UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UNE ADAPTATION DE L'APPROCHE DES MOINDRES CARRÉS PARTIELS MULTIDIMENSIONNELLE AUX ÉTUDES D'ASSOCIATION GÉNÉTIQUE

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES

PAR

FLORENCE LAURE MAGNIFO KAHOU

AOÛT 2018

REMERCIEMENTS

Ce travail n'aurait jamais vu le jour sans les efforts considérables de nombreuses personnes qui m'ont encadrée et encouragée sans relâche, me permettant ainsi de développer ce que j'ai reçu du Très Haut.

Mes remerciements vont d'abord au Très Haut. Reçois l'honneur et la gloire. Tu as été à l'œuvre, de l'admission dans ce programme jusqu'aujourd'hui.

Ma dette est immense envers mon directeur de recherche, Oualkacha Karim, dont l'aide-académique a eu raison des difficultés que j'aurais pu endurer pendant la réalisation de ce travail. Sa rigueur à la tâche, son goût de la précision, son ouverture d'esprit, ses précieux conseils et ses encouragements m'ont sans cesse soutenue dans cette entreprise. Je lui dis également merci pour sa patience envers moi.

Je remercie tout le personnel du département de mathématiques et de statistiques : les professeurs pour leurs enseignements et leurs encouragements, les étudiants pour les échanges fructueux que nous avons partagés, et le personnel administratif, pour sa sympathie et sa disponibilité. Un merci particulier à Mme Gisèle Legault pour l'aspect informatique de ce travail. Je remercie également le Collège Ahuntsic pour le financement partiel de mes droits de scolarité.

Mes remerciements vont aussi à mon époux, Lottin Wekape ; mes filles, Perle Imelda et Maelle Esméralda. Recevez le fruit du sacrifice après toutes ces années !

Je remercie de tout cœur la famille Nobou qui a toujours été là pendant les

moments d'hésitation et de difficulté pour me relever et m'encourager à aller de l'avant. Merci à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

RÉSUMÉ

Nous décrivons une approche multidimensionnelle qui permet de tester l'association entre des variants génétiques et des traits quantitatifs. Cette approche utilise la théorie des valeurs extrêmes, GPD (de l'anglais Generalized Pareto Distribution), pour approximer la distribution de la statistique du test d'association sous l'hypothèse nulle (Knijnenburg *et al.*, 2009). Nous généralisons donc au cas multidimensionnel, l'approche donnée dans Xu et Greenwood (2013), en ce qui concerne l'approche PLS (de l'anglais Partial Least Squares regression). Nous adaptons ainsi la méthode PLS comme technique de réduction de la dimension des données, pour l'étude d'association entre des variants génétiques (covariables) et les traits d'une maladie complexe (variables réponses). À partir des données, généralement corrélées, on va extraire des composantes scores (cotes) et des composantes loadings (charges), à condition que la corrélation entre les nouvelles composantes (scores, loagings) demeure maximale. Ces nouvelles variables (les scores) peuvent donc être utilisées pour des études d'association génétique afin d'augmenter la puissance du test. Ne sachant pas la distribution de la statistique de test sous l'hypothèse nulle, nous l'avons approximée à l'aide des permutations. Nous constatons que cette approche contrôle bien l'inflation de l'erreur de type I. Une étude de simulation nous permet de comparer plusieurs approches existantes à la nôtre. Nous validons aussi notre approche par une analyse d'un jeu de données réelles sur des patients atteints de la maladie de l'Alzheimer.

Mots clés : Régression, Single Nucleotide, GWAS, Tests d'associations génétiques, Composantes latentes, loadings.

INTRODUCTION

En statistique génétique, plusieurs approches statistiques sont utilisées pour analyser et interpréter des données ; on souhaite très souvent établir le lien entre deux ensembles de variables. Par exemple, dans les études génétiques, on s'intéresse à la relation entre plusieurs phénotypes (ensemble des caractères visibles : la couleur des cheveux, les traits...) et de variants génétiques(mutations) du génome. Dans le cadre des maladies génétiques humaines complexes ou rares, on souhaite trouver l'association entre le génome et des phénotypes. Le GWAS (de l'anglais Genome Wide Association Study), est l'étude d'association pangénomique. Le GWAS tente d'identifier l'association qu'il y aurait entre des marqueurs génétiques et une maladie, par exemple. Plusieurs de ces études d'association sont limitées parce qu'elles n'analysent qu'un seul trait à la fois. Nous pensons, comme en analyse multidimensionnelle, qu'il serait préférable de tester l'association d'un groupe de marqueurs génétiques avec plusieurs traits à la fois. Les traits sont le plus souvent corrélés entre eux et le fait de les tester en groupe pourrait augmenter la puissance du test statistique. L'approche PLS (de l'anglais Partial Least Squares) est la plus souvent utilisée en analyse multidimensionnelle pour analyser deux ensembles de variables, parce qu'elle permet de réduire le nombre de variables qui sont fortement colinéaires dans les deux ensembles de variables, en des variables latentes orthogonales, en gardant le maximum d'information utile entre les deux ensembles de variables.

Dans l'analyse des variants rares de Xu et Greenwood (2013), tous les tests mesurent l'association entre plusieurs SNPs et un seul trait. Les auteurs (Xu,*et al.*, 2012), montrent qu'on a un gain de puissance quand on analyse les variants rares

avec la méthode PLS, comparativement à d'autres existantes. Malheureusement, ces approches dans la réduction des données ne tiennent pas compte de la colinéarité qui existerait entre les traits, puisqu'elles analysent un trait à la fois. Nous proposons une approche multidimensionnelle qui permet de tester l'association entre un groupe de variants communs/rares et un ensemble de phénotypes. Cette approche utilise la théorie des valeurs extrêmes, GPD (de l'anglais Generalized Pareto distribution) pour approximer la loi de la statistique du test sous H_0 de façon appropriée. Nous généralisons donc au cas multidimensionnel, l'approche donnée par Xu et Greenwood (2013), en ce qui concerne PLS. Nous validons notre approche par des études de simulations, mais aussi par un jeu de données réelles sur des patients atteints de la maladie d'Alzheimer.

Ce travail est divisé en quatre chapitres. Le **chapitre 1** est celui des préliminaires. Nous donnons les méthodes d'analyse linéaire univariées couramment utilisées pour la réduction des données. Nous insistons particulièrement sur la régression Ridge (RR), la régression LASSO (de l'anglais Least Absolute Selection and Shrinkage Operator), la régression en utilisant l'analyse des composantes principales (PCR, de l'anglais Principal Component Regression) et la régression des moindres carrés partiels (PLS). Le vocabulaire génétique que nous pourrons utiliser dans ce mémoire s'y trouve également. Le **chapitre 2** introduit les analyses d'association linéaires entre deux groupes de données. On distingue les cas où on mesure l'association entre une variable et un ensemble de variables, et le cas où on aurait deux ensembles de variables. La notion de test de signification y est aussi abordée. Le **chapitre 3** est consacré à l'approche que nous proposons pour mesurer l'association linéaire entre deux groupes de variables à la fois. Inspirés du package R de RVtests [1] nous décrivons une autre façon d'analyser les données

1. RVtests (Rare Variant Tests), donne la description de certaines méthodes qui utilisent la régression multiple pour tester l'association entre des variants rares et les traits d'une maladie.

multidimensionnelles. Notre approche utilise les tests de permutation et la théorie des valeurs extrêmes. Le **chapitre 4** est consacré à l'étude des simulations et à une analyse d'un jeu de données réelles des patients atteints de l'Alzheimer. Il nous a donc semblé utile de décrire la maladie d'Alzheimer. Nous analysons et comparons plusieurs des méthodes décrites dans le présent travail et notre approche. Nous terminons par une conclusion. On retrouve en annexe les codes R des différentes fonctions qui ont été utilisées.

Plusieurs de ces fonctions testent l'association entre les variants et un trait à la fois.

CHAPITRE I

PRÉLIMINAIRES

1.1 La régression multiple standard.

La régression multiple est une généralisation de la régression simple qui permet d'expliquer une variable endogène y en fonction de m variables exogènes $x_1, x_2, ..., x_m$, avec $m \geq 2$. On s'intéresse donc à expliquer la variation d'une variable dépendante à l'aide de plusieurs variables explicatives. Si l'on dispose de n observations $(y_i, x_{i1}, ..., x_{im})$, $i = 1, ..., n$, le modèle sous la forme matricielle s'écrit

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{1.1}$$

où $\mathbf{X}$ est une matrice de format $n \times m$ connue, $\boldsymbol{\beta}$ un paramètre à estimer et $\boldsymbol{\varepsilon} = N(\mathbf{0}, \sigma^2\mathbf{I_n})$ l'erreur du modèle. Il faut analyser le lien entre $\mathbf{y}$ et $\mathbf{X}$, afin de prédire les valeurs de la variable y éventuellement. On utilise la **régression multiple standard** pour évaluer la relation entre l'ensemble des variables indépendantes $\mathbf{x} = (x_1, x_2, ..., x_m)$ et la variable dépendante y.

La problématique ici est d'estimer le paramètre $\boldsymbol{\beta}$ à partir des observations. Cet estimateur doit être très proche du vrai paramètre et permettre de prédire des valeurs futures de la variable y. La méthode des moindres carrés consiste à minimiser

la somme des carrés résiduels :

$$||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}).$$

Il est montré que l'estimateur de $\boldsymbol{\beta}$ est donné comme suit (Hastie *et al.*, 2008)

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}.$$

L'estimateur des moindres carrés, $\hat{\boldsymbol{\beta}}$, est l'unique estimateur sans biais qui minimise la somme des carrés des résidus quand la matrice $(\mathbf{X}^\top\mathbf{X})$ est non-singulière. C'est un estimateur à variance minimale parmi tous les estimateurs sans biais d'après le théorème de Gauss-Markov (Hastie *et al.*, 2008).

1.2 Mesures d'association linéaire entre la variable y et les variables $\mathbf{x}$.

Le coefficient de corrélation multiple R, (ou R^2, le coefficient de détermination) est la corrélation maximale entre y et toute combinaison linéaire des variables de $\mathbf{x}$. Si on considère le modèle de l'équation (1.1), nous pouvons écrire R, comme

$$R = \max_{\substack{\beta_i \in \mathbb{R} \\ i \in \{1,2,\cdots,m\}}} \{Corr(\mathbf{Y}, \sum_{i=1}^{m} \beta_i \mathbf{x_i})\} \tag{1.2}$$

Le coefficient de corrélation partielle r, (ou r^2 le coefficient de détermination partiel) mesure la dépendance linéaire entre y et l'une des variables exogènes, conditionnellement aux autres variables de $\mathbf{x}$.

Ces coefficients évaluent l'influence des variables dans le modèle globalement (les m variables en bloc) et individuellement. Un coefficient de détermination élevé (près de 100%) signifie que les variables exogènes expliquent ensemble une bonne part de la dispersion de y. Cependant, un des défauts de ce coefficient est qu'il a tendance à augmenter quand le nombre de variables explicatives augmente. Dans ce cas, il est conseillé de prendre le coefficient R ajusté, qui tient compte du nombre de degrés de liberté dans le modèle (c'est-à-dire le nombre de variables explicatives).

On a vu que pour une régression multiple, l'estimateur de $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}}$, est

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

Ceci n'est vrai que si les colonnes $\mathbf{X}$ sont linéairement indépendantes pour que la matrice $(\mathbf{X}^\top \mathbf{X})^{-1}$ soit bien définie. Quand les variables sont fortement corrélées, cet estimateur reste sans biais. Cependant, il a tendance à avoir une grande variance. Une solution pour le problème de multicolinéarité serait d'omettre certaines de ces variables et de ne considérer que quelques unes pour la régression. C'est une solution très coûteuse, dans le sens qu'il faudrait déjà faire une sélection. Une solution meilleure est d'introduire un biais à l'estimateur dans le but de réduire sa variance. Par exemple, en cas de multicolinéarité dans la matrice $\mathbf{X}$, une méthode consiste à ajouter une constante à la diagonale de $(\mathbf{X}^\top \mathbf{X})$ afin de la rendre inversible. C'est ce que la régression *Ridge* propose.

1.3 Les techniques de réduction de données

1.3.1 Régression *Ridge*

La régression *Ridge* a été proposée en 1970 par Hoerl et Kennard. Les auteurs ont proposé de transformer la matrice $(\mathbf{X}^\top \mathbf{X})$ en une matrice $(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})$ inversible, où λ est un réel positif. Cette méthode est équivalente à minimiser la somme des carrés

$$||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda ||\boldsymbol{\beta}||^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta}.$$

Cela revient à chercher $\boldsymbol{\beta}$ qui minimise

$$||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2$$

sous la contrainte

$$||\boldsymbol{\beta}||^2 \leq C,$$

où C est une constante qui dépend de λ. C'est pour cela qu'on parle aussi de méthode de régression avec rétrécissement ou (de l'anglais, "shrinkage method"). La constante λ ou C contrôle la norme L_2 de $\boldsymbol{\beta}$,

$$||\boldsymbol{\beta}||^2 = \sum_{j=1}^{m} \beta_j^2.$$

Elle permet donc de définir un modèle linéaire en réduisant la longueur du paramètre. L'estimateur de $\hat{\boldsymbol{\beta}}$ par la méthode *Ridge* est donné par

$$\hat{\boldsymbol{\beta}}_{Ridge,\lambda} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}.$$

L'espérance et la variance de $\hat{\boldsymbol{\beta}}$ par la méthode *Ridge* sont

$$E(\hat{\boldsymbol{\beta}}_{Ridge,\lambda}) = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{X} \boldsymbol{\beta},$$

$$Var(\hat{\boldsymbol{\beta}}_{Ridge,\lambda}) = \sigma^2 (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} (\mathbf{X}^\top \mathbf{X}) (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}).$$

Une valeur de λ bien choisie permet d'inverser la matrice $(X^\top X + \lambda I)^{-1}$, mais aussi de contrôler la variance de l'estimateur. Il faut donc bien choisir le λ qui finalement va définir le modèle. Hoerl et Kennard ont proposé de représenter la longueur du paramètre $||\hat{\boldsymbol{\beta}}_{Ridge,\lambda}||$ en fonction de la constante λ et de considérer les valeurs de λ où il n'y avait pas trop de variations de l'estimateur. C'est une technique pas très objective qui a connu beaucoup de critiques. Une méthode standard est d'utiliser la "validation croisée" pour déterminer λ. Nous expliquons la validation croisée après la régression Lasso.

Quand on a un grand nombre de variables prédictives, on souhaite faire une sélection des variables qui représenteraient au mieux le modèle, et donc de rendre certains coefficients de l'estimateur $\hat{\boldsymbol{\beta}}$ nuls. C'est ce que fait le Lasso.

1.3.2 Régression Lasso : Least Absolute Selection and Shrinkage Operator.

Pour la régression Lasso, on cherche donc l'estimateur $\boldsymbol{\beta}$ qui minimise la fonction de perte

$$||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda \sum_{i=1}^{m} |\beta_i|.$$

La différence entre le Lasso et la régression *Ridge* se situe donc au niveau du terme de la pénalité :

$$||\boldsymbol{\beta}||_1 = \sum_{i=1}^{m} |\beta_i|$$

pour le Lasso et

$$||\boldsymbol{\beta}||^2 = \sum_{i=1}^{m} \beta_i^2$$

pour la regression *Ridge*. Bien que ces deux méthodes soient semblables au niveau de la définition, le comportement de leur solution peut bien être différent. Le terme de la pénalité du Lasso permet d'avoir des coefficients de l'estimateur $\hat{\boldsymbol{\beta}}$ nuls (plus λ est grand, plus le nombre de coefficients nuls augmente et plus les coefficients non nul sont réduits), ce qui n'est pas le cas pour la régression *Ridge*. Pour les coefficients nuls, les variables correspondantes ne font pas partie du modèle, et donc le Lasso permet de faire une sélection de variables, ce qui n'est pas le cas pour la *Ridge*. Avec cette sélection de variables, on peut donc mieux interpréter le modèle dans le cas du Lasso.

1.3.3 Validation croisée

Quand on a la possibilité d'avoir plusieurs choix de modèles, comme c'est le cas pour les régressions Ridge et Lasso, il se pose le problème du choix optimal du " bon modèle". La validation croisée ou, (de l'anglais) cross-validation, est une méthode qui permet d'estimer la fiabilité d'un modèle. Elle est basée sur une technique d'échantillonnage. Une des techniques, le "test set-validation" ou "hold-dout method", consiste à diviser l'échantillon en un échantillon d'apprentissage

et un échantillon test. L'échantillon d'apprentissage doit représenter plus de 60% de l'échantillon total. L'échantillon test permet alors de valider le modèle en effectuant un calcul d'erreur mesurant les écarts entre ce qui est observé et ce qui est prédit, dans l'échantillon test.

Une autre technique, la " k-fold cross-validation", consiste à diviser l'échantillon en k échantillons de même taille(généralement, on considère $k = 10$). Il faut considérer un de ces échantillons comme l'échantillon de test et les $(k-1)$ autres comme échantillon d'apprentissage. Ensuite, on effectue le calcul de l'erreur. On procède ainsi pour les k échantillons, obtenant ainsi k termes d'erreur effectués. La moyenne des erreurs est calculée pour avoir l'erreur de prédiction. Le cas particulier où $k = n$, où n représente la taille de l'échantillon est appelé "leave-one-out cross-validation, (LOOCV)". Ce dernier cas n'est pratiqué que si la taille de l'échantillon n'est pas trop grande.

Pour les régressions *Ridge* et Lasso, on se donne un intervalle des valeurs possibles de λ (les valeurs pouvant varier de 1 à 1000 par exemple), et après la validation croisée, on décide de celle qui définirait le mieux le modèle.

Dans le cas où les variables sont fortement corrélées et dans le cas où le nombre de variables explicatives est très élevé, on peut penser à réduire le nombre de variables explicatives afin de choisir celles qui expliquent le mieux la variable endogène. La régression *Ridge* ne fait pas la sélection des variables. Même si elle permet d'avoir une solution unique pour $\hat{\boldsymbol{\beta}}$ et donc de faire la prédiction, l'interprétation du modèle n'est pas toujours évidente. Le problème du choix du λ computationnel suggère de penser aux techniques de réduction de variables comme l'analyse en composantes principales.

1.3.4 Régression sur les composantes principales : PCR

L'analyse en composantes principales (ACP) (ou PCR de l'anglais *principal compoment regression*) est une technique d'analyse des données. On voudrait, à partir des m variables explicatives, $(x_1; x_2; \cdots; x_m)$, construire d'autres variables, au nombre de $k \leq m$ qui expliquent aussi bien la variable y. Ces k nouvelles variables sont alors appelées composantes principales ou facteurs. Soit n sujets sur lesquels on a mesuré les variables. On note alors $\mathbf{X} = \mathbf{X}_{n\times m}$ et $\mathbf{y} = \mathbf{Y}_{n\times 1}$ l'échantillon considéré. Une composante principale est une combinaison linéaire des variables initiales (c'est-à-dire une combinaison linéaire des colonnes de la matrice $\mathbf{X}$). Les composantes principales sont toutes orthogonales entre elles, ce qui permet d'éliminer l'effet de la multicolinéarité des variables initiales. Pour déterminer les composantes principales, on utilise la technique SVD (de l'anglais, singular value decomposition), qui consiste à effectuer la décomposition de la matrice des variables explicatives de la manière suivante :

$$\mathbf{X}_{n\times m} = \mathbf{U}_{n\times r}\mathbf{D}_{r\times r}\mathbf{V}^{\top}_{r\times m},$$

où r est le rang de la matrice $\mathbf{X}$,

$$\mathbf{U}_{n\times r} = (\mathbf{u}_1; \cdots; \mathbf{u}_r)$$

et

$$\mathbf{V}_{m\times r} = (\mathbf{v}_1; \cdots; \mathbf{v}_r)$$

des matrices orthogonales telles que $\mathbf{U}^{\top}\mathbf{U} = \mathbf{I}_r$ et $\mathbf{V}^{\top}\mathbf{V} = \mathbf{I}_r$. La matrice $\mathbf{D}_{r\times r}$ est diagonale et définie par

$$\mathbf{D} = diag(\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq \ldots \geq \sqrt{\lambda_r}).$$

avec $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r$ les valeurs propres de la matrice $\mathbf{X}\mathbf{X}^{\top}$ décomposée par

$$\mathbf{X}\mathbf{X}^{\top} = \mathbf{U}\mathbf{D}^2\mathbf{U}^{\top}.$$

Les vecteurs $(\mathbf{u}_1; \mathbf{u}_2; \cdots; \mathbf{u}_r)$ sont alors les directions des composantes principales. Ils sont, à un facteur près, les composantes principales. La jème composante principale est donnée par

$$\mathbf{z}_j = \mathbf{X}\mathbf{v}_j = \sqrt{\lambda_j}\mathbf{u}_j.$$

On effectue alors la régression de $\mathbf{y}$ sur les composantes principales

$$\mathbf{Z} = (\mathbf{z}_1; \cdots; \mathbf{z}_r),$$

de dimension réduite ($r \leq m$) et dont les nouvelles variables explicatives ne sont plus colinéaires. Le modèle peut s'écrire ainsi comme suit

$$\mathbf{y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\varepsilon} = (\mathbf{X}\mathbf{v})\boldsymbol{\beta} + \boldsymbol{\varepsilon} = \mathbf{U}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\alpha} = \mathbf{D}\boldsymbol{\beta}, \tag{1.3}$$

ce qui permettrait d'estimer $\boldsymbol{\beta}$ par

$$\hat{\boldsymbol{\beta}} = (\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top\mathbf{y}.$$

Quand les variables sont très nombreuses et qu'on voudrait résumer le mieux l'information, on utilise des critères pour le choix des composantes. On peut alors utiliser le *critère de Kaiser* : on considère les composantes dont la variance est supérieure à celle des variables analysées ; ce qui revient, pour des données centrées réduites, à considérer seulement les composantes qui correspondent à une valeur propre strictement plus grande que 1. Un autre critère utilisé est celui du graphique qui donne les valeurs propres en fonction du numéro des composantes. On devrait alors choisir toutes les composantes qui viennent avant "le coude" observé sur le graphique (Stéphane Tufféry, 2012).

La méthode PCR maximise donc la variance entre les prédicteurs. Elle ne tient pas compte de la corrélation qui existerait entre les prédicteurs et la variable à expliquer. La méthode PLS est donc cette méthode qui généralise PCR tout en maximisant la corrélation entre $\mathbf{X}$ et $\mathbf{y}$.

