Research and development of innovative mathematical algorithms using cluster-based interactions of metagenomic data in biomedicine

Camille Champion

▶ **To cite this version:**

Camille Champion. Research and development of innovative mathematical algorithms using cluster-based interactions of metagenomic data in biomedicine. Statistics [math.ST]. INSA de Toulouse, 2021. English. NNT : 2021ISAT0005 . tel-03395321v2

HAL Id: tel-03395321

https://theses.hal.science/tel-03395321v2

Submitted on 22 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)*

Présentée et soutenue le *28/06/2021* **par :**

Camille CHAMPION

Développement et étude mathématique d'algorithmes d'analyse en clusters d'interactions de données métagénomiques en biomédecine

---

JURY

RÉMY BURCELIN	Professeur, INSERM	Directeur
ANTOINE CHAMBAZ	Professeur, Université de Paris	Président du Jury
ADELINE LECLERCQ SAMSON	Professeure, Université Grenoble Alpes	Rapporteur
JEAN-MICHEL LOUBES	Professeur, Université Paul Sabatier	Directeur
MATHILDE MOUGEOT	Professeure, ENSIIE	Rapporteur
LAURENT RISSER	Ingénieur de Recherche, CNRS	Membre du Jury

École doctorale et spécialité :
MITT : Domaine Mathématiques : Mathématiques appliquées

Unité de Recherche :
Institut de Mathématiques de Toulouse (UMR 5219)

Directeur(s) de Thèse :
Jean-Michel LOUBES et *Rémy BURCELIN*

Rapporteurs :
Adeline LECLERCQ SAMSON et *Mathilde MOUGEOT*

Remerciements

Il est venu le temps des remerciements,
Moment très important,
Pour saluer tous ceux qui m'ont aidée,
Ces quatre dernières années.

C'est le jour 1, celui qu'on retient,
Rythmé de va-et-vient,
Entre secrétariat et premier bureau,
Pour régler les papiers du labo.

Merci, merci, je vous dis merci,
A tous ceux qui m'ont accueillie,
INSA, I2MC et IMT,
Pas facile de s'y retrouver.

S'il suffisait d'un mot pour décrire cette expérience,
Je dirais patience et confiance,
Sans oublier travail et rigueur,
Pour être sûre de terminer à l'heure.

Être à la hauteur,
Faire comme mes deux sœurs,
Je l'ai souvent répété,
Pour terminer cette thèse en beauté.

J'ai écrit l'histoire,
De mes travaux de thèse dans un mémoire,
J'en remercie les membres du jury tous réunis,
Pour me permettre de finir cette thèse ainsi.

Mes chers encadrants je pars,
Je vous remercie mais je pars,
J'ai beaucoup appris à vos côtés,
Durant ces quatre dernières années.

Mais si je m'en sors aujourd'hui,
C'est grâce à mes premiers amis,
Ashley, Céline et Yamina,
Puis Paulin, Émilie et Anissa.

Formidables, vous êtes formidables,
José et Trang, une rencontre incroyable,
Que vous soyez au Vietnam, en Colombie,
Vous resterez toujours mes amis.

Évidemment, je n'oublie pas,
Les collègues de l'INSA,
Mélisande, Cathy, Sandrine et Olivier,
Vous m'avez donné l'envie d'enseigner.

On ne m'a pas laissé le temps,
Covid oblige, éloignés pendant longtemps,
De vous remercier l'I2MC,
Pour votre hospitalité.

Comment puis-je oublier,
Les JDS de Saclay,
Qui m'ont permis de rencontrer,
Anouar et toute sa bonté.

Donnez-moi le temps de vous présenter,
Ce qui depuis ces 21 dernières années,
Fait partie intégrante de ma vie,
Et celle de ma fratrie.

J'adore, patiner, entraîner, chorégraphier,
Mais surtout partager,
Jérémy, Cléia et Emmie ont tout donné,
Pour finir tous médaillés.

Je veux, d'l'amour, d'la joie,
Sur la piste il faut que cela se voit,
Eloïse tu t'en sors vraiment bien,
Continue, tu es sur le bon chemin.

L'important c'est d'y croire,
La famille doit s'en apercevoir,
Du soutien je le trouve chez les miens,
On est comme les 5 doigts d'une main.

Petit papa Champion,
Te pose pas trop d'questions,
Ta tête pourrait faire boum,
Ce pourrait bien être la Shkoumoune.

Si maman si,
Je vais bien c'est promis,
Mais ne fais pas la tête,
Je reviendrai dans quelques jours peut être.

Si j'ai besoin d'amour,
Julie panda me fera toujours,
Des bisous, des câlins il suffit de demander,
Elle en donne par milliers.

On joue de la musique,
Avec Magali, quel orchestre symphonique,
Et quand Mickael, Monica, Gracy sont de sortie,
Un grand ballet on aurait dit.

Allez venez chez nous,
On parle vraiment de tout,
L'allemand on a reparlé,
Depuis que gros groot est arrivé.

Un beau jour, ou peut-être une nuit,
Durant cette thèse, on est tous partis,
A La Rochelle toute la famille est venue,
Du courage il nous en a fallu.

Une photo, une larme, c'est à n'y pas croire,
De la tristesse on pouvait voir,
Mais la famille était là,
Pour avancer à petits pas.

J'vous l'dis quand même fière,
Ce n'est pas dans mon habitude première,
Alain, Maryvonne, Cocos, Gigi,
D'avoir été à mes côtés, je vous remercie.

Ah jamais on a vu, jamais on ne nous verra,
Danser, chanter avec un panier au bras,
La boiteuse on s'en souviendra,
Evelyne, Michel on a vraiment ri ce soir là.

Oh Mamie, oh Mamie B et C,
Je ne vous oublierai jamais,
Vous m'avez tant apporté,
Toutes ces années à vos côtés.

Libérée, délivrée, j'avais promis de la chanter,
Fini la chanson des enfoirés,
Comme vous l'aurez compris,
La chanson a une grande place dans ma vie.

Non je ne regrette rien,
Cette thèse prend vraiment fin,
Pour laisser place à l'écriture,
D'une toute nouvelle aventure.

Table des matières

1 Introduction Générale **1**

1 Contexte biologique . . . 1
1.1 Les maladies métaboliques et cardiovasculaires . . . 1
1.2 Les données . . . 2
1.2.1 Génomiques . . . 2
1.2.2 Transcriptomiques . . . 3
1.2.3 Métagénomiques . . . 3
2 Modélisation statistique des données . . . 5
2.1 Caractéristiques des données . . . 5
2.2 Transformation et normalisation des données métagénomiques . . . 6
2.3 Analyse de la diversité bactérienne . . . 7
3 Méthodes d'apprentissage statistique . . . 8
3.1 Méthodes d'apprentissage non supervisé . . . 9
3.1.1 Analyse en Composantes Principales . . . 9
3.1.2 Analyse en Coordonnées Principales . . . 10
3.2 Méthodes d'apprentissage supervisé . . . 11
3.2.1 Régression Lasso . . . 11
3.2.2 Régression des Moindres Carrés Partiels . . . 12
3.2.3 Analyse Discriminante des Moindres Carrés Partiels . . . 13
4 Modélisation et méthodes de classification des données sous forme de réseaux . . . 14
4.1 Introduction à la théorie des graphes . . . 14
4.1.1 Caractéristiques d'un graphe . . . 14
4.1.2 Représentation d'un graphe . . . 15
4.2 Analyse de réseaux . . . 16
4.2.1 Définitions et propriétés d'un cluster . . . 17
4.2.2 Méthodes de clustering . . . 17
5 Contributions de cette thèse . . . 18
5.1 CORE-clustering et détection de variables représentatives en grande dimension 19
5.2 Clustering de graphes par spectral clustering pénalisé . . . 20
5.3 Découverte de signatures bactériennes chez des patients fibrotiques . . . 21
5.4 Méthode fair de Régression des Moindres Carrés Partiels . . . 22

2 CORE-clustering **25**
1 Introduction . . . 25
2 Statistical Methodology . . . 27
2.1 Graph-based representation of the observations . . . 27
2.2 Coherence of a variable set . . . 27
2.3 CORE-clusters . . . 28
2.4 CORE-clustering . . . 29
2.5 Representative variables selection . . . 29
2.6 General guidelines for the choice of ξ and τ . . . 30
3 Computational Methodology . . . 30
3.1 Main interactions estimation . . . 30
3.2 CORE-clustering algorithms . . . 31
3.2.1 Maximum Spanning Tree . . . 31
3.2.2 CORE-clustering algorithm . . . 32
3.2.3 A greedy alternative for CORE-clusters detection . . . 33
3.3 Central variables selection in CORE-clusters . . . 33
4 Results . . . 35
4.1 Core clustering of simulated networks . . . 35
4.1.1 Experimental protocol . . . 35
4.1.2 Measure of clustering quality . . . 35
4.1.3 Results . . . 36
4.2 Application to real biological data . . . 36
4.2.1 Yeast dataset . . . 36
4.2.2 Comparison of the two CORE-clustering algorithms . . . 37
4.2.3 Impact of the number of observations . . . 39
4.2.4 Comparison with spectral-clustering . . . 39
4.3 Application to the U.S. road network . . . 39
5 Conclusion . . . 40
6 Appendices . . . 42
6.1 Pertinence of the representative variable detection model . . . 42
6.1.1 Illustration of the coherence . . . 43
6.1.2 Influence of the undesirable relations . . . 44

3 ℓ_1-spectral clustering **47**
1 Introduction . . . 47
2 Reminders about graph and spectral clustering . . . 49
2.1 Graphs modeling and notations . . . 49
2.2 Graph clustering through spectral clustering . . . 50
3 An ℓ_1-version of the spectral clustering algorithm . . . 51
3.1 General ℓ_0-minimization problem . . . 51
3.2 Relaxed ℓ_1-minimization problem . . . 52
3.3 Generalization of the relaxed ℓ_1-minimization problem . . . 53
4 The ℓ_1-spectral algorithm . . . 54

4.1 Global overview of the algorithm . . . 54
4.2 Solving the ℓ_1-minimization problem . . . 55
4.3 Optimally tuning the number of clusters . . . 56
4.4 Finding the representative elements . . . 57
5 Numerical experiments . . . 57
5.1 Application to toy datasets . . . 58
5.1.1 Numerical settings . . . 58
5.1.2 Effect of the dimension and cluster sizes on perturbed graphs . . . 59
5.1.3 Performance results with respect to state-of-the-art . . . 60
5.2 Application to cancer data . . . 62
5.2.1 The kidney cancer data set . . . 62
5.2.2 ℓ_1-spectral clustering algorithm on kidney cancer data . . . 62
5.2.3 Clusters as hallmarks of kidney cancer . . . 63
6 Conclusion . . . 65

4 Application to metagenomic data **67**
1 Introduction . . . 67
2 Material and Methods . . . 68
2.1 Subjects and Ethics . . . 68
2.2 Liver biopsies and liver fibrosis diagnosis . . . 70
2.3 Clinical assessments . . . 70
2.4 Biochemical and molecular analyses . . . 70
2.5 16S rDNA sequencing and bioinformatic analysis . . . 71
2.6 Linear Discriminant Analysis (LDA) Effective Size (LEfSe) . . . 71
2.7 Beta diversity analysis . . . 71
2.8 Multivariate analysis . . . 72
2.9 Cluster graphical analysis . . . 72
2.10 Functional metagenomic prediction . . . 72
2.11 Data Availaibility Section . . . 73
3 Results . . . 73
3.1 Graphical classification of the clinical variables by PCA . . . 73
3.2 Analyses of the liver bacterial 16S rDNA ecology . . . 73
3.3 Identification of specific bacterial signatures . . . 76
3.4 Identification of clusters of cohort-independent 16S rDNA associated with different mild scores of fibrosis . . . 89
3.5 Low frequency bacterial 16SrDNA gene contains classifying information . . . 94
3.6 Predicted functional metagenome pathways . . . 94
4 Discussion . . . 98
5 Conclusion . . . 105

5 Fair conditional Partial Least Square **107**
1 Introduction . . . 107
2 Standard PCA and PLS . . . 109
2.1 Principal Component Analysis . . . 109
2.2 Partial Least Square . . . 109
3 Fair conditional independence strategies . . . 110
3.1 Overall strategy . . . 110
3.2 Fair PCA based on correlation measure . . . 111
3.3 Fair PLS based on correlation measure . . . 112
3.4 Application to real data . . . 113
3.4.1 Florinash dataset . . . 113
3.4.2 Results . . . 113
3.4.3 Liver fibrosis dataset . . . 116
3.4.4 Results . . . 117
4 Conclusion . . . 120
5 Further developments . . . 121
5.1 PLS under fair constraint based on covariance measure . . . 121
5.2 PLS under fair constraint based on Hilbert-Schmidt Independance Criterion . 122
5.3 PLS through Wasserstein-2 distance . . . 124

6 Conclusion **125**
1 Synthesis . . . 125
2 Scientific production . . . 125
2.1 Published papers . . . 125
2.2 Submitted papers . . . 126
2.3 R packages . . . 126

Chapitre 1

Introduction Générale

1 Contexte biologique

1.1 Les maladies métaboliques et cardiovasculaires

Ces dernières années, les progrès technologiques ont fortement marqué, à l'échelle mondiale, le mode de vie des populations notamment le rythme quotidien, la mobilité et la qualité d'alimentation. Ayant dû s'adapter à un environnement aussi changeant, les organismes, qui ne sont pas programmés biologiquement pour vivre dans ces nouvelles conditions sociétales, ont développé des maladies chroniques notamment métaboliques telles que l'obésité et une de ses conséquences graves, le diabète de type 2. Ce dernier est caractérisé par une augmentation de la concentration de glucose dans le sang le matin, à jeun ou suite à un repas, dû à un défaut de sécrétion et à un défaut d'action de l'insuline. Au-delà des aspects biochimiques et physiologiques, ces dérèglements métaboliques sont des facteurs de risque des maladies cardio-vasculaires, hépatiques et rénales et peuvent accentuer la sensibilité aux infections notamment virales. Ces maladies métaboliques augmentent le tonus inflammatoire de manière prolongée sur plusieurs années ce qui aboutit à des atteintes cellulaires pouvant conduire jusqu'à la mort cellulaire. Ainsi, l'élimination des cellules mortes et la réparation tissulaire dans le foie entraîne une fibrose progressive difficilement réversible. Cette fibrose peut aboutir à une insuffisance hépatique, une cirrhose et un cancer potentiel. Parmi les maladies hépatiques, nous retrouvons notamment la Non-Alcoholic Fatty Liver Disease (NAFLD), caractérisée par l'accumulation de graisses dans le foie qui, chez certains individus, peut altérer le fonctionnement des cellules hépatiques et provoquer une inflammation, la Stéatite Hépatique Non Alcoolique (NASH) (Figure 1.1).

Parmi les causes de ces évolutions pathologiques, la génétique exerce un impact majeur dans leur développement. Cependant, pour beaucoup de pathologies, de multiples facteurs liés à l'environnement entrent aussi en jeu et plus particulièrement la qualité et quantité alimentaire, la sédentarité et le stress. Les interrelations gènes/environnement à l'origine des maladies métaboliques sont d'autant plus difficiles à détecter que de nombreuses modifications moléculaires modulant l'expression des gènes peuvent interférer. Dans le cadre de l'obésité et du diabète de type 2, plusieurs études récentes ont révélé que le microbiote intestinal (l'ensemble des micro-organismes qui peuplent l'intestin) conciliait tous ces paramètres. Sous l'influence des conditions alimentaires, de la génétique et de l'environnement, la composition du microbiote intestinal évolue tout au long de la journée pour

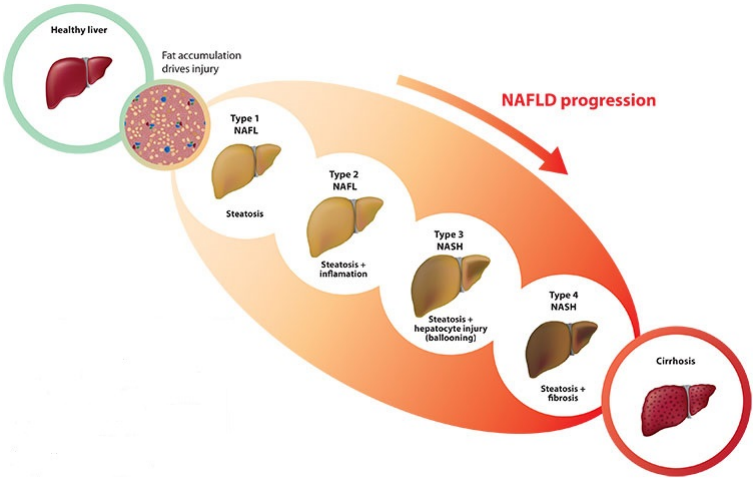

FIGURE 1.1 – Progression de la Non-Alcoholic Fatty Liver Disease (NAFLD)

effectuer des cycles stables au quotidien.

D'un point de vue statistique, afin de rendre compte de manière la plus générale possible du phénomène biologique sous-jacent, une première étape dans l'analyse des données consiste à réaliser une Analyse Exploratoire des Données (AED) cliniques, génétiques et microbiologiques. Elle peut être suivie d'analyses prédictives ou discriminantes dépendant de la nature des variables à étudier. Ceci permet à la fois d'avoir une vision globale d'un ensemble de données, d'en chercher les corrélations sans hypothèse a priori et d'étudier plus en profondeur le processus biologique en question. L'objectif est alors d'identifier des signatures biologiques, aussi appelées biomarqueurs, associés aux premiers stades de la maladie afin de poser un diagnostic précoce qui permettraient de prévenir, par une action thérapeutique, le développement de la maladie.

Dans la section suivante, les différents types de données exploités sont présentés.

1.2 Les données

1.2.1 Génomiques

Le génome représente l'ensemble du matériel génétique nécessaire au développement de l'espèce vivante. Il repose sur l'ADN, support de l'information génétique. Celui-ci est une molécule constituée de deux brins complémentaires en forme de double hélice composés d'une succession de nucléotides, comme représenté sur la figure 1.2. On trouve quatre nucléotides différents, notés A, G, C et T, du nom des bases correspondantes adénine, guanine, cytosine et thymine, dont l'ordre d'enchaînement (codes) détermine l'information génétique. Le génome joue un rôle crucial dans le fonctionnement de l'organisme en permettant la production de protéines qui assurent une multitude de fonctions au sein des cellules et des tissus. Les protéines sont des macromolécules composées d'acides aminés remplissant les fonctions vitales de la cellule qui sont aussi responsables de l'ensemble du métabolisme de l'ADN (synthèse, réplication, réparation). L'ADN détermine la synthèse des protéines par l'intermédiaire de

l'acide ribonucléique (ARN). Comme pour l'ADN, l'ARN est un support moléculaire de l'information génétique mais il est constitué d'un seul brin. L'information est elle aussi encodée par quatre bases nucléiques, la thymine (T) étant remplacée par l'uracile (U). Il existe différents types d'ARN classés selon leur fonction. Certains ARN dit codants, aussi appelés ARN messagers, portent l'information génétique codant pour des protéines. Les autres ARN interviennent dans le fonctionnement de la cellule, la régulation de l'information et l'activité cellulaire. Par des techniques biochimiques, il est possible de déterminer le code génétique (partiel ou complet) d'un organisme donné : c'est le séquençage du génome.

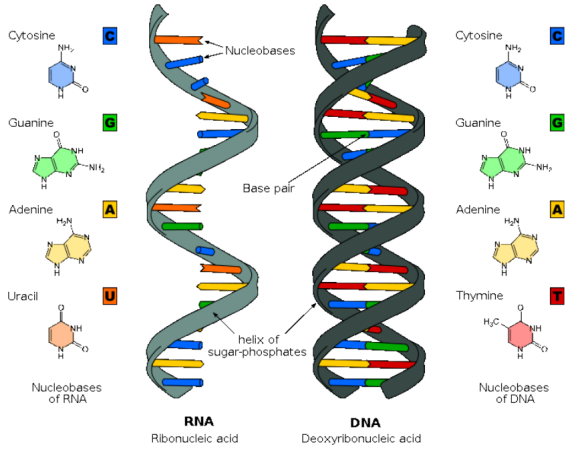

FIGURE 1.2 – Structure de la molécule d'ADN et d'ARN

1.2.2 Transcriptomiques

Le transcriptome est l'ensemble des ARN produits par transcription de l'ADN. Sa caractérisation et quantification dans une cellule donnée se fait par le biais de puces à ADN qui intéragissent avec les molécules d'ARN ou directement par séquençage de l'ensemble des molécules d'ARN. Ces techniques, très largement utilisées, permettent notamment d'identifier les gènes actifs, de déterminer les mécanismes de régulation d'expression des gènes et de définir des réseaux d'expression des gènes.

1.2.3 Métagénomiques

La métagénomique est la méthode d'étude des métagénomes, c'est à dire de l'information génétique (le génome) de l'ensemble des micro-organismes vivant au sein d'un environnement spécifique. Les micro-organismes regroupent l'ensemble des êtres vivants microscopiques, comportant les virus, parasites, champignons microscopiques, levures et bactéries. Cet ensemble d'organismes, le microbiote, est codé par le génome correspondant, appelé microbiome, c'est-à-dire l'ensemble des gènes du microbiote. La génomique est souvent, et par défaut, l'étude du génome des eucaryotes (organismes supérieurs complexes). La métagénomique englobe les génomes des microorganismes dans un milieu donné. Chez l'homme et la souris de laboratoire, il s'agit souvent des génomes bactériens.

Chaque bactérie a son propre génome, dont une partie très variable de l'information est partagée avec les autres bactéries d'un milieu donné. La mise en commun de l'ensemble des génomes, notamment bactérien, est appelé métagénome, même s'il s'agit de plusieurs génomes différents mais dont une partie de l'information peut être commune. Une analyse typique métagénomique bactérienne donne la composition de la taxonomie d'un microbiome bactérien c'est-à-dire les taxa notamment bactériens présents, leur abondance ainsi que leur diversité. Il existe deux grandes stratégies de séquençage en métagénomique : la stratégie globale (Figure 1.3 **a)**) et la stratégie ciblée (Figure 1.3 **b)**).

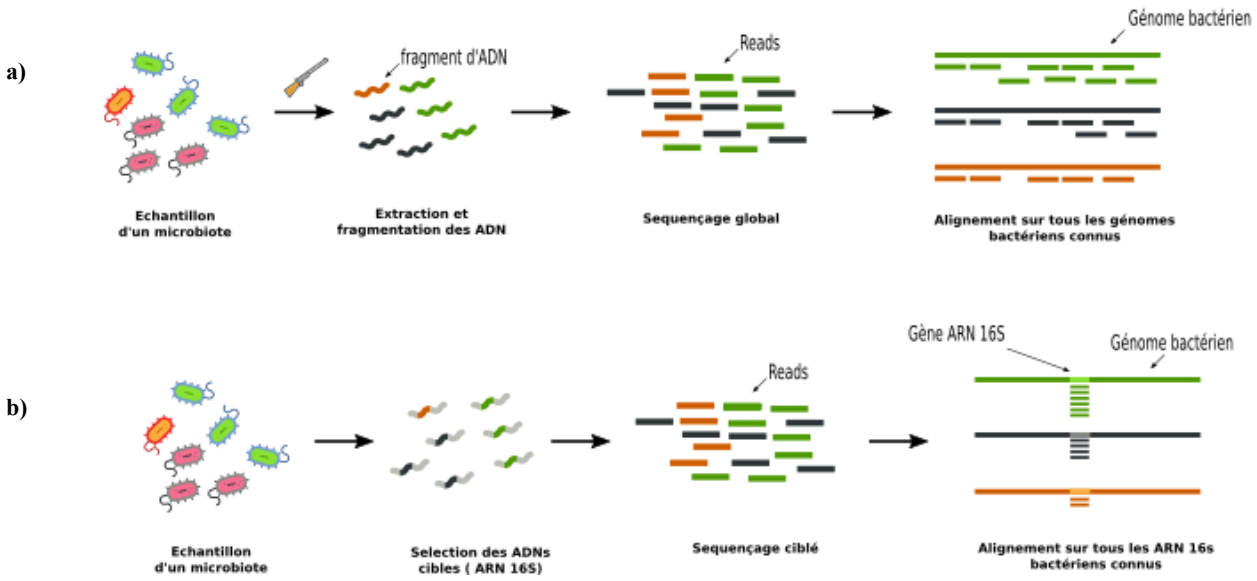

FIGURE 1.3 – Stratégie globale a) et ciblée b) de séquençage en métagénomique.

La métagénomique globale permet d'extraire, fragmenter et déterminer le code ADN de l'ensemble des génomes des micro-organismes présents dans le milieu étudié (la plupart du temps et par défaut bactérien). Les séquences obtenues, aussi appelées reads sont ré-assemblées bioinformatiquement afin de reconstruire les génomes bactériens d'origine. La métagénomique ciblée consiste à étudier un unique gène au lieu du génome complet. Ce gène doit être commun à plusieurs espèces tout en présentant des régions suffisamment variables afin de discriminer une espèce. En bactériologie, par exemple, le gène utilisé est celui de l'ARN ribosomique 16S. Ce gène possède des régions constantes (donc similaires) entre plus de $80 - 90\%$ des bactéries et d'autres régions variables très spécifiques d'une espèce bactérienne donnée. Les régions variables sur ce gène n'ont pas de rôle fonctionnel important et peuvent diverger au cours de l'évolution sous l'effet de mutations neutres. C'est ce qui permet de classer les microorganismes en taxons (famille, genre, espèce), sur la base de distances génétiques (Figure 1.4).

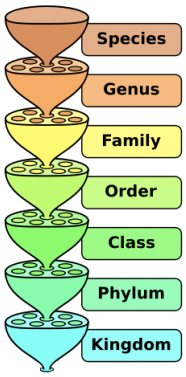

FIGURE 1.4 – Taxonomie des bactéries

Une fois le séquençage réalisé, à chaque read doit être assigné le nom d'une bactérie, on parle d'assignation taxonomique. En particulier, les séquences de nucléotides sont regroupées avec un seuil général de 97% de similarité dans des Unités Taxonomiques Opérationnelles (OTUs). Ces dernières sont ensuite comparées à des séquences de données de référence contenues dans des bases de données publiques pour identifier la taxonomie des bactéries en cause. Une fois la liste des taxons identifiée, leur abondance est estimée en fonction du nombre de séquences alignées sur leur référence. Le résultat final est représenté sous la forme d'un tableau de comptage à double entrée contenant le nombre de séquences par OTU et par échantillon.

2 Modélisation statistique des données

2.1 Caractéristiques des données

On considère un système composé de p variables quantitatives $X = (X^1, \dots, X^p)$ et n observations $(X_1^j, \dots, X_n^j)_{j\in\{1,\dots,p\}}$ de ces p variables. Dans cette thèse, nous étudierons deux types de données : les données transcriptomiques, dont la littérature dans le domaine statistique est très riche et les données métagénomiques, encore assez peu connues, qui nécessitent des développements particuliers dus à leur nature très différente. Les données transcriptomiques se présentent sous la forme d'un tableau à double entrée représentant l'activité génomique, c'est-à-dire son expression en ARNm (ou quantité d'ARN messagers produite) de chaque gène par échantillon. Le nombre de gènes dans un tissu donné chez une centaine d'individus pouvant atteindre jusqu'à plusieurs dizaines de milliers, les données transcriptomiques sont caractérisées par la grande dimension ($p >> n$). Les données microbiologiques se présentent quant à elles sous la forme d'une matrice de comptage correspondant au nombre de reads de chaque espèce bactérienne observée par échantillon. Cette table de comptage présente plusieurs caractéristiques singulières. Il s'agit d'une matrice :

— creuse : elle contient un grand nombre de comptages nuls, pouvant varier de 70 à 90% ce qui peut être interprété comme l'absence ou la présence non détectée d'un OTU (échantillonnage trop faible),
— de grande dimension dans le sens où $p >> n$,

— dont les variables sont dépendantes en raison des interactions écologiques entre les espèces ou à l'appartenance à une même entité biologique,
— avec une grande dispersion : variance d'un OTU généralement plus élevée que sa moyenne.

2.2 Transformation et normalisation des données métagénomiques

Les données de comptage sont des données très hétérogènes et de distribution asymétrique. Des modèles basés sur des lois discrètes comme la loi de Poisson ou la loi binomiale négative ont été proposés afin de les simuler le plus correctement possible (Gary and Bennetts, 1996; Linden and Mäntyniemi, 2011). L'analyse statistique du jeu de données non transformé peut ne pas révéler l'information sous-jacente du fait de ses nombreuses caractéristiques singulières. Certains outils statistiques nécessitent une transformation des données afin de pouvoir être utilisés ou optimisés. C'est le cas notamment de la régression linéaire pour laquelle l'hypothèse de normalité est primordiale. D'autres outils comme l'Analyse en Composantes Principales présentent une certaine sensibilité aux trop grandes dispersions des données. Les méthodes de transformation des données se déclinent en deux grandes familles :

— la standardisation ou l'action de centrer-réduire les données pour diminuer la dispersion des données,
— la normalisation des données qui consiste à rechercher la transformation adaptée au travers de laquelle elles suivront une loi normale.

Parmi les normalisations les plus utilisées, on trouve la transformation log suivante :

$$\forall i \in [\![1, n]\!],\ \tilde{X}_i = log(X_i + 1).$$

En revanche, celle-ci ne permet pas de stabiliser la variabilité du jeu de données. En effet, le nombre de lectures obtenues de chaque séquence, aussi appelé profondeur des séquences, peut varier d'un échantillon à l'autre, notamment à cause d'imprécisions techniques et technologiques. La comparaison de deux échantillons de profondeurs différentes risque alors de donner des écarts de proportions de gènes non nécessairement liés à une réelle distinction biologique. Pour les rendre comparables, la normalisation classique utilisée est la normalisation Total Sum Scaling (TSS) qui divise chaque comptage par le nombre total de variables observées par échantillon :

$$\forall i \in [\![1, n]\!], \quad \tilde{X}_i = \frac{X_i}{\sum_{j=1}^{p} X_i^j}.$$

La matrice de comptage est ainsi convertie en matrice de proportion aussi appelée matrice d'abondance. La normalisation TSS engendre cependant un biais lors de l'estimation des abondances différentielles. Afin de réduire l'influence des OTUs sur-représentés dans une matrice de type creuse, Paulson et al. (2013) ont développé une stratégie alternative, la normalisation par somme cumulée (CSS). Celle-ci consiste à remettre à l'échelle les OTUs en fonction d'un sous-ensemble (quartile) de taxons à faible abondance (relativement constants et indépendants), réduisant ainsi l'impact des taxons à forte abondance (dominants). Dans cette thèse, cette dernière méthode sera appliquée aux données de comptage log-normalisées avant toute analyse statistique.

2.3 Analyse de la diversité bactérienne

Afin de contrôler l'homogénéité des échantillons en terme de composition taxonomique, une étude de la diversité bactérienne est très souvent réalisée. Whittaker (1972) a introduit les deux types d'études suivantes :

— l'étude de la diversité α, dite locale, utilisée pour mesurer la diversité au sein d'un unique échantillon,

— l'étude de la diversité β, utilisée pour estimer la diversité des espèces bactériennes entre les échantillons.

La diversité α est mesurée à partir d'indices parmi lesquels : l'indice richesse qui mesure le nombre d'entités bactériennes (taxons) distinctes présentes dans l'échantillon mais donne parfois trop d'importance aux espèces rares. L'indice de Shannon (1948), ou entropie relative permet par contre de prendre en compte à la fois le nombre d'espèces dans milieu et la répartition des individus au sein de ces espèces. Il est calculé comme suit :

$$\forall i \in [\![1,n]\!],\ \ H_i = -\sum_{j=1}^{p} \frac{X_i^j}{\sum_{j=1}^{p} X_i^j} log\left(\frac{X_i^j}{\sum_{j=1}^{p} X_i^j}\right).$$

Un autre indice très utilisé est l'indice de Simpson (1949) qui mesure la probabilité que deux individus sélectionnés au hasard appartiennent à la même espèce :

$$\forall i \in [\![1,n]\!],\ \ S_i = \frac{\sum_{j=1}^{p} X_i^j(X_i^j-1)}{\left(\sum_{j=1}^{p} X_i^j\right)\left(\sum_{j=1}^{p} X_i^j-1\right)}.$$

et ainsi donne plus de poids aux espèces abondantes qu'aux espèces rares.

Finalement, un dernier indice utilisé est l'indice Chao (1984) qui donne plus de poids aux faibles abondances (présentes seulement une ou deux fois) pour estimer le nombre d'espèces non observées et est calculé comme suit :

$$\forall i \in [\![1,n]\!],\ \ C_i = p + \frac{F_1(F_1-1)}{2(F_2+1)},$$

où F_1 respectivement F_2 correspondent au nombre d'espèces représentées une, respectivement deux fois chez chacun des individus.

Les différents indices présentés ci-dessus reflètent chacun des propriétés spécifiques de la diversité taxonomique. L'analyse de la diversité α se base donc sur la combinaison de tous ces indices.

La diversité β est quant à elle quantifiée à partir de mesures de distance permettant de comparer les échantillons deux à deux et de leur attribuer une valeur résumant leur ressemblance globale. Notons $\Omega = \{1,\dots,n\}$ l'ensemble des individus. On se propose alors de définir sur $\Omega \times \Omega$ différentes mesures d'éloignement entre deux individus. On introduit les notions de dissimilarité et de similarité suivantes :

Définition 1 *Une dissimilarité d est une application de Ω^2 dans $\mathbb{R}_+$ vérifiant pour tout $(i,j) \in \Omega^2$*

— $d(i,j) = d(j,i)$,

— $d(i,j) = 0 \Leftrightarrow i = j$,
— $i \neq j,\ d(i,j) \geq 0$.

Définition 2 *Une similarité s est une application de Ω^2 dans $\mathbb{R}_+$ vérifiant pour tout $(i,j) \in \Omega^2$*
— $s(i,j) = s(j,i)$,
— $s(i,i) = S > 0$,
— $s(i,j) \leq S$.

Une matrice de similarité peut se transformer en matrice de distance à l'aide de la transformation :

$$\forall (i,j) \in \Omega^2,\ d(i,j) = (s(i,i) + s(j,j) - 2s(i,j)) - 1/2.$$

Les mesures de distance les plus utilisées pour estimer la diversité β sont les distances de Bray and Curtis (1957) et de Jaccard (1901). La distance de Bray-Curtis ou indice de dissimilarité de Bray-Curtis est utilisée pour évaluer la dissimilarité entre deux échantillons donnés, en terme d'abondance de taxons (niveau OTU, phylum, famille,...) présents dans chacun de ces échantillons. Elle est définie par :

$$\forall i \in [\![1,n]\!],\ \forall j \in [\![1,n]\!],\ BC_{i,j} = \frac{\sum_{k=1}^{p} |X_i^k - X_j^k|}{\sum_{k=1}^{p} (X_i^k + X_j^k)}.$$

La dissimilarité de Jaccard traite les espèces rares et abondantes de façon égale en comparant uniquement le nombre d'espèces en commun entre les échantillons. Elle est définie par :

$$\forall i \in [\![1,n]\!],\ \forall j \in [\![1,n]\!],\ JC_{i,j} = \frac{\sum_{k=1}^{p} \left(\mathbb{1}_{\left\{X_i^k > 0,\ X_j^k = 0\right\}} + \mathbb{1}_{\left\{X_j^k > 0,\ X_i^k = 0\right\}} \right)}{\sum_{k=1}^{p} \mathbb{1}_{\left\{X_i^k + X_j^k > 0\right\}}}.$$

L'étude couplée de la diversité bactérienne α et β permet d'obtenir des informations sur la composition et la diversité de la communauté microbienne étudiée. Elle ne permet cependant pas de décrire complètement l'effet de certains facteurs sur le microbiote et les interactions qui s'y opèrent.

3 Méthodes d'apprentissage statistique

Lorsqu'un système biologique est trop complexe pour être modélisé de manière déterministe et précise, l'apprentissage statistique propose un ensemble de méthodes et algorithmes pour décrire au mieux son comportement à partir d'une série d'observations. On distingue usuellement deux types de problèmes d'apprentissage : le non supervisé et le supervisé (Vapnik, 1998; Bishop, 2006; Hastie et al., 2001), décrits dans les sections suivantes.

3.1 Méthodes d'apprentissage non supervisé

On suppose que l'on a un tableau de données représenté par une matrice $X = (X_i^j)_{i=1,\dots,n, j=1\dots,p}$ où pour tout $i \in \{1,\dots,n\}$, $j \in \{1,\dots,p\}$,

— la i-ième ligne $X_i := (X_i^1,\dots,X_i^p)$ désigne les observations relatives à l'individu i,

— la j-ième colonne $X^j := {}^t(X_1^j,\dots,X_n^j)$ désigne les observations relatives à la variable j.

En apprentissage supervisé, les observations sont données sous la forme d'une entrée X sans aucune valeur cible Y. L'objectif est alors d'apprendre un modèle capable d'extraire les régularités présentes au sein des observations pour mieux visualiser ou appréhender la structure de l'ensemble des données. Un des principaux défis de la théorie de l'apprentissage statistique est posé par la grande dimension des données (la taille n de l'échantillon négligeable devant le nombre p de variables observées). Pour pallier à ce problème, la réduction de dimension se pose alors comme une étape importante avec des méthodes du type Analyse en Composantes Principales ou encore Analyse en Coordonnées Principales présentées dans les sous-sections suivantes.

3.1.1 Analyse en Composantes Principales

L'Analyse en Composantes Principales (ACP) est une méthode de réduction de dimension très classique pour l'étude exploratoire de tables de données en grande dimension (Pearson, 1901). Elle est basée sur une transformation orthogonale qui convertit une famille de variables quantitatives en de nouvelles variables décorrélées, appelées composantes principales de dimension plus petite. Pour cela, un ensemble de k vecteurs $(w_1,\dots,w_k)$, appelés facteurs principaux, sont tout d'abord créés et ce, de manière successive de telle sorte à restituer le maximum de variance des données originales.

Munissons $\mathbb{R}^p$ de la norme euclidienne et du produit scalaire euclidien. Alors, les facteurs principaux $(w_h)_{h\in\{1,\dots,k\}}$ sont définis comme solutions des problèmes d'optimisation (1.1) suivants :

$$\forall h \in \{1,\dots,k\}\,,\ w_h = \underset{w\in\mathbb{R}^p}{\arg\max}\ \mathrm{Var}(Xw), \tag{1.1}$$

sous la contrainte $\|w_h\| = 1$ et les contraintes d'orthogonalité $\forall l \in [\![1, h-1]\!],\ {}^t w_h^t X X w_l = 0$.

Un ensemble de k composantes principales notées $(T^1,\dots,T^k)$ et définies comme des combinaisons linéaires des X^j centrés sont ensuite créées :

$$\forall h \in \{1,\dots,k\}\,,\ T^h = Xw_h.$$

Notons que ces composantes principales ont la caractéristique d'être orthogonales, gardant le maximum de variance des données originales et ordonnées de la plus informative T^1 à la moins informative T^k. Ces nouvelles variables définissent des plans factoriels servant de base à une représentation graphique plane des variables initiales. L'interprétation de résultats se restreint généralement aux deux premiers plans factoriels, sous réserve que ceux-ci expliquent la majeure partie de la variance du nuage des variables initiales (Figure 1.5).

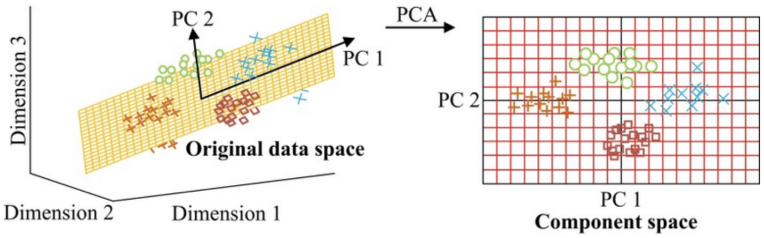

FIGURE 1.5 – Analyse en Composantes Principales

3.1.2 Analyse en Coordonnées Principales

L'Analyse en Coordonnées Principales (PCoA) aussi appelée Positionnement Multi-Dimensionnel (MDS), a pour objectif d'explorer les similarités existant entre les observations $X_i := (X_i^1, \ldots, X_i^p)$ d'un jeu de données (Kruskal, 1964; Cox and Cox, 2008). Cette méthode est notamment très utilisée lors de l'analyse de la diversité β de la communauté microbienne étudiée. Elle construit une représentation euclidienne des observations dans un espace de dimension $k << p$ à partir d'une matrice de dissimilarités/distances que l'on notera D, de terme général $D_{ij} = d(X_i, X_j)$ où d est une mesure de dissimilarité définie en Section 2.3. Notons alors que la PCoA prend tout son sens lorsque les observations sont inconnues ou les distances entre les individus sont non-euclidiennes.

Rappelons la définition d'une matrice euclidienne : une matrice D est euclidienne s'il existe une configuration de vecteurs $(X_1, \ldots, X_n)$ vérifiant :

$$(d(X_i, X_j))^2 = \langle X_i - X_j, X_i - X_j \rangle.$$

On introduit ensuite la matrice A de terme général :

$$\forall i, j \in [\![1, n]\!],\ A_{i,j} = a(X_i, X_j) = -\frac{d(X_i, X_j)^2}{2}.$$

Soit H la matrice de centrage, aussi appelée matrice de projection D-orthogonale, donnée par :

$$H = I_n - 1_n^T 1_n D,$$

où 1_n est le vecteur dont tous les coefficients sont égaux à 1 et I_n est la matrice identité de taille $n \times n$.

