Janoueix-Lerosey I, Boeva V, Cappo J, Delattre O, Bach F, Vert J-P: **Learning smoothing models of copy number profiles using breakpoint annotations.** *BMC Bioinformatics* 2013, **14**(1):164.
11. Picard F, Robin S, Lavielle M, Vaisse C, Daudin J-J: **A statistical approach for array CGH data analysis.** *BMC Bioinformatics* 2005, **6**(1):27.
12. Rigaill G: **Pruned dynamic programming for optimal multiple change-point detection.** arXiv e-print, 2010, arXiv/1004.0887.
13. Killick R, Fearnhead P, Eckley IA: **Optimal detection of changepoints with a linear computational cost.** *J Am Stat Assoc* 2012, **107**(500):1590–1598.
14. Lebarbier E: **Detecting multiple change-points in the mean of gaussian process by model selection.** *Signal Process* 2005, **85**(4):717–736.
15. Birgé L, Massart P: **Minimal penalties for gaussian model selection.** *Probability Theory Related Fields* 2007, **138**(1-2):33–73.
16. Staaf J, Lindgren D, Vallon-Christersson J, Isaksson A, Göransson H, Juliusson G, Rosenquist R, Höglund M, Borg Å, Ringnér M: **Segmentation-based detection of allelic imbalance and loss-of-heterozygosity in cancer cells using whole genome SNP arrays.** *Genome Biol* 2008, **9**(9):R136.
17. van de Wiel MA, Kim KI, Vosse SJ, van Wieringen WN, Wilting SM, Ylstra B: **CGHcall: calling aberrations for array CGH tumor profiles.** *Bioinformatics* 2007, **23**(7):892–894.
18. Tibshirani R: **Regression shrinkage and selection via the lasso.** *J R Stat Soci Series B* 1994, **58:**267–288.
19. Efron B, Hastie T, Johnstone I, Tibshirani R: **Least angle regression.** *Ann Stat* 2004, **32:**407–499.
20. Ravikumar P, Wainwright M. J, Raskutti G, Yu B: **High-dimensional covariance estimation by minimizing l_1-penalized log-determinant divergence.** *Electron J Stat* 2011, **5:**935–980.
21. Ishwaran H, Rao JS: **Spike and slab variable selection: frequentist and bayesian strategies.** *Ann Stat* 2005, **33**(2):730–773.
22. Ishwaran H, Rao JS: **Generalized ridge regression: geometry and computational solutions when p is larger than n.** Technical Report, Department of Public Health Sciences Division of Biostatistics, University of Miami, Miller School of Medecine; 2010.
23. Arlot S, Celisse A: **A survey of cross-validation procedures for model selection.** *Stat Surv* 2010, **4:**40–79.
24. Anscombe FJ: **The transformation of poisson, binomial, and negative-binomial data.** *Biometrika* 1948, **35**(3/4):246–254.

doi:10.1186/s12859-014-0394-y
Cite this article as: Grimonprez *et al.*: MPAgenomics: an R package for multi-patient analysis of genomic markers. *BMC Bioinformatics* 2014 **15**:394.

Table des matières

Sélection de groupes de variables corrélées en grande dimension

Résumé

Le contexte de cette thèse est la sélection de variables en grande dimension à l'aide de procédures de régression régularisée en présence de redondance entre variables explicatives. Parmi les variables candidates, on suppose que seul un petit nombre est réellement pertinent pour expliquer la réponse. Dans ce cadre de grande dimension, les approches classiques de type Lasso voient leurs performances se dégrader lorsque la redondance croît, puisqu'elles ne tiennent pas compte de cette dernière. Regrouper au préalable ces variables peut pallier ce défaut, mais nécessite usuellement la calibration de paramètres supplémentaires. L'approche proposée combine regroupement et sélection de variables dans un souci d'interprétabilité et d'amélioration des performances. D'abord une Classification Ascendante Hiérarchique (CAH) fournit à chaque niveau une partition des variables en groupes. Puis le Group-lasso est utilisé à partir de l'ensemble des groupes de variables des différents niveaux de la CAH à paramètre de régularisation fixé. Choisir ce dernier fournit alors une liste de groupe candidats issus potentiellement de différents niveaux. Le choix final des groupes est obtenu via une procédure de tests multiples.
La procédure proposée exploite la structure hiérarchique de la CAH et des pondérations dans le Group-lasso. Cela permet de réduire considérablement la complexité algorithmique induite par la flexibilité liée à la possibilité de choisir des groupes issus de différents niveaux de la CAH.

Mots clés : group-lasso, classification ascendante hiérarchique, variables corrélées, test hiérarchique, grande dimension, sélection de variables

---

Abstract

This thesis takes place in the context of variable selection in the high dimensional setting using penalized regression in presence of redundancy between explanatory variables. Among all variables, we suppose that only a few number is relevant for predicting the response variable. In this high dimensional setting, performance of classical lasso-based approaches decreases when redundancy increases as they do not take it into account. Firstly aggregating variables can overcome this problem but generally requires calibration of additional parameters.
The proposed approach combines variables aggregation and selection in order to improve interpretability and performance. First, a hierarchical clustering procedure provides at each level a partition of the variables into groups. Then the Group-lasso is used with the set of groups of variables from the different levels of the hierarchical clustering and a fixed regularization parameter. Choosing this parameter provides a list of candidates groups potentially coming from different levels. The final choice of groups is done by a multiple testing procedure.
The proposed procedure exploits the hierarchical structure from hierarchical clustering and some weights in Group-lasso. This allows to greatly reduce the algorithm complexity induced by the possibility to choose groups coming from different levels of the hierarchical clustering.

Keywords: group-lasso, hierarchical clustering, correlated variables, hierarchical testing, high dimension, variable selection

---

Équipe-projet MΘDAL
Inria Lille-Nord Europe – 40 avenue Halley – 59650 Villeneuve d'Ascq – France