1.3.5 Régression sur les composantes principales : PLS

La méthode PLS tire ses origines de l'article de Herman Wold dans les années 70 (Wold, 1966) en sciences sociales. Elle a été très vite généralisée dans d'autres domaines. Pour cette méthode, on va toujours transformer les données en des données centrées et réduites. Pour un échantillon de n individus, 0n dispose d'une matrice

$$\mathbf{X} = \mathbf{X}_{n\times m} = (\mathbf{x}_1, \cdots, \mathbf{x}_m),$$

de m variables explicatives et d'un vecteur

$$\mathbf{y} = \mathbf{Y}_{n\times 1}$$

d'une variable à expliquer. On veut déterminer des combinaisons linéaires des colonnes de $\mathbf{X}$, liées à $\mathbf{y}$ et ensuite les utiliser pour prédire $\mathbf{y}$. On utilise généralement la méthode PLS en présence de multicolinéarité et quand le nombre de variables explicatives est très grand, supérieur au nombre d'observations. Un des objectifs ici est de trouver les combinaisons linéaires des colonnes de $\mathbf{X}$ qui maximisent simultanément la variance totale dans $\mathbf{X}$ (ce qui revient à une PCR) ainsi que la corrélation entre cette combinaison linéaire et le vecteur $\mathbf{y}$.
Formellement, la méthode PLS cherche des combinaisons linéaires des colonnes de $\mathbf{X}$, $\mathbf{t}_j = \mathbf{X}\mathbf{w}_j$ qui maximisent $\frac{\mathbf{y}^\top \mathbf{X}\mathbf{w}_j}{||\mathbf{y}||||\mathbf{X}\mathbf{w}_j||}$, $\mathbf{w}_j \in \mathbb{R}^m$, $j \in \{1, 2, \cdots, r\}$, avec

$$||\mathbf{t}_j||^2 = \mathbf{t}_j^\top \mathbf{t}_j = 1$$

et

$$\mathbf{t}_j^\top \mathbf{t}_k = 0,\ j \neq k.$$

Ces composantes sont déterminées de manière itérative (Robert Sabatier, 2010). En effet, la première composante est déterminée par

$$\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1 = w_{11}\mathbf{x}_1 + w_{21}\mathbf{x}_2 + \cdots + w_{m1}\mathbf{x}_m = \sum_{j=1}^{m} w_{j1}\mathbf{x}_j,$$

avec

$$w_{j1} = \frac{\mathbf{x}_j^\top \mathbf{y}}{(\sum_{k=1}^m (\mathbf{x}_k^\top \mathbf{y})^2)^{\frac{1}{2}}}.$$

On effectue alors la régression simple de $\mathbf{y}$ sur $\mathbf{t}_1$ comme suit

$$\mathbf{y} = C_1 \mathbf{t}_1 + \mathbf{y}_{res1},$$

avec

$$\hat{C}_1 = (\mathbf{t}_1^\top \mathbf{t}_1)^{-1} \mathbf{t}_1^\top \mathbf{y},$$

et le résidu

$$\mathbf{y}_{res1} = \mathbf{y} - C_1 \mathbf{t}_1.$$

Ainsi, la prédiction associée à la première composante est

$$\hat{\mathbf{y}}^{(1)} = \hat{C}_1 \mathbf{t}_1 = \sum_{j=1}^m \hat{\beta}_j^{(1)} \mathbf{x}_j.$$

Les estimateurs des coefficients du modèle de regression de $\mathbf{y}$ sur les colonnes de $\mathbf{X}$ sont données par

$$\hat{\beta}_j^{(1)} = \hat{C}_1 w_{j1}, \quad j = 1, \cdots, m.$$

Pour construire la deuxième composante, $\mathbf{t}_2$, non corrélée à la première, on recommence le même processus, mais cette fois avec le résidu de $\mathbf{y}$, $\mathbf{y}_{res1}$ et les résidus des variables $\mathbf{x}_j$ par régression simple sur la première composante $\mathbf{t}_1$. Autrement dit, on écrit

$$\mathbf{x}_j = P_{1j} \mathbf{t}_1 + \mathbf{x}_j^{(1)},$$

où $\mathbf{x}_j^{(1)}$ est le résidu de la régression simple de $\mathbf{x}_j$ sur $\mathbf{t}_1$. La deuxième composante, $\mathbf{t}_2$, est alors obtenue comme combinaison linéaire des $\mathbf{x}_j^{(1)}$:

$$\mathbf{t_2} = w_{12} \mathbf{x}_1^{(1)} + w_{22} \mathbf{x}_2^{(1)} + \cdots + w_{m2} \mathbf{x}_m^{(1)} = \sum_{j=1}^m w_{j2} \mathbf{x}_j^1,$$

avec

$$w_{j2} = \frac{\mathbf{x}_j^{(1)\top} \mathbf{y}_{res1}}{(\sum_{k=1}^m (\mathbf{x}_k^{(1)\top} \mathbf{y}_{res1})^2)^{\frac{1}{2}}}.$$

On effectue alors la régression simple de $\mathbf{y}_{res1}$ sur $\mathbf{t}_2$,

$$\mathbf{y}_{res1} = C_2\mathbf{t}_2 + \mathbf{y}_{res2},$$

avec

$$\hat{C}_2 = (\mathbf{t}_2^\top \mathbf{t}_2)^{-1}\mathbf{t}_2^\top \mathbf{y}_{res1},$$

et le résidu est

$$\mathbf{y}_{res2} = \mathbf{y}_{res1} - C_2\mathbf{t}_2.$$

Ainsi, la prédiction associée aux deux premières composantes est donc

$$\hat{\mathbf{y}}^{(2)} = \hat{C}_1\mathbf{t}_1 + \hat{C}_2\mathbf{t}_2 = \sum_{j=1}^{m} \hat{\beta}_j^{(2)}\mathbf{x}_j.$$

On recommence le processus jusqu'à l'obtention de r composantes, avec $r \leq m$. Le choix de r se fait par validation croisée, ou en se fixant un seuil pour la variance expliquée dans $\mathbf{X}$ (soit, par exemple, un seuil supérieur ou égal à 80%). Contrairement à la méthode PCR, pour PLS, la décomposition se fait de manière simultanée et itérative entre $\mathbf{X}$ et $\mathbf{y}$.

1.4 Vocabulaire génétique

La **cellule** est l'unité de base de tout organisme. C'est dans le noyau de la cellule qu'on retrouve généralement de l'information génétique. L'information génétique se trouve dans une molécule appelée **ADN (acide déoxyribonucléique)**. Pendant la multiplication des cellules, l'ADN prend la forme de chromosomes.
Le **génome** est l'ensemble de l'ADN présent dans le noyau de chaque cellule d'un individu . Un **gène** est une petite portion ou un segment de l'ADN . Chaque gène détermine une caractéristique spéciale d'un individu et se transmet héréditairement. La position du gène sur le chromosome est appelée **locus** ou **loci** au pluriel.
Les différentes formes prises par le gène sont des **allèles**. L'ensemble de gènes

constitue le génotype, tous les caractères héréditaires. Le génotype va donc produire en partie le phénotype. Le phénotype est l'ensemble de tous les caractères ou traits observables d'un individu. Il est fortement influencé par le génotype, mais dépend aussi de l'environnement et des mutations. Une **mutation** est une modification rare, accidentelle ou provoquée, de l'information génétique (séquence d'ADN) dans le génome.

La **fréquence du génotype** dans une population est la proportion des individus qui possède ce génotype dans la population. La **fréquence allélique** est la proportion dans la population d'un allèle particulier par rapport à toutes les copies possibles d'allèles. Pour un SNP donné, l'allèle le moins fréquent dans la population de ce SNP est appelé **allèle** mineur (on l'identifie aussi à l'allèle qui cause la maladie). En statistique génétique, les **marqueurs génétiques** sont des séquences d'ADN qui ne varient pas d'un individu à l'autre et servent donc de point de repère pour déterminer la position des autres gènes sur le génome. Le marqueur génétique le plus simple est le **SNP** (de l'anglais, **Single-Nucleotide Polymorphism**). Pour n individus, on va représenter l'ensemble de leur SNP par une variable qui prend les valeurs dans $\{0, 1, 2\}$. Par exemple, soit un SNP avec deux allèles (T,G) où l'allèle mineur est noté "G". On note $z_i = 0$ si on observe "TT" chez l'individu i, $i = 1, 2, \cdots, n$. De même, on note $z_i = 1$ si on observe "TG" ou "GT" chez l'individu i et on note $z_i = 2$ si on observe "GG" chez l'individu i.

En statistiques génétiques, on veut identifier les SNPs qui causent les maladies héréditaires. On va alors se poser plusieurs questions comme chercher le lien entre la maladie (variable réponse) et le gène ou les gènes (prédicteurs), quel est la position de ces gènes sur le génome, est-ce qu'il existe une association entre la maladie et l'environnement, l'âge, les gènes,... Pour répondre à ces questions, une étude d'agrégation familiale et autres études peuvent être faites. (On peut consulter le chapitre 2 de (Gang *et al.*, 2012) pour plus de détails).

CHAPITRE II

MESURES D'ASSOCIATION

L'analyse d'association a pour but d'étudier l'existence de relations entre des variables. C'est une technique très souvent utilisée pour identifier les variants génétiques en lien avec certaines maladies complexes comme le diabète, l'obésité, l'Alzheimer, etc. On distingue l'analyse d'association SLAS(de l'anglais, Single Locus Association studies) et l'analyse d'association MLSA(de l'anglais, Multilocus Association Studies). On distingue les cas où l'on mesure l'association entre une variable d'intérêt (variable réponse) et un ensemble de variables explicatives et le cas où on mesure l'association entre deux ensembles de variables.

La corrélation multiple se définit comme la corrélation entre une variable réponse (y_1) et un ensemble de variables explicatives, $(x_1, x_2, \cdots, x_p)$ donnée au chapitre 1 par l'équation (2.4). La mesure la plus utilisée pour tester l'association entre deux ensembles de variables $(x_1; x_2; \cdots; x_p)$ et $(y_1; y_2; \cdots; y_q)$ est la corrélation canonique entre ces variables. C'est une statistique qui mesure la dépendance linéaire entre les deux ensembles de variables et est une extension de la corrélation multiple. Un exemple d'application est la corrélation entre un trait de caractère et un ensemble de SNPs. Dans ce chapitre, nous décrivons une mesure d'association (la corrélation canonique) entre deux ensembles de variables. Nous décrivons également la signification de l'association entre les deux ensembles de variables.

2.1 Mesure d'association entre deux ensembles de variables

Soit $(x_1, x_2, \cdots, x_p)$ et $(y_1, y_2, \cdots, y_q)$ deux ensembles de variables sur lesquels on voudrait mesurer l'association. Soit n sujets sur lesquels on a mesuré les deux ensembles de variables. On obtient un échantillon de taille n dont les matrices des observations sont données par : $\mathbf{X} = \mathbf{X}_{n\times p} = (\mathbf{x}_1; \mathbf{x}_2; \cdots; \mathbf{x}_p)$ et $\mathbf{Y} = \mathbf{Y}_{n\times q} = (\mathbf{y}_1; \mathbf{y}_2, \cdots, \mathbf{y}_q)$. On pose

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_{yy} & \mathbf{S}_{yx}^\top \\ \mathbf{S}_{yx} & \mathbf{S}_{xx} \end{pmatrix},$$

la matrice de variance-covariance échantillonnale entre $\mathbf{X}$ et $\mathbf{Y}$. Les matrices $\mathbf{S}_{yy}$ et $\mathbf{S}_{xx}$ désignent les matrices de variance-covariance de $\mathbf{Y}$ et $\mathbf{X}$ respectivement, tandis que $\mathbf{S}_{yx}$ est la matrice de variance-covariance échantillonnale entre les vecteurs de $\mathbf{Y}$ et ceux de $\mathbf{X}$. De même, on note par

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{yy} & \mathbf{R}_{yx}^\top \\ \mathbf{R}_{yx} & \mathbf{R}_{xx}, \end{pmatrix},$$

la matrice des corrélations échantillonnales entre les variables $(x_1, x_2, \cdots, x_p)$ et $(y_1, y_2, \cdots, y_q)$.

2.1.1 Cas particulier, $q = 1$

Dans le cas particulier où $q = 1$, la matrice $\mathbf{Y}$ se réduit à un seul vecteur $\mathbf{y} = \mathbf{Y}_{n\times 1}$ On peut mesurer l'association en calculant le coefficient de détermination (Rencher, 2002) donné comme suit

$$\begin{aligned} R^2 &= \frac{\mathbf{S}_{yx}^\top \mathbf{S}_{xx}^{-1} \mathbf{S}_{yx}}{\mathbf{S}_{yy}} \\ &= \mathbf{R}_{yx}^\top \mathbf{R}_{xx}^{-1} \mathbf{R}_{yx}. \end{aligned}$$

Au chapitre 1, nous avons vu que le coefficient de détermination est donné par la relation

$$R = \max_{\substack{\beta_i \in \mathbb{R} \\ i \in \{1,2,\cdots,m\}}} \{Corr(y, \sum_{i=1}^{m} \beta_i x_i)\} \tag{2.1}$$

Noter que R^2 est une mesure globale de l'association entre la variable y et toutes les variables $x_j;\ j = 1; \cdots; p$. Nous pouvons également calculer l'association entre $\mathbf{Y}$ et chacune des variables $x_j, (j = 1; \cdots; p)$ sans tenir compte des autres variables $x_k,\ (k \neq j)$, en calculant tout simplement la corrélation entre $\mathbf{y}$ et chacune des variables $x_1, \cdots, x_p$. Ainsi, nous pouvons écrire

$$\begin{aligned} R_{jy} &= \frac{S_{jy}}{S_{jj} S_{yy}} \\ &= \frac{\sum_{i=1}^{n} (x_{ij} - \overline{x}_j)(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_{ij} - \overline{x}_j)^2 \sum_{i=1}^{n} (y_i - \overline{y})^2}}, \end{aligned}$$

où R_{jy} désigne la corrélation entre la variable y et la variable $x_j,\ j = 1, \cdots, p$.

2.1.2 Cas général, $q > 1$

On peut analyser la corrélation canonique pour étudier les relations existant entre deux ensembles de variables. Pour chaque ensemble de variables $\mathbf{x} = (x_1; x_2; \cdots; x_p)$ et $\mathbf{y} = (y_1; y_2; \cdots; y_q)$, on détermine deux nouvelles variables, appellées premières variables canoniques, qui sont des combinaisons linéaires des variables $\mathbf{x}$ et $\mathbf{y}$ respectivement

$$v_1 = \mathbf{a}^\top \mathbf{x},\ \mathbf{a} \in \mathbb{R}^p$$

et

$$u_1 = \mathbf{b}^\top \mathbf{y},\ \mathbf{b} \in \mathbb{R}^q.$$

Les deux premières variables canoniques sont construites avec la condition que leur coefficient de corrélation, r_1, soit maximal.

$$r_1 = corr(v_1, u_1) = \max_{\mathbf{a},\mathbf{b}} \{corr(\mathbf{a}^\top \mathbf{x}, \mathbf{b}^\top \mathbf{y})\}.$$

On détermine ensuite les deuxièmes variables canoniques de la même manière que les précédentes, avec la condition qu'elles ne soient pas correlées aux premières. On construit ainsi $r_1, r_2, \cdots, r_s$ corrélations canoniques, avec $s = min(p, q)$. Dans l'échantillon, les coefficients $\mathbf{a}_i$ et $\mathbf{b}_i$ correspondants aux variables canoniques $v_i = \mathbf{a}_i^\top \mathbf{x}$ et $u_i = \mathbf{b}_i^\top \mathbf{y}$ sont en réalité les vecteurs propres (Rencher, 2002) des matrices

$$\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}$$

et

$$\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}.$$

Les corrélations $r_1, r_2, \cdots, r_s$ sont alors les racines carrées positives des valeurs propres associées. Ces corrélations mesurent l'association entre les deux ensembles de variables ; r_1 est la corrélation maximale parmi toutes les corrélations (entre les variables prises individuellement ou ensemble).

Lorsqu'on a deux ensembles de variables, $\mathbf{x}$ et $\mathbf{y}$, pour mesurer l'association, une généralisation du coefficient de détermination de l'équation (2.1) serait le rapport des déterminants des matrices de variance-covariance,

$$\begin{aligned} R_M^2 &= \frac{|\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}|}{|\mathbf{S}_{yy}|} \\ &= |\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}| \\ &= \prod_{i=1}^{s} r_i^2, \end{aligned} \tag{2.2}$$

avec $s = min(p, q)$, et r_i^2 les valeurs propres de la matrice

$$\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}.$$

On peut aussi mesurer l'association entre $\mathbf{x}$ et $\mathbf{y}$ par la statistique

$$R_V = \frac{tr(\mathbf{S}_{xy}\mathbf{S}_{yx})}{\sqrt{tr(\mathbf{S}_{xx}^2)tr(\mathbf{S}_{yy}^2)}}. \tag{2.3}$$

D'autres mesures d'association sont données par Rencher (2002). Toutes ces mesures ne donnent pas le même degré d'association. Il reste encore beaucoup de

recherches à faire pour que l'une ou plusieurs de ces mesures soient recommandées de façon générale. La corrélation canonique donne plusieurs mesures d'association. La première corrélation canonique est la maximale. La statistique dans l'équation (2.2), qui est le produit des r_i^2, $i \in \{1, 2, \cdots, s\}$ détermine une mesure d'association qui serait très faible parce que les $0 \leq r_i^2 \leq 1$. Dans le cadre de ce travail, nous utilisons comme mesure d'association la statistique R_V donnée par l'équation (3.3). En effet, l'implémentation de cette statistique est facile, et nous montrons aussi au chapitre suivant que dans le cas d'une association entre une variable réponse et des variables explicatives, cette statistique coincide avec l'approche utilisée par Xu et Greenwood (2013). Comment décider s'il y aurait une certaine association ou un lien significatif entre les deux ensembles de variables après l'obtention des mesures d'association ?

2.2 Test de signification

Nous considérons comme hypothèse nulle, H_0, le fait qu'il n'y a pas de relation (linéaire) entre les ensembles de variables $\mathbf{x} = (x_1; x_2; \cdots; x_p)$ et $\mathbf{y} = (y_1; y_2; \cdots; y_q)$. Ainsi, l'hypothèse nulle est définie comme suit :

H_0 : La covariance entre chaque y_k et chaque x_j est nulle.

$\Leftrightarrow$ La correlation entre chaque y_k et chaque x_j est presque nulle.

$\Leftrightarrow$ Toutes les corrélations cannoniques $r_1, r_2, \cdots, r_s$ sont proches de zéro.

Dans le cas unidimensionnel ($q = 1$), on va conclure l'absence de relation linéaire entre y et les variables $x_1, x_2, \cdots, x_p$ si la covariance entre y et chacune des variables de x_k est nulle. Ou de manière équivalente, la corrélation canonique r_1 n'est pas significativement différente de zéro. Nous pouvons également faire le lien entre H_0 défini par la corrélation cannonique et
$H_0 : \beta = \mathbf{0}_p$ dans un modèle de regression qui associe $\mathbf{y}$ à $\mathbf{X}$. En effet, supposons

le modèle de regression

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon,$$

on a

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}$$

et

$$R = \max_{\substack{\beta_i \in \mathbb{R} \\ i \in \{1,2,\cdots,m\}}} \{Corr(y, \sum_{i=1}^{m} \beta_i x_i)\} \tag{2.4}$$

Dans le cas multidimensionnel ($q > 1$), on va conclure l'absence de relation linéaire entre y et les variables $x_1, x_2, \cdots, x_p$ si tous les coefficients du modèle linéaire sont nuls. La corrélation peut être testée par certaines statistiques. On peut, par exemple, tester la signification des corrélations cannoniques $r_1, r_2, \cdots, r_s$ en utilisant la statistique de Wilks. Dans ce cas, pour tester r_1, on calcule

$$\begin{aligned} \Lambda_1 &= \frac{|\mathbf{S}|}{|\mathbf{S}_{yy}||\mathbf{S}_{xx}|} \\ &= \frac{|\mathbf{R}|}{|\mathbf{R}_{yy}||\mathbf{R}_{xx}|} \\ &= \prod_{i=1}^{s}(1 - r_i^2). \end{aligned}$$

qui est distribué selon la loi de Wilks $\Lambda_{p,q,n-1-q} = \Lambda_{q,p,n-1-p}$ sous H_0, avec n qui représente le nombre de sujets. On rejette H_0 si $\Lambda_1 \leq \Lambda_\alpha$; où Λ_α est la valeur critique de la statistique de Wilks (Rencher, 2002). Si le test Λ_1 rejette H_0, on ne sait pas si les autres corrélations sont significatives. Pour tester la significativité des autres corrélations, r_k, on calcule

$$\Lambda_k = \prod_{j=k}^{s}(1 - r_k^2).$$

qui est distribué selon $\Lambda_{p-k+1,q-k+1,n-k-q}$ sous l'hypothèse nulle. Si ces calculs permettent de rejeter l'hypothèse nulle, alors on conclut qu'au moins un des $r_1, r_2, \cdots, r_s$ n'est pas significativement nul.

Un autre test multivarié est le test défini par la statistique de Roy :

$$\theta = r_1^2 \text{ ou } \theta = \frac{r_1}{1+r_1}.$$

cette statistique est distribuée selon une distribution de Fisher avec les degrés de liberté $dl_1 = n - q - d - 1$ et $dl_2 = d$,

$$F = (\frac{n-q-d-1}{d})r_1,$$

avec $d = max(p, q)$.

CHAPITRE III

L'APPROCHE DES MOINDRES CARRÉS PARTIELS MULTIDIMENSIONELLE

Dans le présent chapitre, nous décrivons une approche qui nous permettrait de faire des analyses pour des données de grandes dimensions, en considérant plusieurs variables à la fois comme réponse. On veut donc décrire une nouvelle façon de mesurer l'association entre deux ensembles de variables, une nouvelle approche basée sur la distribution de Pareto généralisée pour évaluer ainsi que tester une telle association. Pour commencer, nous décrivons d'abord la régression PLS dans le cas multidimensionnel..

3.1 La méthode des moindres carrés partiels multidimensionnelle.

On dispose d'une matrice d'observations de p variables explicatives sur n sujets.

$$\mathbf{X} = \mathbf{X}_{n\times p} = (\mathbf{x}_1, \cdots, \mathbf{x}_p);$$

et d'une matrice

$$\mathbf{Y} = \mathbf{Y}_{n\times q} = (\mathbf{y}_1, \cdots, \mathbf{y}_q),$$

de q variables réponses sur les n sujets.

Pour cette méthode, il faut construire les vecteurs latents qui sont des combinaisons linéaires des matrices $\mathbf{X}$ et $\mathbf{Y}$ vérifiant certaines caractéristiques. Ensuite, il faut construire les scores (les X-loadings et les Y-loadings) des matrices $\mathbf{X}$ et $\mathbf{Y}$.

Ces constructions se font de manière itérative (voir les algorithmes descriptifs ci-dessous).Uune forme explicite de ces constructions n'existe pas. L'itération prend fin quand dans le processus de décomposition, l'une des matrices $\mathbf{X}$ ou $\mathbf{Y}$ devient nulle.

Ainsi, l'objectif de la méthode dans un premier temps est de construire deux vecteurs. L'un des vecteurs est une combinaison linéaire des colonnes de $\mathbf{X}$ et l'autre vecteur est une combinaison linéaire des colonnes de $\mathbf{Y}$ tels que la covariance entre les deux vecteurs (composantes) soit maximale. Autrement dit, un des objectifs ici est de maximiser la variation présente dans $\mathbf{X}$ (ce qui revient à une PCR) et aussi de maximiser la corrélation entre les matrices $\mathbf{X}$ et $\mathbf{Y}$. L'objectif se traduit formellement comme suit : on construit deux vecteurs

$$\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1, \text{ et } \mathbf{u}_1 = \mathbf{Y}\mathbf{c}_1$$

tels que $\mathbf{t}_1^\top \mathbf{t}_1 = 1$, et $\mathbf{t}_1^\top \mathbf{u}_1$ maximale. Formellement, nous pouvons écrire

$$\max\ \mathbf{t}_1^\top \mathbf{u}_1 \Leftrightarrow \max_{\substack{w_1 \in \mathbb{R}^p \\ c_1 \in \mathbb{R}^q}} \mathbf{w}_1^\top \mathbf{X}^\top \mathbf{Y} \mathbf{c}_1.$$

Les vecteurs $\mathbf{u}_1$ et $\mathbf{t}_1$ sont appelés vecteurs latents. Ils sont obtenus de manière itérative : on pose $\widetilde{\mathbf{X}} = \mathbf{X}$, $\widetilde{\mathbf{Y}} = \mathbf{Y}$.

Une fois les vecteurs $\mathbf{t}_1$ et $\mathbf{u}_1$ déterminés, on fait la régression de $\mathbf{X}$ sur $\mathbf{t}_1$ et la régression de $\mathbf{Y}$ sur $\mathbf{u}_1$; avec les modèles respectifs :

$$\mathbf{X} = \mathbf{t}_1 \mathbf{p}_1^\top + \widetilde{\varepsilon}_X,$$

$$\mathbf{Y} = \mathbf{u}_1 \mathbf{q}_1^\top + \widetilde{\varepsilon}_Y.$$

On obtient alors deux vecteurs $\mathbf{p}_1$ et $\mathbf{q}_1$, qui constituent les colonnes des matrices X-loadings et Y-loadings, les scores. L'algorithme 2 permet de déterminer ces vecteurs.