La proposition suivante permet alors de déterminer une configuration optimale euclidienne des distances originales entre les individus :

Proposition 1 *Soit D une matrice de distance et $B = HAH^T$ une matrice centrée en lignes et colonnes.*

- *Si D est une matrice de distance euclidienne de points $\{X_1, \ldots, X_n\}$, alors B se met sous la forme $B = (HX)(HX)^T$ et est appelée matrice des produits scalaires de la configuration centrée,*
- *Réciproquement, si B est positive de rang p, une configuration de vecteurs admettant B pour matrice de produits scalaires est obtenue en écrivant sa décomposition spectrale $B = U\Delta U^T$.*

Les différentes lignes de la matrice $Z = U\Delta^{1/2}$ permettent ainsi de définir p coordonnées principales (nouvelle configuration des distances), notées $z_1, \ldots, z_n \in \mathbb{R}^k$ qui minimisent une fonction de coût $S(z_1, \ldots, z_n)$, appelée stress, définie par :

$$S(z_1, \ldots, z_n) = \sum_{i \neq j} (d(X_i, X_j) - \|z_i - z_j\|)^2.$$

Remarquons que dans le cas euclidien, l'ACP et la PCoA sont directement connectées :

Proposition 2 *Soit X la matrice des données sur laquelle est appliquée une ACP. Alors celle-ci fournit les mêmes représentations graphiques que la PCoA calculée à partir de la matrice de distances de terme général $\|X_i - X_j\|$. Si T désigne la matrice des composantes principales, alors les coordonnées principales sont les $\sqrt{n}T$.*

Introduisons à présent les méthodes d'apprentissage supervisé pour de la sélection de variable.

3.2 Méthodes d'apprentissage supervisé

En apprentissage supervisé, les observations d'entrée sont données sous la forme de couples entrée-sortie (X, Y), où la sortie Y, appelée valeur cible, observée sur un même d'échantillon que l'entrée X, est à prédire. Il existe deux types de sous-problèmes en apprentissage supervisé :

— Régression : lorsque la variable cible à prédire est continue,

— Classement ou classification : lorsque la variable cible est discrète.

On suppose que Y est une observation bruitée de p variables explicatives $X^1, \ldots, X^p$ par le biais d'une fonction f inconnue :

$$Y = f(X) + \varepsilon,$$

où ε représente le bruit ou erreur de mesure. Il s'agit alors d'approximer f en commettant le moins d'erreurs possibles sur l'ensemble d'apprentissage tout en garantissant de bonnes prédictions pour des valeurs de Y non encore observées.

Un des principaux défis des méthodes d'apprentissage supervisé est aussi posé par la grande dimension, les méthodes de régression classiques présentant des limites. Pour contourner ce problème, une première solution consiste à utiliser des méthodes de régression pénalisée, une seconde possibilité est d'utiliser des méthodes réduction de dimension.

3.2.1 Régression Lasso

On se place dans le cadre particulier du modèle linéaire pour lequel la fonction f est linéaire. On dispose d'un tableau de données représenté par une matrice $X = (X_i^j)_{i=1,\ldots,n, j=1\ldots,p}$, et d'une variable aléatoire réelle à expliquer $Y =^t (Y_1, \ldots, Y_n)$. Le modèle linéaire s'écrit :

$$\forall i \in \{1, \ldots, n\}\,,\ Y_i = \beta_0 + \beta_1 X_i^1 + \cdots + \beta_p X_i^p + \varepsilon_i,$$

où pour tout $i \in [\![1, n]\!]$, les $X_i := (X_i^1, \ldots, X_i^p)$ sont des vecteurs indépendants et de même loi, les paramètres $\beta_0, \ldots, \beta_p$ des paramètres inconnus supposés constants à estimer et $(\varepsilon_i)_{1 \leq i \leq n}$ est une variable aléatoire centrée, simulant la présence de bruit.

En notant les vecteurs $\varepsilon = {}^t(\varepsilon_1, \ldots, \varepsilon_n)$, $\beta = {}^t(\beta_0, \ldots, \beta_p)$ et $X = (1_n, X^1, \ldots, X^p)$, le modèle s'écrit sous la forme matricielle :

$$Y = X\beta + \varepsilon. \tag{1.2}$$

Dans le cadre de la grande dimension, la méthode Lasso (Tibshirani, 1996) a pour objectif de sélectionner les variables les plus pertinentes. Il s'agit d'une version régularisée des moindres carrés avec une pénalité de type ℓ_1.

Définition 3 *Soit $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$. L'estimateur Lasso de β dans le modèle (1.2) est défini par :*

$$\hat{\beta}_{Lasso} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \ (\sum_{i=1}^{n} (Y_i - \sum_{j=0}^{p} X_i^j \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|).$$

où λ est un paramètre de pénalité positif permettant de calibrer le nombre de coefficients mis à 0.

Ce modèle se réécrit sous la forme :

$$\hat{\beta}_{Lasso} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \ (\|Y - X\beta\|_2^2 + \lambda \|\beta\|_1).$$

La méthode Lasso présente cependant des limites lorsque les variables étudiées sont fortement corrélées. Présentons alors une seconde méthode permettant à la fois de pallier au problème de la grande dimension et à la multicolinéarité des variables explicatives.

3.2.2 Régression des Moindres Carrés Partiels

La PLS, introduite et développée par Wold (1966) a pour objectif de rechercher un modèle de régression linéaire sur un ensemble de composantes décorrélées construites à partir de combinaisons linéaires des p variables explicatives centrées $(X^1, \ldots, X^p)$. La construction des composantes est optimisée pour que celles-ci soient les plus liées à la variable Y à prédire au sens de la covariance empirique. Elles vérifient :

$$\forall h \in \{1, \ldots, k\}, \ T^h = X w_h.$$

où les w_h, appelés facteurs principaux, vérifient les problème d'optimisation (1.3) suivants :

$$\forall h \in \{1, \ldots, k\}, \ w_h = \underset{w \in \mathbb{R}^p}{\arg\max} \, \mathrm{Cov}(Xw, Y), \tag{1.3}$$

sous la contrainte $\|w_h\| = 1$ et les contraintes d'orthogonalité $\forall l \in [\![1, h-1]\!]$, ${}^t w_h^t XXw_l = 0$.

Un total de k composantes orthogonales $T^1, \ldots, T^k$, appelées variables latentes sont ainsi créées. Il suffit ensuite d'effectuer la régression de Y sur les k variables latentes centrées. Se pose alors le problème du choix optimal du nombre de composantes k à conserver. Celui-ci est traditionnellement fixé par validation croisée. Basée sur une technique d'échantillonnage, elle consiste à diviser l'échantillon de taille n en sous-échantillons, le premier dit d'apprentissage et le second dit de validation ou de test. Le modèle est ensuite bâti sur l'échantillon d'apprentissage et validé sur l'échantillon de test avec un score de performance à définir (voir Section 3.2.3).

Lorsque le nombre de variables est important, on souhaite faciliter l'interprétation des données en sélectionnant les variables pertinentes et en mettant de côté les autres. Cependant, la régression PLS agit uniquement sur la compression (regroupement de variables qui se ressemblent dans un même axe) et ne répond pas à la question de sélection des prédicteurs. Dans ce contexte, Le Cao et al. (2008) a développé une méthode variante de la PLS, appelée sparse PLS, qui consiste à introduire une pénalisation lors du calcul des facteurs principaux w_h, en mettant les moins pertinents à 0. Les variables latentes ensuite créées, dépendent seulement d'un sous-ensemble des prédicteurs originaux. Cette méthode s'inspire de celle du Lasso (Tibshirani (1996)), qui adopte elle aussi une stratégie de pénalisation lors du calcul des coefficients de la régression linéaire.

3.2.3 Analyse Discriminante des Moindres Carrés Partiels

La régression PLS peut facilement s'adapter au cas de la classification supervisée (PLS-Discriminant Analysis) (Barker and Rayens, 2003). A la différence de la PLS, cette technique de classification nécessite des groupes d'appartenance de type qualitatif à m modalités (classes) des différents objets du jeu de données. Il suffit de générer un ensemble de m variables indicatrices $Z_1, \ldots, Z_m$ puis d'appliquer la régression PLS en considérant ces nouvelles variables comme des variables quantitatives.

Il existe aussi une version parcimonieuse de la méthode PLS-DA, appelée sPLS-DA (Cao et al., 2011), permettant la sélection des variables prédisant au mieux les variables réponses.

Techniques de validation croisée

Quand on a la possibilité d'avoir plusieurs choix de modèles, comme c'est le cas pour la régression Lasso, il se pose le problème du choix optimal du modèle le plus pertinent. La validation croisée est une méthode qui permet d'estimer la fiabilité d'un modèle. Il existe plusieurs techniques de validation :

- Le test-set validation : l'échantillon de taille n est divisé en un échantillon d'apprentissage, qui doit représenter plus de 60% de l'échantillon total et un échantillon de test, qui permet de valider le modèle,
- La validation croisée à k blocs (k-fold cross-validation) : l'échantillon original est divisé en k échantillons de taille n_k, puis on sélectionne un des k échantillons comme ensemble de validation tandis que les $k-1$ restants constituent l'ensemble d'apprentissage. On peut alors calculer une erreur. L'opération est répétée k fois pour que chaque groupe ne serve qu'une seule fois d'échantillon de validation. Enfin, on moyenne les k erreurs afin d'obtenir une estimation de l'erreur de prédiction. La pénalité qui minimise l'erreur est considérée comme la pénalité optimale.

— La validation croisée un contre tous (Leave-One-Out Cross-Validation) : il s'agit d'un cas particulier de la validation croisée à k blocs où $k = n$. Cela signifie qu'à chaque itération d'apprentissage-validation, l'apprentissage se fait sur $n - 1$ observations et la validation sur l'unique observation restante.

4 Modélisation et méthodes de classification des données sous forme de réseaux

Les réseaux sont utilisés dans de nombreux domaines scientifiques pour représenter les interactions entre les variables d'un système étudié. Les réseaux de régulation décrivent la régulation des gènes par des facteurs de transcription (Milo et al., 2002), les réseaux de co-expression de gènes modélisent les interactions entre gènes (Weirauch, 2011) et les réseaux sociaux modélisent des interactions sociales web, de communication, de collaboration (Easly and Kleinberg, 2010).

4.1 Introduction à la théorie des graphes

4.1.1 Caractéristiques d'un graphe

Un graphe $G(V, E)$ est la donnée d'une ensemble V fini de sommets et d'un ensemble E de couples de sommets $(u, v) \in V^2$. S'il existe une notion d'ordre au sein de ces couples, on parle de graphe orienté, sinon on parle de graphe non orienté (Figures 1.6 **a)** et **c)**). De plus, si un poids est associé à chacune des arêtes composant le graphe, on dit que le graphe est pondéré (Figure 1.6 **b)**).

Définition 4

— *Une paire $(u, v) \in V^2$ est appelée une arête, ou un arc, et est représentée graphiquement par un trait (Figure 1.6 **a)**), ou une flèche (Figure 1.6 **b)**),*

— *Deux sommets u et v sont adjacents si ils sont reliés par une arête,*

— *Une boucle est une arête reliant un sommet à lui-même,*

— *Le degré $d(u)$ d'un sommet $u \in V$ est le nombre d'arêtes incidentes à ce sommet,*

— *Le chemin d'un sommet u vers un sommet v est une séquence $(s_0, \ldots, s_p) \in V^p$ ($p \in \mathbb{N}^*$) tel que :*
  - $u = s_0$, $v = s_p$,
  - $(s_{i-1}, s_i) \in E$ *pour tout* $i \in \{1, \ldots, p\}$.

  Ce chemin $(s_0, \ldots, s_p)$ forme un cycle si $s_0 = s_p$.

— *Un graphe non-orienté est dit simple s'il est sans boucle et s'il admet au maximum une arête entre deux sommets,*

— *Un sous-graphe induit de G est un graphe $G'(V', E')$ où $V' \subset V$ et $E' = \{(u, v) \in E,\ u \in V' \text{et } v \in V'\}$ (Figure 1.7),*

— *Un sous-graphe partiel de G est un graphe $G'(V', E')$ tel que $V' \subset V$ et $E' \subset \{(u, v) \in E,\ u \in V' \text{et } v \in V'\}$ (Figure 1.7),*

— *Un graphe non orienté est dit connexe si chaque sommet est accessible à partir de n'importe quel autre sommet. Autrement dit, si pour tout couple de sommets distincts $(s_i, s_j) \in V^2$, il existe un chemin entre s_i et s_j.*

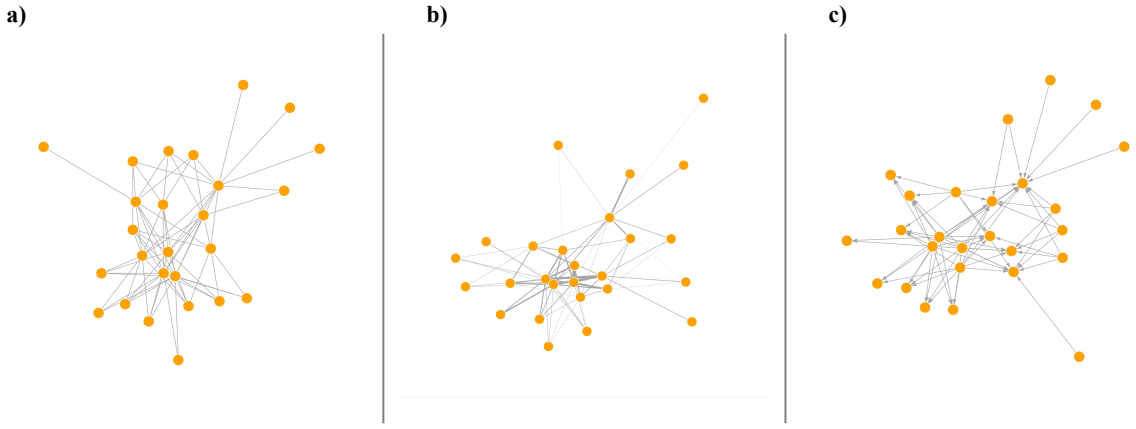

FIGURE 1.6 – Exemple de graphe non orienté **a)**, **b)** non orienté pondéré et orienté **c)**.

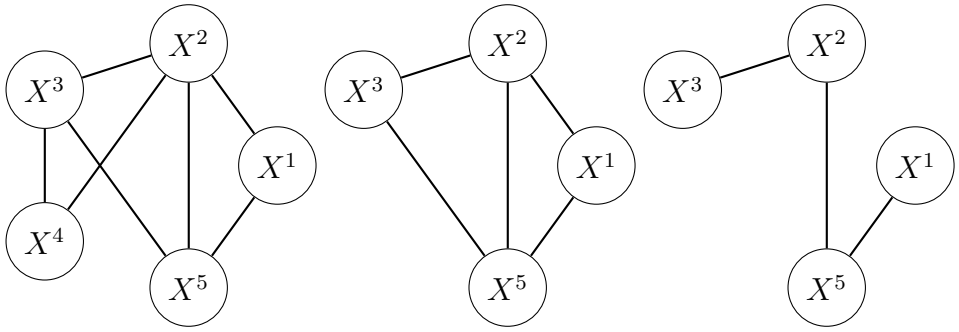

FIGURE 1.7 – Exemple de graphe $G(V, E)$ (gauche), graphe induit par $V' = \{X^1, X^2, X^3, X^5\}$ (milieu) et graphe partiel défini par $V' = V \backslash \{(X^2, X^3), (X^2, X^5), (X^5, X^1)\}$ (droite)

Définition 5 *Une composante connexe d'un graphe non orienté G est un sous-graphe G' de G qui est connexe.*

4.1.2 Représentation d'un graphe

Un graphe $G(V, E)$ peut être représenté par une matrice dite d'adjacence notée A à coefficients dans $\{0, 1\}$ et de dimension $p \times p$ qui décrit la présence (ou absence) d'une arête dans le graphe :

$$\forall i,j \in \{1,p\}\,,\ A_{ij} = \begin{cases} 1 & \text{si } (i,j) \in E, \\ 0 & \text{sinon .} \end{cases}$$

Lorsque le graphe est pondéré, la matrice d'adjacence est remplacée par la matrice des poids $W = (w_{i,j})_{1 \leq i,j \leq p}$ à coefficients dans $\mathbb{R}^+$ tel que

$$w_{i,j} = \begin{cases} 0 & \text{si } (i,j) \notin E \\ w_{i,j} & \text{sinon.} \end{cases}$$

On parle alors de matrice d'adjacence pondérée.

4.2 Analyse de réseaux

L'analyse de réseaux est confrontée à deux problématiques de recherche très actives ces dernières années. La première concerne l'inférence de réseau : comment construire à partir des données un réseau dont les arêtes représentent des " liens directs" entre les variables ? La seconde problématique intervient lorsque le réseau est déjà construit ou donné : il s'agit de l'exploration de réseaux, au travers de méthodes de clustering de graphes, qui vise à en déterminer les principales caractéristiques. L'idée principale est de se focaliser sur des interactions entre groupes de noeuds densément connectés, dont les éléments constitutifs partagent des caractéristiques communes (similarités ou connections) dans un sens prédéfini. Ces groupes de variables, appelés clusters ou communautés, sont identifiés par les méthodes de clustering à partir d'un ensemble de données dont on ne connaît pas au préalable la structure. L'analyse des réseaux offre ainsi la possibilité de mettre en évidence des ensembles de communautés, qui ont de grandes chances d'avoir la même fonction (Figure 1.8).

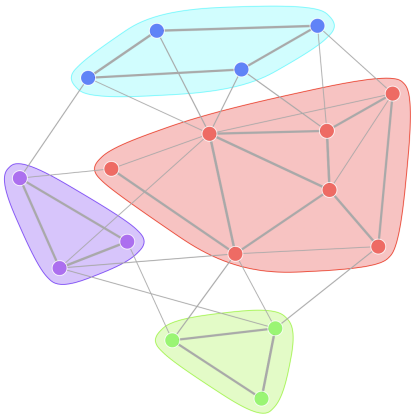

FIGURE 1.8 – Exemple de graphe non dirigé pondéré constitué de clusters

4.2.1 Définitions et propriétés d'un cluster

De manière très générale, les clusters forment des sous-groupes de nœuds densément connectés. Mais, cette définition n'est pas unique puisqu'il existe de nombreuses variantes utilisées dans la littérature. Trouver des communautés dans un réseau peut être une tâche difficile sur le plan computationnel, leur nombre étant inconnu et leurs tailles et densités hétérogènes. Pour faciliter leur détection, on peut souhaiter trouver des clusters satisfaisant des propriétés spécifiques.

La notion de coreset, outil permettant de restituer l'information contenue dans graphe en sélectionnant de petits sous-ensembles représentatifs, a connu un grand succès ces dernières années. Il se base sur la décomposition en cores d'un graphe qui attribue à chaque nœud un entier reflétant sa connectivité avec les nœuds qui l'avoisinent. On parle parfois de k-core décomposition lorsqu'il s'agit de décomposer le graphe en sous graphes dans lequel tous les sommets sont connectés avec au minimum k autres sommets du sous-graphe (Seidman, 1983). Cet outil permet ainsi de quantifier la pertinence, centralité des nœuds qui composent le graphe, notion cruciale dans le cadre de l'analyse de réseaux en grande dimension pour réduire la complexité du problème.

Dans cette thèse, nous utiliserons ces deux définitions comme base des algorithmes de clustering que nous avons développés. Rappelons dans un premier temps, les méthodes de clustering les plus citées dans la littérature.

4.2.2 Méthodes de clustering

On distingue deux grandes familles de méthodes de clustering : les approches hiérarchiques et celles par partitionnement. Les approches hiérarchiques produisent une séquence ordonnée de partitions emboitées dont le premier terme est la partition la plus fine qui ne contient que des singletons, et le dernier terme est la partition la plus grossière qui ne comporte qu'une seule partie. Elles conduisent à des résultats sous forme d'arbres hiérarchiques indicés, aussi appelés dendrogrammes (voir figure 1.9).

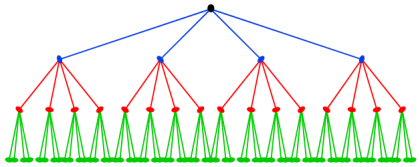

FIGURE 1.9 – Exemple d'arbre hiérarchique indicé

On distingue deux types de classification hiérarchique : les méthodes descendantes (divisives), et les méthodes ascendantes (agglomératives) :

— Les méthodes descendantes considèrent l'ensemble des observations du jeu de données et procèdent par division successive jusqu'à obtenir une partition formée de singletons,

— Les méthodes ascendantes commencent avec la partition la plus fine i.e. chaque nœud est un cluster à lui tout seul. Une fois les deux clusters les plus proches trouvés, ils sont agglomérés en un seul cluster. Cette étape est répétée jusqu'à ce que tous les nœuds appartiennent à un

seul cluster, constitué de l'agglomération de tous les cluters initiaux. On obtient ainsi un arbre binaire dont la racine correspond à la partition à une seule partie et dont les feuilles s'identifient aux différents singletons. Les différents noeuds intermédiaires correspondent à la fusion de deux parties. Ce mode de construction est appelé Classification Ascendante Hiérarchique (CAH) (Ward, 1963).

Contrairement aux approches hiérarchiques, les approches par partitionnement permettent de subdiviser l'ensemble des observations en un certain nombre de clusters fixé a priori. Pour cela, elles emploient une stratégie d'optimisation itérative qui consiste à générer une partition initiale, puis à chercher à l'améliorer en réattribuant les données d'une classe à l'autre. L'algorithme de partitionnement le plus connu est celui des k-moyennes (k-means) (MacQueen, 1967; Lloyd, 1982) qui consiste à répartir les observations $(X_1, \ldots, X_n)$ sur K ensembles $C = \{C_1, \ldots, C_K\}$ $(k \leq n)$ formant une partition C de $E = \{X_1, \ldots, X_n\}$ (telle que $C_k \cap C_l = \emptyset$, $\forall k \neq l$ et $\bigcup_{k=1}^{K} C_k = E$) de façon à maximiser la similarité à l'intérieur des classes et à minimiser celle entre les classes :

$$\sum_{i=1}^{n} \sum_{k=1}^{K} (\mathbf{1}_{C_k})_i ||X_i - \mu_k||^2,$$

où

$$\forall k \in [\![1, K]\!], i \in [\![1, n]\!],\ (\mathbf{1}_{C_k})_i = \begin{cases} 1 & \text{si } x_i \in C_k, \\ 0 & \text{sinon.} \end{cases}$$

et μ_k correspond à la moyenne du $i^{\text{ème}}$ cluster C_k.

D'autres méthode de clustering très connues ont été spécialement développées pour le clustering de graphe. Parmi les plus connues, nous retrouvons notamment le spectral clustering (Shi and Malik, 2000; Ng et al., 2002) qui, en se basant sur les propriétés spectrales du réseau, permet d'identifier des structures de communautés au sein du réseau.

5 Contributions de cette thèse

Les travaux présentés dans cette thèse sont regroupés en quatre chapitres. Le chapitre II concerne l'étude d'un algorithme de core clustering pour la détection robuste de variables représentatives dans le cadre de données en grande dimension. Le chapitre III propose une nouvelle variante de l'algorithme du spectral clustering qui, permet de construire des clusters dans le cadre de graphes bruités. Le chapitre IV présente une étude statistique complète réalisée sur une cohorte de patients atteints de fibrose hépatique dont le but est de découvrir des signatures bactériennes impliquées dans le développement de cette maladie. Enfin, le chapitre V explore une problématique rencontrée dans le chapitre IV qui concerne l'application de méthodes d'apprentissage statistique à des jeux de données issus de deux cohortes différentes. Ce chapitre présente plusieurs approches d'apprentissage plus juste, basées sur des méthodes standards de réduction de dimension dont le but est d'expliquer la variabilité du jeu de données tout en limitant l'effet du biais engendré par les observations.

5.1 CORE-clustering et détection de variables représentatives en grande dimension

On se place dans le cadre d'un problème de clustering de graphe en grande dimension dont l'objectif est de détecter des groupes de variables très connectées formant des structures majeures du graphe. Une très grande variété de méthodes standards de clustering utilisées dans la littérature ont été développées en ce sens. Nombreuses d'entre elles requièrent de fixer en amont le nombre total de clusters désirés qui, dans le cadre de la grande dimension, entraîne souvent des instabilités. Ces dernières années, la notion de coreset est de plus en plus utilisée pour pallier à l'effet de la grande dimension en détectant de manière robuste de petits groupes d'observations représentatives dans des jeux de données en grande dimension. Dans le chapitre II de ce manuscrit est développé un nouvel algorithme, appelé CORE-clustering, permettant de détecter de manière robuste des variables centrales représentatives du jeu de données extraites de CORE-clusters, définis avec comme seuls paramètres d'entrée le nombre minimal de variables et le niveau minimal de similarité (voir Section 2.3) entre les variables de chaque cluster. Une version parcimonieuse alternative de cet algorithme, particulièrement adaptée à la grande dimension est aussi proposée.

Considérons un système complexe composé de p variables quantitatives et n observations de ces p variables ($n << p$) modélisé sous la forme d'un graphe $G(V, E)$ non dirigé, pondéré et sans boucle où $V = (V_1, \ldots, V_p)$ est l'ensemble des noeuds du graphe et $E \subseteq V \times V$ l'ensemble des arêtes reliant chaque paire de nœuds. Notons $e_{i,j} \in E$ l'arête reliant les nœuds V_i et V_j avec un poids $w_{i,j}$ calculé à partir de la valeur absolue du coefficient de corrélation de Pearson.

Commençons tout d'abord par définir une mesure de similarité entre deux nœuds V_i et V_j du graphe. Rappelons pour cela qu'un chemin de longueur Λ est défini comme une liste d'indices $\{d_1, \ldots, d_\Lambda\} \subset \{1, \ldots, p\}$ tel que $V^i = V^{d_1}$, $V^j = V^{d_\Lambda}$, et $w_{d_l, d_{l+1}}$ est connu et non nul pour tout $l = 1, \ldots, \Lambda - 1$. Alors sa capacité $cap(P)$ est définie comme le poids minimal des arêtes qui composent ce chemin :

$$cap(P) = \min_{l=1,\ldots,\Lambda-1} w_{d_l, d_{l+1}}. \tag{1.4}$$

Notons à présent $\mathbf{P}_{i,j}$ l'ensemble de tous les chemins possibles connectant V_i et V_j. La cohérence entre deux nœuds $c(V_i, V_j)$ est définie en considérant le chemin P ayant la capacité maximale parmi tous les chemins possibles $\mathbf{P}_{i,j}$:

$$\begin{aligned} c(N^i, N^j) &= \max_{P \in \mathbf{P}_{i,j}} cap(P), \\ &= \max_{P \in \mathbf{P}_{i,j}} \min_{l=1,\ldots,\Lambda-1} w_{d_l, d_{l+1}}. \end{aligned} \tag{1.5}$$

Notons à présent S un sous-ensemble connecté de nœuds du graphe. La cohérence au sein de ce groupe notée $\mathbf{c}(S)$ est alors définie comme la cohérence minimale entre variables qu'elle contient :

$$\mathbf{c}(S) = \min_{(V_i, V_j) \in S^2} c(V_i, V_j). \tag{1.6}$$

Cette notion de cohérence permet de mesurer le niveau de similarité au sein d'un sous-groupe de nœuds, et donc de décomposer le graphe en sous-graphes dont les variables partagent de fortes similarités. Il s'agit de l'un des paramètres d'entrées de notre algorithme.

L'algorithme de CORE-clustering avec les paramètres d'entrée ξ (niveau de similarité minimal) et τ (nombre minimal de variables) a alors pour objectif de trouver $\widehat{U}$ sous-ensembles de variables $\widehat{\mathbf{S}} = \{\widehat{S^u}\}_{u\in\{1,\dots,\widehat{U}\}}$, où $\widehat{U}$ n'est pas fixé, en optimisant :

$$\left(\widehat{\mathbf{S}}, \widehat{U}\right) = \underset{(\mathbf{S},U)}{\arg\max} \sum_{u=1}^{U} \mathbf{c}(S^u)\,, \tag{1.7}$$

sous les deux contraintes :

1. Tous les S^u sont des ensembles de variables de taille plus grande que τ et une cohérence $\mathbf{c}(S^u_{\xi,\tau}) > \xi$. Ils sont appelés CORE-clusters et peuvent être notés $S^u_{\xi,\tau}$.
2. Il n'y a pas de superposition entre les clusters, *i.e.* $S^{u_1} \cap S^{u_2} = \emptyset$, pour tous les $(u_1, u_2) \in \{1, \dots, U\}^2$.

Le caractère robuste de l'algorithme CORE-clustering qui, a fait l'objet d'une publication dans le journal *Algorithm* et du développement d'un package disponible sur [1], est prouvé au travers de simulations et de plusieurs applications sur jeux de données réelles.

5.2 Clustering de graphes par spectral clustering pénalisé

On se place toujours dans le cadre d'un problème de clustering de graphe dont le but est d'en détecter les structures sous-jacentes puis de les identifier. Un très grand nombre d'algorithmes ont été développés en ce sens et sont communément utilisés dans la littérature comme par exemple le k-means. Les algorithmes de spectral clustering, qui se basent de manière avantageuse sur les propriétés spectrales du graphe, ont également connu du succès dès le début des années 2000. Bien que très performants d'un point de vue théorique et algorithmique, ils restent cependant sensibles à la présence de bruit dans le graphe, de par l'utilisation du k-means, pour extraire des valeurs propres de la structure par bloc sous-jacente. Dans le chapitre III de ce manuscrit est présentée une version ℓ_1-pénalisée du spectral clustering afin d'améliorer ses performances sur des graphes bruités.

Considérons dans un premier temps un graphe idéal $G(V, E)$ non dirigé, non pondéré et sans boucle modélisant un système non bruité. Le graphe ainsi défini est composé d'un ensemble de p noeuds $V = \{1, \dots, p\}$ et d'un ensemble d'arêtes $E \subseteq V \times V$ connectant chaque noeud. Celui-ci peut être modélisé par une matrice dite d'adjacence symétrique notée $A = (A_{ij})_{(i,j)\in E}$ de taille $p \times p$ et vérifiant :

$$\forall (i,j) \in [\![1,p]\!]^2, \;\; A_{ij} = \begin{cases} 1 \;\text{ si }\; (i,j) \in E, \\ 0 \;\text{ sinon.} \end{cases}$$

Supposons de plus que le graphe est formé de K composantes connexes notées $C_1, \dots, C_K$ définies comme des sous-groupes de noeuds de V tel que chaque paire de noeuds de C_k est connectée par un chemin et tel qu'il n'y ait pas de connections entre les noeuds à l'intérieur et à l'extérieur de C_k ($k \in [\![1, K]\!]$). Nous pouvons alors définir les indicateurs des K clusters auxquels appartiennent les p noeuds du graphe de la manière suivante :

1. `https://es.sourceforge.net/projects/core-clustering/`

$$\forall k \in [\![1, K]\!], \forall j \in [\![1, p]\!], \ \ (\mathbf{1}_{C_k})_j = \begin{cases} 1 & \text{si le noeud } j \text{ appartient au cluster } C_k, \\ 0 & \text{sinon.} \end{cases}$$

Alors, les K vecteurs propres $v_{p-K+1}, \ldots, v_p$ associés aux plus grandes valeurs propres $\lambda_{p-K+1} \leq \cdots \leq \lambda_p$ de la matrice d'adjacence A, correspondent, à constante près, à ces indicateurs.

Considérons dans un second temps un graphe bruité dont les propriétés sont les mêmes que le graphe précédemment défini. Nous cherchons alors à détecter de manière robuste les indicateurs des K composantes connexes. Pour cela, nous montrons l'approche sparse suivante : l'indicateur de la première composante est solution du problème de minimisation suivant

$$\mathbf{1}_{C_1} = \underset{\substack{(1,v)\in\mathbb{R}^p \\ Wv=-w}}{\arg\min} \ \|v\|_1,$$

où w et W sont extraits des vecteurs propres correspondant aux $p - K$ plus petites valeurs propres de A et définis par :

$$(v_1, ..., v_{p-k}) = \begin{bmatrix} \boxed{w^T} \\ \boxed{W^T} \end{bmatrix}.$$

Les indicateurs des $K - 1$ clusters restants sont calculés de manière similaire après avoir appliqué une procédure d'orthogonalisation de Gram-Schmidt pour garantir l'orthogonalité des indicateurs. Enfin, ces problèmes de minimisation sont résolus à l'aide de la méthode Lasso.

L'efficacité et le caractère robuste de cet algorithme sont prouvés au travers de nombreuses simulations, comparaisons et applications sur jeux de données réelles.

Cet article a fait l'objet d'une soumission dans le Journal Advances in Data Analysis and Classification et du développement d'un package disponible sur le CRAN (package *l1spectral*).

5.3 Découverte de signatures bactériennes chez des patients fibrotiques

Le chapitre IV de cette thèse est consacré à l'application d'un ensemble de méthodes statistiques sur un jeu spécifique de données réelles. Il s'agit d'une cohorte de patients atteints à divers degrés de Fibrose hépatique, maladie du foie qui, à un stade avancé, peut avoir des conséquences graves et irréversibles tel le cancer du foie. L'objectif de cette étude est d'identifier le profil clinique des patients puis des espèces microbiologiques impliquées dans le début du développement de la maladie afin de prédire et potentiellement de bloquer l'engagement du patient vers des états pathologiques graves en intervenant sur la causalité de la maladie.

L'ensemble des données étudiées provient de la fusion de deux cohortes contenant $n_1 = 36$ patients roumains et $n_2 = 46$ patients provenant de trois pays différents. Tous sont caractérisés par une variable d'intérêt, le stade de fibrose (classé sur une échelle à trois niveaux : F_0 pas de fibrose, F_1 fibrose légère et F_2 fibrose modérée) et ont un profil assez similaire (obésité sévère). Une première étude a été réalisée sur la table des données cliniques contenant $q = 21$ variables caractérisant chacun des $n = 82$ patients de la cohorte. Puis, nous nous sommes focalisés sur la table de comptage de $p = 411$

OTUs (Operational Taxonomic Units) contenant le nombre de séquences observées pour chaque OTU dans chaque échantillon. Cette table a été filtrée puis normalisée en appliquant la transformation CSS avant toute application de méthodes multivariées.

L'étude statistique réalisée sur cette cohorte se décompose en quatre étapes :

— Analyse exploratoire des données cliniques et bactériologiques (ACP, analyse α et β de la diversité bactérienne) pour dresser le profil global des patients à chaque stade de fibrose,

— Analyse discriminante des données bactériologiques (LDA, PLS-DA et sPLS-DA) afin de déterminer les signatures bactériologiques impliquées dans l'évolution de la maladie,

— Analyse discriminante plus juste et fair clustering (fair ACP, fair l_1-spectral clustering, fairlet clustering) pour identifier des groupes d'OTUs discriminants tout en limitant le biais introduit par la fusion des deux cohortes,

— Analyse discriminante des enzymes et pathways (PLS-DA et sPL-DA) afin de prédire les voies métaboliques fonctionnelles impliquées dans le développement de la fibrose.

L'efficacité des diverses méthodes appliquées sur le jeu de données est prouvée au travers de nombreuses simulations et comparaisons mises en lumière sous la forme de figures et tableaux. La combinaison de l'ensemble des méthodes précédemment citées a permis d'identifier des signatures spécifiques du début du développement de la fibrose hépatique tant au niveau clinique que bactériologique.

Cet article a fait l'objet d'une soumission dans le Journal *Microbiome*.

5.4 Méthode fair de Régression des Moindres Carrés Partiels

Au cours de la précédente étude statistique, nous avons identifié des problématiques liées à l'origine des données. En effet, issues de l'agrégation de deux cohortes distinctes, elles sont caractérisées par une très grande variabilité phénotypique et une diversité bactériologique entre les deux groupes d'individus, qui vient interférer dans la découverte de signatures biologiques de la maladie. La précision des méthodes d'apprentissage standards, dépendante de l'information diffusée par les données, est très sensible à l'existence de biais. Pour remédier à cette problématique, nous avons proposé dans le chapitre précédent quelques adaptations de méthodes standards d'apprentissage statistique (exploratoires, prédictives et de clustering) qui, dans le chapitre V de cette thèse, font l'objet d'une recherche plus poussée.

Rappelons que l'objectif principal de cette étude est de comprendre l'ensemble des variabilités qui caractérisent le jeu données et à terme de détecter des biomarqueurs de la maladie. Notre étude s'est alors premièrement portée sur l'Analyse en Composantes Principales, méthode exploratoire souvent utilisée comme analyse préliminaire à toute autre analyse statistique, qui permet de créer une nouvelle représentation des données reflétant le plus justement possible les variabilités qui le composent. L'analyse de données biomédicales étant souvent centrée autour d'une variable clinique d'intérêt, nous nous sommes intéressés à une méthode standard d'apprentissage supervisé : la Régression des Moindres Carrés. Très proche de l'Analyse en Composantes Principales, elle créée une nouvelle représentation des données diffusant largement l'information des données originales tout en conservant un lien avec la variable d'intérêt. Cependant, ces deux méthodes principalement basées sur la notion de variance, sont beaucoup moins efficaces en la présence de biais. Pour cette raison, nous proposons dans ce chapitre deux approches d'apprentissage fair, basées sur les deux méthodes précédemment

citées, afin de pouvoir expliquer l'ensemble des variabilités qui composent le jeu de données en limitant l'effet du biais. Notons $S =^t (S_1, \ldots, S_n)$ la variable, dite sensible, à l'origine du biais, $Y =^t (Y_1, \ldots, Y_n)$ la variable d'intérêt et $X = (X_i^j)_{i=1,\ldots,n,j=1,\ldots,p}$ chacune des observations qui composent le jeu de données. L'idée est alors de fixer un ou plusieurs critères permettant de dissocier la variable sensible des composantes créées par ces méthodes, afin d'éviter les interférences lors de la phase de prédiction. Cette notion d'indépendance est garantie par le rapport de corrélation et contrôlée par un seuil, déterminé à l'issue d'un processus de validation croisée. Dans ce chapitre sont présentés deux versions d'algorithmes fair, fair conditional PCA et fair conditional PLS dont l'objectif est de sélectionner un nombre optimal de composantes naturellement impliquées dans le développement de la maladie sans se soucier du biais généré par les données. L'efficacité de ces algorithmes est prouvée au travers de comparaisons et applications sur deux jeux de données réelles caractérisés par une variable clinique d'intérêt de type binaire et de type continue.

Ce travail est encore en cours et ouvre à de nombreuses perspectives présentées en fin de chapitre.

Chapter 2

Detection of representative variables in complex systems with interpretable rules using CORE-clusters

This chapter describes the main elements of an accepted paper but in a different form from Champion et al. (2018), a joint work with Anne-Claire Brunet, Rémy Burcelin, Jean-Michel Loubes and Laurent Risser.

Abstract

In this paper, we present a new framework dedicated to the robust detection of representative variables in high dimensional spaces with a potentially limited number of observations. Representative variables are selected by using an original regularization strategy: they are the center of specific variable clusters, denoted CORE-clusters, which respect fully interpretable constraints. Each CORE-cluster indeed contains more than a predefined amount of variables and each pair of its variables has a coherent behavior in the observed data. The key advantage of our regularization strategy is therefore that it only requires to tune two intuitive parameters: the minimal dimension of the CORE-clusters and the minimum level of similarity which gathers their variables. Interpreting the role played by a selected representative variable is additionally obvious as it has a similar observed behaviour as a controlled number of other variables. After introducing and justifying this variable selection formalism, we propose two algorithmic strategies to detect the CORE-clusters, one of them scaling particularly well to high-dimensional data. Results obtained on synthetic as well as real data are finally presented.

1 Introduction

Discovering representative variables in high dimensional and complex systems with a limited number of observations is a recurrent problem of machine learning. Heterogeneity between the variables behavior and multiple similarities between variable subsets often make this process ambiguous. A

convenient strategy to solve this task is to associate each representative variable of the complex system to a cluster of variables, and to model the relations between variables in a graph (Liu and Barahona, 2020). The complex systems are indeed typically modeled as undirected weighted graphs (Boccaletti et al., 2006; Newman, 2009), where the nodes (vertices) represent the variables and the edge weights are a measure of the observed similarity between the variables of the dataset. The choice of a specific clustering algorithm over the wide variety of traditional methods often depends on the nature of the data (*e.g.* their structure or size). Determining the granularity level of the clusters is also a common issue in high dimensional data clustering. If the clustering granularity is high, some clusters have a high similarity rate between the nodes/variables they contain, but potentially many other clusters only contain noisy relations. A large amount of selected representative variables can then be meaningless. Conversely, a low granularity leads to few large clusters with high internal heterogeneous behaviors, which makes it hard to identify the representative variables of the system in each cluster. Importantly, this issue is particularly critical when the number of observations n is lower than the observations dimension p, because of the instability related to high dimensionality and high complexity.

As in k-means clustering algorithms (MacQueen, 1967), we will use in this paper the notion of *cores* to address with an interpretable strategy the choice of the granularity level. Based on a distance function, the k-means algorithms indeed form a controlled number of clusters. This notion of *core* is also used in Seidman (1983); Giatsidis et al. (2014), where the graph is partitioned into a maximal group of entities, which are connected to at least k other entities in the group. The method of Batagelj and Zaversnik (2011) is also related to the notion of *coreness* as it hierarchically calculates the core number for each node with a complexity in the order of $\mathcal{O}(p)$. Strong connections also exist between the issues we address and the notion of *coresets* (Agarwal et al., 2005). This notion has recently gained a significant interest in the machine learning community as it deals with finding *representative* observations, and not variables, in large datasets. As described in Claici et al. (2020), it can be used in supervised learning to reduce the size of large training sets. In a similar vein, it can also be used to robustify the generalisation of the trained decision rules (Baharan et al., 2020), for neural network compression (Baykal et al., 2019) or for unsupervised learning (Bachem et al., 2018). Note that Bachem et al. (2018) is also particularly close to core methods, as it specifically deals with $k-$clustering, *i.e.* finding at most k cluster centers. The proposed method however does not address explicit interpretability issues.

The challenge of high-dimensionality is clearly raised in Saeys et al. (2007); Zhao et al. (2010); Li et al. (2018), where the authors proposed different feature selection techniques with an explicit regularization in order to speed up a data mining algorithm and to improve mining performance. In this spirit, Wu et al. (2018) recently developed an approach based on an iterative spectral optimization technique that improves the quality, computation time and scalability to high dimension of an existing alternative clustering method (Kernel Dimension Alternative Clustering). In Chen et al. (2018), the authors also used multinomial regression model to learn automatically the number of clusters, and then to limit strong assumptions required by the model in high dimension. Note that Yu and Liu (2003) also defined a strategy for the detection of representative variables in high-dimensional data but did not explored a regularization strategy when the number of observations is much lower than the problem dimension. Those strategies require as well to make a decision about the number of clusters to determine.

Motivated by the above issues, we propose a new formalism to robustly and intuitively estimate the representative variables of complex systems. This is first made through a graph clustering strategy for which the clusters do not necessarily cover all nodes/variables. This clustering strategy specifically estimates CORE-clusters, which are connected subsets of variables having (1) a minimum number of nodes/variables, and (2) a minimal similarity level between all their variables. The representative variables are then those having the lowest average distance to all other variables in each CORE-cluster. The detection of representative variables is therefore regularized using a control on the minimal CORE-cluster size and not the number of representative variables to be detected, or a LASSO derived penalty term. This point of view has been originally considered in Brunet et al. (2015, 2016). We present here a totally reformulated version of this initial idea, which makes fully interpretable the selection of the representative variables by introducing the notion of CORE-clusters. Fine algorithmic improvements, described in this paper, also make the original clustering algorithm more efficient. A greedy version of this original algorithm, which turns out to scale particularly well to high dimensional data, is additionally proposed. New results on synthetic data as well as comparisons with other methods now shed light on how the proposed strategy is robust and explainable. Finally, we now demonstrate the validity of our formalism on two high dimensional datasets representing the expression of genes and a road network.

Our methodology is described in Sections 2 and 3 and is then tested both on simulated and real data in Section 4.

2 Statistical Methodology

2.1 Graph-based representation of the observations

Let us consider a complex system of p quantitative variables $X = (X^1, \cdots, X^p)$ and n observations $(X_1^j, \cdots, X_n^j)$, $j \in \{1, \cdots, p\}$ of these p variables. The driving motivation of our work is to detect representative variables out of X when $n \ll p$. As mentioned in Section 1, the detection of these representative variables is regularized using a graph-based approach. We then model the relations between the variables using an undirected weighted graph $G(V, E)$, where $V = (V_1, \cdots, V_p)$ is the nodes set corresponding to the p variables, and E is the edges set. We also denote $e_{i,j} \in E$ the edge joining the nodes V_i and V_j with weight $w_{i,j}$.

In order to handle the properties of application-specific similarity measures that can be encoded in the edge weights $w_{i,j}$, we will consider in the remainder of the paper that all $w_{i,j} \geq 0$ and that the higher $w_{i,j}$ the closer the observed behavior of the variables X^i and X^j. The *weights therefore represent a notion of similarity* between the variables X^i and X^j. For instance, if the empirical correlations $cor(X^i, X^j)$ are measured between the pairs of variables X^i and X^j, with $(i, j) \in \{1, \ldots, p\}^2$, then $w_{i,j} = |cor(X^i, X^j)|$ can reasonably be used.

2.2 Coherence of a variable set

To define a notion of distance between two variables X^i and X^j, which are not directly connected in the graph, we use the notion of capacity (Pollack, 1960; Hu, 1961) of a path P between the

corresponding nodes V_i and V_j in $G(V, E)$. A path P of a graph G from X^i to X^j of length Λ is a list of indices $\{d_1, \ldots, d_\Lambda\} \subset \{1, \ldots, p\}$ such that $X^i = X^{d_1}$, $X^j = X^{d_\Lambda}$, and $w_{d_l, d_{l+1}}$ is known and is not equal to 0, for all $l = 1, \ldots, \Lambda - 1$. The capacity $cap(P)$ of path P is then the minimal weight of its edges, *i.e.*

$$cap(P) = \min_{l=1,\ldots,\Lambda-1} w_{d_l, d_{l+1}}. \tag{2.1}$$

We also denote by $\mathbf{P}_{i,j}$ the set of all possible paths connecting X^i to X^j. The coherence $c(X^i, X^j)$ between X^i and X^j is then defined by considering the path P having the maximum capacity among the paths of $\mathbf{P}_{i,j}$ (Hu, 1961), *i.e.*

$$\begin{aligned} c(X^i, X^j) &= \max_{P \in \mathbf{P}_{i,j}} cap(P), \\ &= \max_{P \in \mathbf{P}_{i,j}} \min_{l=1,\ldots,\Lambda-1} w_{d_l, d_{l+1}}. \end{aligned} \tag{2.2}$$

If the weight $w_{i,j}$ is known, its is interesting to remark that the coherence $c(X^i, X^j)$ is not necessarily equal to its value. For instance, both X^i and X^j may be very similar to a third variable X^k but not similar to each other. From a computational point of view, the similarity in $w_{i,j}$ may also be unknown if the edge $e_{i,j}$ is not stored in a sparsified version of the complete graph. Since $n \ll p$, pertinent relations may finally be lost in $w_{i,j}$ but recovered in $c(X^i, X^j)$ thanks to other relations that would be captured. We believe that these points are particularly important to define coherent variable sets in the complex data case.

We now denote by S a connected subset of the variable set X. The coherence $\mathbf{c}(S)$ of this variable subset is the minimal coherence between the variables it contains:

$$\mathbf{c}(S) = \min_{(X^i, X^j) \in S^2} c(X^i, X^j). \tag{2.3}$$

If all the variables of S have a coherent observed behavior, then $\mathbf{c}(S)$ is high. The use of this notion on synthetic data is illustrated in Appendix 6.1. The coherence of a subset measures the strength of the variables it contains. The more coherent S, the more sense it makes to consider that its variables share common features measured by the chosen similarity. Decomposing the graph into maximal groups sharing a strong similarity, *i.e.* finding all the groups of variables with a large enough coherence is the core of the following subsections on CORE-clusters selection.

2.3 CORE-clusters

We recall that the goal of our formalism is to detect the representative variables of complex systems. In our formalism, each representative variable is extracted out of a CORE-cluster, defined as:

Definition 1 *A CORE-cluster $S_{\xi,\tau} \subset X$ is a connected variable subset having a size higher than τ and a coherence $\mathbf{c}(S_{\xi,\tau})$ higher than a threshold ξ.*

The parameters τ and ξ ensure that each representative variable has a non-negligible coherence with at least $\tau - 1$ other variables, which directly regularizes its selection: Large values of τ indeed lead to the detection of large sets of coherent variables. In that case, the representative variables are likely

to be meaningful even if $n < p$. If τ is too large, each CORE-cluster may however contain several variables that would have been ideally representative. On the contrary, too small values of τ are likely to detect all meaningful representative variables but also a non-negligible number of false positive representative variables, especially if the observations are noisy or if $n < p$. A good trade-off, which depends on n, p and the level of noise in the observations has then to be found when tuning τ.

2.4 CORE-clustering

CORE-clustering consists in estimating an optimal set of CORE-clusters, so that the representative variables they contain explain as much information as possible in the observed complex system. We use the following definition:

Definition 2 *CORE-clustering with parameters ξ and τ consists in finding $\widehat{U}$ variable subsets $\widehat{\mathbf{S}} = \{\widehat{S^u}\}_{u\in\{1,\dots,\widehat{U}\}}$, where $\widehat{U}$ is not fixed, by optimizing:*

$$\left(\widehat{\mathbf{S}}, \widehat{U}\right) = \underset{(\mathbf{S},U)}{\arg\max} \sum_{u=1}^{U} \mathbf{c}(S^u)\,, \tag{2.4}$$

under the two constraints:

1. *All S^u are connected variable sets having a size higher than τ and a coherence $\mathbf{c}(S^u_{\xi,\tau}) > \xi$. They therefore correspond to CORE-clusters and can be denoted $S^u_{\xi,\tau}$.*
2. *There is no overlap between the clusters, i.e. $S^{u_1} \cap S^{u_2} = \emptyset$ for all $(u_1, u_2) \in \{1, \dots, U\}^2$.*

It may first seem that $\widehat{U}$ should be as high as possible, so that the union of all $S^u_{\xi,\tau}$ contains all the variables of X. Each CORE-cluster $S^u_{\xi,\tau}$ must however have a coherence higher than the threshold ξ. As illustrated in Appendix 6.1, the variables of X which are not coherent with at least τ other variables should ideally not be contained in any CORE-cluster as they would make drop their coherence. The CORE-clusters in S should then only contain pertinent variables so the optimal value for U is implicitly defined during the CORE-clustering procedure.

It is also important to remark that the potential number of subsets of X to find good CORE-cluster candidates is huge, even for moderate values of p. Moreover computing Eq. (2.4) is particularly demanding. The two optimization algorithms of Section 3 are then aggregative and divisive algorithms designed to optimize Eq. (2.4) without explicitly computing it.

2.5 Representative variables selection

We now present how each representative variable is extracted out of a CORE-cluster $S_{\xi,\tau}$. As mentioned Section 2.2 the pertinent relations between two variables X^i and X^j may be lost in $w_{i,j}$ since $n \ll p$ and recovered by their coherence $c(X^i, X^j)$ using other relations. In this example, the similarity s captures true positive and false negative relations and the notion of coherence makes robust the detection of relations. For the same reasons, it may however also capture false positive relations, so the CORE-clusters may contain undesirable variables. The variables captured by CORE-clusters using false positive relations should however be located at the cluster boundaries if the data are not

too noisy. False positive connections are indeed less common and in average weaker than true positive connections in this case.

In order to limit the impact of the false positive relations, we then define the representative variables as the CORE-cluster centers. More specifically, each representative variable minimizes an average distance with the other variables of a CORE-cluster $S_{\xi,\tau}$. Instead of using distances based on the maximum capacity Eq. (2.2), we use a more standard notion of distance calculated as sums of weighted edges traversed by the optimal paths. This limits the phenomenon of having multiple variables with the same optimal distance due to the min-max strategy. The graph weights $w_{i,j}$, which represent a similarity level, must however be converted into distances, which can be simply done by using $d_{i,j} = 1/(w_{i,j} + \epsilon)$, where $\epsilon > 0$. The representative variable of a CORE-cluster $S_{\xi,\tau}$ is then the one that has the lowest average distance to the other variables of $S_{\xi,\tau}$. The impact of selecting the representative variables as CORE-cluster centers is discussed in Appendix 6.1.

2.6 General guidelines for the choice of ξ and τ

The selection of the representative variables directly depends on the parameters ξ and τ. As explained Section 2.3, a CORE-cluster $S_{\xi,\tau}$ is indeed a connected variable subset having a size higher than τ and a coherence $\mathbf{c}(S_{\xi,\tau})$ higher than a threshold ξ. In order to estimate pertinent representative variables, we recommend to use the following guidelines: **(1)** First compute how the edge weights $w_{i,j}$ of the graph $G(V, E)$ are distributed. The coherence $\mathbf{c}(S_{\xi,\tau})$ represents the weakest connection between the variables of $S_{\xi,\tau}$, so the threshold ξ should be relatively high regarding to the different values of $w_{i,j}$. A value of ξ equal to the 80th percentile of the edge weights $w_{i,j}$ appears as reasonable. **(2)** Choosing a suitable minimal amount of variables τ in $S_{\xi,\tau}$ is more subtle as this choice both depends on the complexity of the relations expressed in $G(V, E)$, and how the number of observations n is low compared with the observations dimension p. In all generality, tuning τ as equal to $p/10$ is reasonable as a first guess. **(3)** If no CORE-clusters are found with the initial parameters, the user may try to run again the CORE-clustering procedure with lower parameters ξ and τ.

From our experience, we recommend in all cases not using values of ξ lower than the 40th percentile of the edge weights or values of τ lower than 10. The CORE-clusters would be likely to contain variables with strongly heterogeneous behaviors or false positive connections in these cases. Note finally that when several *connected* CORE-clusters are identified with a given parametrisation, it is interesting to test whether stronger CORE-clusters would be locally found by using higher the values of ξ or τ. This is illustrated in Section 4.3.

3 Computational Methodology

3.1 Main interactions estimation

The very first step of our strategy is to quantify the similarity between the different observed variables. The similarity is first computed using the absolute value of Pearson's correlation and represented as a dense graph $G(V, E)$, where V contains p nodes, each of them representing one of the observed variables, and E contains $K_E = p(p-1)/2$ undirected weighted edges that model a

similarity level between all pairs of variables. In this approach, the variables with no connection are associated with a correlation coefficient of zero. The algorithmic cost of this estimation can be $\mathcal{O}(n^2p)$ but it can also be easily parallelized using divide and conquer algorithms for reasonably large datasets, as in Randall (1998). For large to very large datasets, correlations should be computed on sparse matrices, using *e.g.* Cysouw (2018) to make this task computationally tractable.

3.2 CORE-clustering algorithms

Inspired by Hu (1961) who solved the maximum capacity problem of Pollack (1960) using optimal spanning tree, we estimate the CORE-clusters on optimal spanning trees. A spanning tree $G(V, T)$ is a subgraph of $G(V, E)$ with no cycle and $T \subset E$. The maximum spanning tree of G is then the spanning tree of G having the maximal sum of edge weights. Using the maximum spanning tree to detect the CORE-clusters strongly limits the potential amount of node combinations to test while preserving the graph edges that are likely to be good candidates for the optimal paths of Eq. (2.2). Conversely, it is straightforward to show that the coherence of a variable subset in $G(V, T)$ is lower or equal to the coherence of the same variable subset in $G(V, E)$. The edges of T are indeed a subset of E, so Eq. (2.2) between two variables X^i and X^j is lower or equal on T than on E. The CORE-clusters computed in $G(V, T)$ are therefore eligible CORE-clusters on $G(V, E)$. By discussing the algorithmic cost of our algorithms, we will make clear that this reasonable domain reduction makes our problem scalable to large datasets. The impact of using maximal spanning trees on the measure of a cluster coherence is also discussed in Appendix 6.1 on synthetic examples.

3.2.1 Maximum Spanning Tree

Algorithm 1 Maximum spanning tree algorithm

Require: Graph $G(V, E)$ with nodes V_i, $i \in 1, \cdots, p$ and edges E_k, $k \in 1, \cdots, K_E$.
Require: Weight of edge E_k is $W(E_k)$.
1: Sort the edges by decreasing weights, so $W(E_1) \geq W(E_2) \geq \cdots \geq W(E_{K_E})$.
2: Assign label $L(V_i) = i$ to each node $V(i)$.
3: Initiate an edge list T as void.
4: **for** $k = 1 : K_E$ **do**
5: We denote V_i and V_j the nodes linked by edge E_k.
6: **if** $L(V_i)! = L(V_j)$ **then**
7: Add edge E_k to the list T
8: Propagate the label $L(V_i)$ to the nodes that have label $L(V_j)$.
9: **end if**
10: **end for**
11: **return** Graph with a tree structure $G(V, T)$.

The maximum spanning tree of G is the simple and reliable modeling of the relationship between the graph nodes (only $p - 1$ links). One of the most famous algorithm developed to find such trees is

called Kruskal's algorithm (Kruskal, 1956). The maximum spanning tree is built by adding step by step partial associations so that there will be no cycle in the partial graph.

We denote by $G(V,T)$ the resulting graph, where T has a tree-like structure. Details of the algorithm are given in Alg. 1. The algorithmic cost of the sort procedure (row 1) is $\mathcal{O}(K_E \log(K_E))$. Then, the for loop (rows 4 to 10) only scans one time the edges. In this for loop, the most demanding procedure is the propagation of label $L(V_i)$, row 8. Fortunately, the nodes on which the labels are propagated are related to V_j in $G(V,E)$. We can then use a Depth-First Search algorithm (Tarjan, 1972) for that task, making the average performance of the for loop $\mathcal{O}(K_E \log(p))$.

3.2.2 CORE-clustering algorithm

Algorithm 2 CORE-clustering algorithm

Require: Graph $G(V,T)$ with nodes V_i, $i \in 1, \cdots, p$; and edges T_k, $k \in 1, \cdots, K_T$.
Require: Weight of edge T_k is $W(T_k)$.
Require: Granularity coefficient τ and threshold ξ.
1: {Initiate the algorithm}
2: Sort the edges by increasing weights;
3: Assign label $L(V_i) = i$ to each node $V(i)$ and set $CORElabel = -1$.
4: {CORE-clusters detection}
5: **for** $k = 1 : K_T$ **do**
6: We denote V_i and V_j the nodes linked by edge E_k.
7: **if** $L(V_i) != L(V_j)$ **then**
8: Propagate the label $L(V_i)$ to the nodes that have label $L(V_j)$.
9: **if** number of nodes with label $L(V_i) \in \{\tau, \cdots, 2\tau - 1\}$ **then**
10: Label $CORElabel$ is given to the nodes with label $L(V_i)$
11: $CORElabel = CORElabel - 1$
12: **end if**
13: **end if**
14: **end for**
15: {Post-treatment of the labels}
16: **for** $v = 1 : V$ **do**
17: **If** $L(V_v) > 0$ **then** $L(V_v) = 0$ else $L(V_v) = -L(V_v)$
18: **end for**
19: {Filter the CORE-clusters $S^u_{\xi,\tau}$ s.t. $\mathbf{c}(S^u_{\xi,\tau}) < \xi$}
20: **for** $u = 1 : U$ **do**
21: **If** $\mathbf{c}(S^u_{\xi,\tau}) < \xi$ **then** Set $L(V_v) = 0$ to the nodes V_v of $S^u_{\xi,\tau}$.
22: **end for**
23: **return** Labels L.

In contrast with other clustering techniques, this CORE-clustering approach detects clusters having an explicitly controlled granularity level, and only gathers nodes/variables with a maximal path capacity. CORE-clusters are detected from the maximal spanning tree $G(V,T)$ by gathering

iteratively its nodes V in an order that depends on the edge weights in T (increasing weight order). A detected CORE-cluster has a size higher than τ, where τ is the parameter that controls the granularity level. Thus, Alg. 2 first generates many small and meaningless clusters and parameter τ should not be too small to avoid considering these clusters as CORE-clusters. Then, the first pertinent clusters will be established using the edges of T with larger weights, leading to pertinent node groups. Finally, the largest weights of T are treated in the end of Alg. 2 in order to split into several CORE-clusters the nodes related to the most influential nodes/variables.

It worth mentioning that the Rows 20 to 25 of Alg. 2 are the only ones that require to compute the coherence $\mathbf{c}$ of the estimated CORE-clusters. Computing a coherence Eq. (2.3) is indeed demanding, so it is considered here as a post-treatment limited to pre-computed CORE-clusters. In practice it is also performed on the maximal spanning tree $G(V, T)$ and not on the whole graph $G(V, E)$.

Remark too that the algorithmic structures of Alg. 1 and Alg. 2 are similar. However Alg. 2 runs on $G(V, T)$ and not on $G(V, E)$. The number of edges K_T in $G(V, T)$ is much lower than K_E since T has a tree structure and not a complete graph structure. It should indeed be slightly higher than p (Steele, 2002), which is much lower than $K_E = p(p-1)/2$. Moreover the propagation algorithm (rows 9 and 11) will then never propagate labels on more than $2\tau - 1$ nodes. The algorithmic cost of the sort procedure (row 1) is then $\mathcal{O}(K_T \log(K_T))$ and the average performance of the for loop $\mathcal{O}(K_T \log(\tau))$.

3.2.3 A greedy alternative for CORE-clusters detection

We propose an alternative strategy to the CORE-clusters detection algorithm: The edge treatment queue may be ordered by following decreasing edge weights instead of increasing edge weights. The nearest edges are then first gathered making coherent CORE-clusters as in Alg. 2, although one CORE-cluster may contain several representative variables. To avoid gathering noisy information, the for loop on the edges (row 6 of Alg. 2) should also stop before meaningless edges are treated. This strategy has a key interest: It can strongly reduce the computational time dedicated to Algs. 1 and 2. By doing so, Alg. 1 and modified Alg. 2 are purely equivalent to Alg. 3.

Again, the structure of this algorithm is very similar to the structure of the Maximum Spanning Tree strategy in Section 3.1. The algorithmic cost of the sort procedure (row 1) is $\mathcal{O}(K_E \log(K_E))$. Then, the loop rows 5 to 14 scans γ edges, where γ is the number of edges having a weight higher than ξ. In most cases, γ should be much lower than K_E, which strongly limits the computational impact of this loop. Labels propagation in this loop (rows 8 and 10) is also limited to $2\tau - 1$ nodes. The average performance of the for loop is therefore $\mathcal{O}(\gamma \log(\tau))$.

3.3 Central variables selection in CORE-clusters

Once a CORE-cluster is identified, we use a straightforward strategy to select its central variable: the distance between all pairs of variables in each CORE-cluster is computed using a Dijkstra's algorithm (Cormen et al., 2001; Zan and Noon, 1998) in $G(V, E)$. The central variable is then the one that has the highest average distance to all other connected variables in the CORE-cluster. As the computed CORE-clusters have less than 2τ nodes, the algorithmic cost of this procedure is $\mathcal{O}(\tau^2)$ times the number of detected CORE-clusters, which should remain low even for large datasets.

Algorithm 3 Greedy CORE-clustering algorithm

Require: Graph $G(V, E)$ with nodes V_i, $i \in 1, \cdots, p$ and edges E_k, $k \in 1, \cdots, K_E$.
Require: Weight of edge E_k is $W(E_k)$.
Require: Granularity coefficient τ and threshold ξ.

Sort the edges by decreasing weights.
Define the number of edges γ having a weight higher than ξ.
Assign label $L(V_i) = i$ to each node $V(i)$.
Set $CORElabel = -1$.
for $k = 1 : \gamma$ **do**
    We denote V_i and V_j the nodes linked by edge E_k.
    if $L(V_i)! = L(V_j)$ **then**
        Propagate the label $L(V_i)$ to the nodes that have label $L(V_j)$.
        if number of nodes with label $L(V_i) \in \{\tau, \cdots, 2\tau - 1\}$ **then**
            Label $CORElabel$ is given to the nodes with label $L(V_i)$
            $CORElabel = CORElabel - 1$
        end if
    end if
end for
for $v = 1 : V$ **do**
    If $L(V_v) > 0$ **then** $L(V_v) = 0$ else $L(V_v) = -L(V_v)$
end for
return Labels L.

4 Results

4.1 Core clustering of simulated networks

In this section, we compare our CORE-clustering algorithms with other standard methods on simulated scale-free networks. Such complex networks are indeed common in the data-science literature. They contain a little amount of highly connected nodes (hubs), that we will assimilate to representative variables, and many poorly connected nodes.

4.1.1 Experimental protocol

To generate the synthetic networks, we first simulated the profile (observations) of representative variables and then the profile of remaining variables around these hubs. We then considered K different clusters of size p_{C_1}, p_{C_2}, ..., p_{C_K}. A simulated expression data set $X \in \mathbb{R}^{n \times p}$ is then composed of $p = p_{C1} + \ldots + p_{C_K}$ variables. In the cluster k, the observations are then simulated as follows: **(1)** Generate the observations $\mathbf{x}^{(1,k)} = (x_1^{(1,k)}, \ldots, x_n^{(1,k)})^\intercal$ of a representative variable using a normal distribution $\mathcal{N}(0,1)$. **(2)** Choose a minimum correlation r_{min} and a maximum correlation r_{max} between the representative variable and the other variables of the predefined cluster. In this paper, we always used $r_{max} = 1$ and $r_{min} = 0.5$. **(3)** Generate the profiles $\mathbf{x}^{(j,k)}$, with $j \in \{2, \ldots, p_{C_k}\}$, such that the correlation of the j-th profile with the profile of $\mathbf{x}^{(1,k)}$ is close to $r_j = r_{min} + \left(1 - \frac{j}{p_C}\right)(r_{max} - r_{min})$. For $i \in \{1, \ldots, n\}$, we then use $x_i^{(j)} = x_i^{(rep)} + (r_j^{-2} - 1)^{\frac{1}{2}} \epsilon_i^{(j)}$, where $\epsilon_i^{(j)} \sim \mathcal{N}(0, \alpha)$.

Three different type of networks were simulated using this protocol with different parameterizations. **(a)** The first type of network consisted in simulating $n = 100$ observations of 40 variables with $K = 2$ clusters of 20 variables. The additional noise was simulated with α ranging from 0.25 to 1.5. **(b)** The same protocol was used for the second type of networks, but $K = 5$ clusters of 7 variables were simulated. **(c)** The third kind of networks consisted in varying the number of the observations from $n = 5$ to $n = 30$, with $K = 5$ clusters having 50 to 100 variables.

Note finally that an amount of 30 networks of each type was generated to assess the stability of our methodology. This will indeed make it possible to draw in Figure 2.1 the box-plots of the clustering quality for each type of network.

4.1.2 Measure of clustering quality

There exists various criteria to measure a clustering quality. External indices such as impurity and Gini indices measure the extent to which the clusters match externally supplied class labels. Internal indices like the modularity and the intra-cluster to inter-cluster distance ratio are also used to measure the quality of a clustering structure without any external information. Such criteria are however not suitable here, as we do not clusterize all variables but rather extract core structures that emphasize representative variables. Thus, we propose the following criterion for simulated data for which the block structure of the similarity matrix is known: Let X^i, $i \in \{1, \cdots, p\}$ be the variables and C_j ($j \in [1, K]$) be the ground-truth CORE-clusters of variables, typically on synthetic data. As in

Section 2, we also denote $\widehat{S^u_{\xi,\tau}}$ ($u \in [1, U]$) the CORE-clusters predicted by our algorithm. In order to evaluate the quality of the prediction, we compute a score R defined as:

$$\forall u \in \{1, \cdots, U\}, R_u = \max_{j \in [1,K]} Card(X^i \in C_j \cap \widehat{S^u_{\xi,\tau}}), \tag{2.5}$$

where $1 \leq i \leq p$ and $R = \frac{1}{p} \sum_{u=1}^{U} R_u$. To compute this score, each R_u is equal to 0 if there is no overlap between C_j and any $\widehat{S^u_{\xi,\tau}}$, and is equal to the number of variables in C_j if a $\widehat{S^u_{\xi,\tau}}$ contains all the variables of C_j. A score R equals to 1 then means that a perfectly accurate estimation of the C_j was reached, and the closer to 0 its values the less accurate the CORE-clusters detection.

4.1.3 Results

We compared the standard and greedy CORE-clustering algorithms (Algs. 2 and 3) on these simulated datasets with two other graph-based clustering algorithms: the spectral clustering (Shi and Malik, 2000; Ng et al., 2002), available in the `R`-package `anocva`, and Louvain method for community detection (Blondel et al., 2008), available in the `R`-package `igraph`. Note that Louvain method requires as input parameter a graph modeling the dataset (the correlation matrix is transformed upstream into a graph) but not the final number of clusters. Boxplots of the computed scores R (see Eq. (2.5)) are shown in Figure 2.1. Note that in each boxplot, the dots represent the outlier scores which are either lower than $q_{0.25} - 1.5(q_{0.75} - q_{0.25})$ or higher than $q_{0.75} + 1.5(q_{0.75} - q_{0.25})$, where $q_{0.25}$ and $q_{0.75}$ are the first and third quartiles of the scores, respectively.

The subplots of Figure 2.1**(a-b)** show that Alg. 2 is more robust than Alg. 3, and gives slightly better results than spectral clustering and Louvain method, when the level of noise is high. The same applies when the sample size decreases in the subplots of Figure 2.1**(c)**.

4.2 Application to real biological data

We now present the results obtained on the classic Yeast dataset [1] (Spellman et al., 1998). With a total of about 1.3×10^6 weighted edges considered when representing the variables correlations in the graph $G(V, E)$, the CORE-clustering procedure required about 160 and 3 seconds with Algorithms 2 and 3, respectively.

4.2.1 Yeast dataset

The well-known synchronized yeast cell cycle data set of Spellman et al. (1998), includes 77 samples under various time during the cell cycle and a total of 6179 genes, of which 1660 genes are retained for this analysis after pre-processing and filtering. The goal of this analysis is then to detect CORE-clusters among the correlation patterns in the time series of yeast gene expressions measured along the cell cycle. Using this dataset, a measure of similarity between all gene pairs was measured with the absolute value of Pearson's correlation. A total of about 1.3×10^6 weighted edges are then considered when representing the variables correlations in the graph $G(V, E)$.

1. `https://archive.ics.uci.edu/ml/datasets/Yeast`

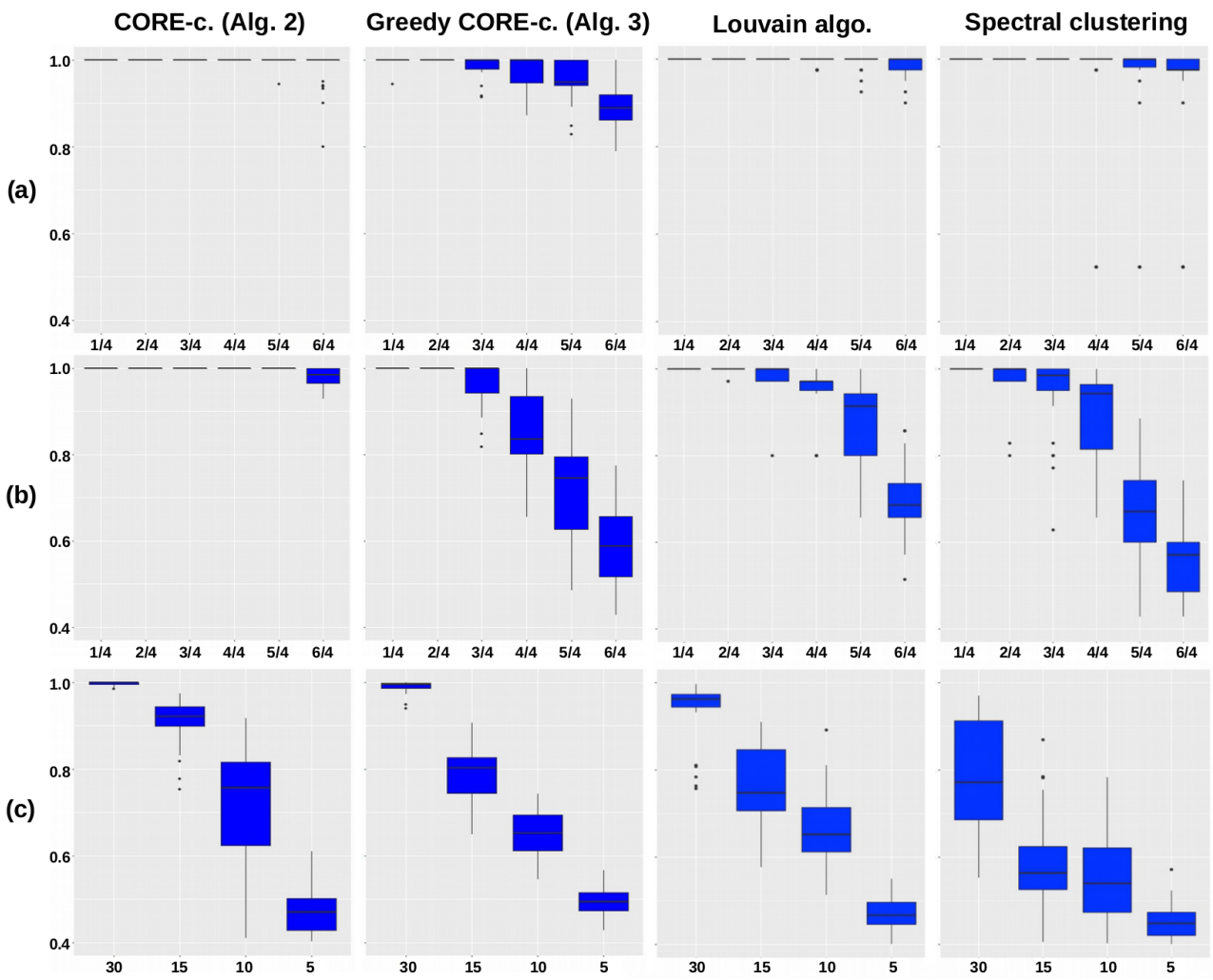

Figure 2.1 – Boxplots of the scores R (see Section 4.1.2) obtained on simulated datasets using the standard and the greedy CORE-clustering algorithms as well as the Louvain and the Spectral clustering algorithms. A score of 1 reflects a purely accurate detection of the simulated clusters and the lower this score, the lower the accuracy. The boxplots were obtained by reproducing 30 times the procedure of Section 4.1.1. (**a**) Two simulated clusters with noise levels ranging from 0.25 to 1.5. (**b**) Same as **(a)** with five simulated clusters. (**c**) Five clusters simulated using 30, 15, 10 and 5 observations and a noise level of 0.5.

4.2.2 Comparison of the two CORE-clustering algorithms

In order to compare the two proposed CORE-clustering algorithms described (Alg. 2 and Alg. 3) we tested them on the yeast dataset with $\tau = 30$ and $\xi = 0.75$. We indeed empirically considered that CORE-clusters containing 30 to 59 variables are reasonable to regularize the problem, and that 0.75 is a threshold above which the absolute value of the correlation between two variables reasonably shows

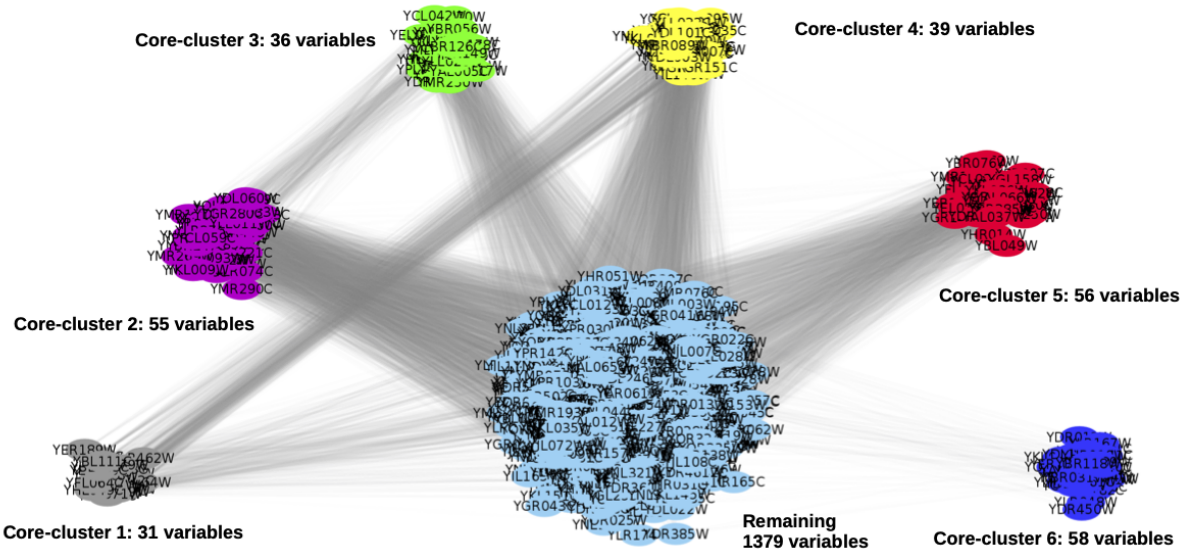

Figure 2.2 – CORE-clusters obtained using Alg. 2 on the yeast dataset of Spellman et al. (1998) and the granularity coefficient $\tau = 30$. CORE-clusters containing 30 to 59 variables are then estimated.

that their behavior is similar.

The clustering obtained using the standard algorithm, rows 1 to 19 of Alg. 2 is shown in Fig. 2.2. In this figure, the represented clusters 1 to 6 have a coherence $\mathbf{c}$ (see Eq. (2.3)) equals to $(0.79, 0.73, 0.76, 0.68, 0.79, 0.82)$. Only the clusters 1, 3, 5 and 6 are then considered as CORE-clusters (rows 20 to 25 of Alg. 2) and their representative variables are RV1=*YER190W*, RV3=*YLL026W*, RV5=*YDL003W* and RV6=*YGL120C*. Computational time for the clustering was about 160 seconds on an Intel(R) Core(TM) i7-6700HQ CPU at 2.60GHz.

The computations were much faster using the greedy algorithm Alg. 3. It indeed required about 3 seconds. An amount of 11 CORE-clusters was found. To interpret this result, we computed the coherence c (see Eq. 2.2) between the 4 representative variables obtained using Alg. 2 and the 11 ones obtained using Alg. 3. Interestingly, Alg. 3 selected *YLL026W*=RV3 and *YDL003W*=RV5. Other variables very close to RV1, RV5 and RV6 (with $c > 0.83$ *i.e.* higher that the highest $\mathbf{c}$ within the CORE-clusters) were also selected: *YHR219W*, *YLR103C*, *YLR276C* and *YLR196W*. Results equal or close to those obtained with Alg. 2 were then obtained. The representative variables *YDR418W*, *YML119W*, *YJL038C*, *YNL283C* and *YGR167W* were additionally found. In this experiment, Alg. 3 therefore selected more representative variables and has then naturally a larger score (see Eq. (2.4)) than by using Alg. 2. However, it also obviously captured different representative variables that would be gathered in the same CORE-cluster using Alg. 2. The two algorithms have therefore slightly different properties but lead to coherent results.

4.2.3 Impact of the number of observations

In order to evaluate the stability of the results with respect to the number of observations, we tested again Alg. 2 with the same parameters, but by using only the 30 first observations of the yeast dataset out of the 77 observations. Interestingly, *YDL003W*=RV5 was selected and *YML093W* which is very close to RV6 ($c > 0.86$) was also selected. The two other representative variables found, *YER190W* and *YLL026W*, were however not similar to RV1 or RV3. The information lost in the 47 observation we removed therefore did not allowed to recover the influence of RV1 and RV3 on the complex system but the 30 remaining observations contained a sufficient amount of information to detect RV5 and RV6 as influent variables. This suggests that the strategy detects stable representative variables, even when the number of observations is very low compared with the dimension of the observations.

4.2.4 Comparison with spectral-clustering

In order to compare our CORE-clustering strategy with a standard clustering approach, we finally estimated representative variables in the Yeast dataset as the center of clusters estimated using spectral clustering. The standard version of the spectral clustering available in *R* was used. Its main parameter is the number of seeds η used in the k-means part of the spectral clustering. When using η seeds, an amount of 1 to η clusters (with more than one variable) are estimated using k-means and all variables are contained in a cluster. In order to fairly compare the spectral clustering and the CORE-clustering approaches, we then clusterized the variables of the Yeast dataset using $\eta = \{5, 30, 50, 70, 110\}$. Note that we only tested an η higher than 77, due to the fact that $n = 77$ observations are known. We tested $\eta = 110$ to evaluate the spectral clustering behaviour with η slightly higher than n.

An amount of $\{3, 3, 6\}$ large clusters were obtained for $\eta = \{50, 70, 110\}$, respectively. For $\eta = \{5, 30\}$ only a single cluster gathering almost all variables was also obtained. The average coherence of the estimated clusters was 0.44 for a standard deviation of 0.06. The highest coherence was 0.63 which is clearly lower than the considered threshold of $\xi = 0.75$ we used with the CORE-clustering. This makes ambiguous the interpretation of the role played by the representative variables.

Finally, it is worth mentioning that all representative variables (genes) obtained using Alg. 2 have a known physiological function and only two variables out of eleven have an unknown function by using Alg. 3. For the spectral clustering with $\eta = 110$ four selected variables out of six have known functions. For $\eta = 70$, three variables with unknown functions were selected. For $\eta = 50$, two variables out of three with known functions were selected. For $\eta = 5$ and $\eta = 30$ the single selected variable was the center of all variables with respect to the center definition in Section 3.3, and turns out to have a known function. It therefore appears in this experiment that the representative variables obtained using CORE-clustering are more interpretable than those estimated using spectral clustering.

4.3 Application to the U.S. road network

We now assess the CORE-clustering algorithms on the U.S. road network out of the 9th DIMACS Implementation challenge [2]. Our goal here is to discuss the pertinence of the detected CORE-clusters, and not specifically their representative variables, on a large scale and straightforwardly interpretable

2. `http://users.diag.uniroma1.it/challenge9/`

dataset. The graph contains here 2.4×10^7 nodes, each of them representing a crossing of the U.S. road/streets network, and 5.8×10^7 arcs representing the part of the roads/streets between two crossings. Interestingly the distance between the crossings is also associated to each edge of the graph.

To assess the CORE-clustering algorithms on the U.S. road network, we first transformed the distances between the crossings into weights that are higher and higher for increasingly closer crossings. A weight $\max\left((4 \times 10^3 - l)/(4 \times 10^3), 10^{-4}\right)$ was specifically given to each edge, where l is its original length in feet.