Algorithme 1 itératif pour trouver $\mathbf{t}_1$ et $\mathbf{u}_1$ les vecteurs latents.

1. Poser $\mathbf{E} = \widetilde{\mathbf{X}}$, $\mathbf{F} = \widetilde{\mathbf{Y}}$;
2. Poser $\mathbf{u} = \mathbf{f}_1$, la première colonne de $\mathbf{F}$;
3. Poser $\mathbf{w}_1 = \mathbf{E}^\top \mathbf{u}/(\mathbf{u}^\top \mathbf{u})$;
4. Normaliser $\mathbf{w}_1$ pour qu'il soit de norme 1 $(\mathbf{w}_1 \to \mathbf{w}_1/\sqrt{\mathbf{w}_1^\top \mathbf{w}_1})$;
5. Poser $\mathbf{t}_1 = \mathbf{E}\mathbf{w}_1$;
6. Calculer $\mathbf{c}_1 = \mathbf{F}^\top \mathbf{t}_1$;
7. Normaliser $\mathbf{c}_1$ pour qu'il soit de norme 1 $(\mathbf{c}_1 \to \mathbf{c}_1/\sqrt{\mathbf{c}_1^\top \mathbf{c}_1})$;
8. Calculer $\mathbf{u}_1 = \mathbf{F}\mathbf{c}_1$;
9. S'il y a convergence (voir A.1 pour les critères de convergenceS), passer à l'algorithme 2 ; sinon reprendre à l'étape 2 avec $\mathbf{u} = \mathbf{u}_1$.

Algorithme 2 itératif pour trouver $\mathbf{p_1}$ et $\mathbf{q_1}$ les colonnes des matrices X-loadings et Y-loadings.

1. **X**-loadings : $\mathbf{p}_1 = \mathbf{E}^\top \mathbf{t}_1$;
2. **Y**-loadings : $\mathbf{q}_1 = \mathbf{F}^\top \mathbf{u}_1/(\mathbf{u}_1^\top \mathbf{u}_1)$;
3. Faire la régression de $\mathbf{u}_1$ par rapport à $\mathbf{t}_1$: $b_1 = \mathbf{u}_1^\top \mathbf{t}_1$;
4. Calculer les matrices résiduelles $\mathbf{E} \to \mathbf{E} - \mathbf{t}_1\mathbf{p}^\top$ et $\mathbf{F} \to \mathbf{F} - b_1\mathbf{t}_1\mathbf{c_1}^\top$;
5. **Retourner** $\mathbf{u}_1$, $\mathbf{t}_1$, $\mathbf{p}_1$, et $\mathbf{q}_1$.

On a ainsi déterminé $\mathbf{u_1}$, $\mathbf{t_1}$, $\mathbf{p_1}$ et $\mathbf{q_1}$. On reprend le processus (les algorithmes 1 et 2) avec les nouvelles matrices $\widetilde{\mathbf{X}} = \mathbf{E}$ et $\widetilde{\mathbf{Y}} = \mathbf{F}$. Le processus prend fin quand l'une des matrices résiduelles est nulle. Les vecteurs latents et les scores sont donc consignés dans des matrices $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_r]$, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r]$, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_r]$ et $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_r]$, avec r le nombre de composantes possibles. Il existe plusieurs formes de PLS. L'algorithme décrit ici est l'algorithme du NIPAL (de l'anglais Nonlinear iterative partial least squares)(Abdi, 2010). Cette décomposition a pour but de former deux ensembles de vecteurs, $\mathbf{T} = \mathbf{XW}$ et $\mathbf{U} = \mathbf{YC}$, qui sont des combinaisons linéaires respectives des colonnes de $\mathbf{X}$ et $\mathbf{Y}$, et dont la covariance est maximale (tels que $\mathbf{w}_i^\top \mathbf{w_i} = 1$, $\mathbf{t}_i = \mathbf{X}\mathbf{w_i} = 1$, $\mathbf{t}_i^\top \mathbf{t_i} = 1$ et $\mathbf{t}_i^\top \mathbf{u_i}$ est maximale). La forme explicite de la décomposition n'existe pas. Cette décomposition se fait suivant un algorithme itératif jusqu'à l'obtention de la matrice nulle (c'est-à-dire $\mathbf{E} \approx \mathbf{0}$ ou $\mathbf{F} \approx \mathbf{0}$. Nous avons programmé cet algorithme avec le logiciel R. Vous trouverez en annexe (A.1) le code R. Le nombre de composantes nécessaires se détermine par la validation croisée.

La méthode de Abdi (2010) peut être vue formellement comme suit. On cherche à décomposer $\mathbf{X}$ et $\mathbf{Y}$ de la manière suivante :

$$\mathbf{X} = \mathbf{TP}^\top + \mathbf{E}, \tag{3.1}$$

$$\mathbf{Y} = \mathbf{UQ}^\top + \mathbf{F}, \tag{3.2}$$

avec $\mathbf{T}^\top\mathbf{T} = \mathbf{I}_r$ et $\mathbf{W}^\top\mathbf{W} = \mathbf{I}_r$ où $\mathbf{I}_r$ représente la matrice identité dont la dimension est le nombre de vecteurs latents. Les matrices $\mathbf{T}$ et $\mathbf{U}$ sont les matrices des scores $\mathbf{X}$ et $\mathbf{Y}$ respectivement. Leurs colonnes sont appelées vecteurs latents. Les matrices de format $p \times r$ et $q \times r$, $\mathbf{P} = \mathbf{P}_{p\times r}$ et $\mathbf{Q} = \mathbf{Q}_{q\times r}$, sont les matrices des loadings de $\mathbf{X}$ et $\mathbf{Y}$ respectivement avec r le nombre de vecteurs latents. Dans ces algorithmes, on a utilisé le modèle $\mathbf{Y} = \mathbf{UQ}^\top + \mathbf{F}$, et on a aussi effectué la régression de chaque vecteur $\mathbf{u}_i$ par rapport au vecteur $\mathbf{t}_i$ correspondant, avec pour coefficients de régression les b_i ; $i \in \{1, 2, \cdots, r\}$ où r désigne le nombre de

composantes. On peut alors écrire

$$\mathbf{U} = \mathbf{TB} + \varepsilon_{\mathbf{Y}},$$

avec $\mathbf{B}$ la matrice diagonale dont les éléments de la diagonale sont les poids de la régression (en anglais "*regression weights*"), les b_i.

Une autre façon d'obtenir les composantes $\mathbf{T}$ et $\mathbf{U}$ serait par la méthode SVD (de l'anglais, Singular Value Decomposition) décrite par Björn-Helge et Wehrens (2007).

On effectue alors la décomposition en valeurs singulières de la matrice $\mathbf{X}^\top\mathbf{Y}$. On note par $\mathbf{w}$ le premier vecteur singulier à droite et $\mathbf{q}$ le premier vecteur singulier à gauche. Les Scores de $\mathbf{X}$ et de $\mathbf{Y}$ sont alors obtenus de la manière suivante respectivement :

$$\mathbf{t} = \mathbf{Xw},$$

$$\mathbf{u} = \mathbf{Yq}.$$

Les $\mathbf{X}$-scores sont normalisés. On remplace ainsi $\mathbf{t}$ par $\mathbf{t}/\sqrt{\mathbf{t}^\top\mathbf{t}}$. Les $\mathbf{X}$-loadings et les $\mathbf{Y}$-loadings sont obtenus respectivement par

$$\mathbf{p} = \mathbf{X}^\top\mathbf{t},$$

$$\mathbf{q} = \mathbf{Y}^\top\mathbf{u}.$$

On peut résumer ce processus dans l'algorithme 3.

On obtient ainsi les colonnes des matrices des scores et loadings en effectuant la décomposition en valeurs singulières. Les vecteurs $\mathbf{t}, \mathbf{u}, \mathbf{w}, \mathbf{q}$ et $\mathbf{p}$ sont donc consignés dans des matrices $\mathbf{T}, \mathbf{U}, \mathbf{W}, \mathbf{Q}$ et $\mathbf{P}$ pour former les matrices des scores $(\mathbf{T}, \mathbf{U})$, la matrice des poids weight $\mathbf{W}$ (de l'anglais weight) et les matrices des loadings $(\mathbf{P}, \mathbf{Q})$. Dans (Björn-Helge et Wehrens, 2007), les coefficients de la régression dans le model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

Algorithme 3 : Détermination des composantes par la méthode SVD.

1. Poser $\mathbf{E} = \mathbf{X} = \mathbf{X}_{n\times p}$, $\mathbf{F} = \mathbf{Y} = \mathbf{Y}_{n\times q}$;
2. Faire la SVD de $\mathbf{M} = \mathbf{M}_{p\times q} = \mathbf{E}^\top\mathbf{F}$: $\mathbf{M} = \mathbf{E}^\top\mathbf{F} = \mathbf{ADB}^\top$;
   avec $\mathbf{A}^\top\mathbf{A} = \mathbf{I}_p$; $\mathbf{A}$ est la matrice des vecteurs propres de $\mathbf{MM}^\top$.
   $\mathbf{B}^\top\mathbf{B} = \mathbf{I}_q$; $\mathbf{B}$ est la matrice des vecteurs propres de $\mathbf{M}^\top\mathbf{M}$.
   $\mathbf{D} = \mathbf{D}_{p\times q} = diag(\lambda_1, \ldots, \lambda_k, 0)$ où $k = min(rang(\mathbf{X}), rang(\mathbf{Y}))$ et $\lambda_1 \geq \ldots \geq \lambda_k \geq 0$ sont les valeurs singulières, c'est-à-dire les racines carrées des valeurs propres de $\mathbf{M}^\top\mathbf{M}$ (ou de $\mathbf{MM}^\top$) ;
3. Poser $\mathbf{w} = \mathbf{a}_1$, la première colonne de $\mathbf{A}$
   et $\mathbf{q} = \mathbf{b}_1$, la première colonne de $\mathbf{B}$;
4. Les $\mathbf{X}$-scores : prendre $\mathbf{t} = \mathbf{Ew}$;
5. Les $\mathbf{Y}$-scores : prendre $\mathbf{u} = \mathbf{Fq}$;
6. Normaliser $\mathbf{t}$ pour qu'il soit de norme 1 $(\mathbf{t} \to \mathbf{t}/\sqrt{\mathbf{t}^\top\mathbf{t}})$;
7. Les $\mathbf{X}$-loadings : prendre $\mathbf{p} = \mathbf{E}^\top\mathbf{t}$, pour la première colonne de $\mathbf{X}$-loadings
   $\mathbf{q} = \mathbf{F}^\top\mathbf{t}$, pour la première colonne de $\mathbf{Y}$-loadings ;
8. Poser $\mathbf{E} = \mathbf{E} - \mathbf{tp}^\top$, $\mathbf{F} = \mathbf{F} - \mathbf{tq}^\top$
   et revenir à l'étape 2 jusqu'à ce que $\mathbf{E} \approx \mathbf{0}$ ou $\mathbf{F} \approx \mathbf{0}$.

sont alors obtenus par

$$\boldsymbol{\beta} = \mathbf{R}(\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{Y} = \mathbf{RQ}^\top, \text{ avec } \mathbf{R} = \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}.$$

En effet, dans l'algorithme précédent, celui qui utilise la décomposition SVD, on

a obtenu les relations matricielles suivantes : $\mathbf{P}^\top \simeq \mathbf{T}^\top\mathbf{X}$, $\mathbf{T} \simeq \mathbf{XW}$, et donc

$$\begin{aligned}\mathbf{XR} &= \mathbf{XW}(\mathbf{P}^\top\mathbf{W})^{-1}\\ &= \mathbf{T}(\mathbf{P}^\top\mathbf{W})^{-1}\\ &= \mathbf{T}(\mathbf{T}^\top\mathbf{XW})^{-1}\\ &= \mathbf{T}(\mathbf{T}^\top\mathbf{T})^{-1}\\ &= \mathbf{T}.\end{aligned}$$

Ainsi, $\mathbf{Y}$ prend les différents modèles suivants :

$$\begin{aligned}\mathbf{Y} &= \mathbf{TQ}^\top + \varepsilon'_{\mathbf{Y}}\\ &= \mathbf{XRQ}^\top + \varepsilon'_{\mathbf{Y}}\\ &= \mathbf{X}\boldsymbol{\beta} + \varepsilon_{\mathbf{Y}}.\end{aligned}$$

Et on a

$$\begin{aligned}\boldsymbol{\beta} &= \mathbf{RQ}^\top\\ &= \mathbf{R}(\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{Y}\\ &= \mathbf{RT}^\top\mathbf{Y}.\end{aligned}$$

Dans la suite, nous décrivons une méthode d'association basée sur la PLS multidimensionnelle.

3.2 Une adaptation des moindres carrés partiels multidimensionnelles : MPLS

3.2.1 Mesure d'association selon la méthode MPLS

Nous avons présenté plusieurs mesures d'association dans le cadre de la régression multiple. Dans cette section, nous introduisons une autre mesure d'association. Notre objectif est de décrire la relation linéaire qui existe entre $\mathbf{Y}$ et $\mathbf{X}$. Ou au mieux, tester l'association entre les deux. Pour cela, nous nous servons de $\hat{\mathbf{Y}}$, la prédiction de $\mathbf{Y}$ obtenue par la méthode MPLS pour construire une statistique qui va mesurer le degré d'association entre les deux matrices $\mathbf{X}$ et $\mathbf{Y}$.

Rappelons que dans le cadre de la régression multiple avec $\mathbf{Y}$ un vecteur et $\mathbf{X}$ une matrice,

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} = \mathbf{P}_\mathbf{X}\mathbf{Y}$$

est la projection de $\mathbf{Y}$ sur le sous-espace vectoriel de $\mathbb{R}^n$ engendré par les colonnes de $\mathbf{X}$, ($Span(\mathbf{X})$). Ainsi, la corrélation entre $\mathbf{Y}$ et $\hat{\mathbf{Y}}$ peut être utilisée pour mesurer l'association entre $\mathbf{Y}$ et $\mathbf{X}$. En effet, la corrélation parfaite de 1 ou -1 entre $\mathbf{Y}$ et $\hat{\mathbf{Y}}$ nous indique que $\mathbf{Y}$ est colinaire avec $\hat{\mathbf{Y}}$ et ainsi $\mathbf{Y} \in Span(\mathbf{X})$. Ceci veut dire que $\mathbf{X}$ explique parfaitement ou complètement $\mathbf{Y}$. Nous pouvons aussi montrer que

$$\begin{aligned}
\widehat{corr}(\mathbf{Y}, \hat{\mathbf{Y}}) &= \frac{\mathbf{Y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}}{(\mathbf{Y}^\top\mathbf{Y})^{\frac{1}{2}}((\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y})^\top(\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}))^{\frac{1}{2}}} \\
&= \frac{\mathbf{Y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}}{(\mathbf{Y}^\top\mathbf{Y})^{\frac{1}{2}}(\mathbf{Y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y})^{\frac{1}{2}}} \\
&= \frac{(\mathbf{Y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y})^{\frac{1}{2}}}{(\mathbf{Y}^\top\mathbf{Y})^{\frac{1}{2}}} \\
&= [\frac{\mathbf{S}_{\mathbf{YX}}S_{\mathbf{XX}}^{-1}\mathbf{S}_{\mathbf{XY}}}{\mathbf{S}_{\mathbf{YY}}}]^{\frac{1}{2}} \\
&= R.
\end{aligned}$$

Dans le cas multivarié, plusieurs statistiques sont utilisées dans la littérature pour mesurer la dépendance linéaire entre deux matrices(Rencher, 2002). Pour notre approche, nous utilisons celle suggérée en 1976 par Robert et Escoufier (1976) donnée par

$$R_V = \frac{tr(\mathbf{S}_{\mathbf{Y}\hat{\mathbf{Y}}}\mathbf{S}_{\hat{\mathbf{Y}}\mathbf{Y}})}{\sqrt{tr(\mathbf{S}^2_{\mathbf{YY}})tr(\mathbf{S}^2_{\hat{\mathbf{Y}}\hat{\mathbf{Y}}})}}, \tag{3.3}$$

avec $\mathbf{S}_{\mathbf{YY}}$, $\mathbf{S}_{\hat{\mathbf{Y}}\hat{\mathbf{Y}}}$ la matrice de variance covariance des matrices $\mathbf{Y}$ et $\hat{\mathbf{Y}}$ respectivement. Nous avons aussi

$$\mathbf{S}_{\mathbf{Y}\hat{\mathbf{Y}}} = \widehat{cov}(\mathbf{Y}, \hat{\mathbf{Y}}) = ((\mathbf{S}_{\mathbf{Y}\hat{\mathbf{Y}}})_{ij})_{2q*2q},$$

avec

$$(\mathbf{S}_{\mathbf{Y}\hat{\mathbf{Y}}})_{ij} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{Y}_{ij} - \overline{\mathbf{y}}_j)(\hat{\mathbf{Y}}_{ik} - \overline{\hat{\mathbf{y}}}_k).$$

$\overline{\hat{\mathbf{y}}}_k$ représente la moyenne de la kème colonne de $\hat{\mathbf{Y}}$. Ce choix est motivé par le fait que lorsque $\mathbf{Y}$ se réduit à un seul vecteur, on retrouve la statistique utilisée par l'approche PLS de Xu et al. (2012), et il s'avère que la statistique dans (3.3) est le carré de celle utilisée par Xu et al. (2012).

En effet, soient $\mathbf{A} = \mathbf{A}_{n\times p}$ et $\mathbf{B} = \mathbf{B}_{n\times q}$ des matrices quelconques et soit $\mathbf{S_{AB}}$ la matrice de covariance échantionnale entre $\mathbf{A}$ et $\mathbf{B}$ de dimension $m\times m$ $(m = p+q)$. Nous avons

$$\mathbf{S_{AB}S_{BA}} = (C_{ij})_{m\times m}$$

avec

$$C_{ij} = \sum_{k=1}^{m} (\mathbf{S_{AB}})_{ik}(\mathbf{S_{BA}})_{kj}.$$

Ainsi, sa trace est donnée par

$$\begin{aligned} tr(\mathbf{S_{AB}S_{BA}}) &= \textstyle\sum_{i=1}^{m}(C_{ii}) \\ &= \textstyle\sum_{i=1}^{m}\sum_{k=1}^{m}(\mathbf{S_{AB}})_{ik}(\mathbf{S_{BA}})_{kj} \\ &= \textstyle\sum_{i=1}^{m}\sum_{k=1}^{m}(\mathbf{S_{AB}})_{ik}^2 \ \text{ car } \ \mathbf{S_{AB}} = \mathbf{S_{BA}^\top} \end{aligned}$$

Nous avons aussi que

$$\begin{aligned} tr(\mathbf{S_{AA}^2}) &= \textstyle\sum_{i=1}^{p}\sum_{k=1}^{p}(\mathbf{S_{AA}})_{ik}^2 \\ &= \textstyle\sum_{i=1}^{p}\sum_{k=1}^{p}\widehat{Cov(\mathbf{a}_i,\mathbf{a}_k)}^2 \quad \text{car } (\mathbf{S_{AA}})_{ik} = \widehat{Cov(\mathbf{a}_i,\mathbf{a}_k)}. \end{aligned}$$

Dans le cas où $\mathbf{A}$ et $\mathbf{B}$ sont des vecteurs, nous avons p=q=1 et p+q=2, donc

$$\begin{aligned} R_V &= \frac{tr(\mathbf{S_{AB}}S_{\mathbf{BA}})}{\sqrt{tr(\mathbf{S_{AA}^2})tr(\mathbf{S_{BB}^2})}} \\ &= \frac{\sum_{i=1}^{m}\sum_{k=1}^{m}\widehat{Cov(\mathbf{a}_i,\mathbf{b}_k)}^2}{\sqrt{\sum_{i=1}^{p}\sum_{k=1}^{p}\widehat{Cov(\mathbf{a}_i,\mathbf{a}_k)}^2\sum_{i=1}^{q}\sum_{k=1}^{q}\widehat{Cov(\mathbf{b}_i,\mathbf{b}_k)}^2}} \\ &= \frac{\sum_{i=1}^{2}\sum_{k=1}^{2}\widehat{Cov(\mathbf{a}_i,\mathbf{b}_k)}^2}{\sqrt{\sum_{i=1}^{1}\sum_{k=1}^{1}\widehat{Cov(\mathbf{a}_i,\mathbf{a}_k)}^2\sum_{i=1}^{1}\sum_{k=1}^{1}\widehat{Cov(\mathbf{B}_i,\mathbf{B}_k)}^2}} \\ &= \frac{\widehat{Cov(\mathbf{A},\mathbf{B})}^2}{\sqrt{\widehat{Cov(\mathbf{A},\mathbf{A})}^2\widehat{Cov(\mathbf{B},\mathbf{B})}^2)}} \\ &= \frac{\widehat{Cov(\mathbf{A},\mathbf{B})}^2}{\sqrt{Var(\mathbf{A})^2Var(\mathbf{B})^2)}} \\ &= \frac{\widehat{Cov(\mathbf{A},\mathbf{B})}^2}{|Var(\mathbf{A})Var(\mathbf{B})|} \\ &= \left(\frac{\widehat{Cov(\mathbf{A},\mathbf{B})}}{\sqrt{Var(\mathbf{A})Var(\mathbf{B})}}\right)^2. \end{aligned}$$

La différence fondamentale entre l'approche MPLS que nous venons de construire et PLS dans Xu et al. (2012) est le fait que MPLS permet de prédire $\mathbf{Y}$ à partir de $\mathbf{X}$, même dans le cas multidimentionnel ; et cette approche donne ainsi une généralisation de l'approche PLS dans Xu et al. (2012). L'approche PLS dans Xu et al. (2012) teste une variable à la fois (dans ce cas $\mathbf{Y}$ est un vecteur), alors que MPLS teste toutes les variables ensemble (dans ce cas $\mathbf{Y}$ est une matrice). Nous pensons qu'il est important de pouvoir tester toutes les variables ensemble pour rendre un test plus puissant. Nous avons codé cette approche par une fonction que nous avons aussi appelée MPLS qui se retrouve en annexe A.5.

3.2.2 Test d'hypothèses pour l'approche MPLS.

Nous avons vu que l'approche MPLS décompose $\mathbf{X}$ et $\mathbf{Y}$ en deux ensembles de combinaisons linéaires des colonnes de $\mathbf{X}$ et $\mathbf{Y}$ respectivement, $\mathbf{T}$ et $\mathbf{U}$, telles que la covariance entre les colonnes respectives soit maximale. C'est-à-dire

$$\rho_i = \widehat{cov(\mathbf{t}_i, \mathbf{u}_i)} = cov(\widehat{\mathbf{x}^\top \mathbf{w}_i}, \mathbf{y}^\top \mathbf{c}_i),$$

maximale. L'hypothèse nulle, H_0 est définie par :

$$H_0 : \mathbf{X} \text{ et } \mathbf{Y} \text{ sont non associés.} \tag{3.4}$$

La statistique de test que nous utilisons est celle définie par l'équation (3.3). Il a été montré dans (Robert et Escoufier, 1976) que cette statistique est une mesure d'association entre deux matrices, et elle est même assimilable à la corrélation qui existerait entre les colonnes des deux matrices. Elle nous permettrait donc de tester l'hypothèse nulle. Ne connaissant pas la distribution théorique sous H_0 de la statistique de test, R_V, ce qui est aussi le cas pour l'approche PLS de (Xu,*et al.*, 2012), nous utilisons, comme dans leur cas, les tests de permutations pour approximer la distribution de la statistique du test sous l'hypothèse nulle, afin de

pouvoir calculer les valeurs-p associées.