We show Figure 2.3, the CORE-clusters [3]obtained using the greedy algorithm with $\tau = 5 \times 10^4$ and $\xi = 10^{-3}$. This means that each CORE-cluster contains a road network of 5×10^4 to 10^5 crossings for which one can travel from any crossing to another one by only using streets of less than 3996 feet (about 1.2 km) between two crossing. As expected, the clusters represented in Figure 2.3 correspond to the 18 largest urban areas in the U.S. Note that two of them are made of three CORE-clusters (New-York City and Los-Angeles), and two other ones (Miami and Chicago) are made of two CORE-clusters. This is due to an algorithmic choice, discussed in Sections 3.2.2 and 3.2.3, which leads to the detection of CORE-clusters containing τ to $2\tau - 1$ nodes by using the proposed CORE-clustering algorithms. As discussed in Section 2.6, an interesting strategy if connected CORE-clusters are found consists in running again the CORE-clustering algorithms with higher values of τ or ξ, to potentially detect more robust CORE-clusters : By reproducing this test with $\tau = 10^5$ instead of $\tau = 5 \times 10^4$, we now try to find CORE-clusters containing 10^5 to 2×10^5 crossings, only 5 urban areas are found: New-York City (2 CORE-clusters), Los-Angeles, Chicago, Miami and San-Fransisco. Now, by using $\tau = 5 \times 10^4$ and now $\xi = 0.5$, *i.e.* with distances between the crossings lower than about 2000 feet (about 0.6 km) instead of 3996 feet, only 3 urban areas are found: New-York City (2 CORE-clusters), Los-Angeles and Chicago.

What is interesting here in terms of interpretability is that we have been able to select specific subparts of the U.S. road network by explicitly controling the size of the clusters or a specific level of density in the network. This notion of interpretability can be further discussed by observing the CORE-clusters obtained in the New-York city urban area, as shown Figure 2.3-(b-c-d). By using $\tau = 5 \times 10^4$ and $\xi = 10^{-3}$, the three CORE-clusters split the densest parts of the New York City urban area into the New-Jersey state, Long-Island and the rest of New-York state. When having larger CORE-clusters, *i.e.* with $\tau = 10^5$, the New-Jersey cluster expands and the densest parts of the two New-York state clusters are merged. Now, when enforcing instead a stronger coherence inside of the clusters, *i.e.* with $\tau = 5 \times 10^4$ and $\xi = 0.5$, the New-Jersey and Long-Island CORE-clusters remain stable, but the Manhattan/mainland New York state cluster is not large and dense enough, so it is not captured as a CORE-cluster. Note that each CORE-cluster estimation required about 50 seconds and 2.4GB here without any parallelization.

5 Conclusion

Although complex systems in high dimensional spaces with a limited number of observations are quite common across many fields, efficient methods to treat the associated problem of graph clustering

3. Clusters represented using *MI Map Tools: GeoPlotter*: `https://mobisoftinfotech.com/tools/plot-multiple-points-on-map/`

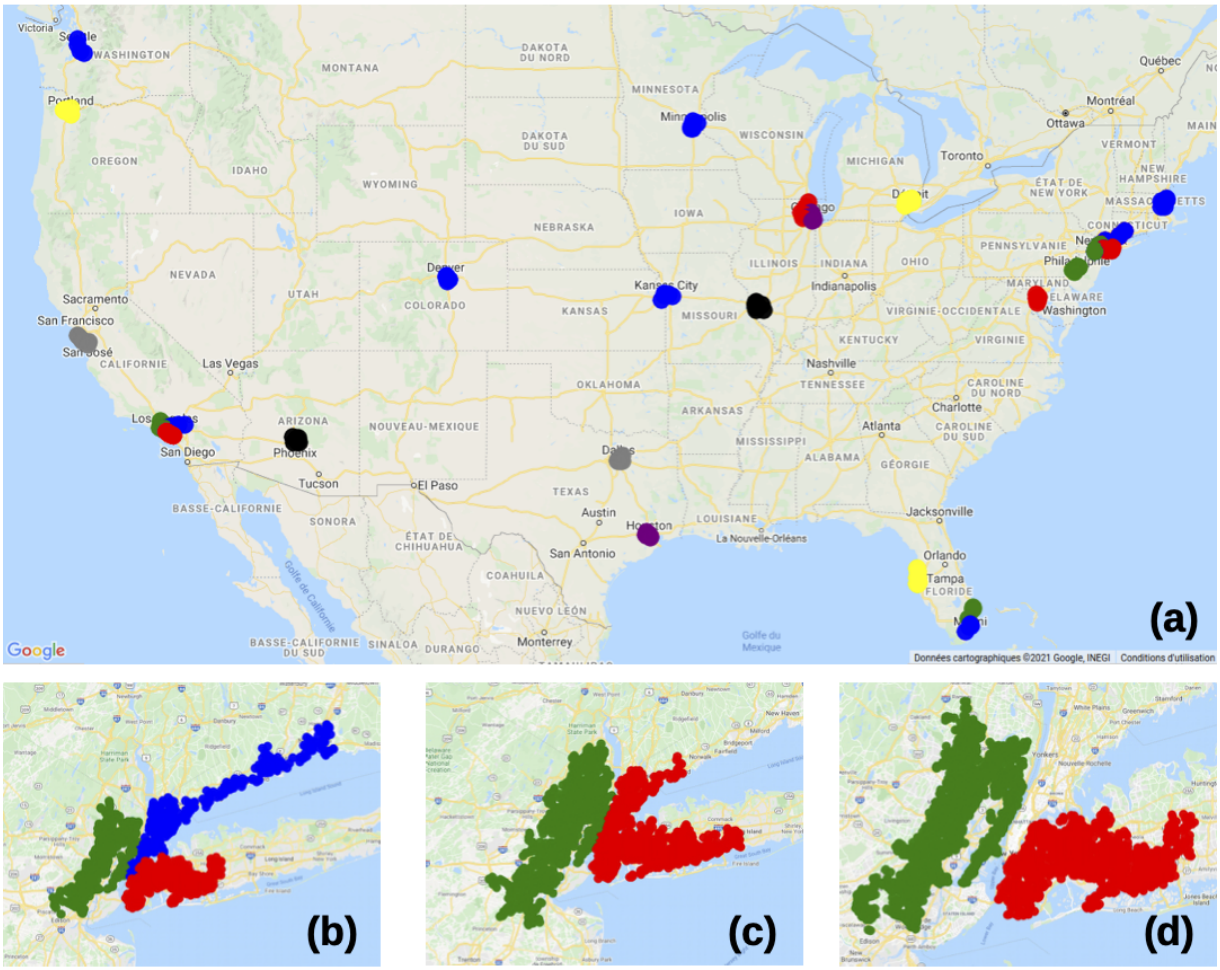

Figure 2.3 – Results obtained using the greedy CORE-clustering algorithm on the U.S. road network ($\approx 2.4 \times 10^7$ nodes) in about one minute. (**a** and **b**) CORE-clusters obtained using $\tau = 5 \times 10^4$ and $\xi = 10^{-3}$ represented in the U.S. and in New-York City urban area. (**c**) CORE-clusters obtained using $\tau = 1 \times 10^5$ and $\xi = 10^{-3}$ represented in New-York City urban area only. (**d**) CORE-clusters obtained using $\tau = 5 \times 10^4$ and $\xi = 0.5$ represented in New-York City urban area only. Background: *Google maps*. Clusters represented using *MI Map Tools: GeoPlotter*

.

is an ambiguous task. Some of these techniques, based on assumptions in view of controlling the variables contribution to the global clustering, often do not allow to select the best graph partition. In reply to this issue, we developed a formalism based on an original graph clustering strategy with specific properties. This formalism makes it possible robustly identify groups of representative variables of the studied system by tuning two intuitive parameters: (1) the minimum number of variables in each CORE-cluster, and (2) a minimum level of similarity between all the variables of a CORE-cluster. Its effectiveness was further satisfactorily assessed on simulated data and on real

datasets.

From a methodological perspective, an interesting research direction would be to mix Algorithms 2 and 3 into a single hybrid top-down and bottom-up optimization scheme. Our goal would be to scale well to very high dimensional datasets, as when using Algorithm 3, while being as robust as Algorithm 2 when potential CORE-clusters are coarsely identified. When the CORE-clusters found by either Algorithm 2 or 3 are very large, a stochastic strategy could also make faster the detection of their representative variables. Although this secondary part of our methodology has been addressed by using a standard Dijkstra's algorithm (see Section 3.3), it could be addressed by using an extension of Gadat et al. (2018) where the optimal paths between all variables would not be pre-computed prior to the central variable detection. Application of our formalism in various fields such as gene regulatory networks, social networks or recommender systems would also be of interest. Our formalism is indeed sufficiently flexible to incorporate different types of similarity measures between the observed variables.

Note finally that our formalism was implemented in *C++* and wrapped in a *R* package, which is freely available on sourceforge [4], so these results can be easily reproduced.

6 Appendices

6.1 Pertinence of the representative variable detection model

We illustrate in this section the notion of coherence defined in Section 2.2. We simulated an adjacency matrix that mimics the absolute values of the Pearson correlations between 15 variables. This symmetric matrix is shown Fig. 2.4(top-left) and represents graph G_1 following the method of Section 2.1. Each of its values is then equal to a similarity level encoded in $w_{i,j}$ between two variables X^i and X^j, where i and j are in $\{0, \cdots, 14\}$. We can clearly remark that it contains two blocks of related variables $S^1_{Ref} = \{X^0, \cdots, X^6\}$ and $S^2_{Ref} = \{X^8, \cdots, X^{14}\}$ and an independent variable X^7. In addition, variables X^3 and X^{11} are slightly more related to other variables in S^1_{Ref} and S^2_{Ref},respectively. They can then be considered as the most pertinent representative variables in these blocks. We also added to G_1 undesirable relations having an intermediate level between the variables $\{X^1, \cdots, X^5\}$ and $\{X^9, \cdots, X^{13}\}$ and saved the result in graph G_2, as shown Fig. 2.4(top-right). More quantitatively, the reference blocks S^1_{Ref} and S^2_{Ref} have inner similarities sampled following the normal law $\mathcal{N}(0.75, 0.1)$, the background ones are sampled following $\mathcal{N}(0., 0.1)$, and in Fig. 2.4, the undesirable relations are sampled following $\mathcal{N}(0.37, 0.1)$. In each reference block, the relation between the simulated reference variables and other variables are finally sampled following $\mathcal{N}(0.9, 0.1)$. The norm of these similarities are then considered and the similarities higher than one are set to one.

We will measure hereafter the coherence of different variable subsamples to illustrate this notion and make clear its interest in the CORE-clustering context. As the CORE-clustering algorithms proposed in Section 3 use maximal spanning trees, we will additionally discuss the corresponding coherences obtained on the maximal spanning trees of G_1 and G_2, which are shown in Fig. 2.4

4. `https://es.sourceforge.net/projects/core-clustering/`

	S^1_{T1}	S^2_{T1}	S^1_{T2}	S^2_{T2}	S^1_{T3}	S^2_{T3}	S^1_{T4}	S^2_{T4}
G_1	0.77	0.85	0.77	0.77	0.18	0.25	0.15	0.19
G_2	0.77	0.85	0.77	0.77	0.18	0.25	0.40	0.47

Table 2.1 – Coherence of the CORE-clusters tested Appendix 6.1 on G_1 and G_2. Corresponding representative variables are given table 2.2.

	$\hat{X}^1_{T1}$	$\hat{X}^2_{T1}$	$\hat{X}^1_{T2}$	$\hat{X}^2_{T2}$	$\hat{X}^1_{T3}$	$\hat{X}^2_{T3}$	$\hat{X}^1_{T4}$	$\hat{X}^2_{T4}$
G_1	4	11	3	11	5	11	6	11
G_2	4	11	3	11	5	11	3	11

Table 2.2 – Representative variables of the CORE-clusters tested Appendix 6.1 on G_1 and G_2. Corresponding coherences are given table 2.1.

(bottom). The estimated representative variables will finally be given in all tests to make sure their estimation is robust.

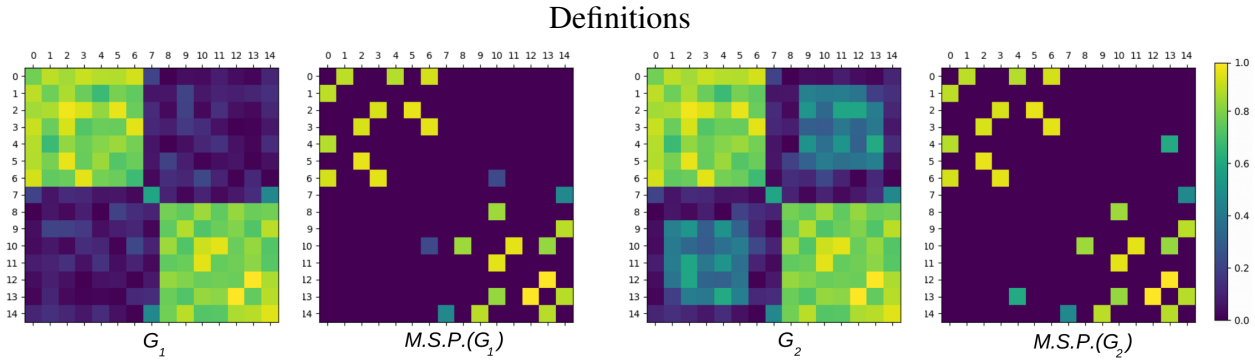

Figure 2.4 – Adjacency matrices of the simulated graphs G_1 and G_2 of Appendix 6.1 and their maximum spanning trees (ST).

6.1.1 Illustration of the coherence

We give Table 2.1 the coherence of four pairs of tested CORE-clusters on G_1, G_2 and their maximal spanning trees. Corresponding estimated representative variables are given in Table 2.2. The tested CORE-clusters are: **(Test 1)** The reference blocks of variables S^1_{Ref} and S^2_{Ref}, *i.e.* $S^1_{T1} = \{X^0, \cdots, X^6\}$ and $S^2_{T1} = \{X^8, \cdots, X^{14}\}$. **(Test 2)** Subsamples of S^1_{Ref} and S^2_{Ref} of size three, *i.e.* $S^1_{T2} = \{X^2, X^3, X^4\}$ and $S^2_{T2} = \{X^{10}, X^{11}, X^{12}\}$. **(Test 3)** Samples S^1_{Ref} and S^2_{Ref} in which a variable was replaced with the independent variable X^7, *i.e.* $S^1_{T3} = \{X^1, \cdots, X^7\}$ and $S^2_{T3} = \{X^7, \cdots, X^{13}\}$. **(Test 4)** Samples S^1_{Ref} and S^2_{Ref} in which the variables X^4 and X^{10} were swapped, *i.e.* $S^1_{T4} = \{X^0, \cdots, X^3, X^{10}, X^5, X^6\}$ and $S^2_{T4} = \{X^8, X^9, X^4, X^{11}, \cdots, X^{14}\}$.

Let us first interpret the results in Table 2.1. The coherences first row on G_1 in tests $T1$ and $T2$ first show that similar coherences were obtained with CORE-clusters of size 7 and 3, although the coherences are slightly higher for smaller CORE-clusters. These coherences are also clearly higher to those obtained in tests $T3$ and $T4$ where all variables are not contained in the same simulated block. Interestingly, the results obtained on graph G_2 are similar to those obtained on G_1. The only difference here is that the coherences of $T4$ are slightly higher than in G_1 but still relatively low. Note that the tested CORE-clusters may lead to disconnected subgraphs when tested on the maximum spanning trees. Eq. (2.2) does not make sense in this case. When the simulated subgraphs were connected (12 cases out of 16), we obtained the same coherences on the maximum spanning trees and the whole graph. As discussed in Subsection 2.4, the coherence of a given CORE-cluster on a maximum spanning tree is indeed lower or equal to the coherence of the same CORE-cluster on the whole graph. We however computed the representative variables in all tested cases as the centrality measure makes sense even on disconnected subgraphs (see Subsection 3.3). The results in Table 2.2 show that the estimated representative variables are always in the reference sets S^1_{Ref} and S^2_{Ref} in the tested configurations. They also correspond to the simulated representative variables, X^3 and X^{11}, in most cases or are very close to these variables. Note that slightly inaccurate reference variables were detected in S^1_{Ref}, and we can clearly see Fig. 2.4(top) that the influence of its simulated representative variable is less obvious than in set S^2_{Ref}.

6.1.2 Influence of the undesirable relations

We further study the influence of the undesirable relations between $\{X^1, \cdots, X^5\}$ and $\{X^9, \cdots, X^{13}\}$ in graph G_2 by simulating these relations with different strengths. In the previous subsection undesirable relations were sampled following $\mathcal{N}(\mu, 0.1)$, where $\mu = 0.37$. Here, we sampled 100 graphs G_2 for each strength $\mu \in \{0.1, 0.40, 0.60, 0.80, 0.90\}$. For each graph, we then measured the portion of representative variable estimates in the true reference block of variables and the portion of representative variable estimates that are the ground truth representative variables.

Results are given in Table 2.3 and show that the representative variables detection is particularly stable in these tests, even for large values of μ. All estimated representative variables are indeed in the true block of variables except in Test 4 (where two variables of the reference sets are swapped) with $\mu = 0.9$, *i.e.* with the same level of similarity as between the blocks representative variables and the other variables they contain. False estimations are however uncommon even in this case. Exact estimates of the representative variables are naturally less frequent as this test is more strict. They are however always clearly higher than random estimations which would have portions equal to 0.14. The estimates of pertinent representative variable therefore appears as robust, even with strong undesirable relations in the variable similarities and tested CORE-clusters which contain undesirable variables.

	Test 1		Test 2		Test 3		Test 4	
	G_2	$ST(G_2)$	G_2	$ST(G_2)$	G_2	$ST(G_2)$	G_2	$ST(G_2)$
$\mu = 0.1$	1. (0.88)	1. (0.82)	1. (0.79)	1. (0.80)	1. (0.40)	1. (0.75)	1. (0.59)	1. (0.79)
$\mu = 0.4$	1. (0.87)	1. (0.79)	1. (0.78)	1. (0.83)	1. (0.46)	1. (0.80)	1. (0.75)	1. (0.84)
$\mu = 0.6$	1. (0.88)	1. (0.79)	1. (0.78)	1. (0.84)	1. (0.44)	1. (0.74)	1. (0.88)	1. (0.78)
$\mu = 0.8$	1. (0.92)	1. (0.72)	1. (0.75)	1. (0.69)	1. (0.47)	1. (0.62)	1. (0.93)	0.96 (0.71)
$\mu = 0.9$	1. (0.92)	1. (0.59)	1. (0.70)	1. (0.48)	1. (0.49)	1. (0.44)	1. (0.93)	0.89 (0.59)

Table 2.3 – Portion of representative variable estimates contained in the proper reference block of variables (*main value*) and corresponding to the ground truth representative variables (*between brackets*). Each portion was computed on 100 simulated graphs (G_2) and their corresponding maximum spanning trees ($ST(G_2)$). For each group of 100 graphs, a different strength μ of the undesirable relations is tested.

Chapter 3

ℓ_1-spectral clustering algorithm: a robust spectral clustering using Lasso regularization

This chapter describes the main elements of a submitted paper but in a different form from Champion et al. (2021b), a joint work with Magali Champion, Rémy Burcelin, Mélanie Blazère and Jean-Michel Loubes.

Abstract

Detecting cluster structure is a fundamental task to understand and visualize functional characteristics of a graph. Among the different clustering methods available, spectral clustering is one of the most widely used due to its speed and simplicity, while still being sensitive to perturbations imposed on the graph. This paper presents a robust variant of spectral clustering, called ℓ_1-spectral clustering, based on Lasso regularization and adapted to perturbed graph models. By promoting sparse eigenbases solutions of specific ℓ_1-minimization problems, it detects the hidden natural cluster structure of the graph. The effectiveness and robustness to noise perturbations of the ℓ_1-spectral clustering algorithm is confirmed through a collection of simulated and real biological data.

1 Introduction

Graphs play a central role in complex systems as they can model interactions between variables of the system. They are commonly used in a wide range of applications, from social sciences (*e.g.* social networks (Handcock and Gile, 2010)) to technologies (*e.g.* telecommunications (Smith, 1997), wireless sensor networks (Akyildiz et al., 2002)) or biology (gene regulatory networks (Davidson and Levin, 2005), metabolic networks (Jeong et al., 2000)). One of the most relevant features when analyzing graphs is the identification of their underlying structures, such as cluster structures, generally defined as connected subsets of nodes that are more densely connected to each other than to the rest of the graph. These clusters can provide an invaluable help in understanding and visualizing the functional components of the whole graph (Girvan and Newman, 2002; Newman and Girvan, 2004;

Abbe, 2017). For instance, in genetics, groups of genes with high interactions are likely to be involved in a same function that drives a specific biological process.

Since the pioneering exploratory works in the early 50s, a large number of clustering methods have launched. Among them, partitioning algorithms, which include the well-known k-means (MacQueen, 1967), classify nodes into a predefined number of groups based on a similarity measure and hierarchical clustering algorithms (Hastie et al., 2001) build a hierarchy of clusters through dendrogram representations. More recently, spectral clustering algorithms, popularized over years by Shi and Malik (2000); Ng et al. (2002), particularly draw the attention of the community research due to their speed, simplicity and numerical performances. As its name suggest, spectral clustering algorithms mainly use the spectral properties of the graph by (i) computing the eigenvectors of the associated Laplacian matrix (or one of its derivatives), which gives information about the structure of the graph, and (ii) performing k-means on it to recover the induced cluster structure. A large number of extensions of the original spectral clustering algorithm, as presented in Luxburg (2007), have been proposed, with applications to different fields (Zelnik-Manor and Perona, 2005; Wang and Davidson, 2010; Li et al., 2019).

While spectral clustering is widely used in practice, handling noise sensitivity remains a tricky point (Bojchevski et al., 2017), mainly due to the k-means algorithm, which is highly sensitive to noise. This issue has been considerably studied with extensions of the k-means to noisy settings so that it recovers the cluster structure in spite of the unstructured part of the input data (Tang and Khoshgoftaar, 2004; Pelleg and Baras, 2007). More generally, the robustness of spectral clustering algorithms has recently been investigated for perturbed graphs derived from stochastic block models (SBM) (Stephan and Massoulié, 2019; Peche and Perchet, 2020). In this paper, we develop an alternative method of the spectral clustering, called ℓ_1-spectral clustering algorithm and based on Lasso regularization (Tibshirani et al., 2001). Note that research papers have explored regularized spectral clustering to robustly identify clusters in large networks. Although Zhang and Rohe (2018) and Joseph and Yu (2016) show the effect of regularization on spectral clustering through graph conductance and respectively through stochastic block models. Equally, Lara and Bonald (2020), shows on a simple block model that the spectral regularization separates the underlying blocks of the graph.

However, in our model, as in the spectral clustering algorithm, we carefully explore the underlying structure of the graph through the Laplacian matrix spectrum to cluster nodes. However, by directly promoting a sparse eigenvectors basis solution to an ℓ_1-norm optimization problem, it does not require the k-means step to extract clustering structures, making it more robust in highly perturbed graph situations.

The paper is organized as follows: in Section 2, we introduce some preliminary concepts about graph clustering and more specifically spectral clustering. In Section 3 and 4, we present the ℓ_1-spectral clustering we developed, from a theoretical and an algorithmic point of view. In Section 5, we finally show its efficiency and accuracy through experiments on simulated and biological real data set and compare it with state-of-the-art clustering methods.

2 Reminders about graph and spectral clustering

2.1 Graphs modeling and notations

This work considers the framework of an unknown undirected graph $\mathcal{G}(V, E)$, with no retroactive loop, consisting of p vertices $V = \{1, \ldots, p\}$ and a set of edges $E \subseteq V \times V$ connecting each pair of vertices. As usual, the graph $\mathcal{G}$ is represented by its associated adjacency matrix $A = (A_{ij})_{(i,j)\in E}$ of size $p \times p$, whose non-zero elements correspond to the edges of $\mathcal{G}$:

$$\forall (i,j) \in [\![1,p]\!]^2, \;\; A_{ij} = \begin{cases} 1 \text{ if } (i,j) \in E, \\ 0 \text{ otherwise.} \end{cases}$$

As $\mathcal{G}$ is undirected with no retroactive loop, the adjacency matrix A is symmetric with zero on its diagonal. Before turning to the next section, we recall some useful graph definitions.

Definition 3 *The degree d_i of a node $i \in V$ of $\mathcal{G}$ is defined as the number of edges that are incident to i: $d_i = \sum_{j=1}^{p} A_{ij}$. The induced degree matrix D is then the $p \times p$ matrix containing $(d_1, \ldots, d_p)$ on its diagonal and zero elsewhere:*

$$D = \textit{diag}\,(d_1, \ldots, d_p).$$

Definition 4 *A connected component C of $\mathcal{G}$ is a subset of nodes from V such that each pair of nodes of C is connected by a path and there is no connection between vertices in C and outside C. Connected components $C_1, \ldots, C_k$ are a k-partition of the set V of vertices if the three following conditions hold:*

1. *they are non-empty: $\forall i \in [\![1,k]\!], C_i \neq \emptyset$,*
2. *they are pairwise disjoints: $\forall (i,j) \in [\![1,k]\!]^2, C_i \cap C_j = \emptyset$,*
3. *their union form the set of all vertices: $\bigcup_{i=1}^{k} C_i = V$.*

Definition 5 *Let $C_1, ..., C_k$ be a k-partition of the set of vertices V of $\mathcal{G}$. Then, the indicators $(\mathbf{1}_{C_i})_{i\in\{1,\ldots,k\}}$ of this partition are defined as the vectors of size p, whose coefficients satisfy:*

$$\forall i \in [\![1,k]\!], \forall j \in [\![1,p]\!], \;\; (\mathbf{1}_{C_i})_j = \begin{cases} 1 \;\; \textit{if vertex } j \textit{ belongs to } C_i, \\ 0 \;\; \textit{otherwise.} \end{cases}$$

In the present paper, we assume that the graph $\mathcal{G}$ is the union of k complete graphs, whose set of vertices $C_1, \ldots, C_k$ form a k-partition of $\mathcal{G}$. We denote by $c_1, \cdots, c_k$ their respective size ($\sum_{i=1}^{k} c_i = p$). To simplify, we assume that the nodes, labeled from 1 to p, are ordered with respect to their block membership and the size of the blocks. From a matrix point of view, the associated adjacency matrix A is a k-block diagonal matrix of size $p \times p$ of the form:

$$A = \begin{bmatrix} \underbrace{\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}}_{c_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \underbrace{\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}}_{c_k} \end{bmatrix}. \tag{3.1}$$

2.2 Graph clustering through spectral clustering

Graph clustering consists in grouping the vertices of the graph $\mathcal{G}$ into clusters according to its edge structure. Whereas some of the most traditional clustering algorithms are based on partitions (*e.g.* k-means) and hierarchies (*e.g.* hierarchical clustering algorithm), spectral clustering takes advantage of the spectral properties of the graph. A large number of spectral clustering algorithms exists in the literature. The most common version, presented in Luxburg (2007) and recapped in Algorithm 4 below, uses the properties of the Laplacian matrix (Definition 6) to detect clusters in the graph.

Definition 6 *Given a graph $\mathcal{G}$, the Laplacian matrix L is defined as:*

$$L = D - A,$$

where A is the adjacency matrix and D the degree matrix associated to $\mathcal{G}$.

Algorithm 4 Spectral clustering algorithm

Require: $\mathcal{G}$ a graph, A its associated adjacency matrix, $\hat{k}$ number of clusters to build.

1: Compute the Laplacian matrix $L = D - A$.
2: Perform the spectral decomposition of L and store the $\hat{k}$ first eigenvectors $U := (u_1, \cdots, u_{\hat{k}})$.
3: Cluster U with the k-means algorithm into clusters $C_1, \cdots, C_{\hat{k}}$.
4: **return** Clusters $C_1, \cdots, C_{\hat{k}}$.

By definition, the diagonal of L is equal to the degrees of the nodes. Moreover, in the ideal case where $\mathcal{G}$ has an underlying partition form with k connected components and a block diagonal adjacency matrix A, as given in Equation (3.1), the eigenvalue 0 of L is of multiplicity k and the associated eigenvectors correspond to the indicator vectors of the k components. These k components can then be recovered only by performing spectral decomposition of L. However, in the perturbed case, any perturbation caused by introducing and/or removing edges between and/or inside the components makes $k - 1$ of the k eigenvalues 0 slightly larger than 0 and changes the corresponding eigenvectors. The final cluster structure is thus no longer explicitly represented. The spectral clustering algorithm then uses the k-means algorithm on these eigenvectors to discover the hidden underlying structure, which is hampered by perturbations.

Since the first development of the spectral clustering algorithm, it has been studied a lot and extended many times in different communities (Hagen and Kahng, 1992; Hendrickson and Leland, 1995; Pothen, 1997; Shi and Malik, 2000; Ng et al., 2002; Zelnik-Manor and Perona, 2005) with powerful results. Refinements include the use of normalized versions of the Laplacian matrix, such as the symmetric and the random walk normalized ones (Luxburg, 2007). Nevertheless, the performances of the spectral clustering have shown to be very sensitive to perturbations, which often occurs when dealing with real data (Bojchevski et al., 2017). To provide more robustness with respect to perturbations, we thus developed the ℓ_1-spectral clustering algorithm, described in Section 3.

3 An ℓ_1-version of the spectral clustering algorithm

In this section, we describe the ℓ_1-spectral clustering algorithm we developed as an alternative to the standard spectral clustering for clustering perturbed graphs. In this context, to ensure a good recovery of the connected components, the eigenvectors basis should be carefully defined. The key point is to replace the k-means procedure, which fails while the perturbation grows, by selecting relevant eigenvectors that provide useful information about the graph structure. As the spectral clustering algorithm, the ℓ_1-spectral clustering focuses on the spectral properties of the graph.

Let $\mathcal{G} = (V, E)$ be a graph formed of k connected components, as defined in Section 2, and A its associated adjacency matrix. We denote by $(\lambda_i)_{1\leq i\leq p}$ the p-eigenvalues of A, sorted in increasing order:

$$\lambda_1 \leq ... \leq \lambda_p,$$

and $v_1, ..., v_p$ their associated eigenvectors. We define by $\mathcal{V}_k$ the eigenspace generated by the k largest eigenvectors:

$$\mathcal{V}_k := \text{Span}(v_{p-k+1}, ..., v_p).$$

In the ideal case, where the graph is not perturbed, the indicators $(\mathbf{1}_{C_i})_{1\leq i\leq k}$ of the connected components $C_1, \ldots, C_k$ correspond exactly to the eigenvectors of the Laplacian matrix L associated to the eigenvalue 0 of multiplicity k (see Section 2.2). As regards the adjacency matrix A, these indicators correspond this time to the k eigenvectors $v_{p-k+1}, \ldots, v_p$, associated to the k largest eigenvalues $\lambda_{p-k+1}, ..., \lambda_p$. In the perturbed case, unlike the traditional spectral clustering, the ℓ_1-spectral clustering algorithm does not directly use the subspace $\mathcal{V}_k$ to recover the k connected components but computes another eigenbasis that promotes sparse solutions, as detailed in the next sections.

3.1 General ℓ_0-minimization problem

Propositions 3 and 4 below show that the connected components indicators $(\mathbf{1}_{C_i})_{i\in\{1,...,k\}}$ are solutions of ℓ_0-minimization problems.

Proposition 3 *The minimization problem*

$$\underset{v\in\mathcal{V}_k\setminus\{0\}}{\arg\min} \ \|v\|_0 \tag{$\mathcal{P}_0$}$$

has a unique solution (up to a constant) given by $\mathbf{1}_{C_1}$.

In other words, $\mathbf{1}_{C_1}$ is the sparsest non-zero eigenvector in the space spanned by the eigenvectors associated to the k largest eigenvalues.

Proof *We recall that, for all* $v \in \mathbb{R}^p$, $\|v\|_0 = |\{j \in [\![1,p]\!],\ v_j \neq 0\}|$. *Let* $v \in \mathcal{V}_k \backslash \{0\}$. *As* $(\mathbf{1}_{C_j})_{1\leq j\leq p} \in \mathcal{V}_k$, v *can be decomposed as* $v = \sum_{j=1}^k \alpha_j \mathbf{1}_{C_j}$ *where* $\alpha = (\alpha_1, \ldots, \alpha_k) \in \mathbb{R}^k$ *and there exists* $j \in \{1, ..., k\}$ *such that* $\alpha_j \neq 0$. *By definition of the* ℓ_0*-norm, we then have:*

$$\|v\|_0 = \mathbf{1}_{\alpha_1 \neq 0} c_1 + \cdots + \mathbf{1}_{\alpha_k \neq 0} c_k, \tag{3.2}$$

with $c_1 \leq ... \leq c_k$ *the sizes of the* k *connected components. The solution of* $(\mathcal{P}_0)$*, which minimizes Equation (3.2), is thus given by setting* $\alpha = (\alpha_1, 0, \ldots, 0)$ *with* $\alpha_1 \neq 0$.

Proposition 3 can then be generalized to iteratively find the indicators associated to the largest connected components introducing sparsity and orthogonality constraints. For $i \in [\![2,k]\!]$, let $\mathcal{V}_k^i$ refers to:

$$\mathcal{V}_k^i := \{v \in \mathcal{V}_k,\ \ \forall l = 1, \ldots, i-1,\ \ v \perp \mathbf{1}_{C_l}\}.$$

Proposition 4 *Let* $i \in [\![2,k]\!]$. *The minimization problem*

$$\underset{v \in \mathcal{V}_k^i \backslash \{0\}}{\arg\min}\ \|v\|_0 \tag{$\mathcal{P}_0^i$}$$

has a unique solution (up to a constant) given by $\mathbf{1}_{C_i}$.

Solving $(\mathcal{P}_0)$ and $(\mathcal{P}_0^i)_{2\leq i\leq k}$ is a NP-hard problem, which is not computationally feasible. To tackle this issue, the classical idea consists in replacing the ℓ_0-norm by its convex relaxation, the ℓ_1-norm, defined for all $v \in \mathbb{R}^p$ as $\|v\|_1 = \sum_{1\leq j\leq p} |v_j|$.

In the next section, we show that the solution of the ℓ_0 optimization problems remains the same by replacing the ℓ_0-norm by the ℓ_1-norm, at the price of slight constraints on the connected components.

3.2 Relaxed ℓ_1-minimization problem

From now on, we assume that we know one representative element for each component, that is a node belonging to each component, denoted by $(i_1, ..., i_k)$ thereafter. Let $\tilde{\mathcal{V}}_k = \{v \in \mathcal{V}_k, v_{i_1} = 1\}$. Then, it is straightforward to see that the indicator vector of the smallest component is solution to the following optimization problem:

Proposition 5 *The minimization problem*

$$\underset{v \in \tilde{\mathcal{V}}_k}{\arg\min}\ \|v\|_1 \tag{$\mathcal{P}_1$}$$

has a unique solution given by $\mathbf{1}_{C_1}$.

Proof *We recall that, for all $v \in \mathbb{R}^p$, $\|v\|_1 = \sum_{j=1}^p |v_j|$. Let $v \in \tilde{\mathcal{V}}_k$. As $(\mathbf{1}_{C_j})_{1\leq j\leq p} \in \mathcal{V}_k$, v can be decomposed as $v = \sum_{j=1}^k \alpha_j \mathbf{1}_{C_j}$ where $\alpha = (\alpha_1, \ldots, \alpha_k) \in \mathbb{R}^k$ and there exists $j \in \{1, \ldots, k\}$ such that $\alpha_j \neq 0$. By definition of the ℓ_1-norm, we then have:*

$$\|v\|_1 = |\alpha_1|c_1 + \cdots + |\alpha_k|c_k, \tag{3.3}$$

with $c_1 \leq ... \leq c_k$ the sizes of the k connected components. The solution of ($\mathcal{P}_1$), which minimizes Equation (3.3), is thus given by setting $\alpha = (\alpha_1, 0, \ldots, 0)$ with $\alpha_1 = 1$.

To simplify and without loss of generality, we assume that i_1 corresponds to the first node. We can then rewrite ($\mathcal{P}_1$) as:

$$\underset{\substack{v\in\mathbb{R}^{p-1}\\(1,v)^T\in\mathcal{V}_k}}{\arg\min} \ \|v\|_1.$$

Constraints in ($\mathcal{P}_1$) can be converted into the following equality contraints:

Proposition 6 *Let $U_k := (v_1, ..., v_{p-k})$ the matrix formed by the eigenvectors associated with the $p-k$-smallest eigenvalues. We denote by w^T its first row and W^T the matrix obtained after removing w^T from U_k:*

$$U_k := (v_1, ..., v_{p-k}) = \begin{bmatrix} \boxed{w^T} \\ \boxed{W^T} \end{bmatrix} \tag{3.4}$$

The minimization problem

$$\underset{\substack{v\in\mathbb{R}^{p-1}\\Wv=-w}}{\arg\min} \ \|v\|_1 \tag{$\tilde{\mathcal{P}}_1$}$$

has a unique solution v^* such that $(1, v^*)^T = \mathbf{1}_{C_1}$.

Proof *Since A is symmetric, its eigenvectors form an orthogonal basis and, for all $v \in \mathcal{V}_k$, we have $U_k^T v = 0$. Let $(1, v)^T \in \mathcal{V}_k$. Using Equation (3.4), we deduce that:*

$$U_k^T \begin{pmatrix} 1 \\ v \end{pmatrix} = w + Wv = 0.$$

The constraint in ($\tilde{\mathcal{P}}_1$) is thus equivalent to the constraint in ($\mathcal{P}_1$), which ends the proof.

3.3 Generalization of the relaxed ℓ_1-minimization problem

Obviously, the indicator vector $\mathbf{1}_{C_1}$ alone is not sufficient to know the complete graph structure. However, Proposition 6 can be extended to find the remaining indicator vectors. To do so, as in Proposition 4, we add the constraint that the target vector is orthogonal to the previously computed vectors, which is done in practice by applying a Gram-Schmidt orthonormalization procedure (see Section 4 below for more details about the procedure).

4 The ℓ_1-spectral algorithm

4.1 Global overview of the algorithm

In this section, we present a global overview of the ℓ_1-spectral clustering algorithm we implemented to recover the components of any perturbed graph (see Algorithm 5 below). It is available as a `R`-package on GitHub at `https://github.com/championcamille/l1-SpectralClustering`. In the next paragraphs, some details about the algorithm and parameters setting are given.

Algorithm 5 ℓ_1-spectral clustering algorithm

1: **Input:** $\mathcal{G}$ a graph, A its associated adjacency matrix, $\hat{k}$ number of clusters to recover and $(i_j)_{j\in\{1,...,\hat{k}\}}$ family of representative elements of each cluster.
2: Perform the spectral decomposition of A, sort the eigenvalues by increasing order and store the associated eigenvectors: $V := (v_1, ..., v_p)$.
3: **for** $j = 1$ **to** $\hat{k}$ **do**
4: Define $U_{\hat{k},j}$ as the matrix that contains the $p - \hat{k} - j + 1$ first columns of V:

$$U_{\hat{k},j} := (v_1, ..., v_{p-\hat{k}-j+1}).$$

5: Split $U_{\hat{k},j}$ into two parts:

$$w^T := U^{i_j}_{\hat{k},j} \text{ the } i_j\text{-th row of } U_{\hat{k},j} \quad \text{and} \quad W^T := U^{-i_j}_{\hat{k},j} \text{ the other rows of } U_{\hat{k},j}.$$

6: Solve the ℓ_1-minimization problem $(\tilde{\mathcal{P}}_1)$:

$$v^* := \underset{\substack{v\in\mathbb{R}^{p-1}\\ Wv=-w}}{\arg\min} \ \|v\|_1.$$

7: Recover the indicator of the j-th component:

$$\hat{\mathbf{1}}_{C_j} = (v_1^*, \ldots, v_{i_j-1}^*, 1, v_{i_j}^*, \ldots, v_p^*).$$

8: Update v_j in V: $v_j \leftarrow \hat{\mathbf{1}}_{C_j}$.
9: Perform Gram-Schmidt orthogonalization on V to ensure orthogonality between v_j and the rest of the columns of V:

$$V \leftarrow \text{Gram-Schmidt}(V).$$

10: **end for**
11: **Output:** $(\hat{\mathbf{1}}_{C_j})_{1\le j\le \hat{k}}$ the indicators of the $\hat{k}$ connected components.

4.2 Solving the ℓ_1-minimization problem

This section is devoted to the resolution of the constrained ℓ_1-optimization problem ($\tilde{\mathcal{P}}_1$) (line 6 of Algorithm 5). To be simplified, it can be equivalently written as the following penalized problem:

$$\underset{v\in\mathbb{R}^{p-1}}{\arg\min}\ \|Wv+w\|_2^2+\lambda\|v\|_1, \qquad (\mathcal{P}_{\text{Lasso}})$$

where, for all $v\in\mathbb{R}^{p-1}$, $\|v\|_2^2=\sum_{j=1}^{p-1}v_j^2$ and $\lambda>0$ is the regularization parameter that controls the balance between the constraint and the sparsity. Two methods are proposed thereafter to solve ($\mathcal{P}_{\text{Lasso}}$).

Lasso solution
The most traditional method to deal with such an ℓ_1-minimization problem is the Lasso procedure, developed by Tibshirani (1996). As for all regularizing methods, the choice of λ is of great importance. Here, especially, taking λ too large will lead to an over-constrained problem and a large number of nodes of $\mathcal{G}$ may not be clustered into components. In practice, K-fold cross validation, as implemented in the `glmnet R`-package, can be used to optimally set λ.