3.2.3 Tests basés sur des permutations.

Les tests basés sur les permutations (c'est-à-dire le réechantillonnage) sont souvent utilisés, dans les tests d'association, pour approximer la distribution de la statistique du test sous l'hypothèse nulle, lorsque cette dernière est difficile à déterminer. En effet, soit $(x_1; x_2; \cdots, x_n)$ un échantillon de taille n identiquement distribué selon une loi de fonction de répartition F_θ. Supposons que nous souhaitons effectuer le test suivant sur le paramètre θ :

$$H_0 : \theta = 0, versus\ H_1 : \theta > 0,$$

un test unilatéral à droite. Soit S_θ la statistique de test. S_θ^{obs} désigne la statistique observée, calculée à partir de l'échantillon observé $(x_1, x_2, \cdots, x_n)$ et on rejette H_0 pour des grandes valeurs de S_θ. Autrement dit, la valeur-p pour ce test est définie comme la probabilité que S_θ soit supérieure à S_θ^{obs}, sachant que H_0 est vraie :

$$\text{valeur-p} = P(S_\theta \geq S_\theta^{obs} | H_0 \text{ vraie}).$$

Lorsque la distribution de la statistique S_θ est inconnue, le calcul de la valeur-p est impossible. Ainsi, les techniques de permutations peuvent s'avérer très utiles pour approximer la distribution de S_θ sous H_0 et calculer ainsi la valeur-p pour un échantillon donné.

La valeur-p empirique est la probabilité d'obtenir une statistique plus élevée après un nombre de permutations que celle obtenue sans permutations. Si m est égal au nombre de résultats supérieurs à la statistique observée ,

$$\widehat{P(S_\theta \geq S_\theta^{obs})} = \text{valeur-p empirique} = \frac{m}{\text{le nombre de permutations}}.$$

Dans l'approche MPLS, nous utilisons la valeur-p empirique ; donc elle dépend du nombre de permutations. On rejette H_0 lorsque la valeur-p empirique est inférieure

ou égale à un seuil fixé d'avance, par exemple 5%. Il serait souhaitable d'obtenir une valeur-p empirique avec toutes les permutations possibles, ce qui n'est pas toujours faisable car le nombre de permutations est bien trop grand et rendrait les calculs très longs. On se pose ainsi la question : comment obtenir des résultats valides avec un nombre pas très grand de permutations ?

3.3 L'approche des moindres carrés partiels multidimentionnelle MPLSGPD.

Avec l'approche MPLS, nous utilisons les permutations pour tester l'hypothèse nulle (voir 3.12). Une question qui se pose est comment améliorer la puissance du test sans faire trop de permutations ? Pour répondre à cette question, nous nous sommes inspirés de l'article de Knijnenburg et al. (2009). Nous décrivons dans un premier temps la procédure utilisée par Knijnenburg et al. (2009) et enfin nous l'adoptons à notre approche.

Dans cet article, les auteurs proposent une approche pour approximer la valeur-p empirique avec moins de permutations. si N représente le nombre de permutations nécessaire pour cette approximation, alors on a $N << N_{all}$ où N_{all} représente le nombre de permutations totales. Les auteurs utilisent la théorie des valeurs extrêmes pour estimer les valeurs-p issues de peu de permutations. ils modélisent alors la queue de la distribution des valeurs-p permutées par la distribution Paréto généralisée (GPD). La distribution GPD a pour fonction de répartition

$$F_{a,k}(z) = \begin{cases} 1 - (1 - kz/a)^{1/k}, & k \neq 0 \\ 1 - e^{-z/a}, & k = 0 \end{cases} \tag{3.5}$$

de fonction de densité

$$f(z) = \begin{cases} a^{-1}(1 - kz/a)^{1/k-1}, & k \neq 0 \\ a^{-1}e^{-z/a}, & k = 0 \end{cases}$$

avec $0 \leq z$ pour $k \leq 0$ et $0 \leq z \leq a/k$ pour $k > 0$, a un paramètre d'échelle et k un paramètre de forme. Pour $k = 0$, GPD est la distribution exponentielle ;
pour $k = 1$, GPD est la distribution uniforme sur $[0, a]$ et
pour $k < 0$, GPD est une Pareto avec des valeurs extrêmes.
Ainsi, soit x_0 une statistique de test observée et $X = \{x_1^*, x_2^*, \cdots, x_{N_{all}}^*\}$, l'ensemble de toutes les statistiques obtenues après les permutations possibles. On sait que la valeur-p empirique avec le test des permutations est donnée par

$$\text{valeur-p empirique} = \frac{\sum_{n=1}^{N_{all}} I(x_n^* \geq x_0)}{N_{all}}, \tag{3.6}$$

où $I(.)$ est la fonction indicatrice . On veut approximer l'équation 3.6 en utilisant de manière aléatoire un sous ensemble Y de X tel que $Y = \{y_1^*, y_2^*, \cdots, y_N^*\}$ et $N << N_{all}$. Dans ce cas, la valeur-p empirique est donnée par

$$\text{valeur-p empirique} = \frac{\sum_{n=1}^{N} I(y_n^* \geq x_0)}{N}. \tag{3.7}$$

Avec N le nombre de permutations ($N << N_{all}$). $Y = \{y_1^*, y_2^*, \cdots, y_N^*\}$ représente toutes les statistiques obtenues après les N permutations. On peut les placer en ordre décroissant et obtenir ainsi $Y = \{y_{(1)}^*, y_{(2)}^*, \cdots, y_{(N)}^*\}$ avec

$$y_{(1)}^* \geq y_{(2)}^* \geq \cdots \geq y_{(N)}^*.$$

La distribution GPD est alors ajustée aux valeurs

$$Z = \{z_1^*, z_2^*, \cdots, z_{N_{exc}}^*\}; \text{ où } \quad z_i^* = y_i^* - t$$

et

$$t = \frac{y_{N_{exc}}^* + y_{N_{exc+1}}^*}{2}. \tag{3.8}$$

L'argument z de GPD (pour la fonction de répartition, voir (3.5)) est un excédent, donc un élément de Z, par rapport au seuil t (voir (3.8)). N_{exc} représente le nombre d'excédents où N_{exc} est un nombre aléatoire fixé au début, mais inférieur au nombre N de permutations pour que l'équation (3.8) ait un sens.

Dans la littérature, ce nombre est généralement pris à être 250 ; mais il est ajusté pour que la distribution GPD soit un bon modèle pour les excédents Z.
On utilise le test d'ajustement (en anglais *Goodness-of-fit*) pour voir si les excédents proviennent d'une GPD. Si tel est le cas, les paramètres k et a de GPD (paramètres de la fonction de répartition, voir (3.5)) sont donc estimés en utilisant le critère du maximun de la vraisemblance avec l'ensemble des excédents Z. Cette estimation a bien du sens parce que pour des échantillons très grands, en statistique appliquée, les valeurs de k devraient être plus petites que $1/2$ (Knijnenburg *et al.*, 2009). La valeur-p obtenue par l'approximation de GPD, P_{gpd}, est donc donnée par

$$P_{gpd} = \frac{N_{exc}}{N}(1 - F_{a,k}(x_0 - t)), \tag{3.9}$$

avec $F_{a;k}$ la fonction de répartition de GPD décrite par l'équation (3.5) et t décrit par l'équation (3.8), x_0 est la statistique de test observée (sans les permutations). Les hypothèses du test d'ajustement pour s'assurer que GPD est un bon modèle pour les valeurs extrêmes des statistiques Z, avec un seuil nominal par exemple $\alpha = 0.05$, sont les suivantes :

$$H_0 : \text{GPD est un bon modèle pour les } Z, \tag{3.10}$$

$$H_1 : \text{GPD n'est pas un bon modèle pour les } Z. \tag{3.11}$$

Le choix du nombre des excédents, N_{exc}, suit une procédure que nous décrivons dans l'algorithme 4. Si l'hypothèse H_0 (de (3.10)) est vraie, (c'est-à-dire H_0 n'est pas rejetée avec les données en main), on retient N_{exc} et on calcule P_{gpd} (voir (3.9)) ; sinon on considère un nouveau seuil en imposant à N_{exc} la valeur $N_{exc} - 10$ jusqu'à ce que H_0 soit vraie ou que $N_{exc} = 10$.

Dans les lignes qui suivent, nous décrivons l'approche MPLSGPD. Rappellons que nous avons un échantillon de données représenté par les matrices $\mathbf{X}$ et $\mathbf{Y}$. On

Algorithme 4 : Choix du nombre des excédents N_{exc}.

Poser N_{exc}=250 ;

Tester H_0 contre H_1 (voir (3.10)) ;

"Tant que" H_0 est rejetée alors **Faire**

"Si" $N_{exc} > 10$ **Alors**

$N_{exc} \leftarrow N_{exc} - 10$,

Tester H_0 contre H_1

Sinon

$N_{exc} = 10$

Fin "Si"

Fin "Tant que"

Retourner N_{exc}.

souhaite tester les hypothèses :

$$H_0 : \mathbf{X} \text{ et } \mathbf{Y}\text{sont non associées versus } H_1 : \mathbf{X} \text{ et } \mathbf{Y} \text{ sont associées.} \qquad (3.12)$$

La statistique de test que nous utilisons est celle définie par l'équation (3.3), R_V empirique. Nous utilisons le test de permutations. Au lieu de faire un très grand nombre de permutations pour avoir à calculer la valeur-p, dans l'approche MPLSGPD, nous optons pour des petites permutations. Les permutations sont faites sur les lignes de la matrice $\mathbf{Y}$. La statistique des valeurs non permutées est notée R_{V0} (ce qui correspondrait à x_0 dans la description précédente). Dans cette approche, pour calculer les valeurs-p, nous utilisons un algorithme. Cet algorithme nous donne le choix entre la valeur-p obtenue dans l'approximation décrite dans (Knijnenburg et al., 2009) qui vient d'être décrite précédemment ou une valeur-p, sans approximation de Knijnenburg. Pour chaque permutation de $\mathbf{Y}$ on note la valeur de la statistique R_V après cette permutation. On constitue ainsi un vecteur

des valeurs de R_V de longueur N (N ici est le nombre de permutations). Ces valeurs obtenues constituent un vecteur que l'on ordonne par ordre décroissant,

$$\mathbf{r_v} = (R_{V1} \geq R_{V2} \geq \cdots \geq R_{VN}).$$

On désigne par M le nombre des valeurs de r_v qui sont supérieures à R_{V0} (la statistique des données non permutées). Si ce nombre est plus grand que 10, nous considérons comme valeur-p le nombre suivant :

$$P_{ecdf} = \frac{M}{N}.$$

Si par contre ce nombre est plus petit que 10 ($M < 10$), nous considérons comme valeur-p celle donnée par l'approximation GPD. Dans ce cas, il suffit donc d'identifier les différents paramètres décrits dans l'approximation décrite précédemment. L'argument z de GPD (pour la fonction de densité) est un excédent d'une valeur du vecteur $\mathbf{r_v}$ par rapport au seuil t, avec

$$t = \frac{R_{VN_{exc}} + R_{V(N_{exc}+1)}}{2}, \tag{3.13}$$

où N_{exc} est un nombre aléatoire fixé au début, mais inférieur au nombre N de permutations (nous prenons $N_{exc} = 250$). On constitue ainsi un vecteur, $\mathbf{Z}$, des excédents de longueur N_{exc}. En effet,

$$\mathbf{Z} = (R_{V1} - t \geq R_{V2} - t \geq \cdots \geq R_{VN_{exc}} - t).$$

Si les excédents proviennent d'une GPD, la valeur-p obtenue par l'approximation de GPD est donc donnée par

$$P_{gpd} = \frac{N_{exc}}{N}(1 - F_{a,k}(R_{V0} - t))$$

avec $F_{a,k}$ la fonction de répartition de GPD décrite par l'équation (3.5) et t décrit par l'équation (3.13). Nous résumons donc l'approche MPLSGPD par l'algorithme 5 suivant :

Algorithme 5 : Algorithme des valeurs-p empiriques

1. R_{V0} valeur observée sans les permutations.
2. $\mathbf{r_v} = (R_{V1} \geq R_{V2} \geq \cdots \geq R_{VN})$, vecteur des valeurs observées après N permutations.
3. M = nombre des valeurs de $\mathbf{r_V}$ supérieures à R_{V0}

"Si" $M \geq 10$ **Alors**

valeur-p = $P_{ecdf} = \frac{M}{N}$

Sinon

valeur-p = $P_{gpd} = \frac{N_{exc}}{N}(1 - F_{a,k}(x_0 - t))$ avec $F_{a,k}$ définie en (3.5) et N_{exc} définie à la page 40

Fin "Si"

Retourner valeur-p.

Avec cet algorithme de détermination des valeurs-p, on améliore la puissance du test par rapport à l'approche MPLS, comme on va le voir dans les simulations au chapitre suivant. Nous avons écrit deux fonctions R, données en annexe, qui permettent de simuler l'approche MPLSGPD. Après description des deux approches, il nous semble important de faire une étude comparative entre celles-ci et les fonctions que nous avons citées au chapitre 1.

CHAPITRE IV

SIMULATIONS ET ANALYSE DE DONNÉES RÉELLES

Nous faisons une étude de simulation ainsi qu'une analyse de données réelles afin d'illustrer la nouvelle méthodologie proposée dans ce mémoire (MPLS et MPLSGPD). Les données réelles sont issues des personnes atteintes de la maladie d'Alzheimer.

4.1 Études de simulations

4.1.1 Simulation des données

Pour l'étude de simulation, nous construisons des données que nous allons analyser. On veut générer deux ensembles de données $\mathbf{X}$ et $\mathbf{Y}$. On distingue le cas où $\mathbf{Y}$ est un vecteur et le cas où $\mathbf{Y}$ est une matrice multidimensionnelle. Nous utilisons la construction proposée dans le rapport de stage Erica Cunningham (2014). On veut ainsi construire deux ensembles de matrices $\mathbf{X}$ et $\mathbf{Y}$ qui respectent les équations (3.1) et (3.2). On construit donc $\mathbf{X}$ et $\mathbf{Y}$ telles que

$$\mathbf{X} = \mathbf{X}_{n\times p} = \mathbf{T}\mathbf{P}^\top + \mathbf{E}_\mathbf{X}, \tag{4.1}$$

$$\mathbf{Y} = \mathbf{Y}_{n\times q} = \mathbf{U}\mathbf{Q}^\top + \mathbf{E}_\mathbf{Y}, \tag{4.2}$$

avec

les lignes de $\mathbf{E}_\mathbf{X}$ et $\mathbf{E}_\mathbf{Y}$ de lois gaussiennes multidimensionnelles de moyenne nulle

et de variance. $\sigma^2_{\mathbf{X}}\mathbf{I}_p$ et $\sigma^2_{\mathbf{Y}}\mathbf{I}_q$ respectivement, représentant les matrices des erreurs. Formellement,

$$\mathbf{E}^{\top}_{\mathbf{X}_i} = \mathcal{N}_p(0_p, \sigma^2_{\mathbf{X}}\mathbf{I}_p), \quad \mathbf{E}^{\top}_{\mathbf{Y}_i} = \mathcal{N}_q(0_q, \sigma^2_{\mathbf{Y}}\mathbf{I}_q).$$

Les matrices $\mathbf{P} = \mathbf{P}_{p\times k}$ et $\mathbf{Q} = \mathbf{Q}_{q\times k}$, les X loadings et Y loadings respectivement, sont les matrices dont les éléments sont des 0.5, qui représentent en quelque sorte le poids qu'une variable de $\mathbf{X}$ (ou de $\mathbf{Y}$) soit informative. Notons qu'on aurait pu prendre différents scénarios, en considérant par exemple le cas où il y aurait des variables non informatives. Dans ce cas, la position correspondante aurait la valeur 0. Le nombre de colonnes de chacune de ces matrices, k, indique le nombre de composantes qui captent la dépendance entre $\mathbf{X}$ et $\mathbf{Y}$. Pour notre cas, nous choisissons deux composantes, $k = 2$.

Une fois les matrices $\mathbf{P}$ et $\mathbf{Q}$ connues, les matrices des vecteurs latents de $\mathbf{X}$ et $\mathbf{Y}$, $\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2)$ et $\mathbf{U} = (\mathbf{u}_1; \mathbf{u}_2)$ respectivement, sont générées. Les vecteurs $(\mathbf{t}_1, \mathbf{u}_1, \mathbf{t}_2, \mathbf{u}_2)$ sont générés comme échantillon de taille n d'une distribution gaussienne multidimensionelle $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ de moyenne nulle et de matrice de corrélations $\mathbf{\Sigma_{ij}} = corr(\mathbf{t}_i, \mathbf{u}_j)$ pour $i, j \in \{1, 2\}$. Nous représentons $\mathbf{\Sigma}$ pour deux composantes comme suit :

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & cov(\mathbf{t}_1, \mathbf{u}_1) & cov(\mathbf{t}_1, \mathbf{t}_2) & cov(\mathbf{t}_1, \mathbf{u}_2) \\ cov(\mathbf{t}_1, \mathbf{u}_1) & 1 & cov(\mathbf{u}_1, \mathbf{t}_2) & cov(\mathbf{u}_1, \mathbf{u}_2) \\ cov(\mathbf{t}_1, \mathbf{t}_2) & cov(\mathbf{u}_1, \mathbf{t}_2) & 1 & cov(\mathbf{t}_2, \mathbf{u}_2) \\ cov(\mathbf{t}_1, \mathbf{u}_2) & cov(\mathbf{u}_1, \mathbf{u}_2) & cov(\mathbf{t}_2, \mathbf{u}_2) & 1 \end{pmatrix}.$$

Pour cette étude de simulations, nous considérons :

$$corr(\mathbf{t_i}, \mathbf{u_i}) = \rho_i, \ i = 1, 2.$$

$$corr(\mathbf{t_i}, \mathbf{u_j}) = 0, \ i \neq j, \ i, j = 1, 2.$$

La matrice $\mathbf{\Sigma}$ devient alors

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & \rho_1 & 0 & 0 \\ \rho_1 & 1 & 0 & 0 \\ 0 & 0 & 1 & \rho_2 \\ 0 & 0 & \rho_2 & 1 \end{pmatrix}.$$

Nous posons $\rho_1 = \rho_2 \in \{0, 0.2, 0.4, 0.6\}$ pour les simulations présentées ici. Cette corrélation est utilisée pour créer une association entre les colonnes de $\mathbf{X}$ et celles de $\mathbf{Y}$. Un exemple de scénario se résume comme dans la figure 4.1.

Nous avons utilisé le logiciel R pour construire deux fonctions qui nous permettent de simuler les données, en précisant les paramètres convenables. Ces fonctions sont données en annexe. La fonction que nous nommons "Multivariée" permet d'avoir des matrices $\mathbf{X}$ et $\mathbf{Y}$. La fonction que nous nommons "Univariée" permet d'avoir une matrices $\mathbf{X}$ et un vecteur $\mathbf{Y}$. Nous simulons $\mathbf{X}$ de format 300×39 et $\mathbf{Y}$ de format 300×16. Nous supposons que les colonnes (variables) de $\mathbf{X}$ sont divisées en deux groupes (deux composantes), de même que celles de $\mathbf{Y}$. Les composantes de $\mathbf{X}$ et celles de $\mathbf{Y}$ sont liées par les corrélations ρ_1 et ρ_2. On peut voir la figure 4.1 pour illustration. Le nombre de réplications est de 1000. Sauf mention du contraire, tous les scénarios de simulations que nous exposons dans la suite auront ce format. Pour les sénarios 1, 2 et 3, nous considérons 5000 permutations pour l'approche MPLS et 1000 permutations pour l'approche MPLSGPD.

4.1.2 Sénario 1 : Risque de première espèce.

Dans ce cas, nous considérons $\rho_1 = \rho_2 = 0$. Comme le montre la figure 4.2, les valeurs empiriques sont proches de celles théoriques, et ceux pour les deux approches MPLS et MPLSGPD. Les tests de permutation contrôlent bien l'erreur de type I (toutes les courbes sont proches de la droite $y = x$), et l'approche MPLSGPD contrôle mieux.

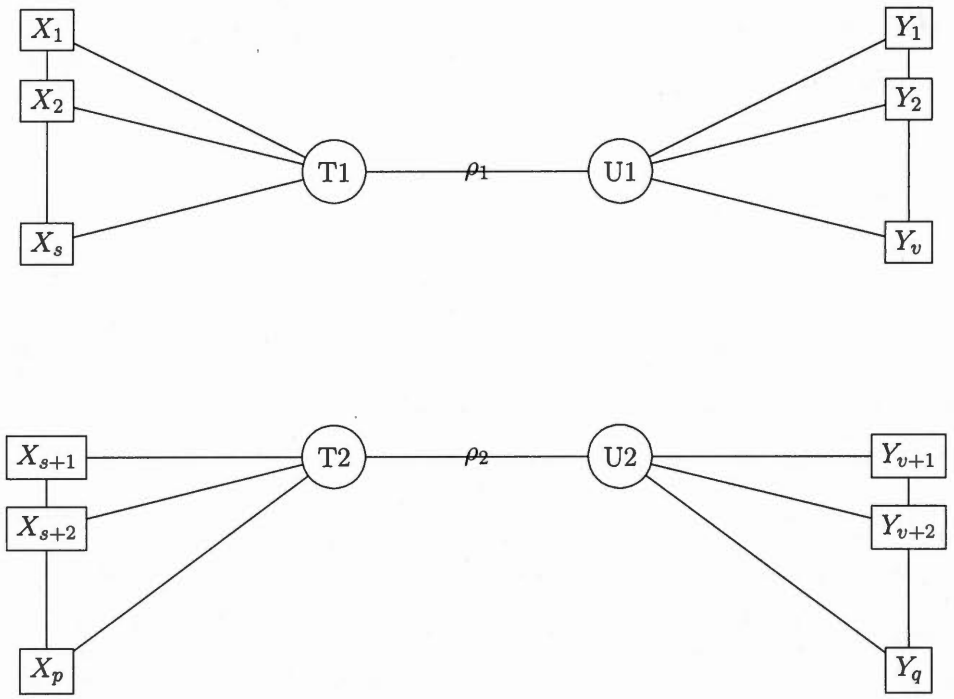

Figure 4.1 Exemple de scénario de simulation. On a deux composantes pour $\mathbf{X}$ ($\mathbf{t}_1$ qui est une combinaison linéaire des colonnes $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_s$ et $\mathbf{t}_2$ qui est une combinaison linéaire des autres colonnes de $\mathbf{X}$. De même, on a deux composantes pour $\mathbf{Y}$ ($\mathbf{u}_1$ qui est une combinaison linéaire des colonnes de $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_v$ et $\mathbf{u}_2$ qui est une combinaison linéaire des autres colonnes de $\mathbf{Y}$; ρ_1 désigne la corrélation qu'il y a entre $\mathbf{t}_1$ et $\mathbf{u}_1$; ρ_2 désigne la corrélation qu'il y a entre $\mathbf{t}_2$ et $\mathbf{u}_2$.

4.1.3 Sénario 2 : Étude de la puissance.

Dans ce cas, nous testons la puissance des approches MPLS et MPLSGPD. Comme le montre la figure 4.3 , la puissance obtenue par l'approche MPLSGPD est légèrement au-dessus de celle reçue par l'approche MPLS, avec $\rho_1 = \rho_2 = 0.2$.

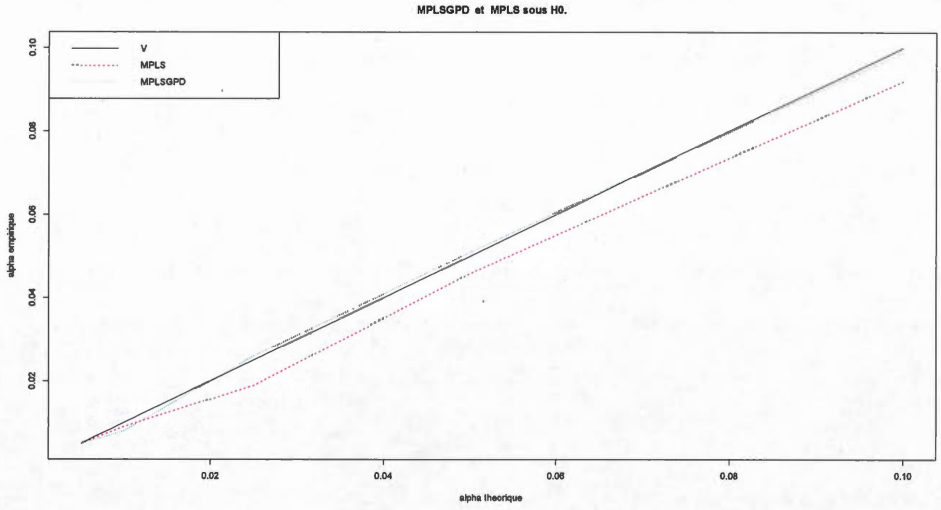

Figure 4.2 Comparaison sous H_0 des approches MPLS et MPLSGPD. L'axe des abscisses représente le seuil de signification $\alpha \in \{0.005, 0.01, 0.025, 0.05, 0.1\}$, l'axe des ordonnées présente les probabilités empiriques sous H_0 dites probabilités d'erreur du type I. La courbe représente les valeurs-p sous H_0 pour un seuil α. $\rho_1 = \rho_2 = 0$

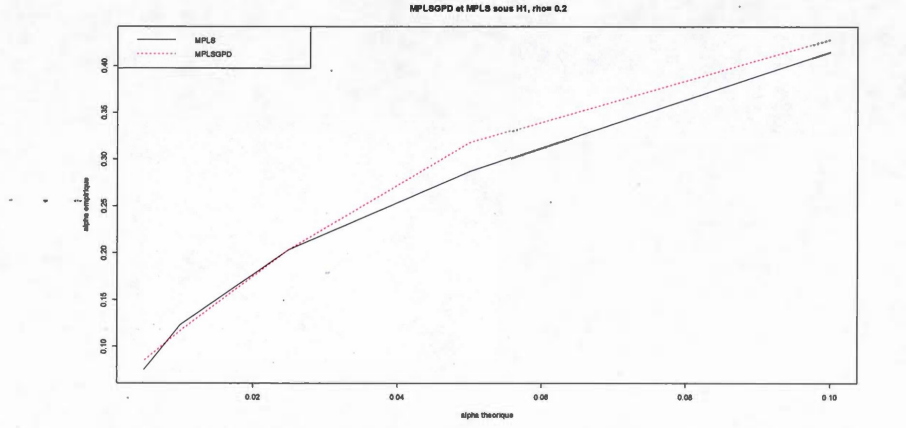

Figure 4.3 Test de puissance par les approches MPLS et MPLSGPD. L'axe des abscisses représente le seuil de signification $\alpha \in \{0.005, 0.01, 0.025, 0.05, 0.1\}$, l'axe des ordonnées présente la puissance sous H_1. La courbe représente les valeurs-p sous H_1 pour un seuil α. Pour ce cas, $\rho_1 = \rho_2 = 0.2$.

Nous avons aussi voulu voir ce qu'on obtiendrait si dans les mêmes conditions, on faisait varier la corrélation entre les composantes. Nous avons simulé les cas $\rho_1 = \rho_2 = 0.4$ (Voir figure 4.4) et $\rho_1 = \rho_2 = 0.6$ (voir figure 4.5). La puissance est presque 1 dans ces deux cas. Dans tous les cas, on conclut qu'il y a un gain de puissance avec l'approche MPLSGPD, et même que ce gain augmente avec l'augmentation de la corrélation (voir figure 4.6). On conclut donc que plus les variables sont corrélées, plus la puissance est grande. La figure 4.6 illustre les cas où les corrélations ρ_i prennent les valeurs 0.2, 0.4 et 0.6.

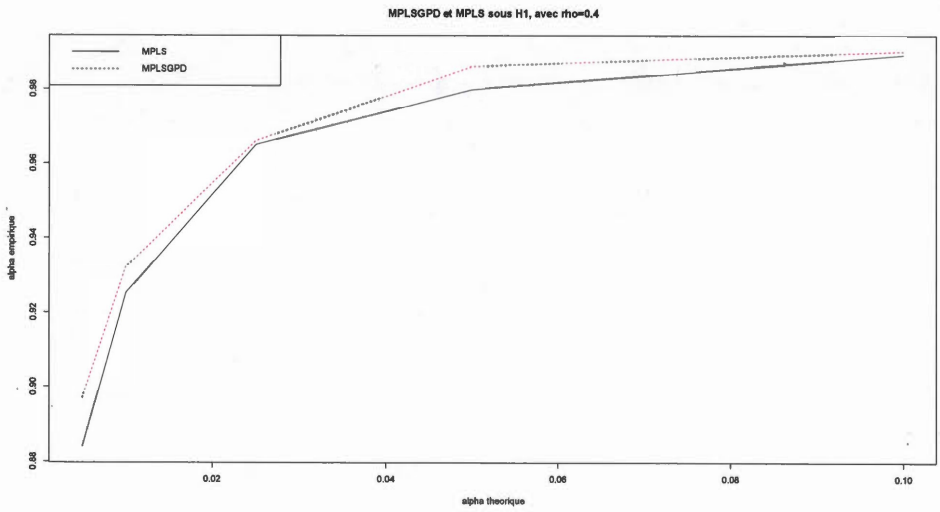

Figure 4.4 Test de puissance par les approches MPLS et MPLSGPD. Dans ce cas, $\rho_1 = \rho_2 = 0.4$.

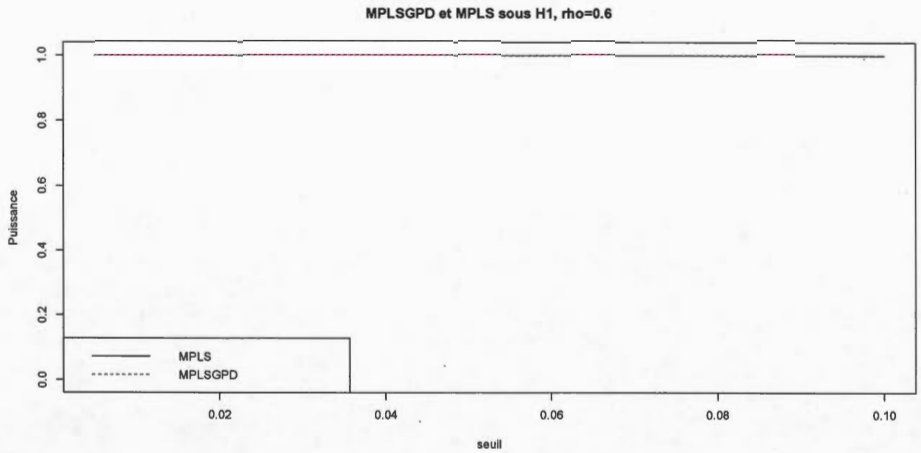

Figure 4.5 Test de puissance par les approches MPLS et MPLSGPD. Dans ce cas, $\rho_1 = \rho_2 = 0.6$.

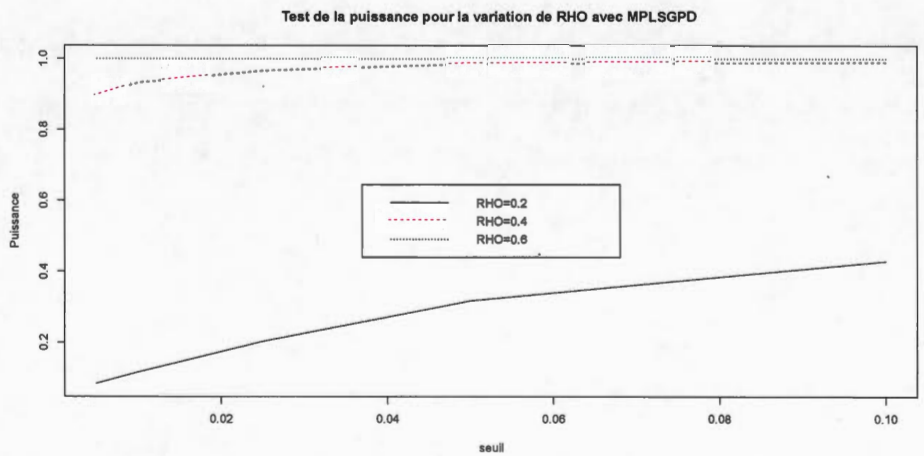

Figure 4.6 Variation de la puissance obtenue par l'approche MPLSGPD en fonction des corrélations $\rho_i \in \{0.2, 0.4, 0.6\}$. Le nombre de permutations est 1000.

4.1.4 Sénario 3 : L'approche MPLSGPD, une généralisation de l'approche PLS de Xu et al. (2012).

Nous avons montré que la statistique, R_V, utilisée par notre approche est le carré de celle utilisée par l'approche PLS de Xu et al. (2012). Dans le cas où $\mathbf{Y}$ est un vecteur, nous montrons que notre approche MPLSGPD et l'approche PLS de Xu et al. permettent d'avoir presque les mêmes résultats, comme le montre la tableau 4.1.4.

α	PLS	MPLSGPD
0.1	0.190	0.211
0.05	0.116	0.123
0.025	0.060	0.066
0.01	0.029	0.029
0.005	0.018	0.017

Tableau 4.1 Pour un seuil α, le tableau donne la moyenne des valeurs-p obtenues par les approches PLS et MPLSGPD lorsque $\mathbf{Y}$ est un vecteur. Pour ce cas, $\rho_1 = \rho_2 = 0.2$ et le nombre de permutations est 1000.

La figure 4.7 donne les résultats obtenus pour le test de la puissance par les deux approches MPLSGPD et PLS, avec $\rho_1 = \rho_2 = 0.2$. Les deux courbes obtenues ne sont pas trop différentes. Les différences sont dues aux permutations, mais surtout à l'approximation GPD de la queue de la distribution de la statistique du test sous H_0.

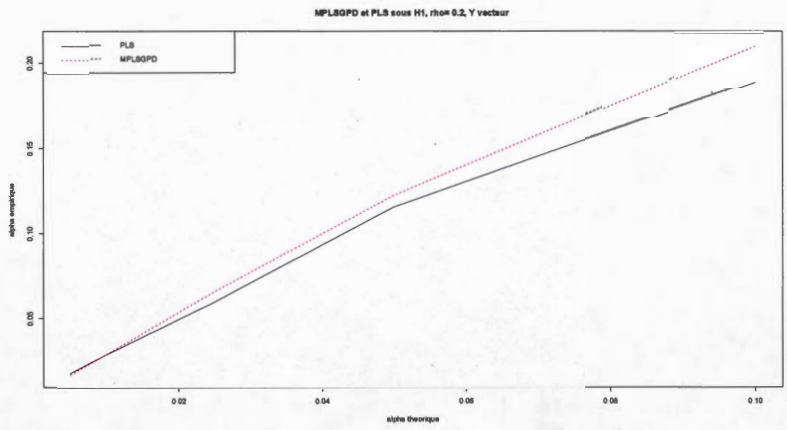

Figure 4.7 Puissance obtenue par les approches MPLSGPD et PLS avec $\mathbf{Y}$ un vecteur. Pour ce cas, $\rho_1 = \rho_2 = 0.2$ et le nombre de permutations est 1000.

Nous avons aussi regardé la variation de la corrélation dans le cas où $\mathbf{Y}$ est un vecteur, pour $\rho_1 = \rho_2 = 0.4$ et $\rho_1 = \rho_2 = 0.6$. Pour ces deux cas, les courbes obtenues ne sont presque identiques. On a ainsi obtenu les graphiques :

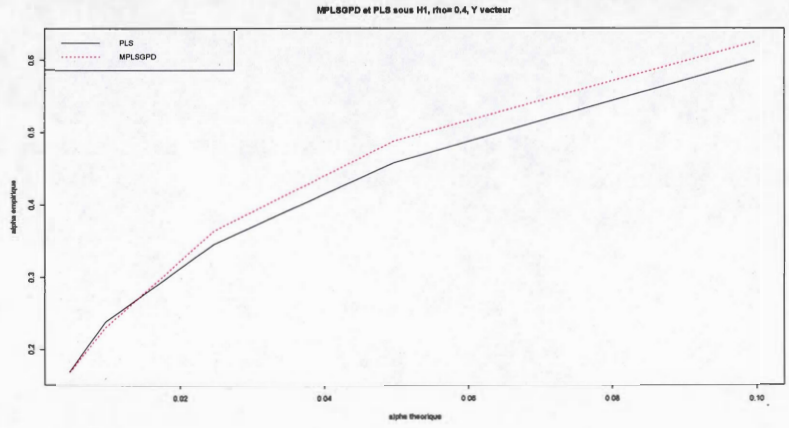

Figure 4.8 Puissance obtenue par les approches MPLSGPD et PLS avec $\mathbf{Y}$ un vecteur. Pour ce cas, $\rho_1 = \rho_2 = 0.4$ et le nombre de permutations est 1000.

et

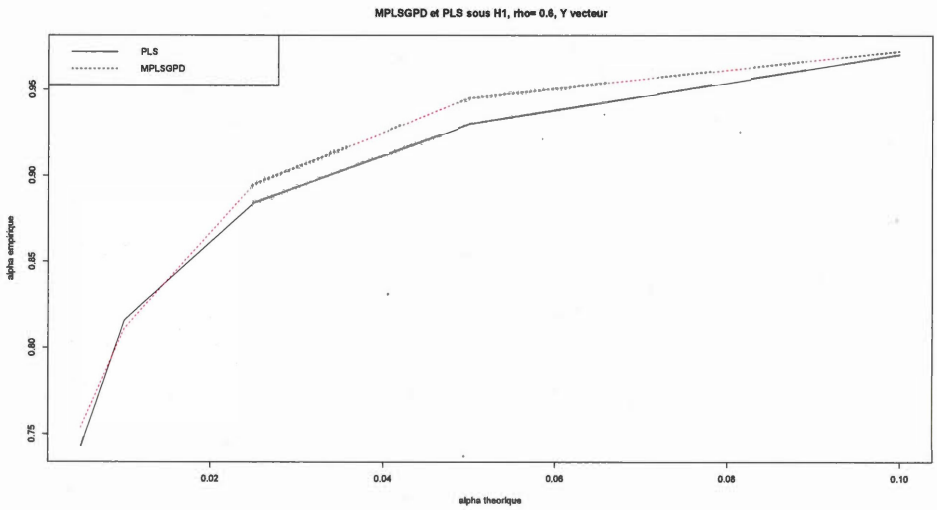

Figure 4.9 Puissance obtenue par les approches MPLSGPD et PLS avec $\mathbf{Y}$ un vecteur. Pour ce cas, $\rho_1 = \rho_2 = 0.6$ et le nombre de permutations est 1000.

4.1.5 Sénario 4 : Comparaison entre le cas unidimensionnel standard et le cas multidimensionnel par l'approche MPLSGPD.

Dans le cas multidimensionnel ($\mathbf{Y}$ est une matrice), nous comparons les résultats obtenus en considérant toutes les colonnes de $\mathbf{Y}$ l'une à la fois et toute la matrice $\mathbf{Y}$ par l'approche MPLSGPD. La puissance obtenue dans ce cas est bien démarquée, comme le montre la figure 4.10. Il est donc préférable de tester toutes les variables globalement et non de les tester une à la fois. On remarque aussi que d'une colonne à une autre de $\mathbf{Y}$, la puissance n'est pas très différente.

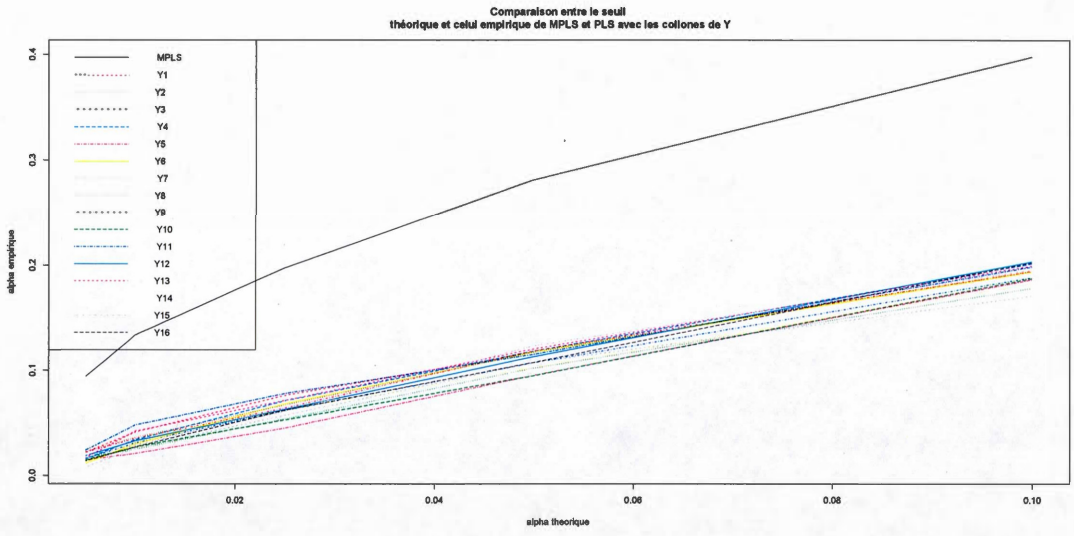

Figure 4.10 Comparaison sous H_1 entre la puissance obtenue par l'approche MPLSGPD avec $\mathbf{Y}$ matrice et chacune de ses colonnes respectivement. Pour ce cas, $\rho_1 = \rho_2 = 0.2$ et le nombre de permutations est 1000.

4.1.6 Sénario 5 : Comparaison avec les méthodes standard.

Dans cette section, nous comparons les résultats obtenus par notre approche, l'approche MPLSGPD, et ceux obtenus par des méthodes standard de réduction de données qui ont été présentées au chapitre 1(LASSO, *Ridge Regression*(RR), PLS, PCR). Rappelons que ces méthodes sont univariées. Autrement dit, elles sont utilisées seulement dans les cas où on a une seule variable réponse. Les données $\mathbf{X}$ et $\mathbf{Y}$ étant obtenues comme précédemment (Sénario 2 par exemple), nous utilisons notre approche pour l'ensemble des colonnes de $\mathbf{Y}$, alors que les autres méthodes sont appliquées sur chacune des colonnes de $\mathbf{Y}$. Pour ne pas alourdir la présentation, on considère seulement la première colonne de $\mathbf{Y}$ pour les simulations dans les méthodes standard. Ce choix d'une seule colonne est aussi motivé par la figure 4.10.

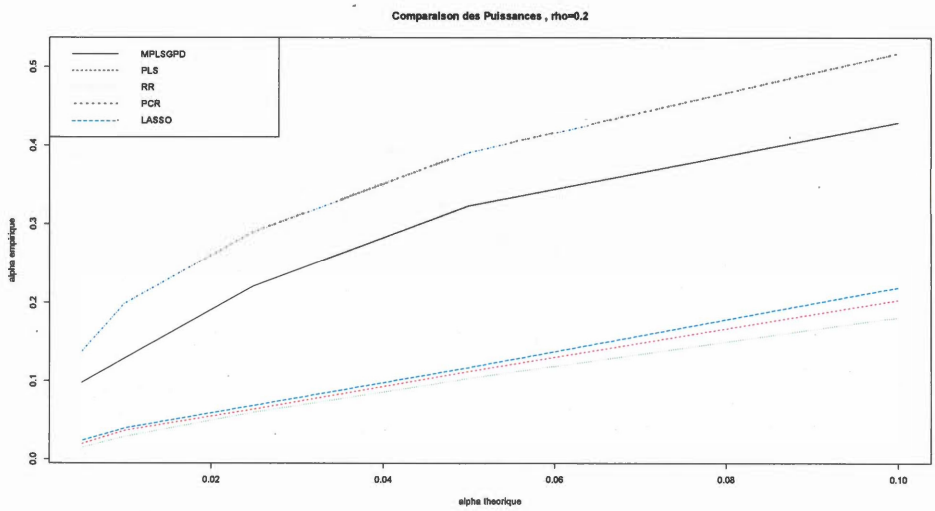

Figure 4.11 Comparaison entre MPLSGPD et les autres méthodes (PLS, *Ridge Regression* (RR), PCR et LASSO) avec $\rho_i = 0.2$. Le nombre de permutations est de 1000.

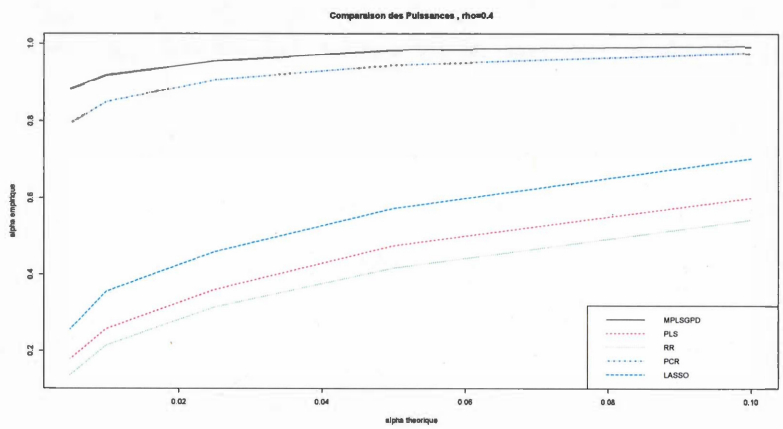

Figure 4.12 Comparaison entre MPLSGPD et les autres méthodes (PLS, *Ridge Regression* (RR), PCR et LASSO) avec $\rho_i = 0.4$. Le nombre de permutations est de 1000.

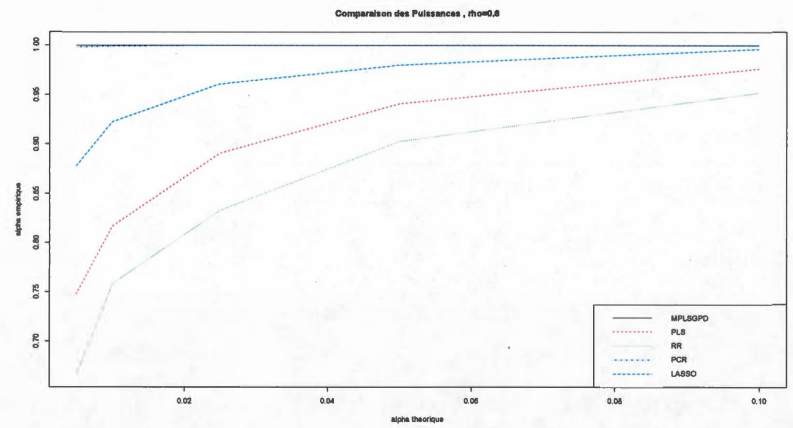

Figure 4.13 Comparaison entre MPLSGPD et les autres méthodes (PLS, *Ridge Regression* (RR), PCR et LASSO) avec $\rho_i = 0.6$. Le nombre de permutations est de 1000.

On constate alors que la puissance obtenue par notre approche est généralement plus élevée que celle des autres cas, comme le montrent les figures 4.11, 4.12 et 4.13 avec les différentes valeur de la corrélation. On remarque que la méthode PCR se démarque des autres méthodes, et que la puissance obtenue est bien proche de la notre.

Pour les données simulées jusqu'à présent, les variables indépendantes de la matrice $\mathbf{X}$ sont continues. Afin de nous mettre dans un contexte génétique où les prédicteurs (les colonnes de $\mathbf{X}$) sont des variables binomiales, nous allons discrétiser la matrice $\mathbf{X}$ afin d'avoir des données dont les éléments sont des 0, 1 ou 2. Cette matrice devient alors assimilable à la matrice des SNPs, et $\mathbf{Y}$ la matrice des phénotypes. Nous programmons une fonction pour discretiser chaque colonne de la matrice $\mathbf{X}$.