Thresholded least-squares solution
Another method consists in solving the least-squares problem:

$$v^*:=\underset{v\in\mathbb{R}^{p-1}}{\arg\min}\ \|Wv+w\|_2^2,$$

and then thresholding v^* given some predefined threshold t:

$$\forall j\in[\![1,p-1]\!],\ \ v_j^*=\begin{cases}1 & \text{if } v_j^*>t,\\ 0 & \text{otherwise.}\end{cases}$$

Of course, this thresholding step imposes sparsity on the solution. However, we can wonder if nodes with large coefficients should really be clustered together. In our model, the ideal parameters to recover (indicators of the components) do not take continuous values. Enforcing the coefficients of all representative elements to be equal to 1, under small perturbations, the coefficients of all other nodes belonging to the same components should then be close to 1. This specific behavior is underlined in Figure 3.1. In this example, we generated a graph $\mathcal{G}$ with 50 nodes, split into 5 connected components. We perturbed the structure of the graph by adding and removing edges with a probability P of 1%, 10%, 25% and 50%. We then solved ($\mathcal{P}_{\text{Lasso}}$) to recover the first component only. As can be seen in Figure 3.1, the Lasso and thresholded least-squares solutions give almost the same results: for small perturbations ($P\leq 10\%$, at the top), the whole component is perfectly retrieved. For $P=25\%$ (at the bottom left), all coefficients are tighter but both methods still work, forgetting only one node. As the perturbation becomes too large ($P=50\%$, at the bottom right), the selection of nodes belonging to the first component fails.

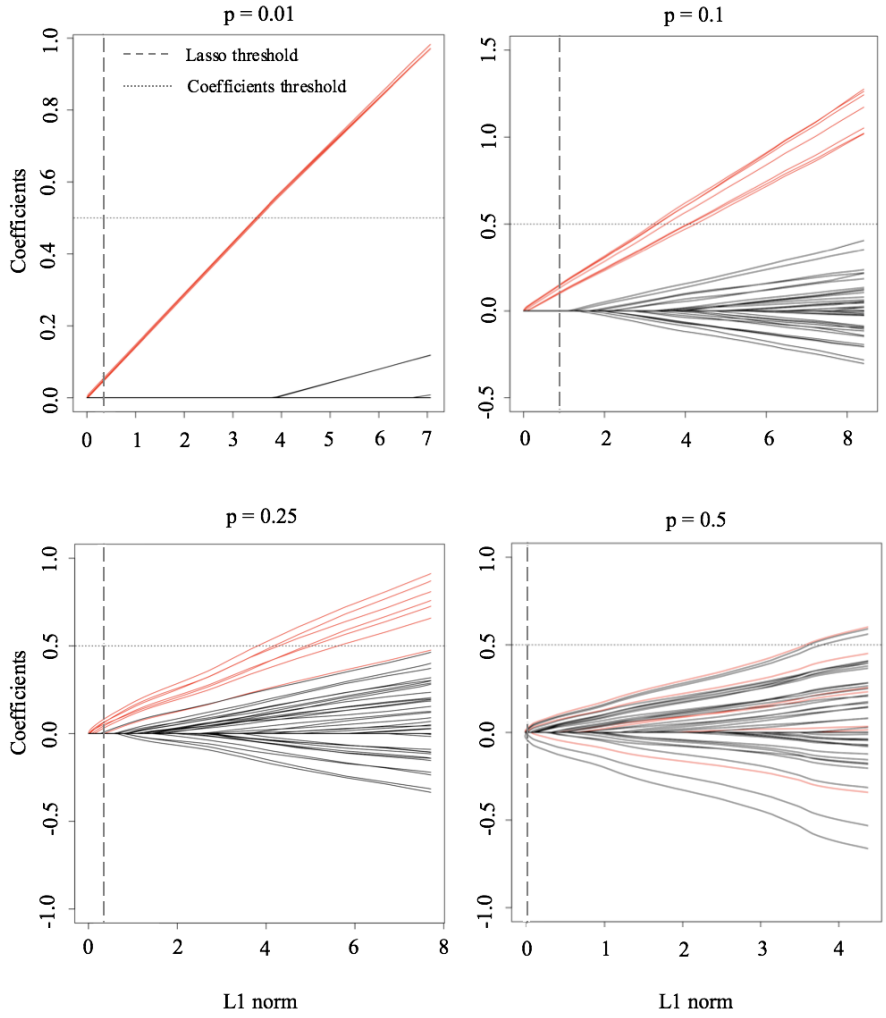

Figure 3.1 – Evolution of the coefficients of v, solution of ($\mathcal{P}_{\text{Lasso}}$), with respect to $\|v\|_1$ for different perturbations of the ideal graph (from top left to bottom right $P = 1\%$, 10%, 25% and 50%). Red lines correspond to the coefficients belonging to the component we aim at recovering, in contrast with black ones. Dotted lines are related to the ℓ_1-norm-threshold (vertical), associated with the Lasso solution, and the threshold on the value of the coefficients (horizontal), associated with the thresholded least-squares solution.

4.3 Optimally tuning the number of clusters

Traditional clustering algorithms, such as k-means, require the user to specify the number of connected components of the graph $\mathcal{G}$ to recover, which is, in practice, unavailable. Determining the optimal number of components $\hat{k}$ thus becomes a fundamental issue. A large number of methods have been developed in this sense: the hierarchical clustering for example looks for a hierarchy of components using dendrograms. The Elbow, average silhouette and gap statistic methods (Tibshirani et al., 2001) are also frequently used in addition to clustering techniques.

In our work, as proposed by Luxburg (2007), we focus on the heuristic eigengap method, which consists in choosing $\hat{k}$ such that it maximizes the eigengap, defined as the difference between consecutive eigenvalues of the Laplacian matrix L. This procedure is particularly well-suited in a spectral context. Indeed, in the ideal case, perturbation theory ensures that there exists a gap between the eigenvalue 0 of multiplicity k and the next $k+1$-th one. In the perturbed case, while being less strong, an eigengap still exists.

4.4 Finding the representative elements

In addition to the number of connected components, to run the ℓ_1-spectral clustering algorithm, we need to know at least one representative element of each component. This assumption may be restrictive when working with real data. However, it makes sense in a large number of situations where clusters are chosen to classify nodes around specific elements of the graph.

To avoid an arbitrary choice of such elements, one solution consists in estimating them using a rough partitioning algorithm. Another solution is to explore the structure of the graph to find hubs of densely connected parts. In this work, this is done by computing the betweeness centrality score of all nodes. In graph theory, the betweeness score S_b measures the centrality of a node based on the number of shortest paths passing through it:

$$\forall \ell \in [\![1, p]\!], \;\; S_b(\ell) = \sum_{1 \leq i,j \leq p} \frac{\#\text{ shortest paths from } i \text{ to } j}{\#\text{ shortest paths from } i \text{ to } j \text{ passing through } \ell}.$$

In practice, the representative elements of the k components are chosen to maximize this score.

Note that the nodes with the highest betweeness scores should be those that connect the densest parts of the graph. The risk of clustering two nodes from different connected components may thus be high. To avoid this, we add a stabilization step to our algorithm. As soon as one of the nodes with the k highest scores is added to a component during the minimization step, it is removed from the list of potential representative elements. We then re-run the algorithm using the k nodes taken among the $k+1$ ones with the highest scores, and so on until stabilization of the list of representative elements.

5 Numerical experiments

This section is dedicated to experimental studies to assess numerical performances of the ℓ_1-spectral clustering algorithm through two datasets. First, we show that it behaves well on simulated data with a variety of different settings and in comparison with state-of-the-art spectral clustering methods. Then, using a gene expression data set from kidney cancer patients, we demonstrate the ability of our algorithm to discover relevant groups of genes that act together to characterize the disease.

5.1 Application to toy datasets

5.1.1 Numerical settings

To explore the capabilities and the limits of the ℓ_1-spectral clustering algorithm with respect to state-of-the-art methods, we first considered simulated data, whose settings are detailed in the next paragraphs.

Simulated data set
We generated random ideal graphs for a given number of nodes p ($p = 20$, 50 and 100) and a given number of connected components k ($k = 2$, 5, 10). The component sizes $(c_j)_{1\leq j\leq k}$ were chosen in a balanced way: $\forall j \in [\![1, k-1]\!], c_j = \lfloor p/k \rfloor$, with $\sum_{j=1}^{k} c_j = p$. With a probability p_{in} and p_{out} of removing an edge from a component and of introducing an edge between two components varying from 0.01 to 0.5, we created multiple perturbed versions of the graph. All experiments were replicated 100 times each for better robustness.

Algorithm parameters
As some of the methods we compare with require the number of components to form, we focus on two versions of the ℓ_1-spectral clustering: the one presented in Algorithm 5, for which the number of clusters and a list of representative elements are assumed to be known, and the self-tuned one, for which both of them are extracted from the graph, as explained in Section 4. The results being very similar, we choose to focus on the thresholded least-squares solution to solve the ℓ_1-minimization problem ($\tilde{\mathcal{P}}_1$) in Algorithm 5. The corresponding threshold parameter t is fixed using 5-fold cross-validation.

Comparison with state-of-the-art
We compare the ℓ_1-spectral clustering with three other graph-based clustering algorithms: first, the spectral clustering (Algorithm 4), which is available in the `R`-package `anocva`, then, SpectACl, which was developped by Hess et al. (2019) with the aim of exhibiting both minimum cut and maximum density of the clusters. This algorithm can be viewed as a combination of DBSCAN, a density-based clustering algorithm (Ester et al., 1996) which is mainly used to identify clusters of any shape in a data set containing noise and outliers, and spectral clustering. SpectACl is publicly available on the Bitbucket platform as a Python code at `https://bitbucket.org/Sibylse/spectacl/src/master/`. Both methods requiring the number of components to cluster as an input, we finally run the Self-Tuning Spectral Clustering from Zelnik-Manor and Perona (2005), available on GitHub as a Python code at `https://github.com/wOOL/STSC`. The latter is an improved version of the spectral clustering, in which the final postprocessing step (k-means) is removed and the structure of the eigenvectors is carefully analyzed to automatically infer the number of clusters. It is thus used to evaluate the performances of the self-tuned version of the ℓ_1-spectral algorithm.

Performance metrics
Performances are measured by comparing the learnt components with the true ones, which are obviously known in the context of simulated data. Among the large number of existing scores, we

		p=20		p=50			p=100	
p_{in}	p_{out}	$k=2$	$k=5$	$k=2$	$k=5$	$k=10$	$k=5$	$k=10$
0.01	0.01	1	1	1	1	1	1	1
	0.1	1	1	1	1	1	0.99	0.99
	0.25	1	0.92	1	1	0.71	0.99	0.88
	0.5	0.99	0.65	1	0.76	0.51	0.96	0.40
0.1	0.01	1	0.99	1	1	1	0.99	1
	0.1	1	0.98	1	1	0.96	0.99	0.99
	0.25	0.99	0.84	1	0.98	0.63	0.99	0.69
	0.5	0.91	0.60	1	0.51	0.49	0.79	0.35
0.25	0.01	0.99	0.96	1	1	0.98	0.99	0.98
	0.1	0.99	0.91	1	0.99	0.79	0.99	0.87
	0.25	0.94	0.69	1	0.78	0.54	0.95	0.47
	0.5	0.54	0.54	0.75	0.32	0.46	0.27	0.30
0.5	0.01	0.95	0.84	0.99	0.93	0.82	0.99	0.86
	0.1	0.83	0.68	0.97	0.73	0.55	0.90	0.51
	0.25	0.52	0.54	0.73	0.34	0.46	0.32	0.30
	0.5	0.22	0.46	0.11	0.21	0.43	0.10	0.25

Table 3.1 – NMI scores obtained after clustering perturbed graphs of different sizes using the ℓ_1-spectral clustering algorithm. All results are averaged over 100 replicates.

used the Normalized Mutual Information (NMI) score, for its ability to compare clusters that could be of different sizes. The closer to 1 the NMI score, the better the classification.

5.1.2 Effect of the dimension and cluster sizes on perturbed graphs

First, we aimed at exploring the effect of the dimension and cluster sizes on the performances of the ℓ_1-spectral clustering algorithm. For p ranging from 20 to 100 and k from 2 to 10, results, in terms of NMI scores, are summarized in Table 3.1. On the one hand, for fixed p and k, when the perturbations are small (p_{in}, $p_{out} < 0.25$), one may note that the ℓ_1-clustering algorithm works well (it is clearly a favorable situation). On the other hand, a crude decrease in performance results can be observed as the perturbation grows (p_{in} or $p_{out} \geq 0.25$). In that case, the perturbed graph is far from the original one, which makes hard the recovery of the components. As expected, this becomes even more significant while the dimension increases (from top left to bottom right for each value of p). For perturbations of 0.5 (last line), the NMI scores do not exceed 0.5, which means that the ℓ_1-spectral clustering algorithm almost fails to recover the components. However, we must keep in mind that imposing a perturbation of 0.5 on a graph strongly affects its structure, with a probability of removing an edge inside a component and introducing an edge between components of 50%.

5.1.3 Performance results with respect to state-of-the-art

To give more credit to the ℓ_1-spectral clustering algorithm, we also evaluated its robustness in comparison with the spectral and SpectACl algorithms (see Section 5.1.1) for clustering different perturbed versions of a graph with $p = 100$ and $k = 10$. For each perturbation, we generated 100 graphs and computed the clustering performances using NMI scores.

Results can be visualized in Figure 3.2, which also indicates the 50% confidence interval. As can be seen, the ℓ_1-spectral and spectral clustering algorithms are very similar, especially for small perturbations ($p_{out} < 0.25$). However, as the perturbations grow, the ℓ_1-spectral clustering algorithm shows a smaller impact to noise sensitivity than the spectral one, being, almost ever, the best method. The results of SpectACl are oddly bad but this may be due to the fact that it was developed for clustering nonconvex shapes, which is beyond the scope of the present work.

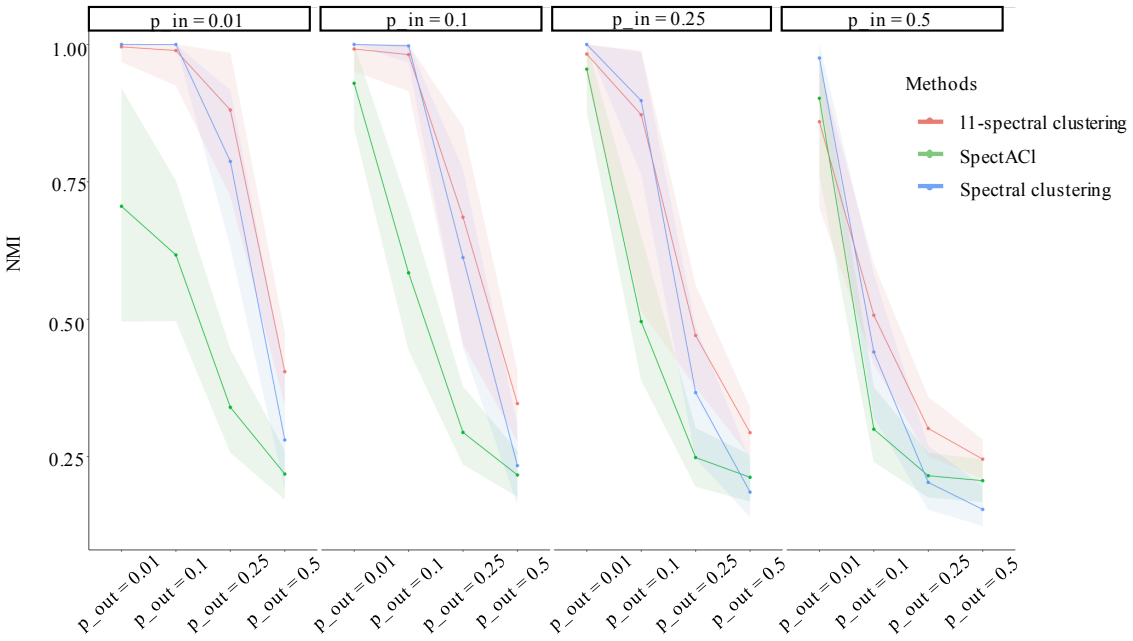

Figure 3.2 – Clustering results, in terms of NMI, of the ℓ_1-spectral clustering (in red), SpectACl (in green) and spectral clustering (in blue) algorithms applied on perturbed graphs for perturbations ranging from 0.01 to 0.5 and the associated 50% confidence intervals.

An interesting question is how the self-tuning version of the ℓ_1-spectral clustering, for which the number of clusters and the representative elements are self-evaluated, compare with the Self-Tuning Spectral Clustering (see Section 5.1.1). For $p = 20$, $k = 2$ and perturbations ranging from 0.01 to 0.5, we generated 100 versions of the same graph. Results are given in Figure 3.3. At the top, the NMI scores indicate that the performances of the self-tuning ℓ_1-spectral clustering (in red) decrease while the perturbations grow. On the contrary, the Self-Tuning Spectral Clustering (in blue) seems to be less sensitive to the increase of p_{in} but provide bad results for $p_{in} < 0.25$ and $p_{out} \geq 0.25$, even though it is a more favorable situation.

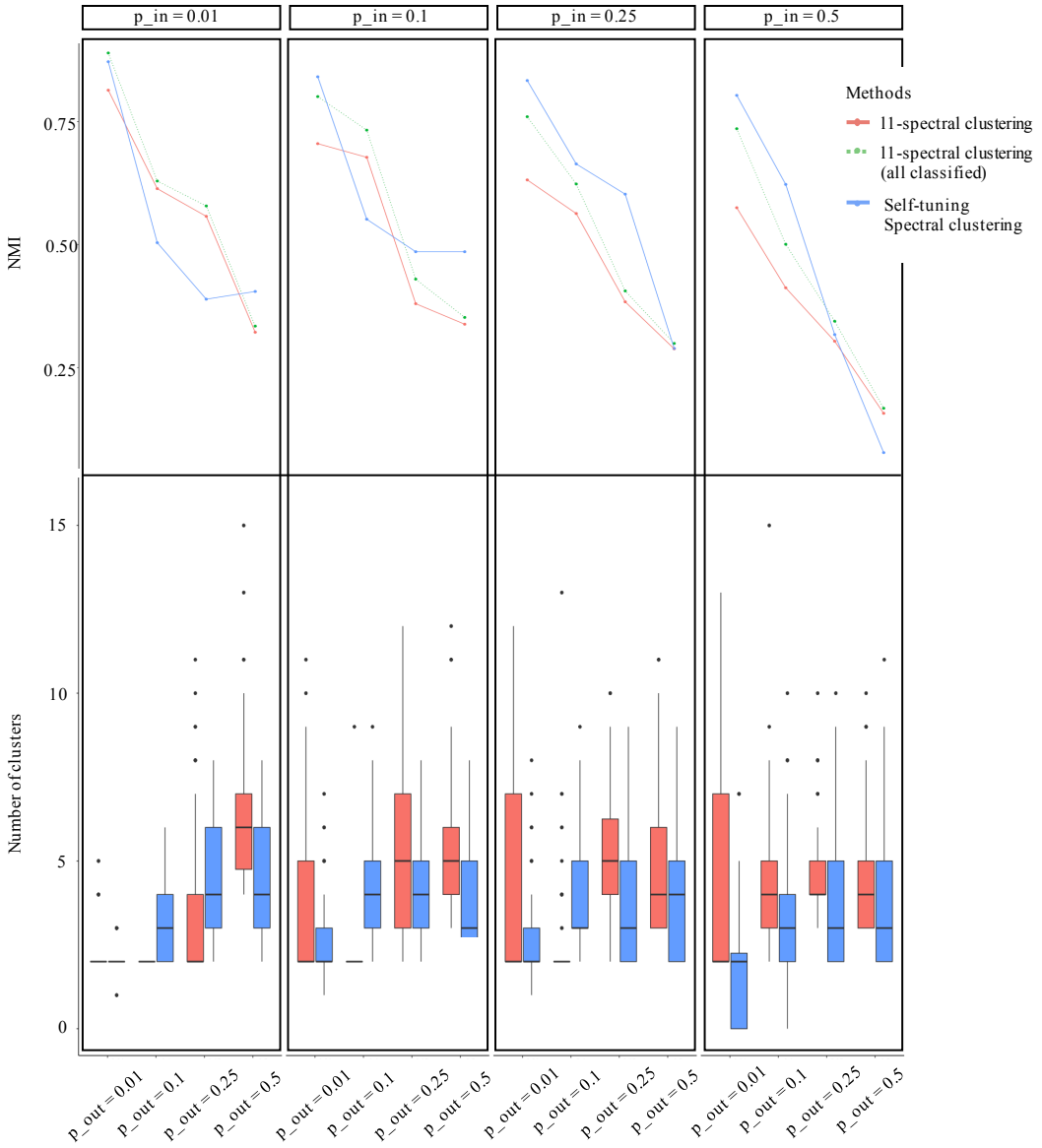

Figure 3.3 – At the top, clustering results, in terms of NMI, of the self-tuning ℓ_1-spectral clustering (in red and in green, where only classified nodes are taken into accounts) and the Self-Tuning Spectral Clustering (in blue) algorithms applied on perturbed graphs for perturbations ranging from 0.01 to 0.5. At the bottom, the associated estimated number of clusters of both methods across the 100 perturbed versions of the graphs.

Some of the lowest performances of the self-tuning ℓ_1-spectral clustering can be explained by observing that this algorithm was developed in a sparse form, that is all nodes from a perturbed graph are not automatically clustered into components. In practice, this has huge consequences on the NMI score, which processes the non-classified nodes as wrongly-classified ones. When considering only

the classified nodes, the NMI scores of the self-tuning ℓ_1-spectral clustering can be found in green, with, of course, better results.

More generally, the NMI scores are particularly sensitive to the number of estimated clusters. At the bottom of Figure 3.3, for each perturbation, the estimated number of clusters of both methods, which should be close to $k = 2$, the true number of clusters, can be visualized. It is clear that the further to 2 the estimation is, the smaller the associated NMI scores. The self-tuning version of the ℓ_1-spectral clustering was not optimized in this sense but it should be the key for an improvement and a stabilization of the performances.

5.2 Application to cancer data

This section is dedicated to the application of the ℓ_1-spectral clustering algorithm on a real kidney cancer data set from The Cancer Genome Atlas (TCGA) project. After describing the data (Section 5.2.1), results are presented in Section 5.2.2 and followed by a discussion (Section 5.2.3).

5.2.1 The kidney cancer data set

The Cancer Genome Atlas (TCGA) is an american project from the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI), which was launched fifteen years ago with the aim of characterizing genetic mutations responsible for cancer using genome sequencing and bioinformatics methods. Since then, millions of data have been produced and made publically available. In this work, we focused on KIdney Renal clear cell Carcinoma, abbreviated to KIRC thereafter. KIRC is one of the most common types of cancer, usually affecting people (mainly men) around 60 years old. Even if the chances of surgical cures are good, KIRC is hard to detect with no early symptoms, which makes it even dangerous.

In this work, we extracted gene expression data for KIRC from the TCGA data portal `http://gdac.broadinstitute.org/`. These data were produced using RNA-sequencing for a total number of $16,123$ genes and 532 cancer patients. After preprocessing by log-transformating, quantile normalizing the arrays and filtering genes based on variance, we only kept 75% of them, i.e. $12,092$ genes.

5.2.2 ℓ_1-spectral clustering algorithm on kidney cancer data

Applying the ℓ_1-spectral clustering algorithm to cluster genes into components require the knowledge of an initial network that models the relationships between genes. The latter are usually represented through Gene Regulatory Networks (GRNs), which are directed graphs that connect genes based on regulations (activations/inhibitions). Here, we focused on co-regulated networks, a simplified version of GRNs, where no causality exists. Edges are thus undirected, representing co-regulations, or correlations in terms of expression between genes. To create such a network, we computed the correlation matrix, based on Pearson's correlation, between all pairs of genes and then thresholded the matrix by removing edges with correlation smaller in absolute value than 0.7. This network is made of 4,982 genes (see Figure 3.4 (a) for an overview of the network).

We then applied ℓ_1-spectral clustering algorithm on the adjacency matrix associated with the co-regulatory network described above. The 4,982 genes were clustered into 186 components, from size 2 to 986, with an averaged number of genes of 27. These components are represented in Figure 3.4 (b), with different colors.

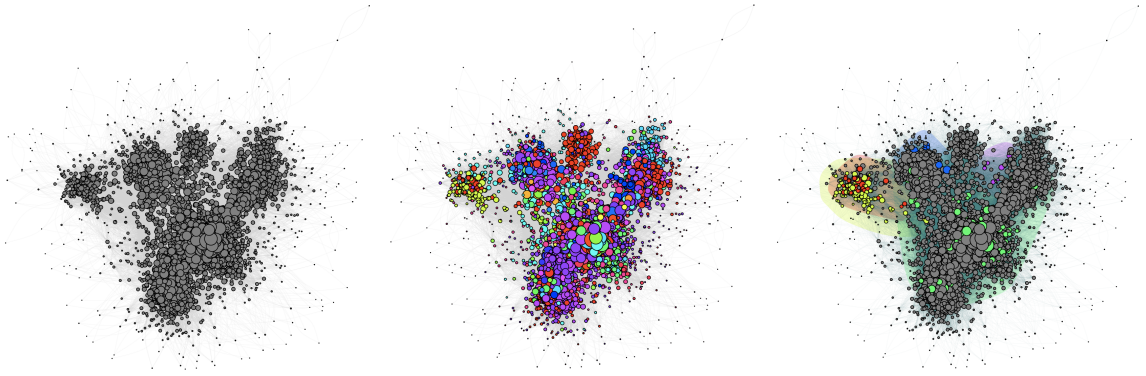

Figure 3.4 – From left to right: (a) the co-regulatory network representing co-regulations between genes in KIRC, (b) the components discovered by applying the ℓ_1-spectral clustering algorithm and highlighted with different colors, (c) the five components we particularly focus on.

5.2.3 Clusters as hallmarks of kidney cancer

In this section, we investigate the biological hypotheses that can be deduced from the network. First, to assign biological meaning to each component of the network, we performed gene set enrichment analysis. To this aim, we used the databases GeneSetDB (Culhane et al., 2010) and MSigDB (Liberzon et al., 2015), restricted to hallmark (H), curated (C2), GO (C5), oncogenic (C6) and immunologic signatures (C7) gene sets, which include the gene sets most relevant to cancer gene expression profiles. Enrichments were evaluated by performing hypergeometric tests, corrected for multiple testing using the FDR (Benjamini and Hochberg, 1995). Among the 186 identified components, four particularly drew our attention (see Figure 3.4 (c)). These components are described in details in the next paragraphs.

Transmembrane activity cluster The first cluster we identified (Figure 3.4 (c), red cluster) is made of 43 genes. This cluster gathers genes involved in the same "transmembrane activity" pathway. Indeed, among the 43 genes of the component, 11 are members of the same group SLC of solute carriers transporters, which aims at facilitating the transport of substrates across membranes. This is confirmed by the gene set enrichment analysis we performed, with p-values ranging from 1.97×10^{-6} from 1.89×10^{-9}.

Epithelial-mesenchymal cluster The second cluster (Figure 3.4 (c), purple cluster) of seven genes is highly enriched in Epithelial-Mesenchymal Transition (EMT) pathways, a natural process that

converts epithelial cells into mesenchymal phenotypes and is often altered in cancers. This cluster includes the gene SERPINH1, a known EMT-related gene, which has been identified as a potential biomarker of kidney cancers (Qi et al., 2018).

T-cells associated cluster The third cluster (Figure 3.4 (c), blue cluster) includes 38 genes, mostly enriched in T-cells and inflammatory response associated pathways. To confirm this, we tested the correlation of the cluster expression (defined as the averaged expression across all genes from the cluster) with CD4+ and CD8+ T-cells, encoded by the genes CD4, CD8A/B, which play a major role in cancer immunotherapy (Tay et al., 2020). With correlations ranging from 0.46 (CD8B) to 0.86 (CD4) and associated p-values smaller than $10e^{-16}$, we validate this relationship.

In addition, Kawashima et al. (2020) recently reported that both CD4+ and CD8+ T-cells were up-regulated in the patients with kidney cancer of high grade. We obtained the same results when comparing the cluster expression with low (grades 1 and 2) and high (grades 3 and 4) kidney cancer grades: the higher the expression, the worst the grade (p-values for t-test of 0.009, see Figure 3.5).

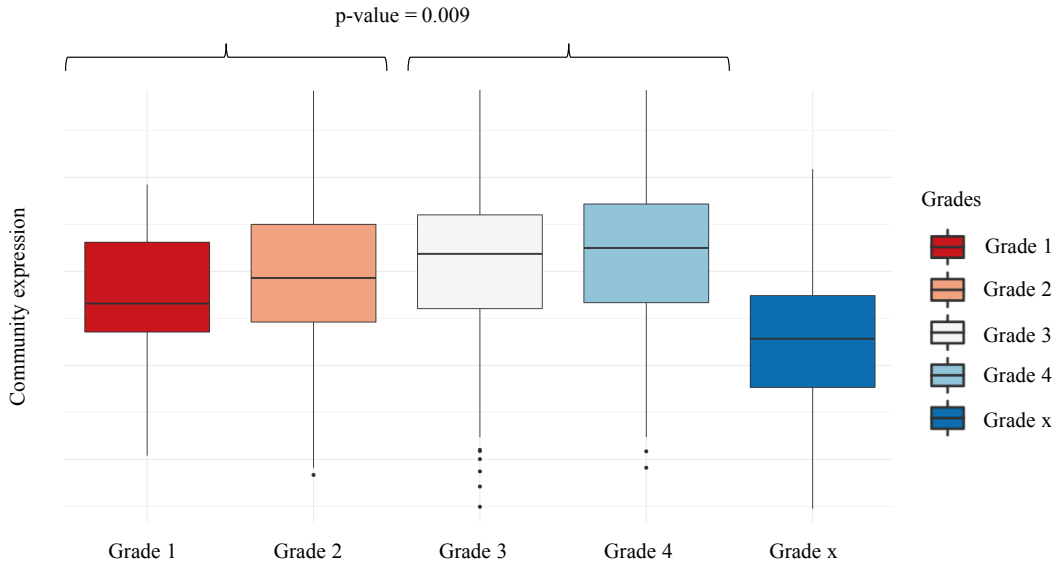

Figure 3.5 – Boxplots representing the association between the cluster expression (averaged expression across all genes from the cluster) and kidney cancer grades, which range from 1 (low grade) to 4 (high grade), grade x indicating that the grade could not be evaluated.

Liver-signature cluster In the last cluster (Figure 3.4 (c), yellow cluster), we found 96 genes, most of which belong to liver gene signatures, characterizing liver cancers. To a little extent, these genes are also associated with glutathione, an antioxidant that protects cells from important damages. Xiao and Meierhofer (2019) have recently shown that an increased level of glutathione is a hallmark of kidney

cancer. To go further, we investigated whether the cluster expression could be used to predict survival in kidney cancer. To this aim, we used Cox proportional hazards modelling. Hazard ratios were used to report the direction of the survival effect and the Wald test was used to determine its significance. As shown in Figure 3.6, high gene expression is significantly associated with good survivals (Hazard ratio of 0.66, p-value of 0.009). This indicates that this cluster may be used as a prognosis of kidney cancers.

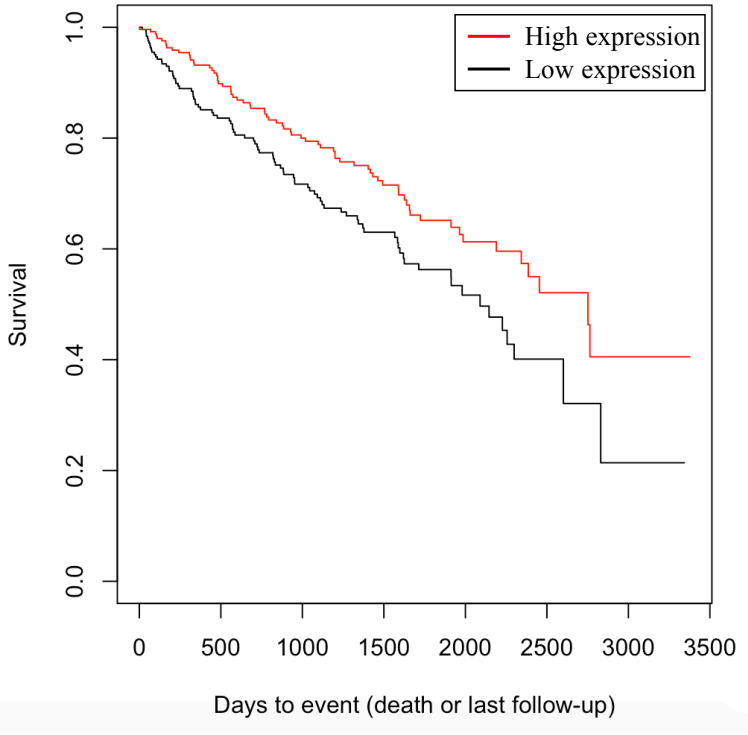

Figure 3.6 – Kaplan-Meier curves representing the association between high/low cluster expression and survival.

6 Conclusion

In this paper, we proposed a new spectral clustering algorithm, called ℓ_1-spectral clustering, for detecting cluster structures in perturbed graphs. To tackle the noise robustness issue of the traditional spectral clustering, the k-means is removed and replaced by writting the indicators of the components as solutions of explicit ℓ_1-constrained minimization problems. The performances of this algorithm are highlighted through numerical experiments, with competitive results when compared to state-of-the-art. Nevertheless, many opportunities for further improvements can be considered. Firstly, from an algorithmic point of view, it would be interesting to better explore solutions for calibrating the optimal number of clusters and its representative elements. Secondly, future works include theoretical

study of the eigenvectors stability, in order to validate the performances of the algorithm. A particular attention may be paid to the more general stochastic block model (SBM), where the edge probabilities depend on the community membership.

Chapter 4

Liver microbiota modeling at the early onset of human fibrosis

This chapter describes the main elements of a submitted paper, as joint work with Radu M. Neagoe, Maria Effernberger, Daniela T. Sala, Florence Servant, Jeffrey E. Christensen, Maria Arnoriaga-Rodriguez, Jacques Amar, Benjamin Lelouvier, Fabrice Gamboa, Herbert Tilg, Massimo Federici, Jose-Manuel Fernández-Real, Jean Michel Loubes and Rémy Burcelin.

Abstract

To understand the pathophysiological impact of liver microbiota on the early stages of fibrosis we identified the corresponding microbiota sequences and overcome the impact of different group size and patient origins with adapted statistical approaches. Liver samples with low liver fibrosis scores (F_0, F_1, F_2) were collected from Romania ($n = 36$), Austria ($n = 10$), Italy ($n = 19$), and Spain ($n = 17$). The 16SrDNA gene was sequenced. We considered the frequency, sparsity, unbalanced sample size between cohorts to identify taxonomic profiles and statistical differences. Multivariate analyses, including adapted spectral clustering with ℓ_1-penalty fair discriminant strategies, and predicted metagenomics were used to identify that 50% of liver taxa were Enterobacteriaceae and Pseudomonadaceae. The Caulobacteraceae, Flavobacteriaceae and Propionibacteriaceae discriminated between F_0 and F_1. The preQ0 biosynthesis and pathways involving glucoryranose and glycogen degradation were negatively associated with liver fibrosis $F_{1/2}$ vs F_0. Altogether, our results suggest a role of bacterial translocation to the liver in the progression of fibrosis. This statistical approach can identify microbial signatures and overcome issues regarding sample size differences, the impact of environment, and sets of analyses.

1 Introduction

Non-alcoholic fatty liver disease (NAFLD) is a common consequence of obesity and type 2 diabetes (DiMaira et al., 2018; Me, 2015). In NAFLD, the origin of inflammation and hepatocyte

injury is related to dietary lipids, bile acids, adipokines and cytokines, to cite a few. Furthermore, gut microbiota seems to be one of the key players of NAFLD development (Brandl and Schnabl, 2017; Hoyles et al., 2018). Markers and receptors of microbiota-related injury features have been described in this disorder such as TLRs, NODS, and NLRP3 (Denou et al., 2015; Miura et al., 2013; Pierantonelli et al., 2017; Roh and Seki, 2013) as well as the activation of the innate and adaptive immune systems (Bieghs and Trautwein, 2014). In early sets of experiments, we initially showed that hepatic steatosis in the obese diabetic mouse was due to an increased circulating concentration of lipopolysaccharides (LPS) i.e. metabolic endotoxemia (Cani et al., 2007). Lipoproteins transport LPS (Verges et al., 2014) to tissues, triggering the CD14/TRL4 pathway that increases liver inflammation and fat deposition (Cani et al., 2007). Gut bacteria were also reported to translocate through the intestinal tract to tissues (Berg et al., 1988) such as the adipose depots and the liver, establishing a tissue microbiota as observed in rodents (Amar et al., 2011a; Garidou et al., 2015; Pomie et al., 2016) and humans (Sookoian et al., 2020; Amar et al., 2011b; Burcelin et al., 2013) which could trigger liver inflam- mation and the onset of fibrosis (Amar et al., 2011a). This mechanism activates immune cells, including Kupffer cells, to release various pro-inflammatory cytokines and chemokines (Balmer et al., 2014) damaging the surrounding tissues initi- ating fibrosis. This hypothesis is now largely supported by recent major advances in NAFLD research, which show gut and blood microbiota dysbiosis of patients with advanced stages of NAFLD (Lelouvier et al., 2016; Loomba et al., 2017; Schierwagen et al., 2019). Hence, the identification of specific groups of translocated bacteria from dysbiotic gut microbiota could aid in the design of novel therapeutic strategies. To address this issue, we have sequenced and identified the bacterial 16S rDNA from liver tissue of a cohort of 36 Romanian, 17 Spanish, 19 Italians and 10 Austrian patients with different stages of liver fibrosis, notably at their early stages. We could design hypoth- eses regarding the putative causal role of liver microbiota in the development of liver fibrosis. We used this database to evaluate the efficacy of Principal Coordinate Analysis (PCoA) to visualize the different liver fibrosis group scores using Wilcoxon-Mann-Whitney statistical tests (Cox and Cox, 2008; Kruskal, 1964). Eventually, since the overall database of patients issued from different separated cohorts we anticipated some degree of heterogeneity of the overall cohort therefore, we adapted and developed a specific statistical approach i.e. ℓ_1-spectral clustering with fairness. This approach establishes inter-relations between liver microbiota and low scores of liver fibrosis that allowed the identification of the translocated bacteria putatively causal to the disease and independent from the group size, the patient origins and sets of sequencing. Overall, we drew a European microbial profile of patients at early stages of liver fibrosis.

2 Material and Methods

2.1 Subjects and Ethics

A multicentric observational study was conducted in the Second Department of Surgery, Emergency Mureş County Hospital of Romania, the Department of Systems Medicine of the Tor Vergata University of Rome, the Institut d'Investigacio Biomedica de Girona IdibGi, the Endocrinology and Nutrition Department of Dr. Josep Trueta University Hospital, and the University Hospital of Innsbruck. All research procedures performed in this study were in strict accordance with a

Characteristics	All patients $N = 82$	Stage F_0 $N = 34$	Stage F_1 $N = 37$	Stage F_2 $N = 11$	p-value F_0 vs F_1	p-value F_2 vs F_0	p-value F_2 vs F_1
Age (years)	41.50 ± 11.52	39.5 ± 12.77	39 ± 9.53	50 ± 9.15	0.99	10.16	$0.03*$
Female (n)	$47(57\%)$	$15(18\%)$	$26(32\%)$	$6(7.3\%)$	0.65	0.99	0.97
Height (m)	1.67 ± 0.08	1.67 ± 0.08	1.7 ± 0.08	1.62 ± 0.07	0.99	0.61	0.14
Smoker (n)	$22(27\%)$	$10(12\%)$	$10(12\%)$	$2(3\%)$	0.99	0.99	0.97
Weight (kg)	118.5 ± 23.99	120 ± 22.55	118 ± 21.59	115.8 ± 35.77	0.99	0.99	0.97
BMI (kg/m^2)	42.65 ± 7.73	43.25 ± 6.9	41.6 ± 7.2	41.52 ± 11.41	0.99	0.99	0.97
Waist (cm)	121 ± 18.37	124.5 ± 19.4	120 ± 15.61	120 ± 24.12	0.99	0.99	0.99
Blood Glucose (mg/dl)	95.7 ± 25.76	95 ± 27.46	99 ± 21.22	95 ± 34.63	0.99	0.99	0.97
Treated Diabetes (n)	$7(8.5\%)$	$1(1.2\%)$	$2(2\%)$	$4(4.7\%)$	0.99	0.02^*	0.027^*
Systolic (mm Hg)	130 ± 19.47	130.5 ± 20.76	124 ± 17.43	134 ± 18.45	0.65	0.99	0.73
Diastolic (mm Hg)	80.0 ± 11.59	80.5 ± 11.4	75 ± 10.31	90 ± 15.3	0.88	0.99	0.97
Treated hypertension (n)	$20(24\%)$	$8(9.7\%)$	$5(6\%)$	$7(8.2\%)$	0.99	0.15	0.027^*
Treated dyslipidemia (n)	$6(7.3\%)$	$2(2\%)$	$3(3.6\%)$	$1(1.2\%)$	0.99	0.99	0.99
Total Cholesterol (mg/dL)	189.1 ± 39.78	190.0 ± 36.93	200.0 ± 43.11	167.0 ± 38.71	0.99	0.99	0.97
HDL Cholesterol (mg/dL)	43.91 ± 13.38	47 ± 11.73	43 ± 13.48	42 ± 16.62	0.99	0.61	0.97
GOT (U/l)	20.85 ± 17.56	18.50 ± 18.14	22 ± 18.97	22 ± 7.54	0.99	0.99	0.97
GPT (U/l)	27.50 ± 25.23	23.50 ± 17.50	29 ± 31.91	30 ± 14.16	0.65	0.99	0.97
GGT (U/l)	29 ± 23.04	27.50 ± 18.04	30 ± 25.84	32 ± 23.3	0.65	0.61	0.99
HCT $(\%)$	41 ± 4.03	40 ± 4.09	41.1 ± 3.05	40.5 ± 6.13	0.99	0.99	0.94
Leukocytes (G/L)	7.84 ± 2.63	7.48 ± 2.4	8.1 ± 2.39	7.8 ± 3.7	0.99	0.61	0.97
Neutrophils (G/L)	5 ± 2.44	4.8 ± 2.36	5.15 ± 2.28	5.3 ± 3.2	0.99	0.99	0.99

Table 4.1 – Baseline characteristics of patients with biopsy-proven fibrosis

predefined protocol and adhered to the Good Clinical Practice guidelines and the Declaration of Helsinki. The study was approved by the Coordinating Ethics Committee of the Emergency Mureş County Hospital, Romania (registration $4065/2014$), the Institutional review board $\&$ Ethics Committee and the Committee for Clinical Research (CEIC) of Dr. Josep Trueta University Hospital, Girona, Spain; the Policlinico Tor Vergata Ethics Committee, Rome, Italy as part of the FLORINASH Study the Institutional Ethics Commission at the medical University of Innsbruck (amendment to AN$20170016369/4.21$). All participants provided informed consent prior to participation. The patients who gave their consent to perform a liver biopsy during the procedure were eligible. Exclusion criteria were serious liver diseases (eg hemochromatosis, alcoholic fatty liver disease, Hepatits B and Hepatitis C infection, chronic diseases, inflammatory systemic diseases, acute or chronic infections in the previous month, use of antibiotic, antifungal, antiviral drugs, proton-pump inhibitors, anti-obesity drugs, laxatives, excessive use of vitamin D supplementation, fiber supplements or probiotics or participation in a weight loss program or weight change of 3 kg during the previous 6 weeks, pregnancy or breastfeeding, or major psychiatric antecedents; neurological diseases, history of trauma or injured brain, language disorders, and excessive alcohol intake (≥ 40 g/day in women or 80 g OH/day in men) or intravenous drug abuse, and previous bariatric surgery.