Pour discrétiser, on se donne un vecteur et une valeur de la fréquence allélique et au retour on obtient un vecteur dont les valeurs sont des 0, 1, 2. À partir de

la fréquence allélique, on construit les fréquences du génotype (AA, Aa ou aA, aa, où a est l'allèle mineur) et les valeurs du vecteur deviennent des 0, 1, 2 en les comparant avec la fréquence du génotype aa suivant cet algorithme :

Algorithme 6 Algorithme de discretisation d'un vecteur

v vecteur (donné) ;

P_a = fréquence allélique de l'allèle mineur (donné) ;

Assurer

Poser $P_A = 1 - P_a$

$P_{AA} = P_A \cdot P_A$ (fréquence du génotype AA)

$P_{aa} = P_a \cdot P_a$ (fréquence du génotype aa)

$P_{Aa} = 2 \cdot P_a \cdot P_A$ (fréquence du génotype Aa (aA)) ;

Poser **DisV** $= (0, 0, \cdots, 0)$, vecteur nul de même longueur que **v** ;

"Pour" $i \in (1 : longueur(\mathbf{v}))$ **Faire**

"Si" $\mathbf{v}[i] \leq qnorm(P_{aa})$ [1] **Alors**

DisV$[i] \leftarrow 0$

Sinon "Si" $qnorm(Paa) < \mathbf{v}[i] \leq qnorm(P_{AA})$ **Alors**

DisV$[i] \leftarrow 1$,

Sinon

DisV$[i] \leftarrow 2$.

Fin "Si"

end "Pour"

Retourner DisV (vecteur discretisé).

1. $qnorm(p)$ désigne la fonction du logiciel R qui permet, à partir d'une probabilité p, d'obtenir le quantile qui correspond à p, lorsque la fonction de répartition est normale. Ou encore, cette fonction détermine le quantile d'une loi normale centrée réduite.

Nous avons programmé une fonction que nous utilisons pour discrétiser la matrice $\mathbf{X}$ en utilisant le logiciel R. Cette fonction est donnée en annexe A.7.

Nous simulons donc les données $\mathbf{X}$ et $\mathbf{Y}$ comme dans le scénario 4.1.3. La matrice $\mathbf{X}$ est alors discretisée pour la transformer en une matrice dont les valeurs sont des 0, 1, 2. Notons que la discrétisation de la matrice $\mathbf{X}$ dépend de la fréquence de l'allèle mineur. Nous distinguons deux cas, selon qu'on a des variants rares et des variants rares et communs. Dans le cas des variants rares, la fréquence de l'allèle mineur provient d'un échantillon aléatoire obtenu d'une loi uniforme dans les intervalles $[10^{-4}, 0.1]$. Dans le cas des variants rares et communs, la fréquence des de l'allèle mineur provient d'un é aléatoire obtenu d'une loi uniforme dans l'intervalle $[10^{-4}, 0.3]$. On détermine alors les puissances obtenues par notre approche, MPLSGPD et les autres méthodes standard précédemment décrites (avec X discretisée). Les figures 4.14 et 4.15 montrent les résultats obtenus. On constate que pour une corrélation faible, notre approche donne aussi une puissance faible autant que les autres ; alors que, dès que la corrélation augmente, elle prend largement le dessus. De manière générale, la puissance obtenue par notre approche est plus élevée que celle obtenue par les autres, même après discrétisation.

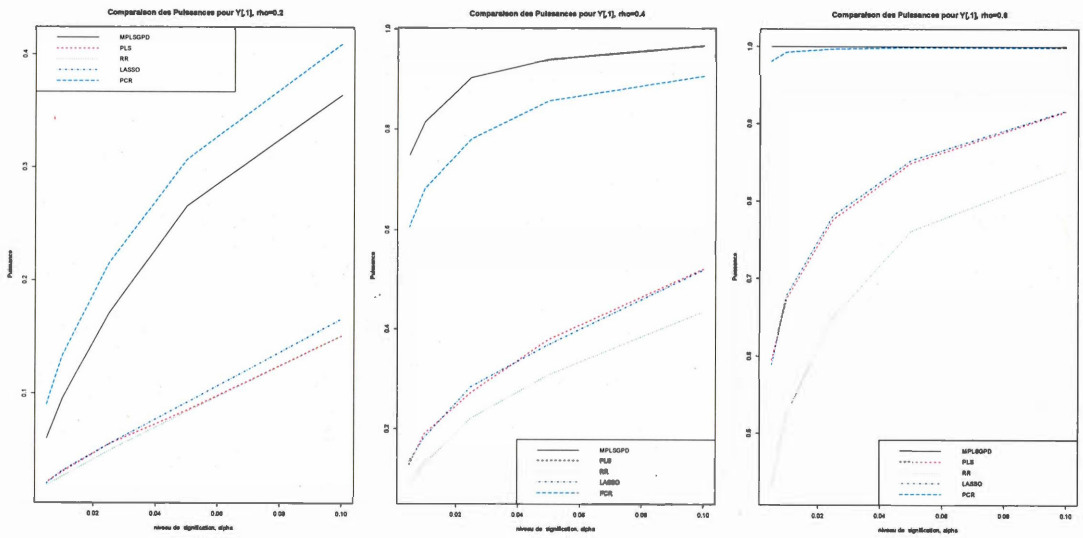

Figure 4.14 Les puissances obtenues avec $\mathbf{X}$ discretisée après l'obtention des données avec la fonction "multivariée", et ceux pour chaque valeur de la corrélation avec l'allèle mineur pris dans $[10^{-4}, 0.1]$ (variants rares). Les graphiques de la gauche vers la droite représentent respectivement les cas de corrélation 0.2, 0.4 et 0.6. Le nombre de permutations est de 1000.

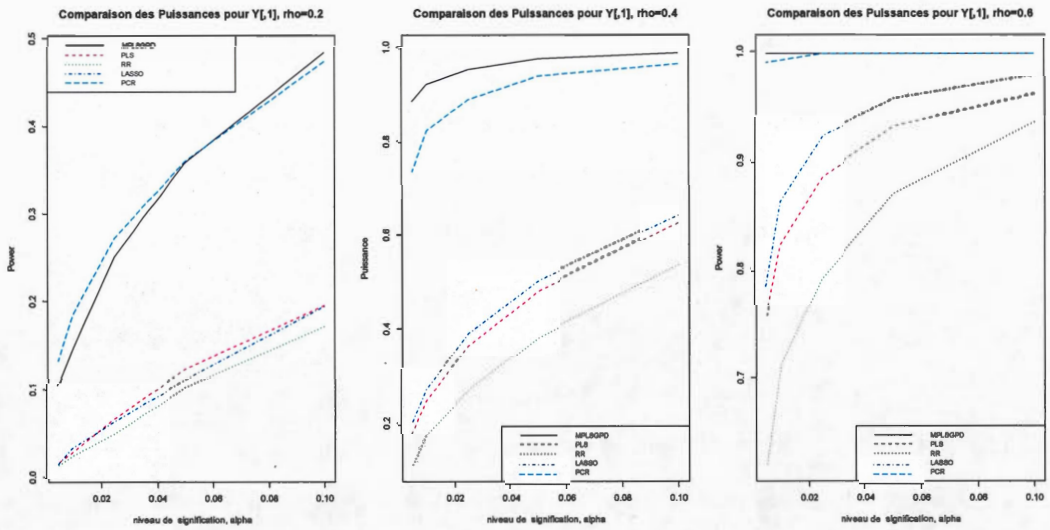

Figure 4.15 Les puissances obtenues avec $\mathbf{X}$ discretisée après l'obtention des données avec la fonction "multivariée", et ceux pour chaque valeur de la corrélation avec l'allèle mineur pris dans $[10^{-4}, 0.3]$ (variants communs). Les figures de la gauche vers la droite représentent les cas de corrélation 0.2, 0.4 et 0.6. Le nombre de permutations est de 1000.

4.2 Analyse des données réelles

Dans cette session, nous analysons un jeu de données issu des personnes atteintes de la maladie d'Alzheimer. Ces données proviennent d'une des bases de données de l'Alzheimer's Disease Neuroimaging Initiative" (abrégé par ADNI) (?). Nous utilisons l'approche MPLSGPD proposée dans ce mémoire pour essayer de détecter les gènes responsables de la maladie. Dans un premier temps, nous expliquons la maladie d'Alzheimer, ensuite décrivons les données soumises à notre analyse et enfin effectuons l'analyse.

4.2.1 La maladie Alzheimer

La maladie d'Alzheimer est une maladie qui attaque le cerveau. Cette maladie neurodégénérative doit son nom au Dr Alzheimer en 1907. Avec l'âge, le cerveau subit des transformations qui peuvent être anormales chez certains et causer des maladies. Les cellules du cerveau, des neurones, peuvent alors se détériorer. Cette détérioration progressive est souvent définitive pour certaines cellules nerveuses et provoque une démence sénile. Selon l'Organisation mondiale de la santé(OMS), le diagnostic de la maladie d'Alzheimer est donné ainsi : «une altération progressive de la mémoire ainsi que de la formation et de l'enchaînement des idées, suffisamment marquée pour handicaper les activités de la vie quotidienne depuis au moins six mois». Cette maladie entraine donc une détérioration intellectuelle, des pertes de mémoire et des troubles de comportement qui conduisent à une perte d'autonomie.

Les symptômes de la maladie les plus connus sont : les pertes de mémoire ; le malade ne communique pas bien ; le malade peut répéter le même scénario (poser la même question, faire le même achat...) plusieurs fois sans s'en rendre compte ; le malade a des problèmes d'orientation dans le temps et dans l'espace et il devient difficile pour lui de se situer et de se déplacer sans un guide ; les planifications abstraites deviennent très compliquées pour lui, etc... Dû à la perte de la mémoire, le malade a tendance à oublier les mots, les dates importantes, et même comment utiliser certains appareils avec lesquels il était pourtant familier. Le malade peut vivre plusieurs émotions d'affilée, sans transition, il a donc des sauts d'humeur. Il est alors plus souvent très triste et agressif (verbalement ou physiquement), a un dégoût pour les loisirs, s'éloigne de ses amis, plus rien ne lui fait plaisir, etc. Plusieurs recherches portent sur comment guérir la maladie et quelles sont les causes potentielles de celle-ci. Certains médicaments et approches thérapeutiques permettent d'arrêter ou de soulager certains symptômes, et donc d'améliorer la

qualité de vie du malade, mais on ne sait pas encore comment arrêter l'évolution de la maladie, encore moins la guérir. Tout récemment, certains chercheurs ont établi une corrélation entre la maladie d'Alzheimer et l'accumulation de certains acides gras sur certaines cellules souches du cerveau. Ils ont établi que toutes les personnes atteintes avaient une accumulation élevée. Cette accumulation commence vers l'âge de vingt ans chez les humains, ils pensent que si l'on réduit cette accumulation, cela permettrait aussi de réduire la maladie. Des recherches ont permis de conclure que c'est une maladie qui peut être génétique, mais qu'il existe aussi des cas isolés provenant des facteurs environnementaux. Ces deux facteurs de risque combinés peuvent avoir une grande influence avec le temps. Certaines protéines malformées peuvent s'accumuler et devenir toxiques pour les cellules, et entrainer la mort de celles-ci.

Plusieurs hypothèses tentent d'expliquer les origines de la maladie d'Alzheimer. Les plus connues sont : l'hypothèse cholinergique (la synthèse insuffisante de l'acétylcholine[2] serait à l'origine de l'Alzheimer), l'hypothèse amyloïde (le dépôt anormal de la protéine bêta-amyloïde sous forme des plaques amyloïdes entre les neurones serait la cause première de l'Alzheimer) et l'hypothèse tau (le dépôt de la protéine Tau, Tubulin associated unit, dans le cortex cérébral serait à l'origine de la maladie). On peut retrouver le détail de chacune de ces hypothèses sur le site de l'Université McGill.[3]

Plusieurs gènes sont connus comme des facteurs à risque pour le développement de la maladie (Hollingworth *et al.*, 2011). On les appelle des gènes de prédisposition. L'un des plus connus est l'allèle 4 du gène de l'apolipoprotéine E (APOE4) qui se

---

2. "Substance chimique faisant partie des neurotransmetteurs, c'est-à-dire sécrétée par certains neurones pour transmettre l'influx nerveux vers d'autres cellules", selon le dictionnaire Larousse.

3. *http : //lecerveau.mcgill.ca/flash/d/d_08/d_08_p/d_08_p_alz/d_08_p_alz.html.*

trouve sur le chromosome 19. La présence ou non de cet allèle ne permet pas de conclure si l'individu va développer ou non la maladie. L'APOE possède quatre allèles. Le gène APOE fabrique ou participe à la fabrication d'une protéine qui est néfaste pour le corps, particulièrement pour l'Alzheimer.

4.2.2 Présentation des données

Les études d'association entre les gènes (genome wide association studies : GWAS) montrent qu'il existerait un lien entre l'Alzheimer et certains gènes. C'est dans ce cadre qu'une étude a été faite sur 418 sujets d'âge moyen de 72,62 ans (225 femmes et 193 hommes). Ces sujets ont une démence légère (au nombre de 27 codés AD pour "alzheimer's disease"), une déficience cognitive légère (au nombre de 265 codés MCI pour "mild cognitive impairment") ou une cognition normale (au nombre de 125 codés CN pour "cognitively normal"). Pour chaque individu, on a observé et mesuré certaines régions du cerveau avec des techniques d'imagerie cérébrale. Au total, on compte 96 variables continues qui sont ce que nous appelons ici des traits ou phénotypes. Ces phénotypes sont des mesures de la protéine bêta-amyloïde[4] et ce sont ces variables qui vont être analysées. L'ensemble de ces variables dans notre analyse constitue ce que nous appelons $\mathbf{Y}$. (Chaque variable représente une colonne dans la matrice $\mathbf{Y}$). On note aussi 11 covariables dont l'identité familiale, l'identité génétique, l'identité pour les données d'imagerie, les dates des examens, la valeur du ratio global, le diagnostique à la date de l'examen d'imagerie, l'année de naissance, le sexe, le niveau d'éducation, l'âge où l'on a été diagnostiqué malade et un examen mental total.
Pour chacun de ces patients, on a observé leurs SNPs du chromosome 19, consti-

4. "La bêta-amyloïde est un peptide néfaste pour le système nerveux. La présence d'agrégat de bêta-amyloïde et de protéine tau sont les signes caractéristiques de la maladie d'Alzheimer". Wikipédia

tuant ainsi l'autre ensemble de variables que nous notons $\mathbf{X}$. À ces SNPs, on note aussi la présence de certaines covariables (elles sont au nombre de 6). Ces données proviennent d'une des bases de données de l'Alzheimer's Disease Neuroimaging Initiative" (ADNI).

On souhaite alors déterminer le lien qui existerait entre $\mathbf{Y}$ et $\mathbf{X}$. À cause des valeurs manquantes ou aberrantes, notre analyse va porter sur 416 de ces sujets.

L'APEO4, comme d'autres gènes, est connu comme un des facteurs à risque de la maladie. Est-ce que notre approche nous permettrait de détecter ce gène ? Existeraient-ils d'autres gènes qui peuvent être détectés par notre approche ? Pour répondre à ces questions, nous effectuons dans un premier temps une analyse descriptive de ces données. Ensuite, dans le jeu de données qui est soumis à notre analyse, nous utilisons notre approche pour voir si nous pouvons détecter la présence de l'APOE4 chez des patients atteints de l'Alzheimer ou tout autre gène.

4.2.3 Analyse descriptive

On dispose de deux ensembles de données. Nous nous intéressons dans un premier temps aux valeurs de la matrice $\mathbf{Y}$ des traits. Nous faisons une analyse descriptive des données. Le tableau 4.2 donne une description de l'interprétation des covariables. Les codes et leur signification se trouve dans (?).

Tableau 4.2 Code et signification des covariables.

Code	Signification	
FID	Family ID	Identité familiale
PTID	Genetic ID	Identité Genetique
FID	Family ID	Identité familiale
RID	ID usually used in imaging data	Identité habituellement utilisée en imagerie
EXAMDATE	Date of the exam	Date de l'examen
av45_suvr_global	[18F]Florbetapir acquisition - global value ratio	Valeur moyenne globale des 96 régions
DXCHANGE	Exam date	Date de l'examen
PTDOBYY	Year of birth	Année de naissance
PTGENDER	Gender	Sexe
PTEDUCAT	Years of education	Niveau d'éducation
Age diagn	Age at diagnose	Âge au diagnostique
MMSCORE	MMSE[5](mimi mental state examination) total score at the diagnose date	Score du test à la date du diagnostique

Nous regardons la corrélation entre les colonnes de $\mathbf{Y}$ et constatons que ces dernières sont corrélées (voir figure 4.16).

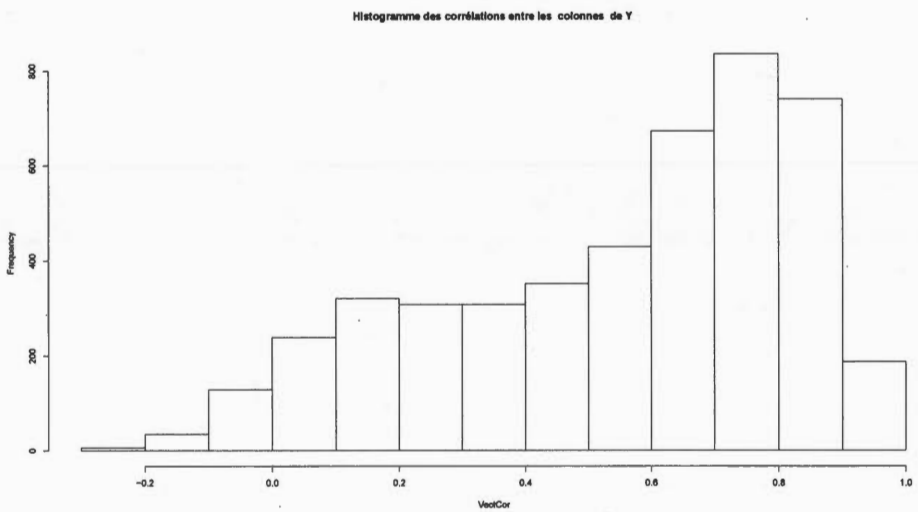

Figure 4.16 Histogramme des corrélations entre les 96 colonnes de $\mathbf{Y}$. Il y a 4656 telles corrélations.

Le test d'association entre les colonnes de $\mathbf{Y}$, deux-à-deux nous permet de confirmer ce fait, d'après la figure 4.17. En effet, le pourcentage des valeurs-p inférieures

à 0.05 est de 91.69%. Les variables sont donc fortement corrélées.

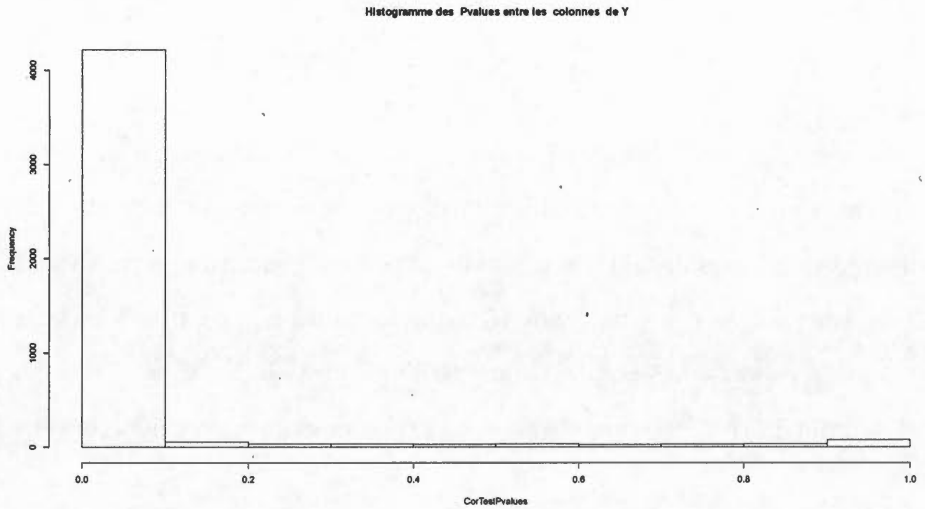

Figure 4.17 Histogramme des valeurs-p obtenues lors du test d'association entre les colonnes de $\mathbf{Y}$, deux-à-deux.

Donc, pour faire notre analyse, le fait que les colonnes de $\mathbf{Y}$ soient fortement corrélées nécessite l'utilisation des techniques de réduction de la dimension. Nous les analysons avec l'approche MPLSGPD présentée dans ce mémoire et d'autres approches présentées au chapitre 1.

4.2.4 Effet d'autres covariables sur les colonnes de Y.

Dans la description des données, nous avons vu que les variables de $\mathbf{Y}$ étaient fortement corrélées. Ces données peuvent être divisées en deux catégories : celles qui ont été mesurées sur le cerveau, qui constituent les 96 variables, et des covariables. Parmi les covariables, nous considérons trois (le sexe, le niveau d'éducation, l'âge où l'on a été diagnostiqué malade) qui, à notre avis, influenceraient les 96 variables de $\mathbf{Y}$. Ces variables sont notées : "ptgender", "pteducat" et "Age.diagn" respec-

tivement. Nous effectuons donc la régression linéaire de chacune des 96 mesures du cerveau en fonction d'elles. Le modèle est alors le suivant :

$$\mathbf{y}_j = \mathbf{X}\beta + \epsilon,$$

avec $\mathbf{y}_j$ une des 96 variables mesurées sur le cerveau, $\mathbf{X}$ la matrice dont les colonnes représentent les trois covariables. Après la régression de chacune des variables, nous considérons la matrice des valeurs-p ainsi constituée et représentons les résultats sous forme d'histogramme comme le montre la figure 4.18 ou sous forme de boîte à moustaches comme le montre la figure 4.19.

Pour déterminer parmi les trois covariables celles qui influenceraient le plus les

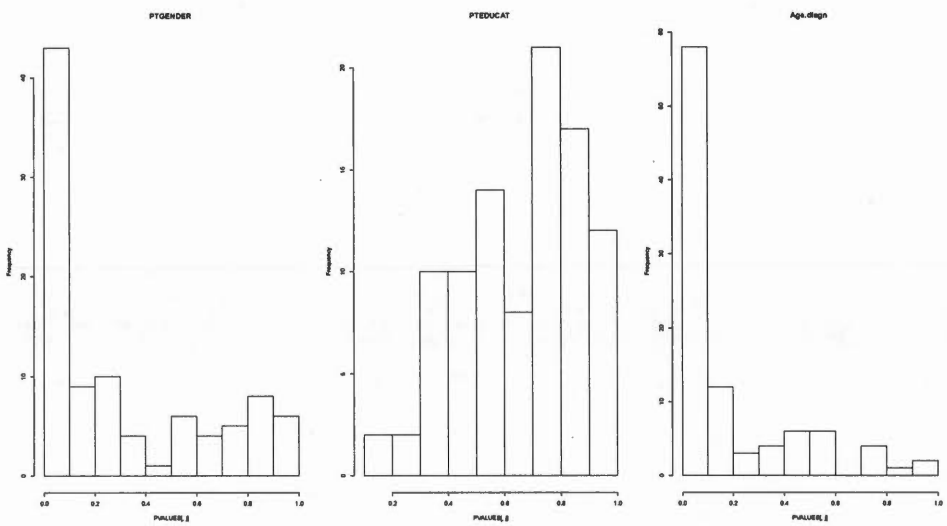

Figure 4.18 Histogramme des valeurs-p pour chacune des trois covariables cibles de $\mathbf{Y}$. Les graphiques de la gauche vers la droite représentent respectivement les covariables "ptgender" ou le sexe, "pteducat" ou le niveau d'éducation et "Age.diagn" ou l'âge au diagnostique.