The cohort consists of 82 Caucasian patients where 34 were diagnosed with fibrosis stage 0 (F_0); 37 stage 1 (F_1) and 11 stage 2 (F_2), as diagnosed from histological analyses of liver biopsies (Table 4.1). The patients suffered from morbid obesity with a mean BMI 42.6 (± 7.3). The mean waist circumference was 121.49 (± 18.73) in male and 123.23 (± 18.26) in female participants.

2.2 Liver biopsies and liver fibrosis diagnosis

Liver biopsies were performed during laparoscopic surgical bariatric procedures or via ultrasound guided liver biopsy. No energy devices were used for collecting the samples since hemostasis was done afterwards when the samples were extracted from the abdomen. Ultrasound (US) guided percutaneous liver biopsy (UPLB) was performed in 10 patients. In all patients, antiplatelet drugs and oral anticoagulation therapy was paused 1 week before UPLB was performed. One experienced physician (> 3000 US- exams and > 100 UPLB) performed the US-examinations with the Philips EPIQ $5^{®}$ (Philips Corporation, Amsterdam, The Netherlands). UPLB was performed using an 18 G Temno II semi-automatic tru-cut biopsy needle (Cardinal Health, Dublin, Ohio, USA). After UPLB, all patients were monitored for any signs of pain or clinically suspected bleeding by nursing staff over a 6-h period. If no serious complica- tions were evident, all patients would be discharged after the mandatory 6-h observation, a stable blood count and a normal ultrasound examination. All patients were follow-up in 2 weeks to review the results of the histology. All the samples were stored in a sterile container and kept at $-80°C$ until assayed. Furthermore, NAFLD was confirmed histologically by an independent pathologist.

2.3 Clinical assessments

Anthropometric measurement of each subject was performed by trained nurses in the morning after fast- ing for at least 8 h. Body height was recorded to the nearest 0.5 cm and body weight to the nearest 0.1 kg. BMI was defined as body weight (kilograms) divided by the square of body height (meters). Waist cir- cumference was measured in the horizontal plane midway between lowest rib and the iliac crest to the nearest 0.1 cm at the end of a normal expiration repeatedly in men and women by 3 trained nurses on 3 consecutive days. Blood pressure was recorded to the nearest 2 mmHg by a mercury sphygmomanometer with the arm supported at heart level after sitting quietly for 10 min. Fasting plasma glucose was meas- ured after fasting for at least 8 h. A standard oral 75-g glucose tolerance test was performed to measure 2-h postprandial plasma glucose. Hypertension was defined in accordance to the Guidelines of the Europe- an Heart Association or if the subject was taking medication for hypertension. Diabetes was diagnosed when fasting plasma glucose was ≥ 126 mg/dL (7 mmol/L), 2-h postprandial plasma glucose ≥ 200 mg/dL (11.1 mmol/L), and HbA1c $\geq 6.5\%$ or if the subject was taking medication for diabetes.

2.4 Biochemical and molecular analyses

Plasma parameters:

Biochemical analyses including total fasted plasma glucose, cholesterol, high-density lipoprotein (HDL) cholesterol, plasma liver enzymes i.e. aspartate aminotransferase (AST/GOT), alanine aminotransferase (ALT/GPT), gamma-glutamyl transferase (GGT), hematocrit and leukocytes were determined by Cobas 8000, (Roche, Basel, Switzerland) according to the manufacturer's specification. Elevated liver enzymes were defined as aspartate aminotransferase and alanine aminotransferase. HbA1c was measured by high- performance liquid chromatography (Bio-Rad, Muenchen, Germany)

and a Jokoh HS-10 autoanalyzer.

2.5 16S rDNA sequencing and bioinformatic analysis

The $V3$-$V4$ hypervariable regions of the 16S rDNA were amplified by two steps PCR using $v1$ primers (Vaiomer) and sequenced using MiSeq Reagent Kit $v3$ (2×300 bp Paired-End Reads, Illumina, San Diego, CA, USA) as previously described (Luche et al., 2013). The MiSeq sequences were then analyzed using the bioin- formatics pipeline established by Vaiomer using FROGS $v1.4.0$ (Escudie et al., 2018). Briefly, after demultiplexing of the bar-coded Illumina paired reads; single read sequences are cleaned and paired for each sample independently into longer fragments. Operational taxonomic units (OTU) are produced with via single-linkage clustering and taxonomic assignment is performed in order to determine community profiles (generated by Blast+ $v2.2.30+$ against the Silva $v128$ Parc databank restricted to the bacterial kingdom).

2.6 Linear Discriminant Analysis (LDA) Effective Size (LEfSe)

The bacterial profiles were further compared between the three groups using LEfSe pairwise analysis with an alpha cut-off of 0.05 and an effect size cut-off of 2.0. The bacterial diversity analyses (alpha and beta diversity, MDS ordinations and taxonomic composition barplots) were generated using the 'Phyloseq $(v1.14.0)$, 'vegan' $(v2.4.0)$ and 'ape' $(v3.5)$ packages under R environment $v3.3.1$. LEfSe analysis was performed on the OTU table using the online Galaxy interface to identify bacterial taxa that were differ- entially abundant in the three liver fibrosis groups (Segata et al., 2011). Respective cladograms were generated with genus at the lowest level. Quantitative plots of differential features were generated from genus level percent relative abundance data showing means with standard deviation using GraphPad Prism 6 software. Using the LEfSe algorithm, bacterial taxa that were differentially abundant in analysis of liver fibrosis groups were first identified and tested using the Kruskal Wallis test.

2.7 Beta diversity analysis

The bacterial diversity (alpha and beta diversity) was analyzed and represented using the 'phyloseq' $(v1.14.0)$, 'vegan' $(v2.4.0)$, 'ape' $(v3.5)$, and 'ggplot' $(3.3.0)$ packages under R environment $v3.5.1$ with Chao, Inverse Simpson, Simpson and Shannon as indexes. The alpha diversity statistical significance was de- termined by Wilcoxon rank-test. The beta diversity was calculated for every pair of variables to generate a matrix of distance using Bray-Curtis, Jaccard, Unifrac, and weighted Unifrac indexes. From distance matrices, Multiple Dimension Scale (MDS) and hierarchical clustering were conducted for graphical representation of beta diversity. PERMDISP2 procedure was used for the analysis of multivariate homo- geneity of group dispersions. The Kruskall-Wallis test was performed to compare abundance across the three groups.

2.8 Multivariate analysis

To visualize the distribution of patients according to their clinical parameters, we performed a Principal Component Analysis (PCA) using 'FactoMineR' and 'factoextra' R packages. For the study of 16S rDNA diversity, we first filtered the less abundant OTUs to reduce the noise within the matrix before running the PCA. We eliminated those with abundance < 0.01. We then normalized the OTU table by using the Cumulative Sum Scaling normalization followed by a log transformation, using 'mixOmics' package (Rohart et al., 2017). To explore the metagenomic data and identify the largest sources of variation, another Principal Component Analysis was conducted. Also based on the projection of the dataset into a space of lower dimension and originally designed for regression we performed a Partial Least Square Discriminant Analysis (PLS-DA) and its sparse version (sPLS-DA) on the normalized OTU table count to predict and select the most discriminative features in the data that help to classify the samples according to the fibrosis variable (package 'mixOmics'). Since we observed the influence of the metagenomic data on the outcome, we used alternative method of classification such as random forest (package 'randomForest'). The random forest is built from a multitude of different decision trees and classifiers at training time thereby predicting and storing the predicted target outcome.

2.9 Cluster graphical analysis

The abundance matrix of OTUs can be modeled by a graph using 'PLNmodels' package under R where nodes represent OTUs and edges interactions between each pair of nodes. We developed an analysis in clusters i.e. the ℓ_1-spectral clustering, implemented in R, a robust variant of the well-known spectral clustering that aims to detect the natural structures of a graph by taking advantage of its spectral properties. The adjacency matrix modeling the variable associations of the graph is used as an input of the ℓ_1-spectralclustering algorithm. In front of the influence of the origin of the cohort on the graphical classification through clusters we applied "fair" technics with k-median clustering objectives. We identified k centers and assign each input point to one of the centers so that the average distance of points to their cluster center is minimized. In the fair-variant, the points are colored while the goal is to minimize the same average distance objective ensuring all clusters to have an approximately equal number of points of each color. This technique called 'fairtree' and developed in python takes as input the desired number of clusters, the desired cluster balance and the normalized table count.

2.10 Functional metagenomic prediction

Metagenome inference and predicted functional analysis were initiated by analysis of the OTU clustered 16S sequence count table data and the OTU representative sequences using the PICRUSt2 tool (Douglas et al., 2020) version $2.3.0b$ for each sample. The metagenome predic- tion process included four main steps: 1) The input OTU representative sequences were aligned against the PICRUSt2 reference alignment, 2) From this alignment, the input OTU were placed into the PICRUSt2 reference phylogenetic tree, 3) The metagenome functions were inferred by the hidden state prediction method using this phylogenetic tree. During this inference process, the abundance values of each OTU were normalized to their respective predicted 16S rDNA copy numbers and then multiplied by the

respective gene counts of the target bacteria, 4) The predicted functions were mapped to the MetaCyc database to determine the minimum set of pathways present in the samples. The resulting core output was a list of enzyme functions (Enzyme Commission numbers) with predicted count data for each sample from step 3 as well as a list of MetaCyc pathways with predicted count data for each sample from step 4.

2.11 Data Availaibility Section

MiSeq 16S rDNA sequences were deposited under the primary accession number PRJEB41831 and a secondary number ERP125667 on December 9th 2020 with a release date on the 31st of December 2021.

3 Results

3.1 Graphical classification of the clinical variables by PCA

We aimed at identifying liver 16S rDNA profiles associated with the early onset of fibrosis. We aggregated together a library of liver biopsies from patients from four cohorts of different European countries. We first visualized the distribution of the patients according to the cohorts by performing a principal component analysis using the anthropomorphic and clinical data where the projection of the different clinical variables is represented (Figures 4.1 A,B). The ellipses calculated for each cohort show some degree of differential distribution suggesting that specific environmental factors have influenced the clinical outcomes. Interestingly, the Romania cohort was unifying all cohorts and could be used as a reference. In addition, we could detect numerous outlier patients from each cohort.

It is noteworthy that we voluntarily included all anthropomorphic and biochemical data, even if some were redundant and confounding, to remain within the frame of a non-a priory statistical approach. The age, diabetes and hypertension variables were the main drivers of the F_2 classification while HDL cholesterol and liver enzymes were drivers for the F_1 histological phenotype. These observations are supported by significant ANOVA tests (Table 4.1).

3.2 Analyses of the liver bacterial 16S rDNA ecology

To identify whether the graphical differences between the three liver fibrosis scores are associated with a differential liver bacterial DNA signature, we then performed PCA on the OTUs as entries in the database. The analysis using countries as groups shows that the different cohorts poorly overlapped suggesting the existence of specific environmental factors specific of each country cohort (Figure 4.2 A). Using the liver fibrosis scores as groups we could not clearly graphically discriminate the fibrosis scores since the distribution of the patients according to their OTU profiles were too scattered and seemed to be depending upon the largest Romanian cohort (Figure 4.2 B). To analyze differently the putative signatures according to the cohorts or the liver fibrosis scores, we studied the frequencies of the phylum and family taxonomic levels. The barplot analysis shows first a large degree of heterogeneity between all individuals at the phylum level (Figure 4.3) but still, we identified that

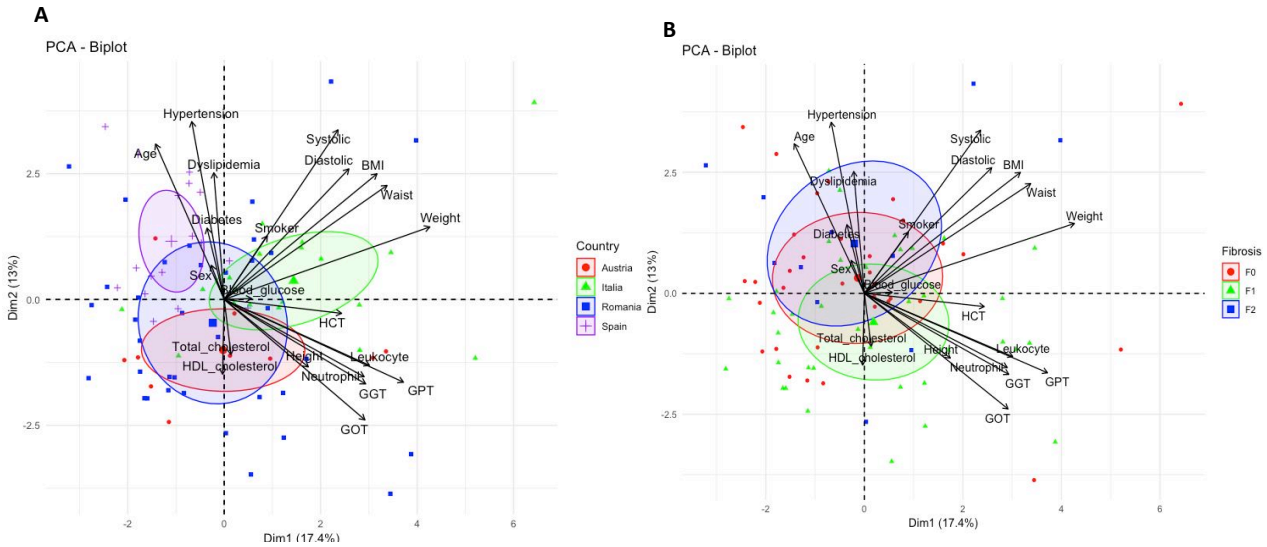

Figure 4.1 – Visualization of clinical variables by principal component analysis according to countries and fibrosis scores. The clinical variables were used as entries for a principal component analysis (PCA). PCA-biplot from package 'Factoextra' and 'FactoMineR' of individuals for the first two principal components are shown. They sum up 30.4% of the total variance of the dataset. Patients were grouped by A, countries (red dots=Austria, green triangle=Italy, blue square=Romania, purple cross=Spain) and by B, fibrosis scores (red dots= F_0, green triangle= F_1, blue square= F_2). The vectors corresponding to the clinical variables are shown as arrows.

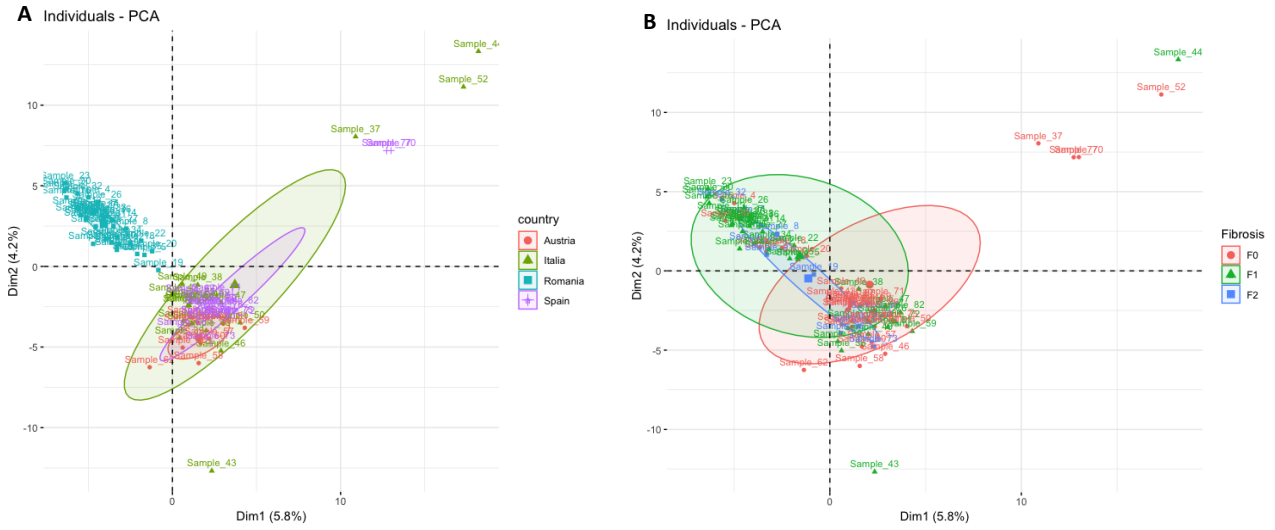

Figure 4.2 – Visualization of liver 16S rDNA sequences by principal component analyses according to countries and fibrosis scores. The 16S rDNA OTUs sequences were used as entries for a principal component analysis (PCA). PCA-biplot from package 'Factoextra' and 'FactomineR' of individuals for the first two principal components are shown. They sum up 10.0% of the total variance of the dataset. Patients were grouped by A, countries (red dots=Austria, green triangle=Italy, blue square=Romania, purple cross=Spain) and by B, fibrosis scores (red dots= F_0, green triangle= F_1, blue square= F_2). The vectors corresponding to the clinical variables are shown as arrows.

the liver microbiota of the overall cohort was composed mostly of Proteobacteria, ($> 75\%$) (Figure 4.4). Group comparisons showed that statistical differences were observed between the F_0 and F_1 groups for the Proteobacteria, Actinobacteria and Firmicutes phyla (Figures 4.5 A,B,C). At the family taxonomic level, the most prominent taxa were the Enterobacteriaceae and the Pseudomonadaceae which accounted for more than 50% of the overall taxa (Figure 4.6). Group comparisons showed that the Caulobacteraceae, Flavobacteriaceae and Propionibacteriaceae families were statistically different when comparing F_0 and F_1 (Figures 4.7 A,B,C).

To further identify whether liver fibrosis scores could be characterized by specific signatures we explored indexes of alpha and beta diversity of 16S rDNA in liver tissue. The data show that differences in abundances at the phylum, and family taxonomic levels were also associated with differences of the alpha diversity (Figure 4.8 A,B,C). Notably, the Observed, Shannon and Simpson indexes were significantly different between the F_0 and F_1 groups at the phylum and family levels. In addition to alpha diversity, we analyzed beta diversity and performed a principal coordinate analysis (PCoA) considering distances between variables (using Bray-curtis distance). The PCoA analyses showed that the F_0 group was distant from the two others which suggests a specific 16S rDNA signature (Supplementary Figure 4.9 A,B). It is noteworthy that outlier patients were also detected. Although, when analyzed together the three groups could not be classified clearly. The F_0 group differed graphically from the F_1, F_2 groups suggesting a specific signature discriminating between F_0 and F_1, F_2. To determine if the ellipse centers of the F_0 group differs from the ellipse center of the other groups, a Permutational Multivariate Analysis of variance (PERMANOVA) followed by a Kruskall-Wallis test were performed and found a difference between F_0 and F_1 groups ($p < 0.03$). Along the same line of investigation, we performed different graphical representations such as heatmaps and Venn diagrams.

3.3 Identification of specific bacterial signatures

To identify the variables that are specific to Fibrosis scores we performed a first Venn diagram on the overall set of variables (Figure 4.10 A). Eighty-nine variables were common to all groups and considered as the core of the cohort while 21, 77, and 108 OTUs were specific of the F_2, F_1, F_0 groups, respectively. To isolate extremely rare variables and unbalanced distribution between groups we next considered only OTUs with more than 25% of non-zero counts and an average number of counts per group higher than 150 and similarly drew a second Venn diagram. We identified 12, 5, and 5 OTUs specific to F_2, F_1, and F_0 scores, respectively (Figure 4.10 B) and (Tables 4.2, 4.3, 4.4). To identify if these specific OTUs could be picked up using another approach we generated a heatmap where each OTUs was positioned while the fibrosis scores was fixed (Figure 4.11 A). We noted that the frequencies of the majority of OTUs equal 0 or are extremely low ($< 0.01\%$) thereby, most of these variables do not bring information. Similarly, a minority of the variables of high frequencies were common to all liver fibrosis groups and did not provide discriminant information neither. Such OTUs could be considered as the core variable of liver microbiota. Conversely, a subset of OTUs could be considered as discriminant that was identified on a different heatmap following the removing of the non-informative OTUs (Figure 4.11 B).

To refine the identifications of the discriminant bacteria we performed a Linear discriminant analysis (LDA) coupled with effect size measurements (Figure 4.12, Figure 4.13, 4.14). The data

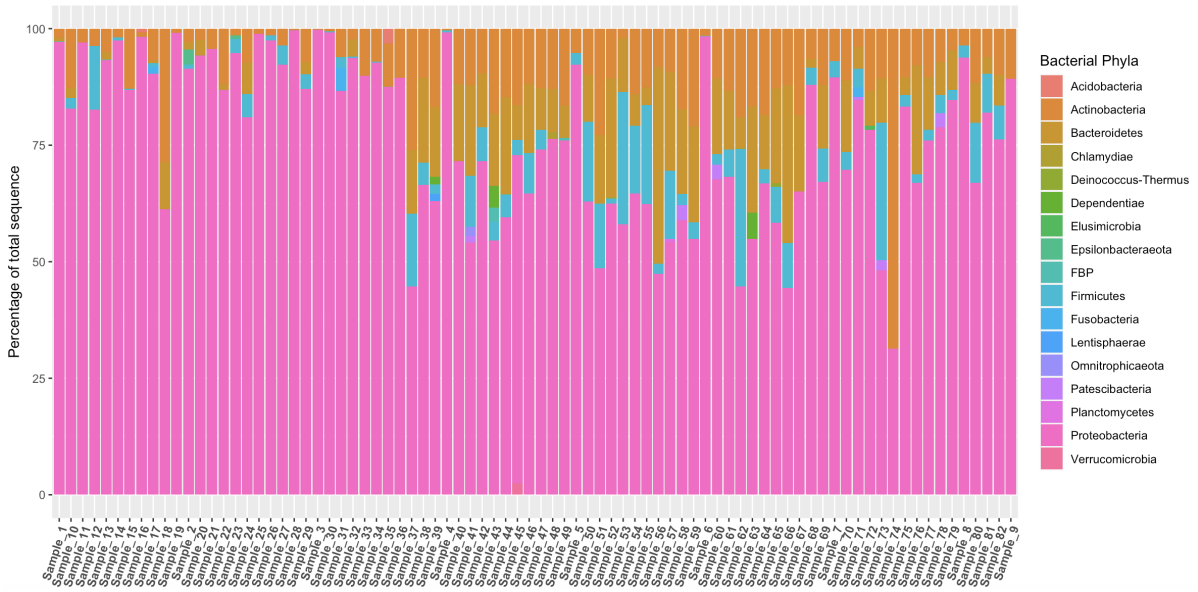

Figure 4.3 – Visualization of liver 16S rDNA sequences by principal component analyses according to countries and fibrosis scores. Barplot depicting the frequencies of liver microbial composition of each patient at the phylum level for the overall cohort (total) or according to the fibrosis scores (F_0, F_1, F_2).

OTU name	Phylum	Family	Genus
Cluster_{30}	Proteobacteria	Moraxellaceae	Acinetobacter
Cluster_{74}	Firmicutes	Ruminococcaceae	Faecalibacterium
Cluster_{35}	Actinobacteria	Microbacteriaceae	Rhodoluna
Cluster_{28}	Actinobacteria	Micrococcaceae	Kocuria
Cluster_{21}	Proteobacteria	Caulobacteraceae	Caulobacter

Table 4.2 – Identification of specific bacterial signatures (unfair analyses) in F_0 patients

OTU name	Phylum	Family	Genus
Cluster_{43}	Proteobacteria	Pseudomonadaceae	Pseudomonas
Cluster_{31}	Proteobacteria	Pseudomonadaceae	Pseudomonas
Cluster_{25}	Proteobacteria	Enterobacteriaceae	Multi-affiliation
Cluster_{37}	Proteobacteria	Rhodobacteraceae	Paracoccus
Cluster_{40}	Bacteroidetes	Chitinophagaceae	Ferruginibacter

Table 4.3 – Identification of specific bacterial signatures (unfair analyses) in F_1 patients

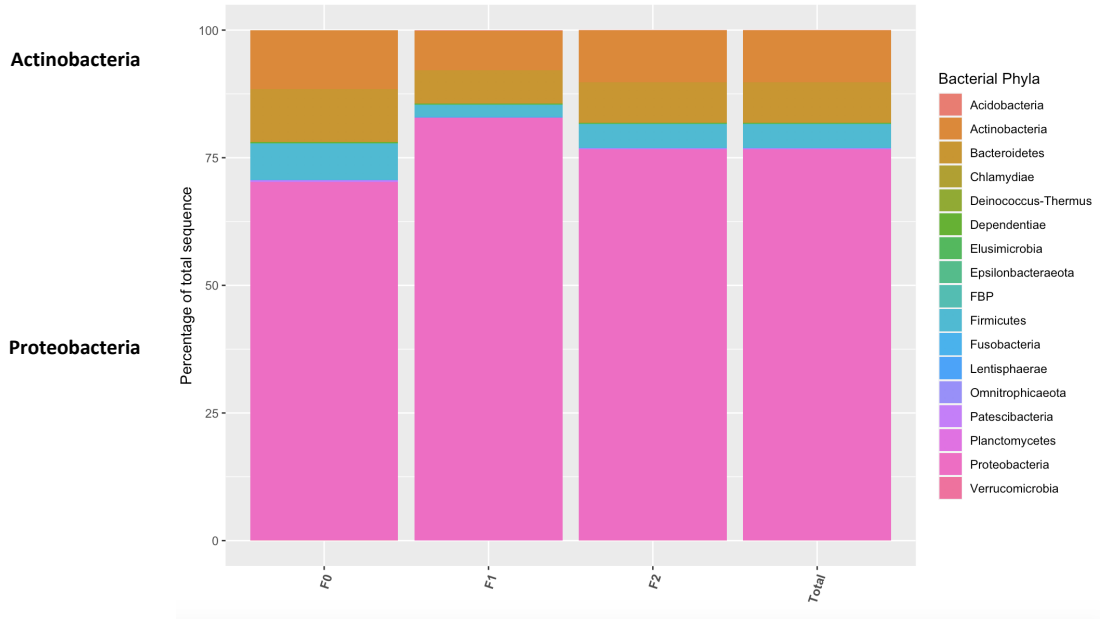

Figure 4.4 – Visualization of liver 16S rDNA sequences by principal component analyses according to countries and fibrosis scores. Barplot depicting the frequencies of liver microbial composition of each patient as means of the phyla frequencies for the overall cohort (total) or according to the fibrosis scores (F_0, F_1, F_2).

OTU name	Phylum	Family	Genus
Cluster_{122}	Firmicutes	Lachnospiraceae	Multi-affiliation
Cluster_{59}	Actinobacteria	Corynebacteriaceae	Corynebacterium 1
Cluster_{34}	Bacteroidetes	Weeksellaceae	Cloacibacterium
Cluster_{48}	Firmicutes	Peptostreptococcaceae	Romboutsia
Cluster_{42}	Proteobacteria	Burkholderiaceae	Tepidimonas
Cluster_{26}	Firmicutes	Streptococcaceae	Lactococcus
Cluster_{53}	Bacteroidetes	Weeksellaceae	Cloacibacterium
Cluster_{18}	Proteobacteria	Enterobacteriaceae	Pantoea
Cluster_{115}	Proteobacteria	Burkholderiaceae	Delftia
Cluster_{50}	Actinobacteria	Microbacteriaceae	Clavibacter
Cluster_{91}	Actinobacteria	Corynebacteriaceae	Corynebacterium
Cluster_{54}	Proteobacteria	Burkholderiaceae	Ralstonia

Table 4.4 – Identification of specific bacterial signatures (unfair analyses) in F_2 patients

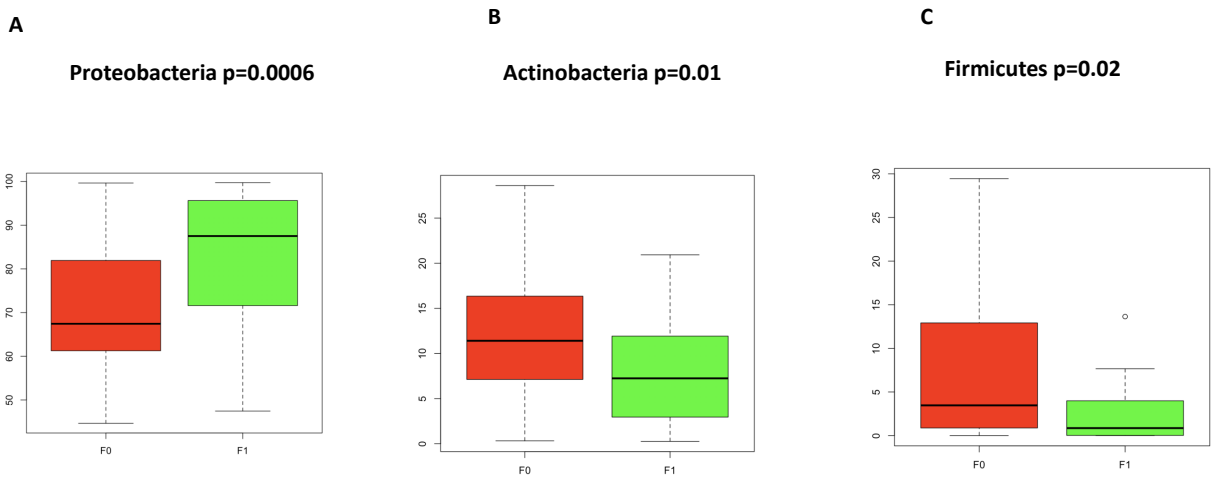

Figure 4.5 – Mean frequencies of discriminating taxa. Boxplot representing the frequencies of A Proteobacteria, B Actinobacteria C Firmicutes phyla throughout two groups of liver fibrosis scores (red= F_0, green= F_1, blue= F_2).

show that most of the discriminant information was identified when comparing between F_0 and F_1. The Firmicutes, Flavobacteriaceae, Caulobacteraceae and Actinobacteria were specific to F_0 group and the Proteobacteria was specific to F_1 (Figures 4.4 A,B,C, Figures 4.6 A,B,C). On the boxplot the taxa enriched in patients with no fibrosis are indicated with a negative score and mild fibrosis enriched taxa are indicated with a positive score. We performed LEFSe between each score pairs and identified much less differences between F_1 F_2 suggesting that they could have a similar liver microbiota, as suggested in Figure 4.2 B despite the discriminant clinical variables identified in Figure 4.1 B.

On these first sets of analyses, the number of fibrosis scores of each patient was too heterogeneous to perform a discriminant analysis (overfitting). As shown on Figures 4.15 A,B, 4.16 A,B they were almost no difference between F_1 and F_2, therefore we merged F_1 and F_2 scores as $F_{1/2}$ group, increasing hence the number of patients of that group.

To validate the pertinence of such strategy we performed a partial least square discriminant analysis i.e. PLS-DA. To select the most discriminative features in the model we used its sparse version sPLS-DA based on a Lasso penalization. The number of variables to be selected per component involved in the visualization is optimized using leave-one-out cross-validation. On the sample plot (Figure 4.9 A), we observe a slight separation of the two fibrosis scores ellipses compared to the unsupervised PCA. From the most discriminant OTUs selected on each sPLS-DA component, a dissociation between the two groups can be visualized using a Clustering Image Map (CIM) (Figures 4.9 B, 4.10 A). The graphs show a clear classification of the patients based on the identified discriminant variables. Eventually, we calculated the ROC curve with all discriminant variables that shows an increased specificity and

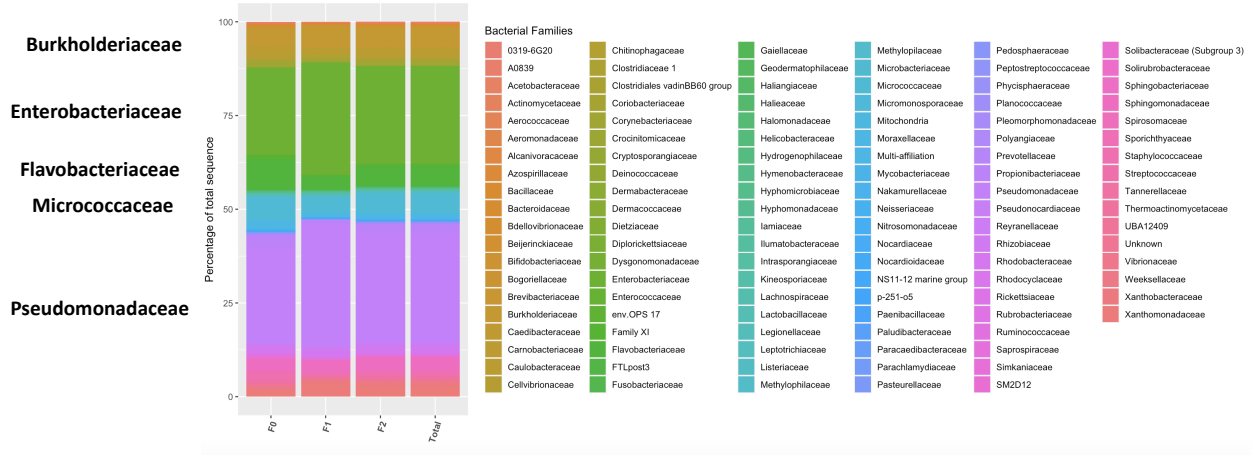

Figure 4.6 – Visualization of liver 16S rDNA sequences by principal component analyses according to countries and fibrosis scores. Barplot depicting the frequencies of liver microbial composition of each patient as means of the family frequencies for the overall cohort (total) or according to the fibrosis scores (F_0, F_1, F_2).

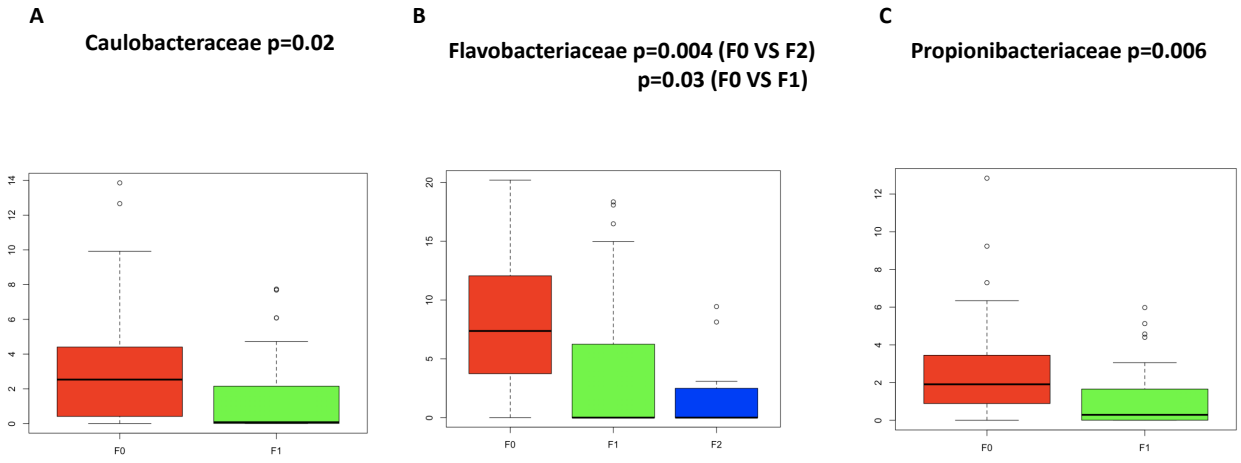

Figure 4.7 – Mean frequencies of discriminating taxa. Boxplot representing the frequencies of A Caulobacteraceae, B, Flavobacteriaceae, and C, Propionibacteriaceae families throughout two groups of liver fibrosis scores (red= F_0, green= F_1, blue= F_2).

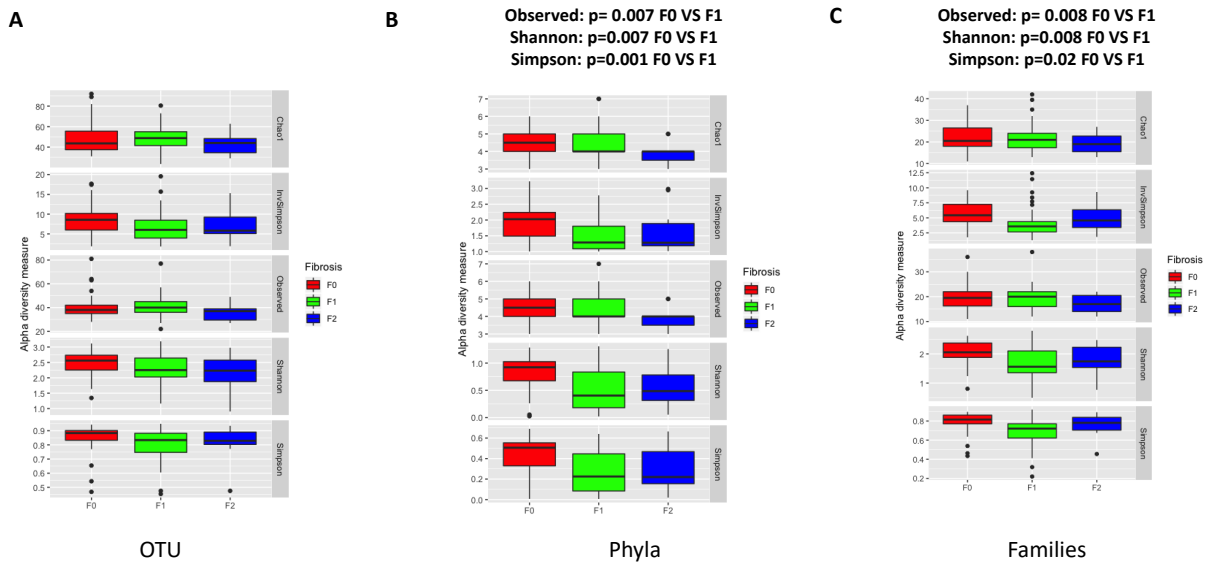

Figure 4.8 – Alpha microbial diversity. Boxplot showing microbial alpha diversity A at the OTU, B, phylum, C, and family taxonomic level calculated according to the Chao, Shannon, Simpson, inv Simpson indexes for the 3 liver fibrosis scores.

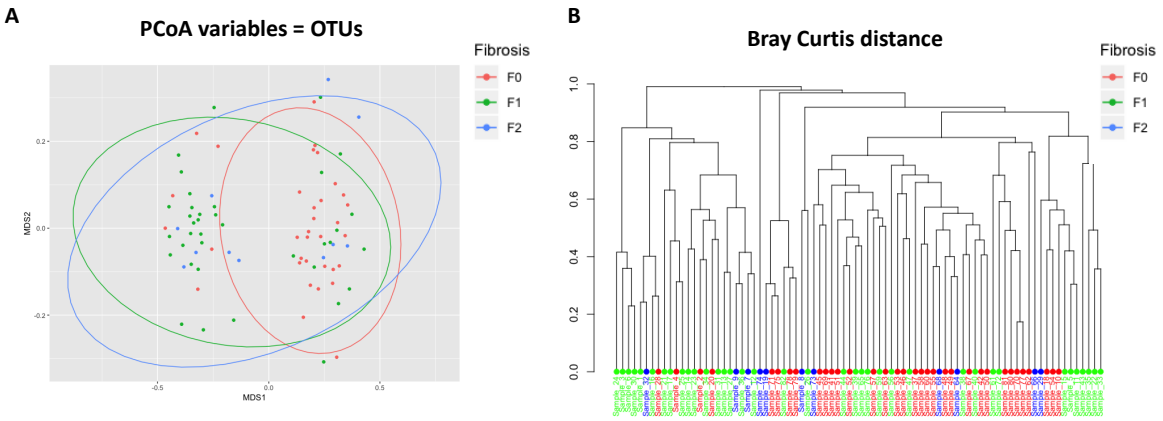

Figure 4.9 – Beta microbial diversity. A PCoA showing Bray Curtis beta diversity of the normalized OTU table count. Dots are assigned to indi- vidual patients and colored according to their fibrosis score (red= F_0, blue= F_2, green= F_1). B Hierarchical clustering of patients colored according to their fibrosis score (red= F_0, blue= F_2, green= F_1) based on Bray Curtis OTU distance.