96 variables, nous avons déterminé, pour chacune des covariables, le nombre des valeurs-p qui étaient inférieures ou égales à 0.05, sans et avec la correction de

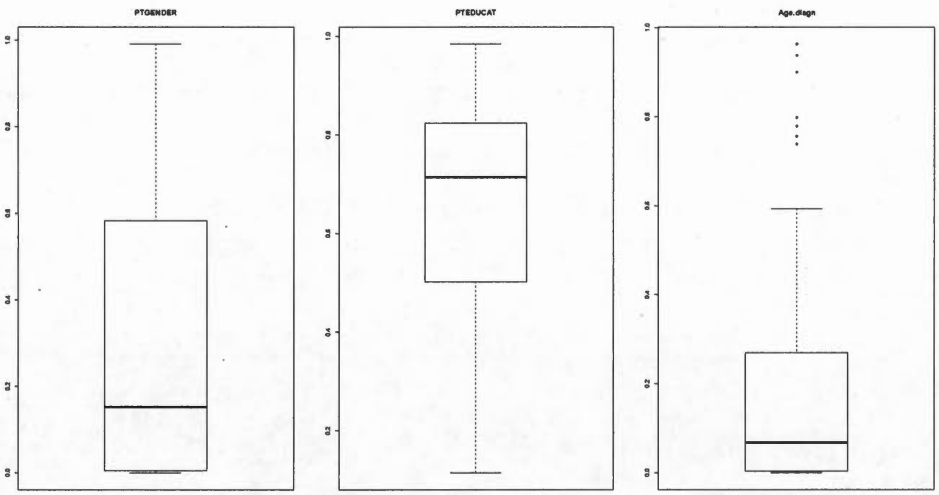

Figure 4.19 Boîte à moustaches des valeurs-p pour chacune des trois covariables de $\mathbf{Y}$. Les graphiques de la gauche vers la droite représentent respectivement les covariables "ptgender" ou le sexe, "pteducat" ou le niveau d'éducation et "Age.diagn" ou l'âge au diagnostique.

Bonferroni. [6]

On a ainsi obtenu le tableau 4.3.

6. correction de Bonferroni est une méthode pour corriger le seuil de signification lors des comparaisons multiples. C'est donc un ajustement qui est fait sur les valeurs-p lorsque plusieurs tests (n tests) statistiques (dépendants ou non) ont lieu, simultanément, sur un même ensemble de données. Dans ce cas, le seuil critique α des valeurs-p (généralement $\alpha = 0.05$) est donc modifié par un nouveau seuil, α/n, qui dépend du nombre de tests effectués. On l'utilise donc pour réduire les chances d'avoir des erreurs de type I.

Tableau 4.3 Comparaison du nombre des valeurs-p obtenues de la régression linéaire de chacune des 96 mesures du cerveau en fonction des trois covariables ("ptgender" ou le sexe, "pteducat" ou le niveau d'éducation et "Age.diagn" ou l'âge au diagnostique) sans et avec la correction de Bonferroni.

Covariables	Nombre de valeurs-p sans correction de Bonferroni ≤ 0.05	Nombre de valeurs-p avec correction de Bonferroni $\leq 0.05/96$
ptgender	35	15
pteducat	0	0
age.diagn	45	11

On constate que les covariables "ptgender" et "age.diagn" sont les plus significatives. Nous retenons donc les covariables "ptgender" et "age.diagn" comme significatives et allons enlever leurs effets sur les 96 variables. On effectue donc la régression des 96 variables par rapport aux deux covariables significatives.
En effet, notre but est d'analyser les 96 mesures par rapport aux SNP. Puisqu'il y a d'autres covariables qui peuvent être confondantes et donc avoir de l'influence sur les 96 mesures, nous allons enlever leurs effets sur ces mesures pour ainsi utiliser notre approche avec $\mathbf{Y}$ correspondant à la matrice des valeurs ajustées et les SNP. Pour la régression linéaire de chacune des 96 mesures du cerveau en fonction des deux covariables ("ptgender", "age.diagn"), le modèle est alors le suivant :

$$\mathbf{y}_j = \mathbf{X}\beta + \epsilon,$$

avec $\mathbf{y}_j$ l'une des 96 variables mesurées dans le cerveau, $\mathbf{X}$ la matrice dont les colonnes représentent les covariables "ptgender" et "age.diagn". On va donc enlever l'effet de ces deux variables sur les 96 mesures et ne considérer que les résidus. Ainsi, pour chaque mesure du cerveau, on obtient la variable estimée

$$\hat{\mathbf{y}}_j = \mathbf{X}\hat{\beta},$$

et le résidu correspondant

$$\tilde{\mathbf{y}} = \mathbf{y}_j - \mathbf{X}\hat{\beta}.$$

Dans la suite de ce travail, nous utilisons donc la matrice des résidus, celle obtenue en enlevant l'effet des covariables significatives :

$$\mathbf{Y} = \tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \cdots, \tilde{\mathbf{y}}_{96}].$$

4.2.5 L'approche MPLSGPD

Avec la description des données, la mesure "av45_suvr_global" est une mesure globale des 96 mesures du cerveau, équivalente à leur moyenne. Alors, comme l'approche PLS est univariée, nous proposons de faire une comparaison entre les approches PLS et MPLSGPD, en considérant comme seul vecteur la mesure "av45_suvr_global" pour l'approche PLS et comme traits les 96 mesures du cerveau pour l'approche multivariée MPLSGPD. Pour l'analyse, dans un premier temps, nous avons divisé la matrice des SNP en plusieurs fenêtres ou sous matrices, qui se suivent avec une intersection de dix SNP à la fois. Chaque fenêtre a vingt SNP. Ce qui donne un total de 998 fenêtres. Les résultats obtenus sont présentés sur la figure 4.20. Dans un second temps, nous avons procédé de la même manière, mais cette fois avec un seul SNP à la fois. Les résultats obtenus sont présentés sur la figure 4.21. Dans les figures 4.20 et 4.21, pour PLS, la variable réponse est celle de la mesure globale,"av45_suvr_global", et pour MPLSGPD la matrice $\mathbf{Y}$ est celle des 96 mesures. Le nombre de permutations ici est 1000. Sur le graphique, la légende 2 représente la distribution des valeurs-p obtenues par l'approche MPLSGPD et 1 représente la distribution par l'approche PLS.

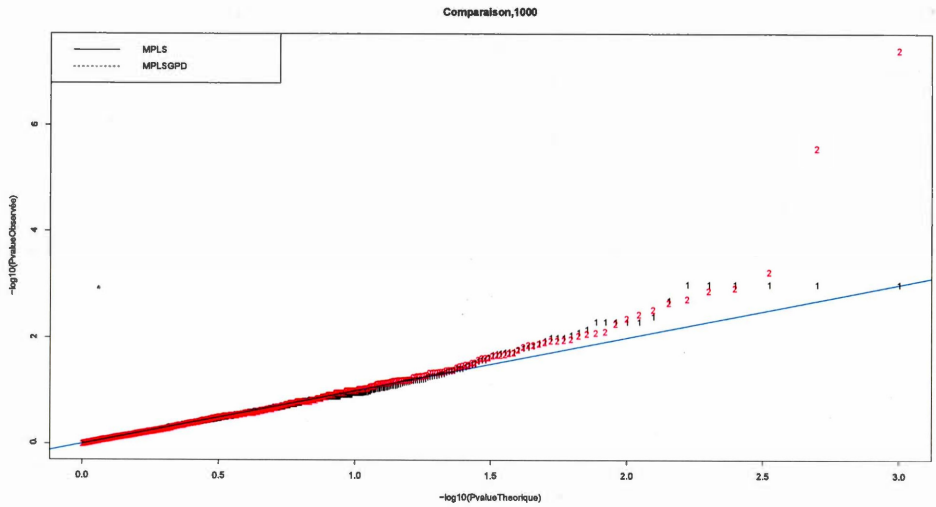

Figure 4.20 QQ-plots des valeurs-p observées versus les valeurs-p espérées sous l'hypothèses nulle pour les méthodes PLS et MPLSGPD. Sur le graphique, la légende 2 représente la distribution des valeurs-p obtenues par l'approche MPLSGPD et 1 représente la distribution par l'approche PLS. On considère 20 SNP à la fois. Le nombre de permutations ici est 1000.

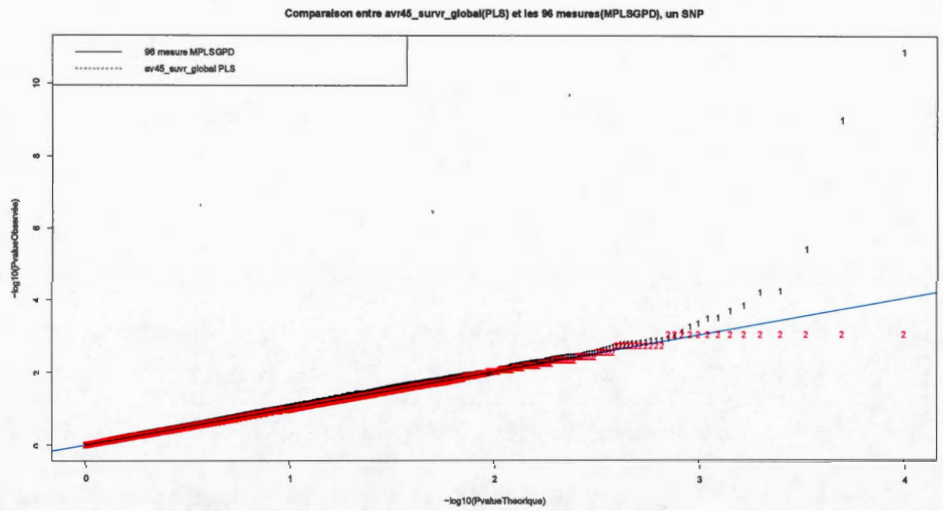

Figure 4.21 QQ-plots des valeurs-p observées versus les valeurs-p espérées sous l'hypothèses nulle pour les méthodes PLS et MPLSGPD. Sur le graphique, la légende 1 représente la distribution des valeurs-p obtenues par l'approche MPLSGPD et 2 représente la distribution par l'approche PLS. On considère un SNP à la fois. Le nombre de permutations ici est 1000. .

Que l'on considère un SNP à la fois ou une fenêtre de 20 SNP, nous n'avons aucun signal pour la mesure globale. En regardant ces figures, nous pouvons dire que les distributions des valeurs-p sont uniformes, et qu'il y a peu de SNP qui sont significatifs. Le QQ-plot montre une vraie association. En effet, le QQ-plot ne montre une déviation de la distribution des valeurs-p de la distribution uniforme (c'est-à-dire leur distribution sous l'hypothèse nulle) que à l'extrémité droite, ce qui suggère une association forte entre les quelques SNP correspondants aux valeurs-p dans l'extrémité droite et les variables réponses. Notons que par l'approche MPLSGPD, il y a un gain de puissance, ce qui montre une fois de plus l'efficacité de notre approche.

La puissance est plus élevée pour l'analyse avec un SNP à la fois par rapport à l'analyse avec les 20 SNP. Nous avons alors déterminé quelles sont les fenêtres où le signal est plus élevé, appelées fenêtres cibles, et les SNP correspondants. Les fenêtres cibles 701 et 702 correspondent aux SNP suivants : le tableau 4.4 donne les SNP et les valeurs-p obtenues pour les fenêtres 701 et 702. Il y a dix SNP qui appartiennent aux deux fenêtres. Certains gènes correspondant à ces SNP seraient des facteurs de risque de la maladie. Le tableau A.1 décrit le gène correspondant à chacun de ces SNP. Pour l'analyse avec un SNP à la fois, nous avons obtenus trois SNP dont les valeurs-p sont au dessus du seuil du manhattan plot : "rs2075650_G", "rs157582_A" et "rs769449_A". Malgré la longueur des calculs, l'avantage avec l'analyse d'un SNP à la fois est que cette méthode dégage directement quel sont les SNP cibles et donc les gènes cibles comme facteurs à risque de la maladie d'Alzheimer. Nous obtenons les figures 4.22 et 4.23 respectivement.

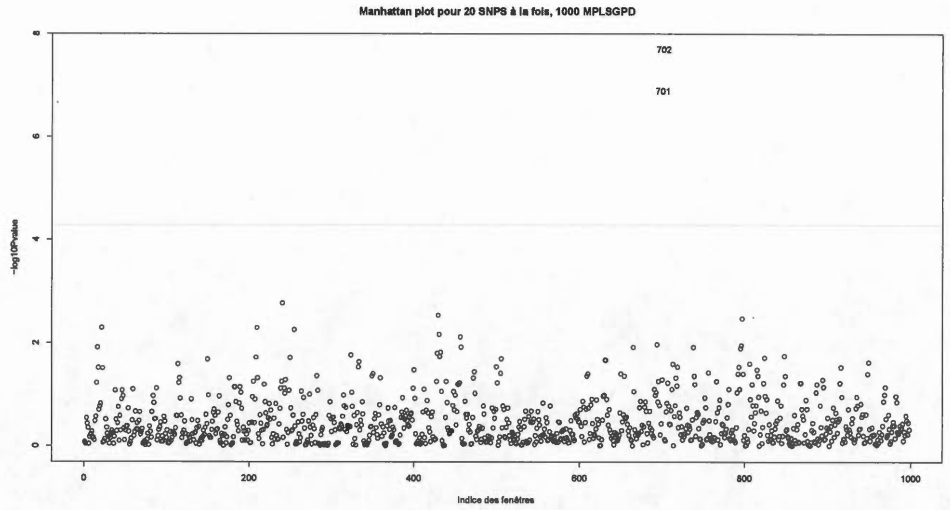

Figure 4.22 Manhattan plot des valeurs-p obtenues avec 20 SNPs à la fois. Le nombre de permutations ici est 1000. L'axe de y représente $-\log 10(valeur - p)$ et l'axe de x désigne les indices des marqueurs selon l'ordre de la matrice des génotypes. La ligne bleue indique un seuil de $-\log\{\frac{0.05}{longueur\ du\ vecteur\ des\ valeur-p}\}$

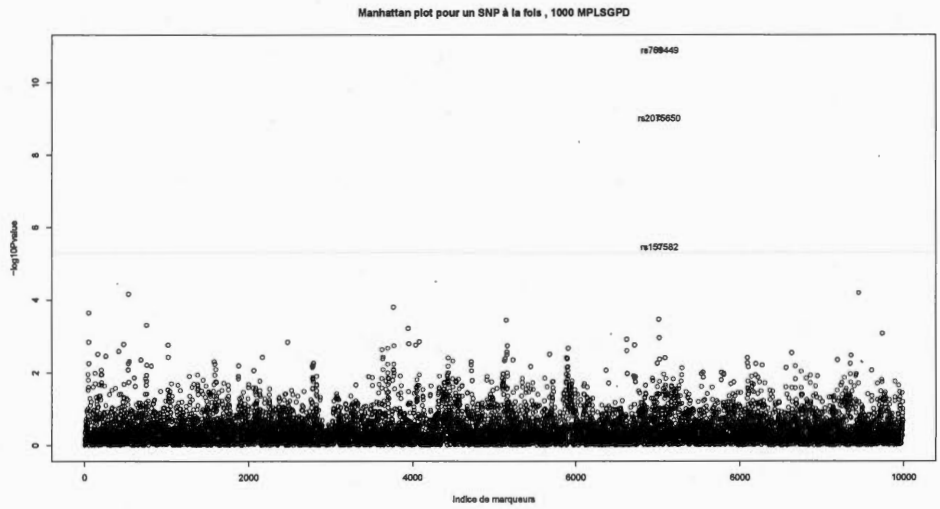

Figure 4.23 Manhattan plot des valeurs-p obtenues avec un SNPs à la fois. Le nombre de permutations ici est 1000. L'axe de y représente $-\log 10(valeur - p)$ et l'axe de x désigne les indices des marqueurs selon l'ordre de la matrice des génotypes. La ligne bleue indique un seuil de $-\log 10\{\frac{0.05}{longueur\ du\ vecteur\ des\ valeur-p}\}$

Nous avons déjà décrit l'APOE comme un des gènes associés à la maladie d'Alzheimer. Les autres gènes sont-ils aussi des facteurs de risque ? Nous sommes donc allées dans la littérature pour voir quels sont les gènes associés à la maladie l'Alzheimer. Ou autrement, est-ce que ces gènes (PVRL2, TOMMM40, APOE, APOC1, APOC4-APOC2, LOC105372418) obtenus après notre analyse ont effectivement un lien avec la maladie ? Une étude a été faite par Logue et al, (2011) sur les Afro-américains pour voir l'association génétique qu'il y aurait entre la maladie d'Alzheimer et cette population. Il y ressort que les gènes PVRL2, TOMMM40, APOE, APOC1 sont des facteurs de risque de la maladie, et plus précisément que APOC1 affecte la mémoire. On obtient le même résultat avec une population de blancs, mais dans un pourcentage différent. Nous pouvons donc dire que l'ap-

proche MPLSGPD donne une autre façon de retrouver cette association génétique multidimensionnelle.

Tableau 4.4 Tableau donnant les SNP des fenêtres 701 et 702 et les valeurs-p obtenues. En gras nous avons les valeurs-p inférieures au seuil : $-\log 10\{\frac{0.05}{longueur\ du\ vecteur\ des\ valeur-p}\}$ = -4.30016053

	fenêtre701	Valeur-p 701	fenêtre702	Valeur-p 702
1	rs3852856_A	0.368	rs3729640_T	0.984
2	rs519825_G	0.08	rs6859_A	0.00104609
3	rs8105340_C	0.02347418	rs283814_T	0.268
4	rs12610605_A	0.01851852	rs157580_G	0.001165542
5	rs4803766_A	0.928	rs2075650_G	**2.94649e − 09**
6	rs8104483_G	0.336	rs157582_A	**1.014262e − 07**
7	rs11879589_A	0.472	rs8106922_G	0.03521127
8	rs395908_T	0.216	rs1160985_T	0.004669119
9	rs2075642_A	0.54	rs405509_C	0.0101833
10	rs387976_G	0.316	rs769449_A	**1.356843e − 11**
11	rs3729640_T	0.944	rs439401_T	0.136
12	rs6859_A	0.001533103	rs445925_T	0.792
13	rs283814_T	0.292	rs1064725_G	0.728
14	rs157580_G	0.001282702	rs5157_T	0.456
15	rs2075650_G	**1.084464e − 09**	rs1132899_T	0.74
16	rs157582_A	**1.701908e − 07**	rs5167_G	0.812
17	rs8106922_G	0.02849003	rs2288911_A	0.44
18	rs1160985_T	0.002988286	rs10413089_C	0.856
19	rs405509_C	0.007332963	rs3760627_C	0.504
20	rs769449_A	**2.207548e − 10**	rs204905_C	0.328

CONCLUSION

Les maladies complexes comme le cancer, le diabète, l'Alzheimer, les maladies cardio-vasculaires, etc sont des maladies qui sont causées par des facteurs génétiques. Plusieurs gènes peuvent être impliqués dans l'étiologie ces maladies ; ainsi, des défis est d'identifier la composante génétique, trouver les gènes responsables. Quelle est l'interaction que ces gènes exerceraient avec l'environnement ou avec d'autres gènes ? Existe-t-il une association entre ce gène et la maladie ? Plusieurs mesures d'association font des analyses individuelles des marqueurs, mesurant l'association qui existerait entre un ensemble de SNP et un trait, par exemple. De telles analyses sont très limitées, parce qu'elles ne tiennent pas compte de la relation qui existerait entre plusieurs traits. Les SNP ne sont pas indépendants, de même que les traits : deux SNP peuvent donner la même information. Une analyse individuelle alourdit les tests parce que les bases de données sont des matrices de l'ordre des milliers et le gain en puissance est bas.

Dans le cadre de ce travail, nous avons décrit une approche qui nous permet de réduire la dimension des données en gardant la corrélation maximale, et donc toute l'information utile pour notre analyse, mais aussi d'avoir un gain en puissance. Le jeu de données sur la maladie d'Alzheimer nous a permis de valider notre approche en détectant la plupart des gènes (APOE, PVRL2, TOMM40) que la littérature de spécialité indique avoir une association avec la maladie. La comparaison avec les autres approches nous a aussi permis de confirmer que notre approche serait plus efficace que plusieurs existantes. Nous aurions souhaité faire plusieurs autres comparaisons, particuliè- rement avec celle de PRM (de l'anglais Projection Regression Model) décrite dans (Lin et al., 2012). Dans cet article, les auteurs

mesurent l'association génétique qui existerait entre un ensemble de phénotypes et des covariables. Ils donnent une généralisation des méthodes statistiques d'analyse en composantes principales de l'héritabilité basées sur le "wild-bootstrap". Nous pensons qu'il serait aussi judicieux d'explorer d'autres approches (LASSO, RR, PCR, etc) données par Xu et Greenwood (2013) afin de les généraliser aux cas multidimensionnels.