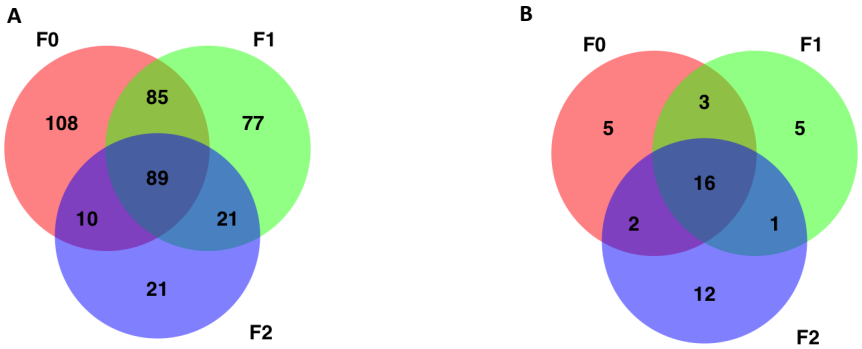

Figure 4.10 – Discriminant analysis strategies of the liver microbiota 16S rDNA OTUs according to the fibrosis scores. Venn diagrams where A all the 16S rDNA taxa or B data after removing those extremely rare and with unbalanced distribution within the 3 groups of patients with liver fibrosis, were used as entry variables characterizing the 3 liver fibrosis scores (red= F_0, green= F_1, blue= F_2).

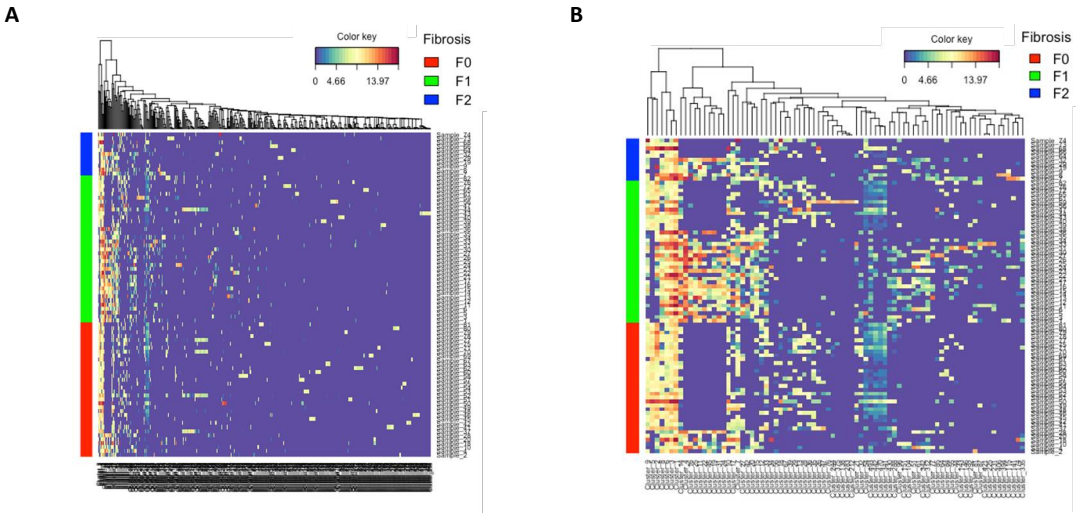

Figure 4.11 – Discriminant analysis strategies of the liver microbiota 16S rDNA OTUs according to the fibrosis scores. A Heatmap of normalized OTU counts according to the 3 groups of patients with liver fibrosis scores and B a corresponding subset of normalized OTU counts with groups of patients fixed.

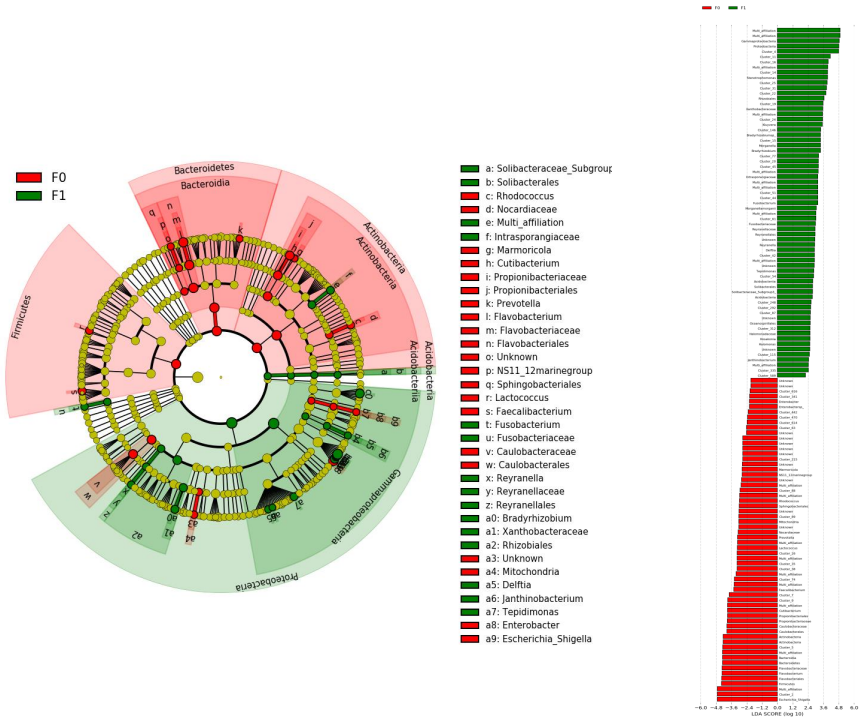

Figure 4.12 – Discriminant analysis strategies of the liver microbiota 16S rDNA OTUs according to the fibrosis scores. LEfSe cladogram of taxonomic assignments from 16S rDNA sequence data of the two liver biopsy fibrosis groups (F_0 and F_1). The cladogram shows the taxonomic levels represented by rings with phyla at the innermost ring and genera at the outermost ring, and each circle is a member within that level. Taxa at each level are shaded according to the liver fibrosis group in which it is more abundant ($P < 0.05$; LDA score ≥ 2.0). LDA scores are shown on the right panel for each taxon.

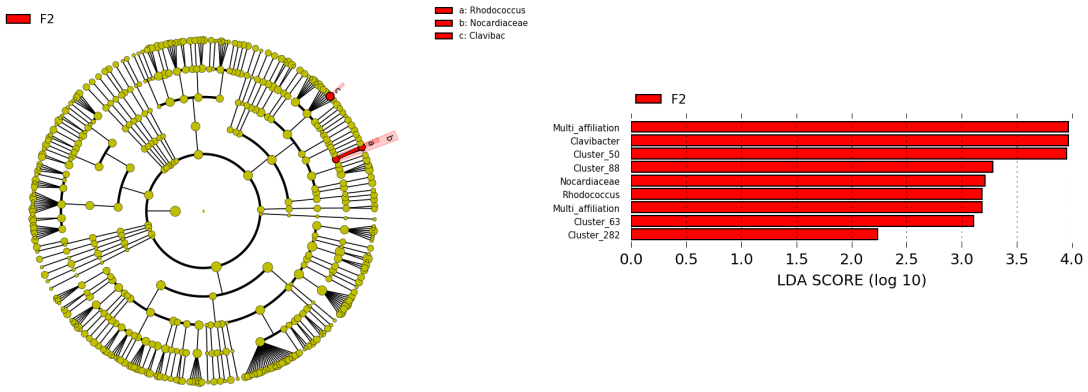

Figure 4.13 – Discriminant microbial signatures identified by linear effect size. LEfSe cladogram and LDA scores of taxonomic assignments from 16S rDNA sequence data of two liver biopsy fibrosis groups F_1 vs F_2

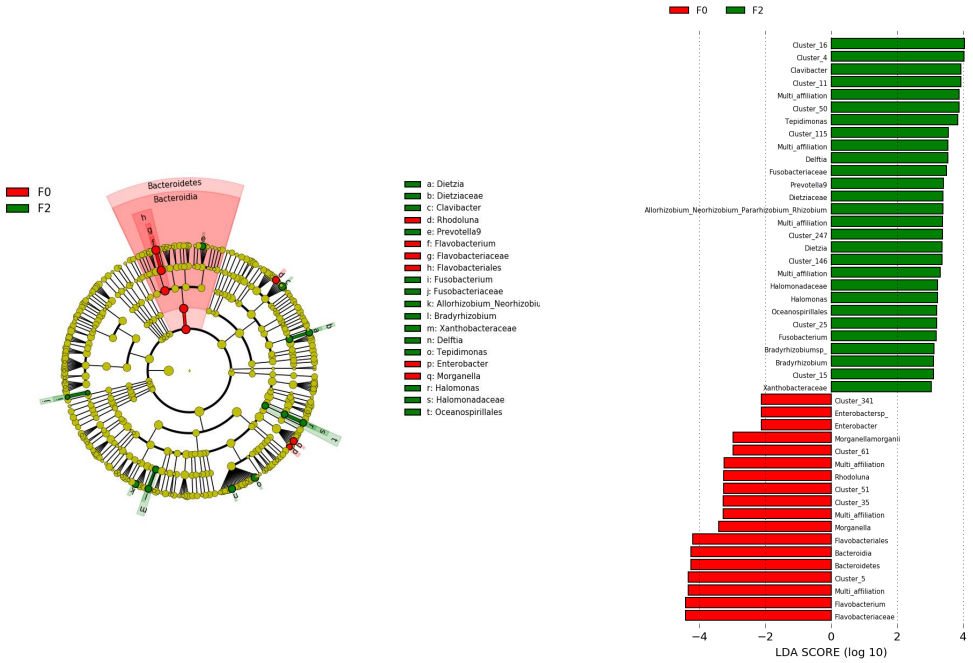

Figure 4.14 – Discriminant microbial signatures identified by linear effect size. LEfSe cladogram and LDA scores of taxonomic assignments from 16S rDNA sequence data of two liver biopsy fibrosis groups F_0 vs F_2

sensitivity above baseline (Figure 4.10 B).

Altogether, some degree of graphical classification of the liver fibrosis score could be observed using the clinical database and the 16S rDNA database. However, in both instances the individuals appear to be still distributed according to the countries. Therefore, to overcome this issue we developed an ad hoc fairness statistical strategy allowing the classification of variables i.e. OTUs independently from the cohort.

3.4 Identification of clusters of cohort-independent 16S rDNA associated with different mild scores of fibrosis

In front of these numerous signatures and the influence of confounding factors such as the impact of the cohort set there is a need to identify clusters of variables specific to each liver fibrosis score but independent of the cohort impact. To this aim we considered three different fair approaches on the overall cohorts and then defined clusters of OTU variables independent from the cohort. The first fair approach consists in identifying principal components from the metagenomic dataset as signatures of the cohorts and removing them to generate a new dataset where no components would be cohort sensitive. To this aim we compared the largest cohort i.e. from Romania to the others. Principal components conditional distributions with respect to the cohorts were visualized (Figure 4.17 A). Then, we removed the principal components the most correlated with the cohorts when the absolute value of Pearson correlation was above a threshold. The remaining non-overlapping components are cohort-insensitive and used to identify the variables associated with the mild fibrosis score. Remarkably, more than 78% of the variation from the original data was still included into the selected principal components suggesting that the discriminant information was only marginally affecting our previous results. On this "fair" dataset we applied the standard random forest classification to predict fibrosis scores. From the variable importance plot, indicating the contribution of the variables to classify the data, we selected the 10 more predictive principal components and identified 3 significantly associated with fibrosis scores (Figures 4.17 B,C,D).

The second fair clustering approach directly selects OTUs which are the most influenced by the cohorts and removes them from the analysis. The associated matrix is then modeled by a graph and subjected to a spectral clustering algorithm to which we applied an ℓ_1 penalty. The nodes represent OTUs and the edges show interactions between each pair of variables (Figure 4.18 A). Using this novel l_1-spectral clustering algorithm we identified 5 clusters of OTUs among which 3 were significantly associated with the liver fibrosis scores (Figure 4.18 B).

Eventually, we performed the fair clustering method called "fair-tree". We used the 16S rDNA normalized table count to identify clusters with approximately equal number of patients from each cohort. Two of the three clusters found containing respectively 36 and 97 OTUs, were statistically significant when comparing F_0 versus F_1 scores (Figures 4.18 C,D).

To summarize all the identified OTUs significantly associated with the different low scores of fibrosis, we compiled them in (Tables 4.5, 4.6, 4.7, 4.8) and identified their respective taxa. From the fair principal components identified, we only considered the five OTUs that contribute most to create each of these components. Then, from the Venn diagram we identified common OTUs signatures of low fibrosis scores from standard (sPLS-DA) and fair approaches (fair-tree, random forest, ℓ_1-spectral

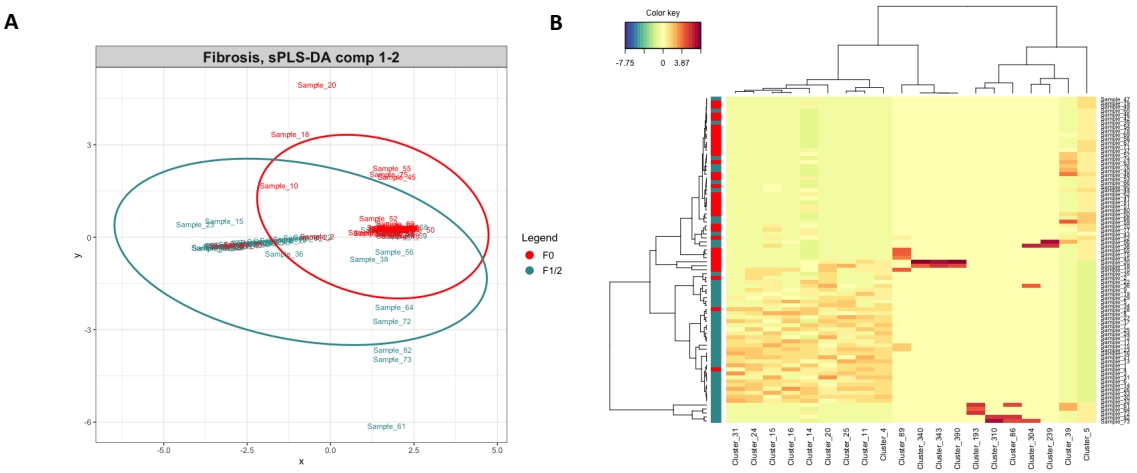

Figure 4.15 – Discriminant analysis strategies of the liver microbiota 16S rDNA OTUs according to the fibrosis scores. A sPLSDA classification performance on a CSS normalized microbial table count of the F0 versus $F_{1/2}$ groups of patients. Sample plot, each point corresponds to an individual and is colored according to its fibrosis score (red= F_0, green= $F_{1/2}$). B Clustering Image Map (CIM) of the OTUs selected on each sPLS-DA component.

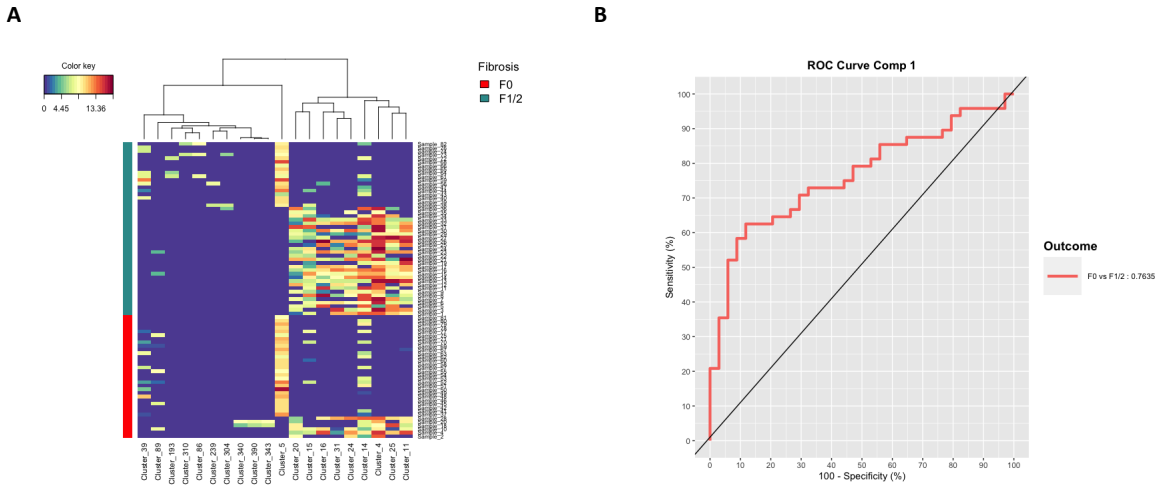

Figure 4.16 – Discriminant analysis strategies of the liver microbiota 16S rDNA OTUs according to the fibrosis scores. A Heatmap of the OTUs selected on each sPLS-DA component with groups of patients fixed. B ROC calculated on the predicted scores obtained from the sPLSDA model.

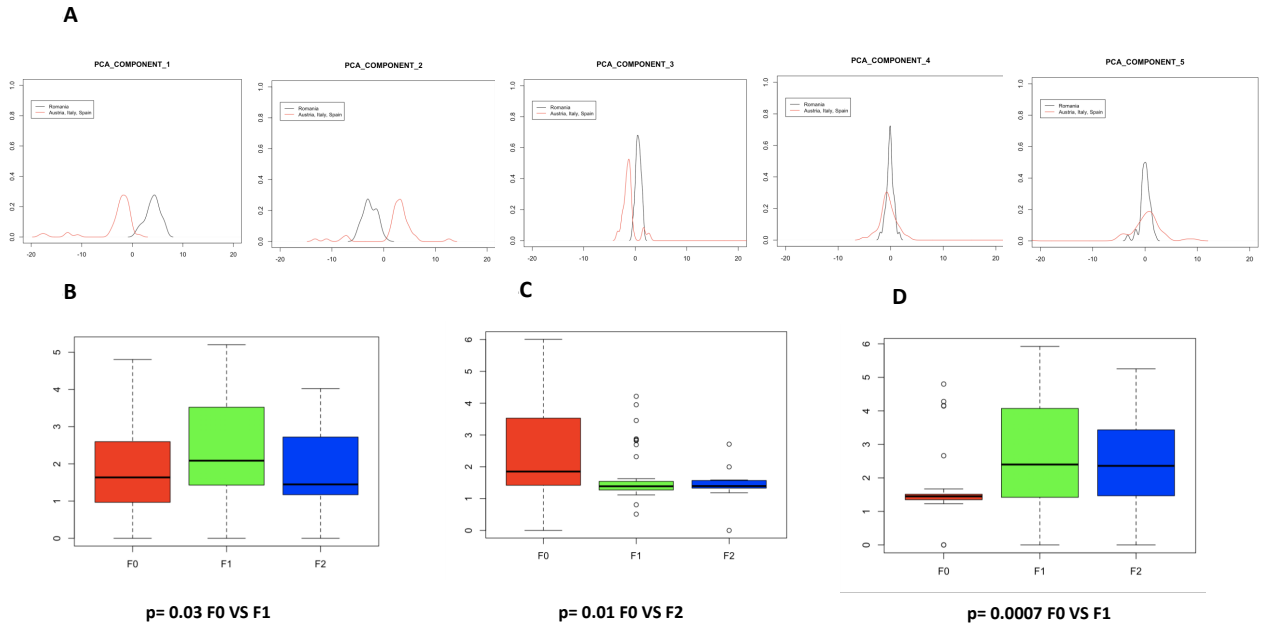

Figure 4.17 – Discriminant analyses of the 16S rDNA OTUs variables using fairness strategies. A Distribution curves (or densities) of the coordinate of individuals, split into two cohort types (black=Romania, red= the other countries: Italy, Austria, and Spain), when projected on the five first principal components built from the 16S rDNA OTUs normalized table count. The non-overlapping plots (for example components $1, 2, 3$) correspond to cohort discriminant components and will be removed from the final analysis to identify the liver fibrosis discriminant variables. Boxplot representing the frequencies of the most significant OTUs contributing to B the 6th, C the 24th, D the 52nd principal components for the different groups of liver fibrosis scores (red= F_0, green= F_1, blue= F_2).

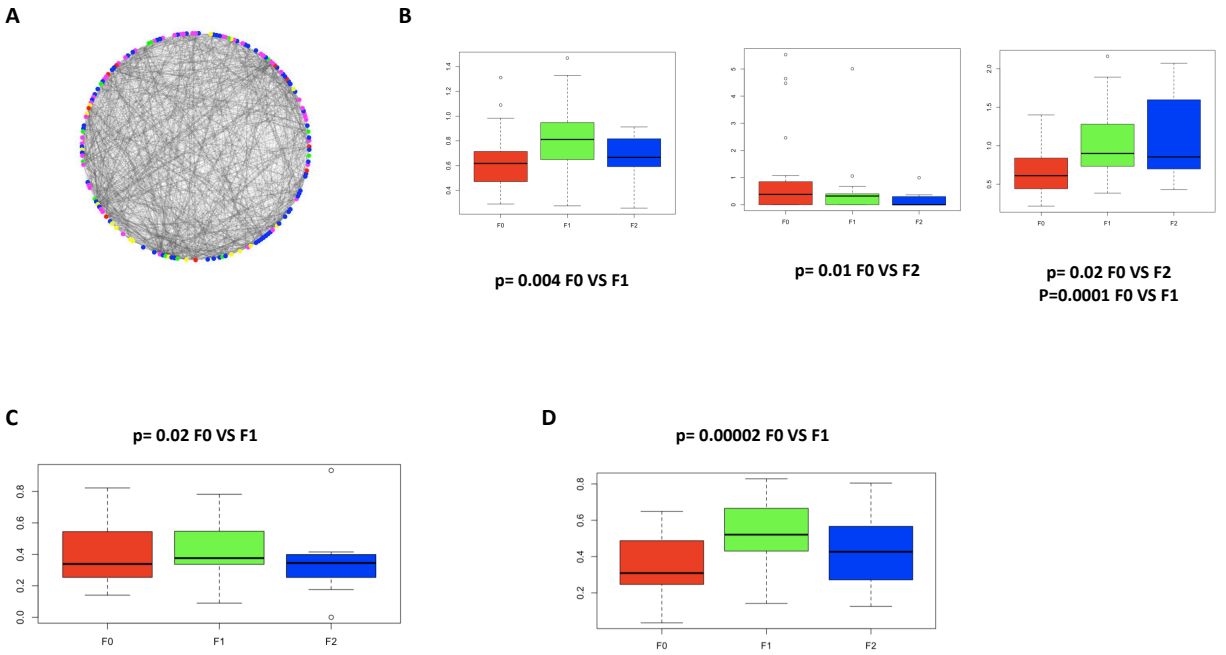

Figure 4.18 – Discriminant analyses of the 16SrDNA OTUs variables using fairness strategies. A Graphical representation of the normalized OTU table counts whose nodes are colored according to the 5 clusters identified by the l_1-spectral clustering algorithm (red= 1, green= 2, blue= 3, pink= 4 and yellow= 5). B Boxplot representing the mean frequencies of the OTUs in cluster 3, 4 and 5, identified by the l_1- spectral clustering algorithm, for the different groups of liver fibrosis scores (red= F_0, green= F_1, blue= F_2). C, D Boxplot representing the frequencies of OTUs in cluster 1, and 2, identified by fair-tree algorithm, for the different groups of liver fibrosis scores (red= F_0, green= F_1, blue= F_2).

OTU name	Phylum	Family	Genus	Significance
Cluster_{20}	Proteobacteria	Burkholderiaceae	Ralstonia	F_0 VS F_1
Cluster_{15}	Proteobacteria	Xanthobacteraceae	Bradyrhizobium	F_0 VS F_1
Cluster_{16}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	F_0 VS F_1
Cluster_{31}	Proteobacteria	Pseudomonadaceae	Pseudomonas	F_0 VS F_1
Cluster_{24}	Proteobacteria	Enterobacteriaceae	Kluyvera	F_0 VS F_1
Cluster_{14}	Proteobacteria	Xanthomonadaceae	Stenotrophomonas	F_0 VS F_1
Cluster_{4}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	F_0 VS F_1
Cluster_{25}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	F_0 VS F_1 and F_0 VS F_2
Cluster_{11}	Proteobacteria	Pseudomonadaceae	Pseudomonas	F_0 VS F_1
Cluster_{89}	Actinobacteria	Corynebacteriaceae	Corynebacterium 1	F_0 VS F_1 and F_0 VS F_2
Cluster_{5}	Bacteroidetes	Flavobacteriaceae	Flavobacterium	F_0 VS F_1 and F_0 VS F_2

Table 4.5 – Identification of clusters of cohort-independent 16S rDNA associated with the different low scores of fibrosis from sPL-SDA

clustering) (Figure 4.19). Interestingly, from all selected OTUs eight common OTUs were from the same phylum i.e. Proteobacteria (Table 4.9) suggesting that most of the discriminant information could be due to these taxa. However, there is still most likely some information that this predominant family could be hiding. We therefore set a new mathematical strategy to exemplify the low frequency and meaningful bacteria by using the TF-IDF (Term frequency-inverse document frequency) approach.

3.5 Low frequency bacterial 16SrDNA gene contains classifying information

From the table count of all significant OTUs detected we generated a "word-cloud" (Figure 4.20 A,B) to visualize the most abundant TF-IDF transformed OTU counts, regardless of fibrosis scores when com- pared to those non-normalized. Cluster 2 emerged as the most important discriminant OTU (taxonomic identifaction=Bacteria|Proteobacteria|Gammaproteobacteria|Enterobacteriales |Enterobacteriaceae|Escherichia- Shigella) further confirming the important amount of information contained in the Proteobacteria.

Based on the identified specific signatures the next step was to generate hypotheses regarding their putative mode of action to the induction of the early events of liver fibrosis. We therefore performed predicted functional metagenomics.

3.6 Predicted functional metagenome pathways

To identify the pathways and enzymes involved in the early development of fibrosis, we run predicted functional metagenomics algorithms based on the fairness-selected bacterial taxa. The heatmap shows clusters of enzymes that are associated with the F_0 vs $F_{1/2}$ fibrosis scores (Figure 4.21 A). Eventually, sPLS-DA showed also a clear discrimination between the F_0 vs $F_{1/2}$ fibrosis scores. To evaluate the accuracy and sensitivity of our analyses as potential diagnostic tool, we drew a ROC and quantified the urea under curve with a score of 81.4% of accuracy (Figures 4.21 B,C). We performed a similar analysis on pathways and showed specific clusters also discriminately

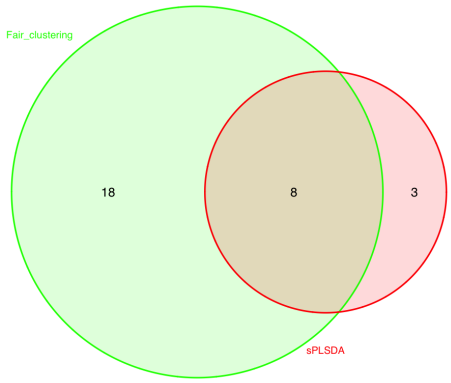

Figure 4.19 – Discriminant analyses of the 16S rDNA OTUs variables using fairness strategies. Venn diagram depicting the liver microbial taxonomies of common OTUs identified by standard (sPLS-DA) and fair approaches (fairtree, random forest, ℓ_1-spectral clustering) as signatures of low fibrosis scores (green= sPLSDA, red= fair algorithms).

OTU name	Phylum	Family	Genus	Significance
Cluster_{31}	Proteobacteria	Pseudomonadaceae	Pseudomonas	F_0 VS F_1
Cluster_{66}	Proteobacteria	Pasteurellaceae	Haemophilus	F_1 VS F_2 and F_0 VS F_2
Cluster_{335}	Proteobacteria	Burkholderiaceae	Janthinobacterium	F_0 VS F_1
Cluster_{42}	Proteobacteria	Burkholderiaceae	Tepidimonas	F_0 VS F_1
Cluster_{341}	Proteobacteria	Enterobacteriaceae	Enterobacter	F_0 VS F_1
Cluster_{248}	Proteobacteria	Reyranellaceae	Reyranella	F_0 VS F_1 and F_0 VS F_2
Cluster_{231}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	F_0 VS F_1
Cluster_{36}	Bacteroidetes	Flavobacteriaceae	Flavobacterium	F_1 VS F_2 and F_0 VS F_2
Cluster_{64}	Actinobacteria	Corynebacteriaceae	Lawsonella	F_1 VS F_2 and F_0 VS F_2
Cluster_{339}	Proteobacteria	Burkholderiaceae	Limnohabitans	F_1 VS F_2
Cluster_{15}	Proteobacteria	Xanthobacteraceae	Bradyrhizobium	F_0 VS F_1
Cluster_{44}	Actinobacteria	Intrasporangiaceae	Multi-affiliation	F_0 VS F_1 and F_0 VS F_2
Cluster_{25}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	F_0 VS F_1
Cluster_{45}	Proteobacteria	Burkholderiaceae	Comamonas	F_0 VS F_1
Cluster_{14}	Proteobacteria	Xanthomonadaceae	Stenotrophomonas	F_0 VS F_1

Table 4.6 – Identification of clusters of cohort-independent 16S rDNA associated with the different low scores of fibrosis from Fair-tree

OTU name	Phylum	Family	Genus	Significance
Cluster_{14}	Proteobacteria	Xanthomonadaceae	Stenotrophomonas	F_0 VS F_1
Cluster_{77}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	F_1 VS F_2 and F_0 VS F_2
Cluster_{35}	Actinobacteria	Microbacteriaceae	Rhodoluna	F_0 VS F_1
Cluster_{36}	Bacteroidetes	Flavobacteriaceae	Flavobacterium	F_0 VS F_1
Cluster_{16}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	F_0 VS F_1 and F_0 VS F_2
Cluster_{312}	Proteobacteria	Enterobacteriaceae	Kosakonia	F_0 VS F_1
Cluster_{24}	Proteobacteria	Enterobacteriaceae	Kluyvera	F_0 VS F_1

Table 4.7 – Identification of clusters of cohort-independent 16S rDNA associated with the different low scores of fibrosis from Random Forest

OTU name	Phylum	Family	Genus	Significance
Cluster_{31}	Proteobacteria	Pseudomonadaceae	Pseudomonas	F_0 VS F_1
Cluster_{15}	Proteobacteria	Xanthobacteraceae	Bradyrhizobium	F_0 VS F_1
Cluster_{341}	Proteobacteria	Enterobacteriaceae	Enterobacter	F_0 VS F_1
Cluster_{20}	Proteobacteria	Burkholderiaceae	Ralstonia	F_0 VS F_1
Cluster_{4}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	F_0 VS F_1
Cluster_{2}	Proteobacteria	Enterobacteriaceae	Escherichia-Shigella	F_0 VS F_1
Cluster_{27}	Proteobacteria	Enterobacteriaceae	Serratia	F_0 VS F_1
Cluster_{3}	Proteobacteria	Pseudomonadaceae	Pseudomonas	F_0 VS F_1
Cluster_{24}	Proteobacteria	Enterobacteriaceae	Kluyvera	F_0 VS F_1
Cluster_{22}	Proteobacteria	Burkholderiaceae	Multi-affiliation	F_0 VS F_1
Cluster_{16}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	F_0 VS F_1

Table 4.8 – Identification of clusters of cohort-independent 16S rDNA associated with the different low scores of fibrosis from Fair l_1-spectral clustering

OTU name	Phylum	Family	Genus	Species	Significance
Cluster_{31}	Proteobacteria	Pseudomonadaceae	Pseudomonas	Pseudomonas putida	F_0 VS F_1
Cluster_{15}	Proteobacteria	Xanthobacteraceae	Bradyrhizobium	Bradyrhizobium sp.	F_0 VS F_1
Cluster_{25}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	Multi-affiliation	F_0 VS F_1
Cluster_{14}	Proteobacteria	Xanthomonadaceae	Stenotrophomonas	Multi-affiliation	F_0 VS F_1
Cluster_{16}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	Multi-affiliation	F_0 VS F_1
Cluster_{24}	Proteobacteria	Enterobacteriaceae	Kluyvera	Multi-affiliation	F_0 VS F_1
Cluster_{20}	Proteobacteria	Burkholderiaceae	Ralstonia	Multi-affiliation	F_0 VS F_1
Cluster_{4}	Proteobacteria	Enterobacteriaceae	Multi-affiliation	Multi-affiliation	F_0 VS F_1

Table 4.9 – Microbial signatures common to all strategies

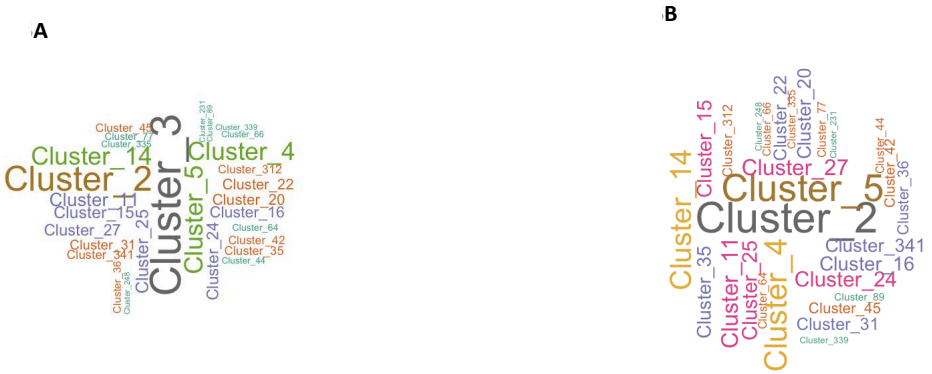

Figure 4.20 – Identification of clusters by wordclouds representation with or without TFIDF normalization. Wordclouds representing taxa of all significant bacteria according to A, their frequencies or B, after TF-IDF normalization. The size of the name of bacteria is proportional to the frequency of the cluster in the cohorts.

associated with the fibrosis scores with a score of accuracy of 81.2% (ROC curve) (Figures 4.22 A-C). We then represented an listed all selected enzymes and pathways highly expressed in the two major discriminant components (Figures 4.23 A-D) and (Tables 4.10, 4.11). Three pathways were highly and negatively associated with the liver fibrosis score of $F_{1/2}$ when compared to the F_0. We identified from the MetaCyc database that the preQ0 biosynthesis (PWY-6703), specific to Enterobacteriaceae such as E. coli, is involved notably in the synthesis of tetrahydrofolate and a class of nucleoside analogues that often possesses antibiotic, antineoplastic, or antiviral activities (Dunphy et al., 1971; Farrand and Taber, 1973) (Figure 4.24 A). In addition, two other pathways related to glucoryranose (PWY 6737) and glycogen (GLYCOCAT-PWY) degradation were identified probably providing energy to the main preQ0 biosynthesis pathway.

On the other hand, six major metabolic pathways were positively associated with the F_0 score from both components. One involves the glycolysis and pentose phosphate pathway (PWY-6629), while the 5 others are all involved in the menaquinones and demethylmenaquinones pathway (Figure 4.24 B). The low-molecular weight lipophilic components of the cytoplasmic membrane are considered vitamin $K2$ components that is found in most aerobic Gram-positive bacteria and are the main quinones that function as a reversible redox component of the electron transfer chain, mediating electron transfer between hydrogenases and cytochromes. Altogether the functional metagenomics prediction suggests that gram negative bacteria from the Proteobacteria family composed of preQ0 biosynthesis and glycolytic pathway are signature of $F_{1/2}$ fibrosis scores while the vitamin $K2$ biosynthesis pathway from gram negative bacteria such as Actinobateriaceae (Iwata-Reuyl, 2003; McCarty et al., 2009) would be signing F_0 liver fibrosis score.

4 Discussion

We here report a mathematical approach to identify a bacterial 16S rDNA signature in liver tissue and corresponding putative biochemical pathways in patients with low scores of fibrosis. Our main finding is that even low scores of fibrosis (F_0 vs $F_{1/2}$) can be classified by biomarkers from the Proteobacteriaceae family within the liver. The second observation is related to the importance of cohort heterogeneity in term of size and data variability which could be major confounding factors that must be taken into account in multi-centric clinical trials or database. We here present a mathematic approach that could help solving this major and common issue.

A gut metagenomics signature of liver fibrosis in humans has been recently described, suggestive of its causal role in the disease (Loomba et al., 2017). However, such patients where mostly characterized by a high score of liver fibrosis questioning the putative causal role of the liver microbiota in the disease. We here focused our attention on low scores of liver fibrosis to putatively identify causal factors. We identified mostly gram negative bacteria and notably the Proteobacteria as signature of the $F_{1/2}$ liver fibrosis scores. Among the families the Proteobacteriaceae, Flavobacteriaceae, and Propionibacteriaceae were discriminating the low fibrosis scores from each other's. They synthesize LPS, a dramatically inflammatory suggesting a pathophysiological role in development of liver fibrosis, probably via the maintenance of a certain degree of immune vigilance. We further refined our analyses and mostly selected Enterobac- teriaceae family from the Proteobacteria phylum suggesting that the liver proinflammation observed dur- ing fibrosis would be due or associated with genera

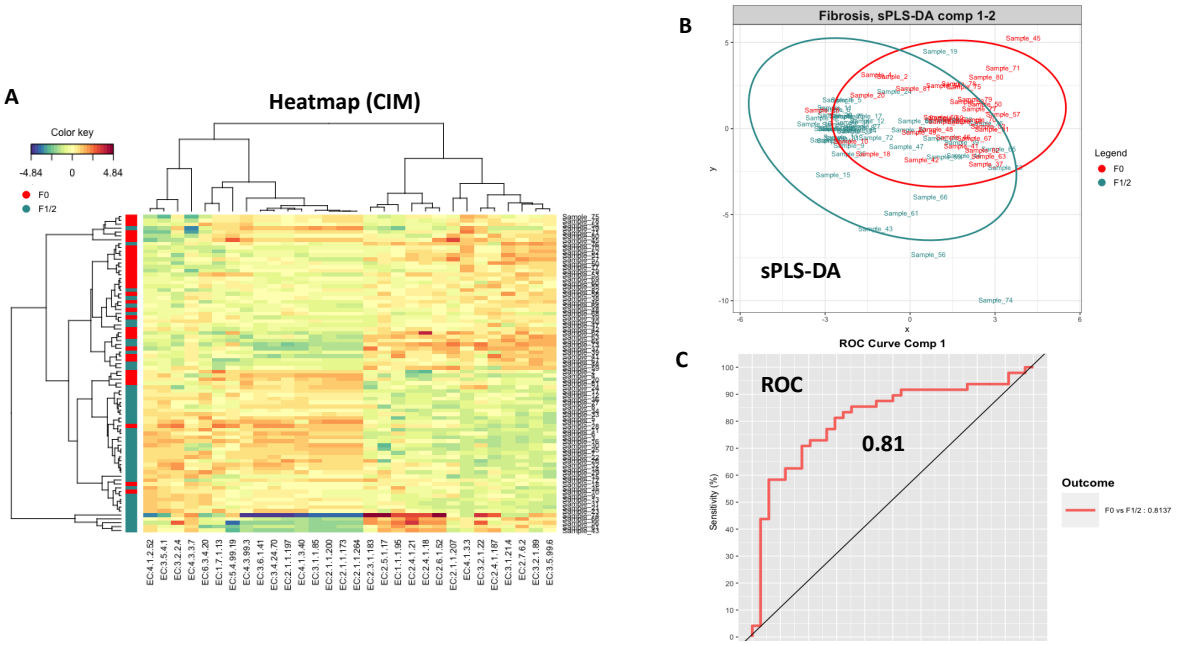

Figure 4.21 – Predicted functional metagenomics analyses of discriminant enzymes according to the fibrosis score. A: Heatmap (Clustering Image Map (CIM)), B: Sample plot, each point corresponds to an individual and is colored according to its liver fibrosis score (red= F_0, green= $F_{1/2}$), C: ROC classification performances of enzymes on a CSS normalized enzyme table count of the F_0 versus $F_{1/2}$ groups of patients.

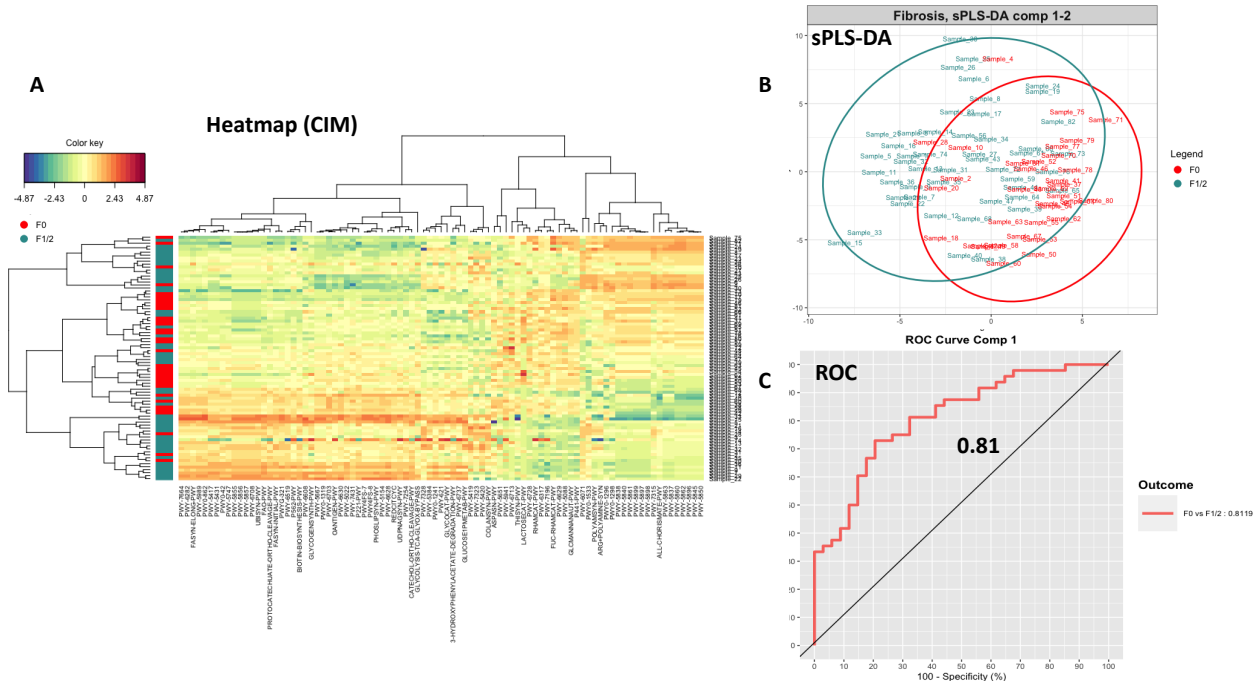

Figure 4.22 – Predicted functional metagenomics analyses of discriminant pathways according to the fibrosis score. A: Heatmap (Clustering Image Map (CIM)), B: Sample plot, each point corresponds to an individual and is colored according to its liver fibrosis score (red= F_0, green= $F_{1/2}$), C: ROC classification performances of pathways on a CSS normalized pathways table count of the F_0 versus $F_{1/2}$ groups of patients.