ANNEXE A

CODE R

A.1 Algorithme de PLS (NIPALS, Nonlinear iterative partial least squares)

```
#Abdi Algo
##############################
Y=matrix(c(14,10,8,2,6,7,7,5,4,2,8,6,5,7,4),5)
colnames(Y)=c('Hedonic','Goes with meat','Goes with dessert')
X=matrix(c(7,4,10,16,13,7,3,5,7,3,13,14,12,11,10,7,7,5,3,3),5)
colnames(X)=c('Price','Sugar','Alcohol','Acidity')
X0=scale(X, center = TRUE,  scale = TRUE)#Pour centraliser et normaliser la
#matrice. la moyenne des colones est 0 et l'ecart-type1.
Y0=scale(Y, center = TRUE, scale = TRUE)
E <- X0
F <- Y0
u<-Y0[,1]
#Etape 1:  Estimer X weights
W=t(E)%*%u
W=W/norm(W, type='2')
#Etape 2 Estimer X factor scores
t0= E%*%W
t0=t0/norm(t0, type='2')
```

```
#Etape 3 Estimer Y weights
c=  t(F)%*%t0
c=W/norm(W, type='2')
#algo de convergence pour t
#CONVERVENCE DE t
u<-Y0[,1]
#Etape 1:  Estimer X weights
W=t(E)%*%u
W=W/norm(W, type='2')
#Etape 2 Estimer X factor scores
t0= E%*%W
t0=t0/norm(t0, type='2')
#Etape 3 Estimer Y weights
c=  t(F)%*%t0
c=c/norm(c, type='2')
epsi <-10^-2
        while(epsi >= 10^(-8) ){
        u=F%*%c
        W=t(E)%*%u
        W = W/norm(W, type='2')
        tt<-E%*%W
        tt = tt/norm(tt, type='2')
        #Estimer Y weights
        c= t(F)%*%tt
        c = c/norm(c, type='2')
        epsi<-as.numeric(t(t0-tt)%*%(t0-tt))
        print(epsi)
        t0<-tt
```

```
        }
#Algorithm
epsi2<-10^-2
Wmatrix<-NULL
Tmatrix<-NULL
Cmatrix<-NULL
Umatrix<-NULL
Pmatrix<-NULL
dE<-1
while(dE> epsi2){
         epsi<-10^(-2)
        t0<-E%*%t(E)%*%F[,1]
        t0<- t0/norm(t0, type='2')

       while(epsi >=10^(-3) ){
      u<-u
      W<-t(E)%*%u
       W= W/norm(W, type='2')

      t<-  E%*%W
      t<- t/norm(t, type='2')

      #Estimer Y weights
      c=  t(F)%*%t
      c<- c/norm(c, type='2')
      #Estimer Y score
      u=  F%*%c
      epsi<-as.numeric(t(t0-t)%*%(t0-t))
```

```
      epsi
      t0<-t
      }
b=t(t)%*%u
b<-as.numeric(b)
p=t(E)%*%t
E=E-t%*%t(p)
F=F-b*(t%*%t(c))
V<-round(E%*%t(E),4)
         for(i in 1:nrow(V)){
          sumV=sum(V[i,i])
          }
dE<-sum(V)
print(dE)
Tmatrix<-cbind(Tmatrix,t)
Umatrix<-cbind(Umatrix, u)
Wmatrix<-cbind(Wmatrix,W)
Cmatrix<-cbind(Cmatrix,c)
Pmatrix<-cbind(Pmatrix,p)
}
T<-round(Tmatrix,4)
U <-round(Umatrix,4)
W <-round(Wmatrix,4)
C <-round(Cmatrix,4)
P <-round(Pmatrix,4)
```

A.2 La fonction "Multivariée"

Cette fonction permet de simuler des matrices $\mathbf{X}$ et $\mathbf{Y}$, en imposant une certaine corrélation entre les colonnes de $\mathbf{X}$ et celles de $\mathbf{Y}$. Il faut alors spécifier les paramètres convenables : le nombre de lignes des matrices, le nombre de colonnes voulues pour chacune des matrices, la corrélation entre les composantes latentes de $\mathbf{X}$ et $\mathbf{Y}$, (ρ_1 et ρ_1), et la variance des matrices des erreurs.

```
Multivarié <- function(rowx, colx, rowy, coly, Rho1, Rho2, sigmax,
sigmay, sigmaxg,sigmayg)
{
#INPUT
#Rho1 <- correlation entre t1 et u1
#Rho2 <- correlation entre t2 et u2
#rowx: le nombre de ligne de la matrice X
#colx: le nombre de colonnes de la matrice X
#rowy: le nombre de ligne de la matrice Y
#coly: le nombre de colonnes de la matrice Y
#sigmaX <- Variance de la matrice des erreurs  de X
#sigmaY <- Variance de la matrice des erreurs de Y
#sigmaxg <- facteur de la matrice de Kinship de X
#sigmayg <- facteur de la matrice de Kinship de Y
#Détermination de la matrice P, X loading
P= matrix(c(rep(0.5, 2*colx)), colx)
#Détermination de la matrice Q, Y loading
Q= matrix(c(rep(0.5, 2*coly)), coly)
#Détermination des  Scores, T et U
#Matrice des moyennes des ti et ui
```

```
mu  <- c(rep(0,4))
#Matrice de covariance des ti et ui
SigmaScores <- matrix(c(1,Rho1,0,0,Rho1,1,0,0,0,0,1,Rho2,0,0,Rho2,1),4)
#SigmaScores <- matrix(c(1,0.2,0,0,0.2,1,0,0,0,0,1,0.2,0,0,0.2,1),4)
#Détermination des Scores = (t1,u1, t2, u2)
Scores <- mvrnorm(rowx, mu, SigmaScores)
# Déduction des XScores T = (t1,t2)
T<- matrix(c(Scores[,1],Scores[,3]),rowx)
ScoresX <- T
#Déduction des YScores, U = (u1, u2)
U<- matrix(c(Scores[,2],Scores[,4]),rowx)
ScoresY <- U
#Détermination de la matrice des erreurs de la matrice X: SigmaX
SigmaX <- t(  mvrnorm(colx,mu=c(rep(0,rowx)),  sigmax*diag(1,rowx)))
#Détermination  de la matrice de Kingship PHI
phi<- matrix(c(1,0,0.5,0,1,0.5,0.5,0.5,1),3)
PHI <- list(length=(rowx/3))
for( i in 1:(rowx/3)) PHI[[i]] <- phi
PHI <- bdiag(PHI)
#Détermination de la matrice   des erreurs de la matrice X:  SigmaXg
SigmaXg <-   t(  mvrnorm(colx,mu=c(rep(0,rowx)), sigmaxg*PHI))
#Détermination de la matrice  X
X <- T%*%t(P)   + SigmaX + SigmaXg
#Détermination de la matrice   des erreurs des variables expliquées SigmaY
SigmaY <-  t(mvrnorm(coly, mu=c(rep(0,rowy)),  sigmay*diag(1,rowy)))
#Détermination de la matrice   des erreurs des variables expliquées SigmaYg
SigmaYg <- t(mvrnorm(coly, mu=c(rep(0,rowy)), sigmayg*PHI))
#Détermination de la matrice Y
```

```
Y <- U%*%t(Q)    + SigmaY  + SigmaYg
 return(list(X=X, Y=Y, ScoresX= ScoresX, ScoresY= ScoresY ))
 }
```

A.3 La fonction "Univariée"

Cette fonction permet de simuler une matrice $\mathbf{X}$ et un vecteur $\mathbf{Y}$, en imposant une certaine corrélation entre les colonnes de $\mathbf{X}$ et celles de $\mathbf{Y}$. Il faut alors spécifier les paramètres convenables : le nombre de lignes des matrices, le nombre de colonnes voulues pour la matrice $\mathbf{X}$, la corrélation entre les composantes latentes de $\mathbf{X}$ et $\mathbf{Y}$, (ρ_1), et la variance des matrices des erreurs.

```
Univarié <- function(rowx, colx, rowy, coly=1, Rho1,  sigmax, sigmay,
sigmaxg, sigmayg)
{
#INPUT
#Rho1 <- correlation entre t1 et u1
#rowx: le nombre de ligne de la matrice X
#colx: le nombre de colonnes de la matrice X
#rowy: le nombre de ligne de la matrice Y
#coly: le nombre de colonnes de la matrice Y = 1
#sigmax <- Variance de la matrice des erreurs  de X
#sigmay <- Variance de la matrice des erreurs de Y
#sigmaxg <- facteur de la matrice de Kinship de X
#sigmayg <- facteur de la matrice de Kinship de Y
#Détermination de la matrice P, X loading
P= matrix(c(rep(0.5, 2*colx)), colx)
#Détermination de la matrice Q, Y loading
Q= matrix(c(rep(0.5, 2*coly)), coly)
```

```
#Détermination des  Scores, T=(t1,t2) et U=u1
#Détermination des  Scores, T et U
#Matrice des moyennes des ti et ui
mu  <- c(rep(0,4))
#Matrice de covariance des ti et ui
SigmaScores <- matrix(c(1,Rho1,0,0,Rho1,1,0,0,0,0,1,Rho2,0,0,Rho2,1),4)
#Détermination des Scores = (t1,u1, t2, u2)
Scores <- mvrnorm(rowx, mu, SigmaScores)
# Déduction des XScores T = (t1,t2)
T<- matrix(c(Scores[,1],Scores[,3]),rowx)
ScoresX <- T
#Déduction des YScores, U = (u1, u2)
U<- matrix(c(Scores[,2],Scores[,4]),rowx)
ScoresY <- U
#Détermination de la matrice des erreurs de la matrice X: SigmaX
SigmaX <- t(mvrnorm(colx,mu=c(rep(0,rowx)),  sigmax*diag(1,rowx) ))
#Détermination  de la matrice de Kingship PHI
phi<- matrix(c(1,0,0.5,0,1,0.5,0.5,0.5,1),3)
PHI <- list(length=(rowx/3))
for( i in 1:(rowx/3)) PHI[[i]] <- phi
PHI <- bdiag(PHI)
#Détermination de la matrice   des erreurs de la matrice X:
 SigmaXg
SigmaXg <-   t(mvrnorm(colx,mu=c(rep(0,rowx)), sigmaxg*PHI))
#Détermination de la matrice  X
X <- T%*%t(P)   + SigmaX + SigmaXg
#Détermination de la matrice   des erreurs des variables expliquées
 SigmaY
```

```
SigmaY <-  mvrnorm(coly, mu=c(rep(0,rowy)),  sigmay*diag(1,rowy))
SigmaY<- as.matrix(SigmaY)
#Détermination de la matrice   des erreurs des variables expliquées
SigmaYg
SigmaYg <- mvrnorm(coly, mu=c(rep(0,rowy)), sigmayg*PHI)
SigmaYg<- as.matrix(SigmaYg)
#Détermination de la matrice Y
Y <- U%*%t(Q)   + SigmaY  + SigmaYg
 return(list(X=X, Y=Y, ScoresX= ScoresX, ScoresY= ScoresY ))
 }
```

A.4 La fonction GPD

```
 library(POT)
 library(ADGofTest)
#Test de la P.value>0.05, on accepte
GPD<-function(y, x0, Nexc){
N<-length(y)
y<-rev(sort(y))
M<-length(y[y>=x0])#parce que la statistique de test est une student
if (M>= 10) {
Pprime<- M/N #Pour  le calcule de Pprimecdf
 } else {
t <- (y[Nexc] + y[Nexc  + 1])/2
z <- y-t
z<-z[z>0]
fitpar<-fitgpd(y,t,"mle")$param ##pour estimer les parametres
PVALUE<- ad.test(z, pgpd, scale=fitpar[1], shape=fitpar[2])$p.value
while (PVALUE<=0.05){
```

```
Nexc<- Nexc-10
t <- (y[Nexc] + y[Nexc  + 1])/2
z = y-t
z<-z[z>0]
fitpar<-fitgpd(y,t,"mle")$param ##pour estimer
 les parametres
PVALUE<- ad.test(z, pgpd, scale=fitpar[1],
shape=fitpar[2])$p.value
if  (Nexc<=0) break
}
a<-fitpar[1]
b<- fitpar[2]
PVALUE
Nexc
F<- pgpd(x0-t , scale=a, shape=b)[[1]]
Pprime<- (Nexc/length(y))*(1-F) #Pour le calcul de Pgpd
}
return(Pprime)
}
```

A.5 La fonction MPLS

```
MPLS <-
function(x, y, scale = FALSE, ncomp, varpercent, npermutation=100 ,
 npermutation.max, min.nonsignificant.counts)
{
#varpercent = 0.80
#Détermine le pourcentage de la variance expliquée par X
#always centralizing x and y
```

```
x<- scale(x, scale = scale)
y<-as.matrix(y)
y<-scale(y , scale = scale)
#missing() : teste si un argument a été  fourni
if (missing(min.nonsignificant.counts)) min.nonsignificant.counts<- 10
 #?20
if (missing(npermutation.max)) npermutation.max<- npermutation
if (missing(ncomp) & missing(varpercent)) ncomp<- min(dim(x))
if (!missing(ncomp)) {
stopifnot(ncomp >= 1)
ncomp<- floor(ncomp)
names(ncomp)<- paste("MPLS", ncomp, sep="")
}
else ncomp<- NULL
if (!missing(varpercent)) stopifnot(varpercent >0 & varpercent <= 1)
#(1) test score and pvalue (nominal pvalue)
yp<- y
if (missing(varpercent)) {
ncompvar<- NA
fit<- plsr(yp~x, scale = FALSE, ncomp =  max(ncomp))
ncomptest<- ncomp
}
else {
fit<- plsr(yp~x, scale = FALSE)
indx<- cumsum(explvar(fit)) >= (100*varpercent)#donne un vecteur de
 TRUE et FALSE
ncompvar<- min(which(indx))
names(ncompvar)<- paste("PLS", ncompvar, ".v", round(varpercent,
```

```
digits=2), sep="")
ncomptest<- c(ncomp, ncompvar)
}
sco<- sapply(ncomptest, function(k) score(x%*%fit$coefficients[ , ,k], yp))

testscore<- abs(sco)
testpvalue<- NULL
#testscore<- sco*sqrt( (nrow(x)-2)/(1-sco^2) ) #t-distribution with df of n-2.
# not correct??? cor(y, Ay)?
#testpvalue<- 2*pt(abs(testscore), df = nrow(x)-2, lower.tail = FALSE)
#(2) permutation pvalue (empirical pvalue)
#permutation
permpvalue<- NULL
counts<- NULL
jth<- 0
if (npermutation >= 1) {
counts<- rep(0, length(ncomptest))
while ((jth < npermutation) | ((jth < npermutation.max) & (min(counts) <
 min.nonsignificant.counts)))
{
jth<- jth + 1
yp <- y[sample(nrow(x)),]#Pour permuter les lignes de la matrice y
if (missing(varpercent)) {
fit<- plsr(yp~x, scale = FALSE, ncomp =  max(ncomp))
ncomptest<- ncomp
} else {
fit<- plsr(yp~x, scale = FALSE)
indx<- cumsum(explvar(fit)) >= (100*varpercent)
```

```
ncomptest<- c(ncomp, min(which(indx)))
}
sco<- sapply(ncomptest, function(k) score(x%*%fit$coefficients[ , ,k], yp))
permscore<- abs(sco)
#permscore<- sco*sqrt( (nrow(x)-2)/(1-sco^2) ) #t-distribution with df of n-2.
counts<- counts + (permscore >= testscore)
}#while
permpvalue<- (1+ counts)/(1 + jth)
}#if
list(score = testscore, nonsignificant.counts = counts,
pvalue.empirical = permpvalue,
 pvalue.nominal = testpvalue,
 total.permutation = jth, ncomp.varp = ncompvar)
}
```

A.6 La fonction MPLSGPD

```
MPLSGPD <-
function(x, y, scale = FALSE, ncomp, varpercent, npermutation=250 ,
 npermutation.max,
 min.nonsignificant.counts)
{
#varpercent = 0.80
#Détermine le pourcentage de la variance expliquée par X
#always centralizing x and y
x<- scale(x, scale = scale)
y<-as.matrix(y)
y<-scale(y,center=TRUE)
#missing() : teste si un argument a été  fourni
```

```
if (missing(min.nonsignificant.counts)) min.nonsignificant.counts<- 10
 #?20
if (missing(npermutation.max)) npermutation.max<- npermutation
if (missing(ncomp) & missing(varpercent)) ncomp<- min(dim(x))
if (!missing(ncomp)) {
stopifnot(ncomp >= 1)
ncomp<- floor(ncomp)
names(ncomp)<- paste("MPLS", ncomp, sep="")
}
else ncomp<- NULL
if (!missing(varpercent)) stopifnot(varpercent >0 & varpercent <= 1)
#(1) test score and pvalue (nominal pvalue)
yp<- y
if (missing(varpercent)) {
ncompvar<- NA
fit<- plsr(yp~x, scale = FALSE, ncomp =  max(ncomp))
ncomptest<- ncomp
}
else {
fit<- plsr(yp~x, scale = FALSE)
indx<- cumsum(explvar(fit)) >= (100*varpercent)#donne un vecteur
de TRUE et FALSE
ncompvar<- min(which(indx))
names(ncompvar)<- paste("PLS", ncompvar, ".v", round(varpercent,
digits=2), sep="")
ncomptest<- c(ncomp, ncompvar)
}
sco<- sapply(ncomptest, function(k) score(x%*%fit$coefficients[ , ,k], yp))
```

```
testscore<- abs(sco)
obsSCO <- abs(sco)[1]
testpvalue<- NULL
#testscore<- sco*sqrt( (nrow(x)-2)/(1-sco^2) ) #t-distribution with df
of n-2.
# not correct??? cor(y, Ay)?
#testpvalue<- 2*pt(abs(testscore), df = nrow(x)-2, lower.tail = FALSE)
###################################
#(2) permutation pvalue (empirical pvalue)
#permutation
permpvalue<- NULL
permpvalueGPD <-NULL
counts<- NULL
PermSCO <- NULL
jth<- 0
if (npermutation >= 1) {
counts<- rep(0, length(ncomptest))
while ((jth < npermutation) | ((jth < npermutation.max) & (min(counts) <
 min.nonsignificant.counts)))
{
jth<- jth + 1
yp <- y[sample(nrow(x)),]#Pour permuter les lignes de la matrice y
if (missing(varpercent)) {
fit<- plsr(yp~x, scale = FALSE, ncomp =  max(ncomp))
ncomptest<- ncomp
} else {
fit<- plsr(yp~x, scale = FALSE)
indx<- cumsum(explvar(fit)) >= (100*varpercent)
```

```
ncomptest<- c(ncomp, min(which(indx)))
}
sco<- sapply(ncomptest, function(k) score(x%*%fit$coefficients[ , ,k], yp))
permscore<- abs(sco)
      PermSCO[jth]<- abs(sco)[1]
#permscore<- sco*sqrt( (nrow(x)-2)/(1-sco^2) ) #t-distribution with df of n-2.
counts<- counts + (permscore >= testscore)
}#while
permpvalue<- (1+ counts)/(1 + jth)
}#if
permpvalueGPD<- GPD(PermSCO,obsSCO,250)
list(score = testscore, nonsignificant.counts = counts,
pvalue.empirical = permpvalue,
 pvalue.empiricalGPD = permpvalueGPD, pvalue.nominal = testpvalue,
 total.permutation = jth,
  ncomp.varp = ncompvar,PermSCO=PermSCO)
}
#MPLSGPD(x ,y, scale = FALSE, ncomp=2, npermutation.max=10)
```

A.7 Discretisation

Cette fonction permet de transformer un vecteur qui a des valeurs continues en un vecteur qui a des valeurs discrètes (0, 1, 2). Les valeurs vont donc dépendre de la fréquence fixée au départ de l'allèle mineur.

```
######################Discretisation############################
#Fréquece allélique de l'allèle mineur a: Pa
#X est un vecteur
```

```
Discrete<- function(X, Pa ){
 X <-as.matrix(X,ncol=1)
X<- scale(X, center=TRUE,   scale = TRUE)
PA <- 1-Pa
PAA <- PA*PA # fréquence du génotype    AA
Paa <- Pa*Pa  # fréquence du génotype  aa
PAa <- 2* Pa*PA  # fréquence du génotype  Aa
qnorm(Paa) #represente le percentile de Paa pour  la loi normale
#centrée réduite
qnorm(PAA) #represente le percentile de PAA pour  la loi normale
#centrée réduite
DisX <- c(rep(0,   length(X)))
for(i in 1:length(X)){
if (X[i]<= qnorm(Paa)) {DisX[i]=0}
else {if ((X[i] > qnorm(Paa))&(X[i] <= qnorm(PAA))) {DisX[i]=1}
else {DisX[i]=2}
}
}
return(DisX)
#DisX est le vecteur discrétisé de X
}
```

Tableau A.1 Tableau donnant les SNPs des fenêtres 701 et 702 et les gènes correspondants.

	fenêtre701	Genes701	fenêtre702	Genes702
1	rs3852856_A	PVRL2	rs3729640_T	PVRL2
2	rs519825_G	PVRL2	rs6859_A	PVRL2
3	rs8105340_C	PVRL2	rs283814_T	PVRL2
4	rs12610605_A	PVRL2	rs157580_G	TOMM40
5	rs4803766_A	PVRL2	rs2075650_G	TOMM40
6	rs8104483_G	PVRL2	rs157582_A	TOMM40
7	rs11879589_A	PVRL2	rs8106922_G	TOMM40
8	rs395908_T	PVRL2	rs1160985_T	TOMM40
9	rs2075642_A	PVRL2	rs405509_C	APOE
10	rs387976_G	PVRL2	rs769449_A	APOE
11	rs3729640_T	PVRL2	rs439401_T	-
12	rs6859_A	PVRL2	rs445925_T	APOC1
13	rs283814_T	PVRL2	rs1064725_G	APOC1
14	rs157580_G	TOMM40	rs5157_T	APOC4-APOC2
15	rs2075650_G	TOMM40	rs1132899_T	APOC4-APOC2
16	rs157582_A	TOMM40	rs5167_G	APOC4-APOC2
17	rs8106922_G	TOMM40	rs2288911_A	APOC4-APOC2
18	rs1160985_T	TOMM40	rs10413089_C	LOC105372418
19	rs405509_C	APOE	rs3760627_C	CLPTM1
20	rs769449_A	APOE	rs204905_C	CLPTM1

RÉFÉRENCES

Abdi, H. (2007). RV coefficient and congruence coefficient. *Encyclopedia of measurement and statistics*, 849-853.

Abdi, H. (2010). Partial least squares regression and projection on latent structure regression (PLS Regression). *Wiley Interdisciplinary Reviews : Computational Statistics*, 2(1), 97-106.

Alzheimer's Disease Neuroimaging Initiative. (2016). Récupéré de http ://adni.loni.usc.edu/study-design.

Mevik, B. H. et Wehrens, R. (2007). The pls package : principal component and partial least squares regression in R. *Journal of Statistical software*, *18*(2), 1-24.

Cunningham, E. (2014).*Estimating and correcting optimism bias in multivariate Partial Least Squares (PLS) regression.*
Application to the study of the association between Single Nucleotide Polymorphisms (SNPs) and multivariate traits in Attention Deficit Hyperactivity Disorder (ADHD) :rapport de stage. [Document non puplié]. Université de McGill.

Gang, Z. Yanning, Y. Xiaofeng, Z. et Robert, C. E. (2012). *Analysis of Genetic Association Studies.* Springer-Verlag New-York.

Hastie, T. Tibshirani, R. et Friedman, J. (2008). *The Elements of Statistical Learning Data Mining, Inference, and Prediction* (2e éd.). Springer Series in Statistics.

Hoerl, A. E. et Kennard, R. W. (1970). Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55-67.

Hollingworth, P. Harold, D. Jones, L. Owen, M. J. et Williams, J. (2011). *Alzheimer's disease genetics : current knowledge and future challenges.* International journal of geriatric psychiatry, *26*(8), 793-802.

Ildiko, E. F. et Friedman, J.H. (1993). Reviewed : A Statistical View of Some Chemometrics Regression Tools. *Technometrics*, *35*(2), 109-135.

Knijnenburg, T. A. Lodewyk, F. A. Wessels, Marcel, J. T. Reinders et Ilya Shmulevich, (2009). Fewer permutations, more accurate P-values. *Bioinformatics, 25*, i161–i168.

Lin, J. A. Zhu, H. Knickmeyer, R. Styner, M. Gilmore, J. et Ibrahim, J. G. (2012). Projection regression models for multivariate imaging phenotype. *Genetic epidemiology, 36*(6), 631-641.

Logue, M. W. Schu, M. Vardarajan, B. N. Buros, J. Green, R. C. Go, R. C. P. ... Farrer, L. A. (2011). A Comprehensive Genetic Association Study of Alzheimer Disease in African Americans. *Archives of Neurology, 68*(12), 1569–1579.

MandelSource, J. (1982).Use of the Singular Value Decomposition in Regression Analysis. *The American Statistician, 36*(1), 15-24.

Rencher, A. C. (2002). *Methods of Multivariate Analysis Second Edition*, A John Wiley & sons, INC. Publication.

Robert, P. et Escoufier, Y. (1976). A unifing tool for linear multivariate statistical methods : The RV-coefficent. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 25*(3), 257-265.

Robert, S. (2010). *La méthode PLS à travers quelques application, modifications et extensions.* Récupéré de www-irma.u-strasbg.fr/~fbertran/PLS_17122010_Sabatier.pdf.

Société Alzheimer. (2014 -). *La génétique et la maladie de l'alzheimer.* Récupéré de http ://www.alzheimer.ca//media/Files/national/Research/understanding_genetics_f.pdf.

Stéphane, T. (2012). *Data Mining et Statistique décisionnelle l'intelligence des données Quatrième édition actualisée et augmentée*, Editions Technip.

Université McGill.[s.d]. *Le cerveau à tous les niveaux.* Récupéré de http ://lecerveau.mcgill.ca/flash/d/d_08/d_08_p/d_08_p_alz/d_08_p_alz.-html.

University of California. (2013 -). *Alzheimer's Disease Neuroimaging Initiative.* Récupéré de // http ://www.adni-info.org.

Wold, H. (1966). Estimation of Principal Components and Related Models by Iterative Least Squares. Dans P.R. Krishnaiah (dir). *Multivariate Analysis.* New York : Academic Press (p. 391-420).

Xu, C. Ladouceur, M. Dastani, Z. Richards, J. B. Ciampi, A. Greenwood, C. M. T. (2012). Multiple Regression Methods Show Great Potential for Rare

Variant Association Tests. (2012). *PLoS ONE* 7(8) : e41694. doi :10.1371/journal.pone.0041694.

Xu, C. et Greenwood, C. M. (2013). *Rare variant Tests*. Récupéré de http ://cran.r-project.org/web/packages/RVtests/RVtests.pdf.