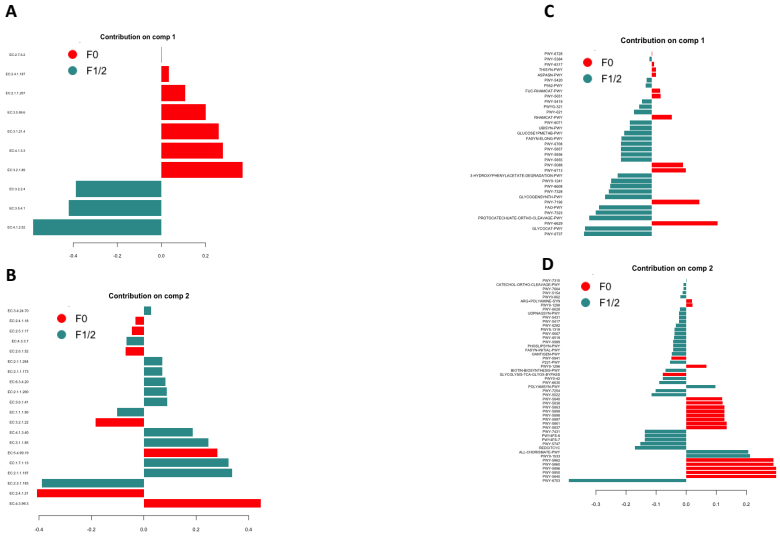

Figure 4.23 – Predicted functional metagenomics analyses of discriminant enzymes and pathways according to the fibrosis score. Loading plot representing the contribution of each enzyme (A,B), and pathways (C,D) selected to build the first and second components (red= F_0, green= $F_{1/2}$).

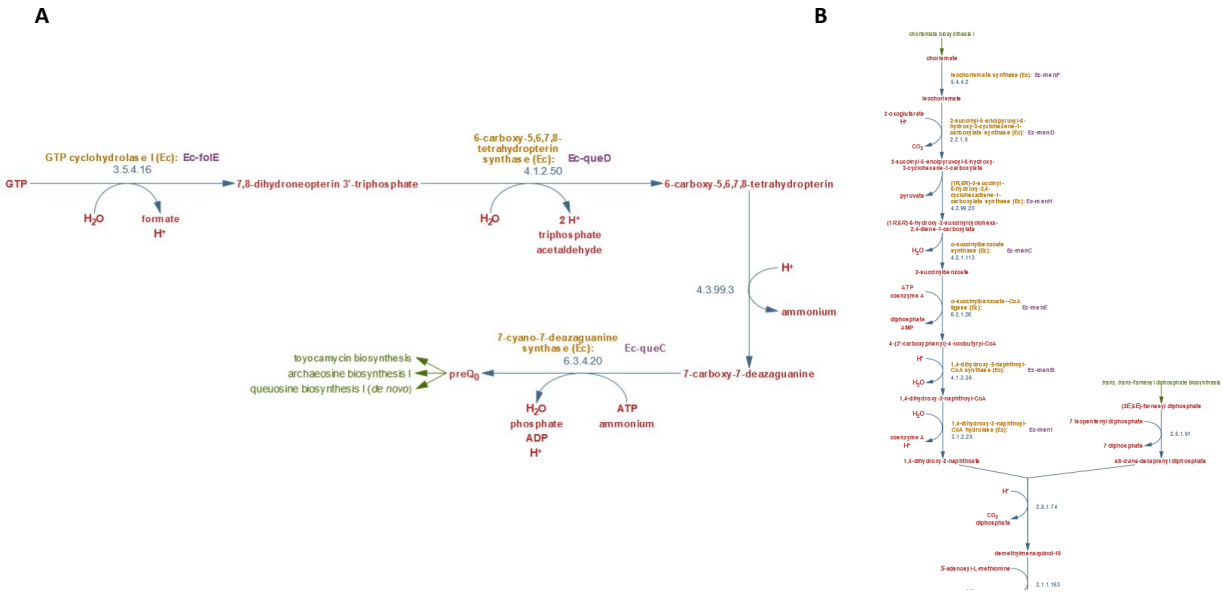

Figure 4.24 – Predicted functional metagenomics analyses of discriminant enzymes and pathways according to the fibrosis score. A,B: main metabolic pathways from the MetaCyc database identified from the Loading plots for the A: $F_{1/2}$ and B: F_0 liver fibrosis scores.

Name	Function
EC:4.1.2.52	4-hydroxy-2-oxoheptanedioate aldolase
EC:3.5.4.1	Cytosine deaminase
EC:3.2.2.4	AMP nucleosidase
EC:4.3.3.7	4-hydroxy-tetrahydrodipicolinate synthase
EC:6.3.4.20	7-cyano-7-deazaguanine synthase
EC:1.7.1.13	PreQ(1) synthase
EC:5.4.99.19	16S rRNA pseudouridine(516) synthase
EC:4.3.99.3	7-carboxy-7-deazaguanine synthase
EC:3.6.1.41	Bis(5'-nucleosyl)-tetraphosphatase (symmetrical)
EC:3.4.24.70	Oligopeptidase A
EC:2.1.1.197	Malonyl-[acyl-carrier protein] O-methyltransferase
EC:4.1.3.40	Chorismate lyase
EC:3.1.1.85	Pimeloyl-[acyl-carrier protein] methyl ester esterase
EC:2.1.1.200	tRNA (cytidine(32)/uridine(32)-2'-O)-methyltransferasev
EC:2.1.1.173	23S rRNA (guanine(2445)-N(2))-methyltransferase
EC:2.1.1.264	23S rRNA (guanine(2069)-N(7))-methyltransferasev
EC:2.3.1.183	Phosphinothricin acetyltransferase
EC:2.5.1.17	Cob(I)yrinic acid a,c-diamide adenosyltransferase
EC:1.1.1.95	Phosphoglycerate dehydrogenase
EC:2.4.1.21	Starch synthase
EC:2.4.1.18	1,4-alpha-glucan branching enzymev
EC:2.6.1.52	Phosphoserine transaminase
EC:2.1.1.207	tRNA (cytidine(34)-2'-O)-methyltransferase
EC:4.1.3.3	N-acetylneuraminate lyase
EC:3.2.1.22	Alpha-galactosidase
EC:2.4.1.187	mannosaminyltransferase
EC:3.1.21.4	Type II site-specific deoxyribonuclease
EC:2.7.6.2	Thiamine diphosphokinase
EC:3.2.1.89	Arabinogalactan endo-beta-1,4-galactanase
EC:3.5.99.6	Glucosamine-6-phosphate deaminase

Table 4.10 – Identification of specific enzymes, signatures of low score of fibrosis

Name	Function
PWY-7664	oleate biosynthesis IV (anaerobic)
PWY-6282	palmitoleate biosynthesis I (from (5Z)-dodec-5-enoate)
FASYN-ELONG-PWY	fatty acid elongation – saturated
PWY-5989	stearate biosynthesis II (bacteria and plants)
PWY0-862	(5Z)-dodec-5-enoate biosynthesis
PWY-5417	catechol degradation III (ortho-cleavage pathway)
PWY-5431	aromatic compounds degradation via &beta,-ketoadipate
PWY0-42	2-methylcitrate cycle I
PWY-5747	2-methylcitrate cycle II
PWY-5855	ubiquinol-7 biosynthesis (prokaryotic)
PWY-5856	ubiquinol-9 biosynthesis (prokaryotic)
PWY-5857	ubiquinol-10 biosynthesis (prokaryotic)
PWY-6708	ubiquinol-8 biosynthesis (prokaryotic)
UBISYN-PWY	superpathway of ubiquinol-8 biosynthesis (prokaryotic)
FAO-PWY	fatty acid &beta,-oxidation I
PROTOCATECHUATE- ORTHO-CLEAVAGE-PWY	protocatechuate degradation II (ortho-cleavage pathway)
FASYN-INITIAL-PWY	superpathway of fatty acid biosynthesis initiation (E. coli)
PWYG-321	mycolate biosynthesis
PWY-6519	8-amino-7-oxononanoate biosynthesis I
P562-PWY	myo-inositol degradation I
BIOTIN-BIOSYNTHESIS- PWY	biotin biosynthesis I
PWY-6608	guanosine nucleotides degradation III
GLYCOGENSYNTH-PWY	glycogen biosynthesis I (from ADP-D-Glucose)
PWY-5667	CDP-diacylglycerol biosynthesis I
PWY0-1319	CDP-diacylglycerol biosynthesis II
PWY-6703	preQ0 biosynthesis
OANTIGEN-PWY	O-antigen building blocks biosynthesis (E. coli)
PWY-6630	superpathway of L-tyrosine biosynthesis
PWY-5022	4-aminobutanoate degradation V
PWY-7431	aromatic biogenic amine degradation (bacteria)
P221-PWY	octane oxidation
PWY4FS-7	phosphatidylglycerol biosynthesis I (plastidic)
PWY4FS-8	phosphatidylglycerol biosynthesis II (non-plastidic)
PHOSLIPSYN-PWY	superpathway of phospholipid biosynthesis I (bacteria)
PWY-5154	L-arginine biosynthesis III (via N-acetyl-L-citrulline)
PWY-6628	superpathway of L-phenylalanine biosynthesis
REDCITCYC	TCA cycle VIII (helicobacter)
UDPNAGSYN-PWY	UDP-N-acetyl-D-glucosamine biosynthesis I
PWY-7254	TCA cycle VII (acetate-producers)
CATECHOL-ORTHO- CLEAVAGE-PWY	catechol degradation to &beta,-ketoadipate
GLYCOLYSIS-TCA-GLYOX- BYPASS	superpathway of glycolysis, pyruvate dehydrogenase, TCA, and glyoxylate bypass
PWY-7328	superpathway of UDP-glucose-derived O-antigen building blocks biosynthesis
PWY-5384	sucrose degradation IV (sucrose phosphorylase)
PWY0-1241	ADP-L-glycero-&beta,-D-manno-heptose biosynthesis
PWY-621	sucrose degradation III (sucrose invertase)
GLYCOCAT-PWY	glycogen degradation I (bacterial)
3- HYDROXYPHENYLACETATE- DEGRADATION-PWY	v4-hydroxyphenylacetate degradation
PWY-6737	starch degradation V
GLUCOSE1PMETAB-PWY	vglucose and glucose-1-phosphate degradation
PWY-5419	catechol degradation to 2-oxopent-4-enoate II
PWY-7323	superpathway of GDP-mannose-derived O-antigen building blocks biosynthesis
PWY-5420	catechol degradation II (meta-cleavage pathway)
COLANSYN-PWY	vcolanic acid building blocks biosynthesis
ASPASN-PWY	superpathway of L-aspartate and L-asparagine biosynthesis
PWY-5651	L-tryptophan degradation to 2-amino-3-carboxymuconate semialdehyde
PWY-5941	glycogen degradation II (eukaryotic)
PWY-6713	L-rhamnose degradation II
THISYN-PWY	superpathway of thiamin diphosphate biosynthesis I
LACTOSECAT-PWY	lactose and galactose degradation I
PWY-6728	methylaspartate cyclev
RHAMCAT-PWY	L-rhamnose degradation I
PWY-6317	galactose degradation I (Leloir pathway)
PWY-7196	superpathway of pyrimidine ribonucleosides salvage
FUC-RHAMCAT-PWY	superpathway of fucose and rhamnose degradation
PWY-6629	superpathway of L-tryptophan biosynthesis
PWY-5088	L-glutamate degradation VIII (to propanoate)
GLCMANNANAUT-PWY	superpathway of N-acetylglucosamine, N-acetylmannosamine and N- acetylneuraminate degradation
P441-PWY	superpathway of N-acetylneuraminate degradation
PWY-6071	superpathway of phenylethylamine degradation
PWY0-1533	methylphosphonate degradation I
POLYAMSYN-PWY	superpathway of polyamine biosynthesis I
ARG+POLYAMINE-SYN	superpathway of arginine and polyamine biosynthesis
PWY0-1296	purine ribonucleosides degradation
PWY0-1298	superpathway of pyrimidine deoxyribonucleosides degradation
PWY-5838	superpathway of menaquinol-8 biosynthesis I
PWY-5840	superpathway of menaquinol-7 biosynthesis
PWY-5861	superpathway of demethylmenaquinol-8 biosynthesis
PWY-5899	superpathway of menaquinol-13 biosynthesis
PWY-5897	superpathway of menaquinol-11 biosynthesis
PWY-5898	superpathway of menaquinol-12 biosynthesis
PWY-7315	dTDP-N-acetylthomosamine biosynthesis
ALL-CHORISMATE-PWY	superpathway of chorismate metabolism
PWY-5863	superpathway of phylloquinol biosynthesis
PWY-5837	1,4-dihydroxy-2-naphthoate biosynthesis I
PWY-5860	superpathway of demethylmenaquinol-6 biosynthesis I
PWY-5862	superpathway of demethylmenaquinol-9 biosynthesis
PWY-5896	superpathway of menaquinol-10 biosynthesis
PWY-5845	superpathway of menaquinol-9 biosynthesis
PWY-5850	superpathway of menaquinol-6 biosynthesis I

Table 4.11 – Identification of specific Pathways, signatures of low score of fibrosis

from the Enterobacteriaceae family (Cani et al., 2007). The Enterobacteriaceae encompass the genera Arthrobacter and Acinetobacter. The mechanisms through which such bacteria could induce inflammation might be linked to the unique structures of their LPS or peptidoglycans (Farrand and Taber, 1973). Furthermore, since such bacteria are motile with flagella, one could also contemplate that the flagellar proteins are involved in the liver fibrosis process. However, data report that the TLR5 receptor of flagellin is rather associated with protection against metabolic syndrome, putatively ruling out this hypothesis (Latino et al., 2007). Through functional metagenomics production we identified the preQ0 biosynthesis pathway as a signature of $F_{1/2}$ fibrosis scores. Such pathway is notably identified from gram negative bacterial such as Proteobacteriaceae (Kanehisa et al., 2012; Iwata-Reuyl, 2003). Conversely, the menaquinones and demethylmenaquinones pathways involved in $K12$ vitamin synthesis were the signature of the F_0 score. They are notably produced by the gram positive Actinobacteriaceae such as Bacillus subtilis (Dunphy et al., 1971), therefore coherence with our metagenomics findings.

A major hurdle of aggregation of different cohort altogether is related to the heterogeneity of the size of the groups and of the diversity of the variables considered. Regarding invasive analyses such as liver biopsies the group size at completion of the inclusions could be different from what predicted during the calculation of power of the trial. Eventually, the distribution of the variables to be studied could be highly heterogeneous for a given disease. Altogether, we here faced several statistical challenges which are linked to liver fibrosis. The first major step preceding the microbial analysis was a prefiltering and then an adapted pathway to normalize the data to deal with their sparse nature. The package Mixomics (Le Cao et al., 2008) used for this study recommends CSS normalization on sparse OTU table counts that could prevent the bias included in the TSS normalization. In addition, it includes multivariate methods for microbiome studies and addresses its limits. In addition, we observed a strong impact of the cohort of origin since the largest cohort from Romania could discriminate the patients from the others based on the 16S rDNA OTU variables. The patients could even be classified by cohort when we used the clinical data as entries showing that this issue also has to be taken into account when analyzing the data. Mathematical approaches to overcome this issue are currently being developed however, little has been done regarding the handling of the 16S rDNA data now widely used by the scientific community that addresses the role of microbiota on diseases and notably liver diseases. Therefore, we here developed several approaches of fairness to overcome the classical cohort impact. Eventually, we noticed that two patients from the F_1 groups were distributed with the F_2 group. This ectopic distribution could be due to the extreme BMI (> 55) featuring a specific clinical phenotype. Conversely, a patient from the F_2 group was associated with the F_0/F_1 distribution. This patient was characterized by his young age (< 40 years old) while the mean age of the F_2 group was of 54 years old.

The statistical approach required to properly analyze microbial data sets needed to be better fitted to the nature of the data. As a preliminary analysis we performed PCoA since better adapted than PCA to dissimilar and sparse data then followed by a sPLS-DA to identify subsets of 16S rDNA that are discriminatory for the liver fibrosis scores. PLS-DA aims to classify a data set according to the values of a qualitative variable by maximizing the covariance between linear combinations of the observed variables and the qualitative outcome. The sparse version, on the other hand, delivers variables per each component, only selected in the OTU dataset, that are the most discriminatory for

the liver fibrosis scores. We focused our attention on the identification of the OTU frequencies within and across each group of patients and on the understanding of the importance that OTUs carry within and across the cohort. We found that the data set is mostly populated by a few high frequency OTUs. However, beside the level of information gained form this approach where overrepresented OTUs we identified cannot rule out that some more information could be obtained from OTUs rarely represented. Therefore, some information could be hidden in the low frequency OTUs. To test this hypothesis we introduced a new normalization approach called TF-IDF (Wang et al., 2017) originally developed for text mining, to attenuate the effects of the high frequencies OTUs in the data set. Furthermore, aside from the fibrosis scores, it reveals some new predominant taxa at the different taxonomic levels.

5 Conclusion

In conclusion, the first evidence of the existence of a liver microbiota opens alternate routes for novel therapeutic strategies since specific bacteria could be involved in the process of liver fibrosis. However, to generate information which could serve as a substratum to reach this aim, we here adapted predicted metagenomics and mathematical approaches to the original and novel nature of the tissue metagenomics data set. We here found that these data are constituted of high heterogeneity variables which are dominated by a few high frequency taxa such as Proteobacteria, signature of $F_{1/2}$ liver fibrosis scores, and Actinobacteria/Firmucutes, signature of F_0 liver fibrosis scores. These major taxa are masking information residing in the lower frequency taxa. Predicting metabolic pathways from selected 16S rDNA-based taxa revealed a role of folate metabolism in $F_{1/2}$ liver fibrosis scores while a role of vitamin $K12$ biosynthesis was characterizing F_0 liver fibrosis score. Altogether, the combined use of metagenomics, sPLS-DA, TF-IDF and fairness strategies appeared useful since we identified signatures specific to the lower scores of liver fibrosis i.e. at the onset of the disease.

Chapter 5

Fair conditional Partial Least Square

1 Introduction

Machine learning algorithms aim to find statistical patterns in a training dataset that can be generalized to the whole population. Here, the term generalization refers to the model's faculty to adapt and make accurate prediction of the outcome from previously unseen new data, which have been drawn from the same distribution as those used to build the model. Over the last past years, machine learning algorithms became efficient tools for the development of various technological applications, first mimicking human performances and then performing better. They are now widely used in many fields, among the most well-known: finance (Trippi and Turban, 1992), education (McArthur and Bishary, 2005), social media (Zeng et al., 2010), business (Dirican, 2015), robotics (Dirican, 2015; Brady, 1985) automotive or industry (Gusikhin et al., 2007; Luckow et al., 2016). This list is not exhaustive and can be expanded to healthcare (Morik, 2010; Hamet and Tremblay, 2017; Yu et al., 2018; Sidey-Gibbons, 2019), which raised in recent years a growing interest with a wide potential of applications: radiology (Hosny et al., 2018; Pisarchik et al., 2019), screening (Bellemo et al., 2019; McKinney et al., 2020), primary care (Blease et al., 2019) or disease diagnosis (Liang et al., 2019).

As Machine learning algorithms' accuracy highly depends on how the dataset conveys reliable information, handling algorithmic bias is a major concern (Parikh et al., 2019; DeCamp and Lindvall, 2020; Tat et al., 2020; Nelson, 2019). In addition, given the high inherent stakes in medical trials, researchers have to rethink how their algorithms are certified before being used at larger scales. To tackle this issue, it is particularly needed to understand where the bias comes from. First it may come directly from models that have been learnt from unbalanced data and will then fit the majority but behave badly for others. In a medical context, bias is particularly stringent since data depend on how it is collected. This is especially true when it comes from distinct cohorts, aggregated into a single one, where the nature of the phenotypes differs. Age, sex, ethnic origin are well known examples of variables that may introduce harmful bias to machine learning models. Yet, as quoted in Dias and Torkamani (2019), bias due to underrepresented population leads to the propagation of non-causal bias and thus bad prediction. It is the case in the diagnosis of Down syndrome which is influenced by the origin of the different populations, as pointed by Lumaka et al. (2017). It may also come from variables with a higher variance in some populations or the presence of a confounder effect

in a subgroup. An example of biased dataset which influences the prediction by detecting spurious correlations or confounders is presented in Esteva et al. (2017) as part of a skin cancer study. Finally, the evolution of the characteristics of the population, leading to shifts in the distribution of the patients, may jeopardize the mid or long term use of the machine learning algorithms, as described in Challen et al. (2019). To tackle the bias issue, new methods were developed in recent years, leading to the emergence of a new field called fairness (Pérez-Suay et al., 2017; Dwork et al., 2012; Gillen et al., 2018; Kearns et al., 2019; Kleindessner et al., 2019), which is attracting increasing attention and is progressing in a wide range of applications (Joseph et al., 2016, 2018; R. et al., 2018; Buolamwini and Gebru, 2018; Siegel, 2014; Doherty et al., 2012; Johndrow and Lum, 2019). Studying the effect of some variables, which are more susceptible to change across time, and building fair models that mitigate their effect may improve the stability of machine learning algorithms and guarantee their performance. Clinical evaluation of such methods should include a careful study of the performances of the algorithms for each subgroup of the population, as pointed out by Kelly et al. (2019), and should be understandable by medical experts of the corresponding field. Hence, explainable models should be preferred over black box models.

In this chapter, we faced this problem when statistically analyzing two biomedical datasets: the Florinash genomic and the liver fibrosis microbial datasets, already presented in chapter IV. The common guideline of their statistical studies was to provide a new representation of the datasets that can be used to understand their total variability and, ultimately, to find, within the variables, biological signatures of the patients' liver pathology. Both were preliminary analyzed through an exploratory technique: Principal Component Analysis (PCA) (Pearson, 1901). However, the generated graphic representations show some degree of differential distributions, suggesting the existence of specific environmental factors. The florinash dataset includes patients with singular phenotypic profile due to their nationality, whereas the liver fibrosis one stems from two distinct cohorts. As regard the induced bias, we must remove this intrinsic bias before further analysis using an unbiased algorithm. In this biological framework, an algorithm is unbiased when a variable $S = (S_1, \ldots, S_n)^T$, also called *protected* or *sensitive attribute*, chosen prior to the analysis, does not interfere in the discovery of the biomarkers associated with the target $Y = (Y_1, \ldots, Y_n)^T$ from the observations $X = (X_i^j)_{i=1,\ldots,n,j=1,\ldots,p}$. We thus developed unbiased algorithms based on PCA exploratory analysis and on Partial Least Square (PLS) regression technique (Wold et al., 1984; Wold, 1975) to provide fair representations of the dataset related to the clinical variable of interest.

The most naive approach is to simply not include the biased features in the model but it is not enough to mitigate the bias. In fact, model's accuracy may be impacted by the lack of informative variables as some variables strongly correlated to the omitted features will propagate the bias they contain. In this work, fairness is guaranteed by controlling the independence between the orthogonal features built by standard PCA or PLS and the sensitive variables using a threshold term. The final insensitive transformed variables are selected with an optimal threshold through a cross-validation procedure.

The chapter is organized as follows: in Section 2, we introduce some preliminary concepts about standard PCA and PLS methods. Then, in Section 3 we present in general terms the fair strategy we developed, the theoretical and algorithmic development of the fair methods. As our work was driven by medical projects, we illustrated the efficiency and accuracy of the algorithms through experiments

on the datasets.

2 Standard PCA and PLS

2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a traditional multivariate statistical method that aims to reduce the dimension of a dataset by extracting few linear combinations of the variables without losing too much information (Pearson, 1901; Jolliffe and Cadima, 2016). This popular technique for dimensionality reduction is also commonly used as a preliminary step before applying any regression or classification technique to better visualize the variation that exists in the dataset.

Consider a system of p quantitative variables $X^1, \ldots, X^p$ and n observations $X^j := (X_1^j, \ldots, X_n^j)^T$, $j \in \{1, \ldots, p\}$ of these p variables. PCA creates a set of decorrelated variables, called principal components, by orthogonally transforming the p variables. Therefore, a set of p vectors $(w_h)_{h\in\{1,\ldots,p\}}$, called principal factors, is firstly found such that the maximal variance in the original dataset is preserved. Principal factors, are successively built as solutions of the following optimization problems:

$$\forall h \in \{1, \ldots, p\}\ ,\ w_h = \underset{\|w\|=1}{\arg\max}\ \text{Var}(Xw), \tag{5.1}$$

under the constraint that for all $l \in [\![1, h-1]\!]$, $w_h^T X^T X w_l = 0$. Then, a set of p principal components $T^1, \ldots, T^p$, defined as linear combinations of the centred variables X^j, are built:

$$\forall h \in \{1, \ldots, p\}\ ,\ T^h = Xw_h.$$

Note that the principal components are ordered from the most informative T^1 to the least informative T^p.

One significant challenge in using PCA technique is to select the optimal number of principal components. To adress this issue, a large number of graphical methods have been developed in recent years. In what follows, we will note $f_{\text{PCA}}(w) = Var(Xw)$ the objective function to optimize in PCA (equation 5.1).

2.2 Partial Least Square

Partial Least Square (PLS) is a technique closely related to PCA. While PCA reduces the dimension of a single dataset by iteratively projecting the data onto components of maximum variance, PLS reduces the dimension of a dataset by projecting the data onto components of maximum covariance with a second data set. It is a supervised technique developed especially for regression, discrimination and classification (Wold et al., 1984; Wold, 1975). For an application to large omics datasets, we refer to the mixOmics package (Le Cao et al., 2008), dedicated to the development of multivariate methods. Note that a mathematical framework has been provided by Blazère et al. (2014).

Consider the matrix X of p predictors previously introduced and a random variable $Y = (Y_1, \ldots, Y_n)^T$ representing the target variable we want to predict from the observed X. PLS iteratively creates decorrelated components $T^1, \ldots, T^p$, called latent variables, defined as linear combinations of the p original centred variables $(X^1, \ldots, X^p)$:

$$\forall h \in \{1, \ldots, p\}\,,\ T^h = Xw_h,$$

where the vectors $(w_1, \ldots, w_p)$, called principal factors, are solutions of the following optimization problems :

$$\forall h \in \{1, \ldots, p\}\,,\ w_h = \underset{\|w\|=1}{\arg\max}\, \mathrm{Cov}(Xw, Y), \tag{5.2}$$

under the constraint $\forall l \in [\![1, h-1]\!],\ w_h^T X^T X w_l = 0$.

Contrary to PCA, the optimal number of PLS latent variables is automatically determined by cross-validation (Lachenbruch and Mickey, 1968). Note that PLS is commonly combined with a regression step. The resulting method, called Partial Least Square Regression (PLSR) yields two variables B and B_0 such that $\hat{Y} = B_0 + TB$ is the best approximation of Y, where T is the matrix obtained by concatenation of the centred latent variables.

In what follows, we will note $f_{\mathrm{PLS}}(w) = Cov(Xw, Y)$ the objective function to optimize in PLS (equation 5.2).

The biomedical studies conducted so far on biased dataset led us to develop fair learning approaches which rely on such standard dimensionality reduction techniques. Based on specific independence criteria, these strategies aim to explain the total variablities in the dataset while limiting the bias effect.

3 Fair conditional independence strategies

In this section, we present two direct fair strategies based on PCA/PLS techniques that aim to find fair components, containing enough variance to predict a clinical outcome. In the following subsections, we consider a response $Y \in \mathbb{R}^n$, p unprotected variables $X^1, \ldots, X^p$ and one sensitive variable $S \in \{0, 1\}^n$ inducing bias.

3.1 Overall strategy

Fairness is guaranteed in a post-processing step by imposing independence between the constructed orthogonal features and the sensitive variable using a correlation measure (see Figure 5.1 for a global overview). From all components created by PCA/PLS, we aim to select a set of variables both decorrelated to the variable causing bias and related to the clinical response. To this end, each correlation coefficient is thresholded to keep the less dependent features. The optimal threshold is selected using a cross validation procedure on the basis of the two following criteria:

— the MSE when predicting the outcome from a fitted linear model to guarantee a good prediction,
— the total explained variance of the selected components to restitute enough variability of the original dataset.

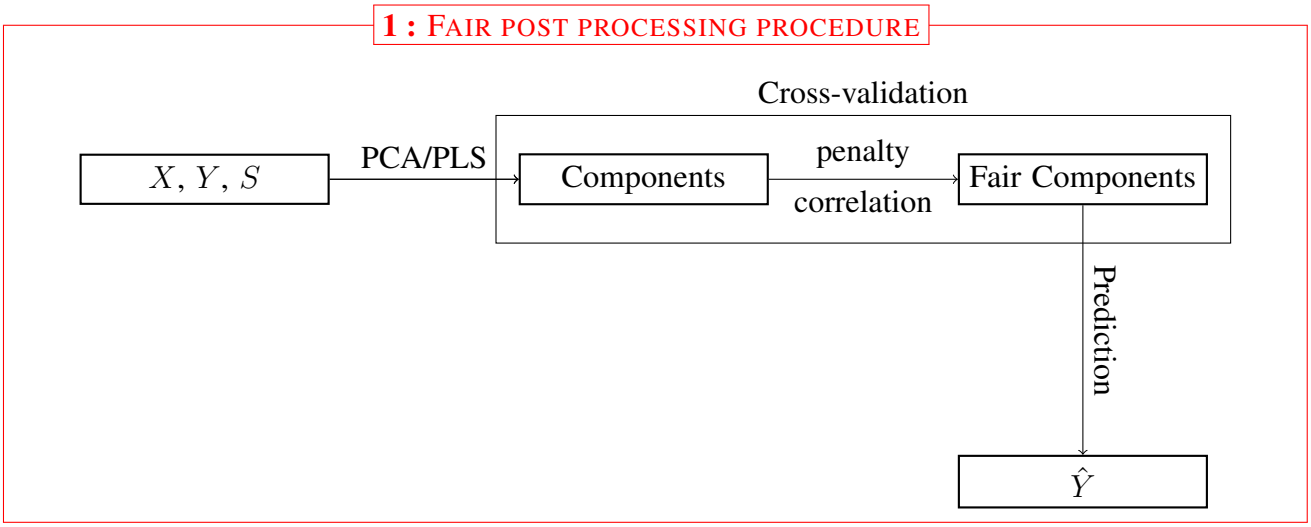

Figure 5.1 – Fair post processing procedure

3.2 Fair PCA based on correlation measure

Fairness in PCA has recently been introduced by Samadi et al. (2018) who showed how to find a single d-dimensional representation for all groups present in the input data and inducing bias, for which the maximum variance is optimized in a balanced manner. It is guaranteed by maximizing the minimum variance over the groups of the projection to a d dimensional subspace. Olfat and Aswani (2019) published another version of fair PCA to reduce the projected data's dependence on a sensitive attribute through a Semi-Definite Programming. Here, we developed a fair algorithm, called fair conditional PCA algorithm, that directly uses the set of principal components to impose fairness.

A global overview of this algorithm we implemented to select principal components independent from the sensitive variable is presented in Algorithm 6. Fair conditional PCA algorithm relies on a standard PCA technique where a set of p principal components $T^1, \ldots, T^p$ is created. We choose to quantify fairness using a threshold μ applied on the correlation ratio $\eta^2_{T^h/S}$ (Fisher, 1925) between each component T^h and the sensitive variable S, which enables to characterize the association between a quantitative and a qualitative variable. In this study, the sensitive variable can only take $K = 2$ values. Hence, the empirical correlation ratio is defined as:

$$\eta^2_{T^h/S} = \frac{\sum_{k=1}^{K} n_k (\overline{T}^h_k)^2}{\sum_{i=1}^{n} (T^h_i - \overline{T}^h)^2},$$

where n_k is the number of observations in category k and $\overline{T}^h_k$ is T^h mean in category k. Then, components with a correlation coefficient C^h ($h \in \{1, \ldots, p\}$) lower than a threshold μ are pre-selected. Once 80% of the cumulative explained variance of the pre-selected components is reached, the remaining pre-selected components are removed. To select the best sample set of fair principal components that are representative of the original variabilities and good predictors of the outcome,

we calibrated the threshold coefficient using a cross-validation procedure. Obviously, PCA does not deal with prediction, but fitting a linear model between the fair components and the clinical outcome enables to quantify the predictive power of the fair components. Fair components are then identified by determining an optimal threshold as the one that minimizes the MSE (or 10% of the minimum when the MSE always decreases with the number of components) after predicting the clinical outcome. In addition to the predictibility of the final fair components, we checked their representativity of the variance by calculating the cumulative explained variance. In summary, the final insensitive principal components are selected by computing standard PCA algorithm, thresholding the correlation ratio and predicting the outcome.

Algorithm 6 Fair conditional PCA algorithm

Require: Centered matrix X of predictors, range of penalty coefficients μ, sensitive variable S.

1: Solve $\underset{\|w\|=1}{\arg\max}\ f_{PCA}(w)$,
2: Extract principal components $T^h = Xw_h$, $h \in \{1, \ldots, p\}$,
3: Compute the correlation ratio $C^h = \eta^2_{T^h/S}$, $h \in \{1, \ldots, p\}$,
4: Run a cross-validation procedure for calibrating the threshold μ,
5: Return $\hat{\mu}$ the optimal penalty coefficient,
6: Select the fair components $P_h, h \in \{1, \ldots, k\}$,
7: **Output:** $(P_h)_{1 \leq h \leq p}$ the fair principal components.

3.3 Fair PLS based on correlation measure

In recent years, fair regression has been extensively studied. It deals with the prediction of a target Y while guaranteeing the notion of fairness with respect to a protected variable S. Several research studies aim at finding predictors that exhibit some form of independence from the protected variable. Kamishima et al. (2012) seeks to fit a probabilistic model that satisfied statistical independence while Pérez-Suay et al. (2017) uses the notion of correlation in a RKHS to capture statistical independence. Finally, gouic et al. (2020) seeks a fair estimator using optimal transport. Yet, there have been no or really few research on the notion of fair Partial Least Square.

As Partial Least Square technique is somewhat similar to Principal Component Analysis, we developed an extension of the Fair conditional PCA algorithm (see Algorithm 7). This new algorithm, called fair conditional PLS algorithm relies on standard PLS technique that creates latent variables $T^1, \ldots, T^p$ which variance and correlation with the clinical outcome is maximal. Fairness is then imposed by thresholding the correlation ratio between each latent variable and the sensitive variable. Then, variables with a correlation coefficient C^h ($h \in \{1, \ldots, p\}$) lower than a predefined threshold μ and with the higher explained variance are chosen. To select the optimal number of latent variables independent from the sensitive variable i.e. to calibrate the optimal threshold coefficient, we used a cross-validation procedure.

Algorithm 7 Fair conditional PLS algorithm

Require: Centered matrix X of predictors, outcome Y, range of penalty coefficients μ, sensitive variable S.

1: Solve $\underset{\|w\|=1}{\arg\max}\ f_{PLS}(w)$,
2: Extract latent variables $T^h = Xw_h$, $h \in \{1, \ldots, p\}$,
3: Compute the correlation ratio $C^h = \eta^2_{T^h/S}$, $h \in \{1, \ldots, p\}$,
4: Run a cross-validation procedure for calibrating the threshold μ,
5: Return $\hat{\mu}$ the optimal penalty coefficient,
6: Select the fair components $P_h, h \in \{1, \ldots, k\}$,
7: **Output:** $(P_h)_{1 \leq h \leq p}$ the fair components.

3.4 Application to real data

3.4.1 Florinash dataset

The first medical dataset we used belongs to the project FLORINASH, which proposes an innovative research concept to address the role of intestinal microfloral activity in Non-Alcoholic Fatty Liver Disease (NAFLD). Hepatic steatosis, often observed in obese and diabetic patients, is a preliminary stage to NAFLD. The studied cohort (Hoyles et al., 2018) is made of obese patients featured with hepatic steatosis, a disease that results from an accumulation of genetic disorders. In this context, our objective is to find genes related to one indicator of the insuline-resistance, called clamp, that will help us to better understand the development of the disease.

The florinash dataset includes a 57×17827 gene expression matrix from the liver of 57 patients who come from Spain and Italy, of which 13370 genes are retained for this analysis after preprocessing and filtering. As this dataset gathers patients with two different profiles (genetically and phenotypically), it introduces a bias in the analysis.

Let X be the gene expression matrix, Y the clamp variable and S the following variable:

$$\forall i \in \{1, \ldots, n\}\ ,\ S_i = \begin{cases} 0 \text{ if patient } i \text{ comes from Italy} \\ 1 \text{ if patient } i \text{ comes from Spain.} \end{cases}$$

To overcome the bias issue, we applied algorithms 6 and 7 to identify, from the genomic dataset, country-insensitive components linked with the insuline-resistance variable.

3.4.2 Results

This section is dedicated to experimental studies conducted on florinash dataset to assess numerical performances of fair conditional PCA/PLS algorithms. We show how to select the final fair components, by finding a reasonable trade-off between fairness and high explained variance through a cross-validation procedure to select the optimal fair threshold μ among a large range of values. The choice of μ is of great importance because taking μ too large will lead to the selection of almost the total number of components.

Results of fair PCA based on correlation measure

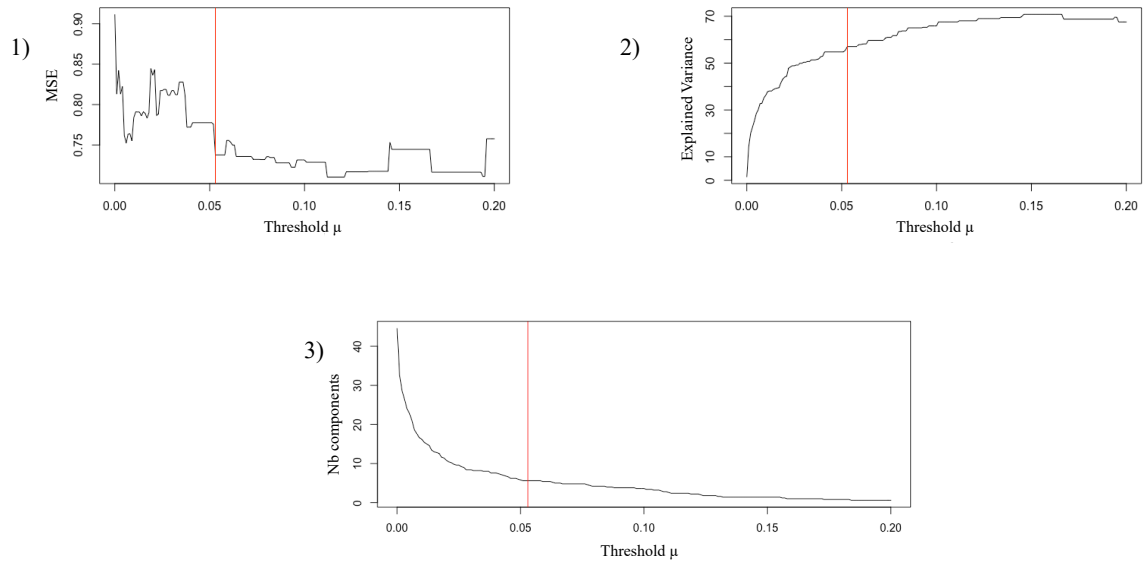

Figure 5.2 – 1) Mean Squared Error of the fair predictions, 2) Total explained variance of the fair principal components and 3) Number of components removed after thresholding, depending on the different threshold values.

The results of fair PCA experiments can be visualized in Figure 5.2 which represents the Mean Squared Error, the number of components removed after thresholding and the explained variance according to the threshold values. These figures reflect the following tendency: a high threshold leads to a better prediction of the outcome and a high explained variance. With the trade-off introduced in the previous subsections, we get an optimal threshold of $\mu = 0.05$, marked with a vertical line, which enables to select twenty principal components explaining 50% of the variance of the original data and predicting the outcome with a MSE equals to 0.75.

To assess the independence between the selected components and the sensitive variable, we can compare the plots of Figure 5.3, representing the marginal distributions conditioned to the country variable of the first four unfair and fair principal components selected. Note a separation between the marginals of the unfair components whereas this effect is attenuated or nonexistant for the fair components. We can observe the same tendency on the individuals factor maps (see Figure 5.4) where the ellipses, representative of the patient nationality, overlap on each other when the individuals are projected on the fair principal components. Obviously, these results are directly dependent on the value of the optimal threshold, which can hardly be decreased as it would lead to bad predictions of the outcome.

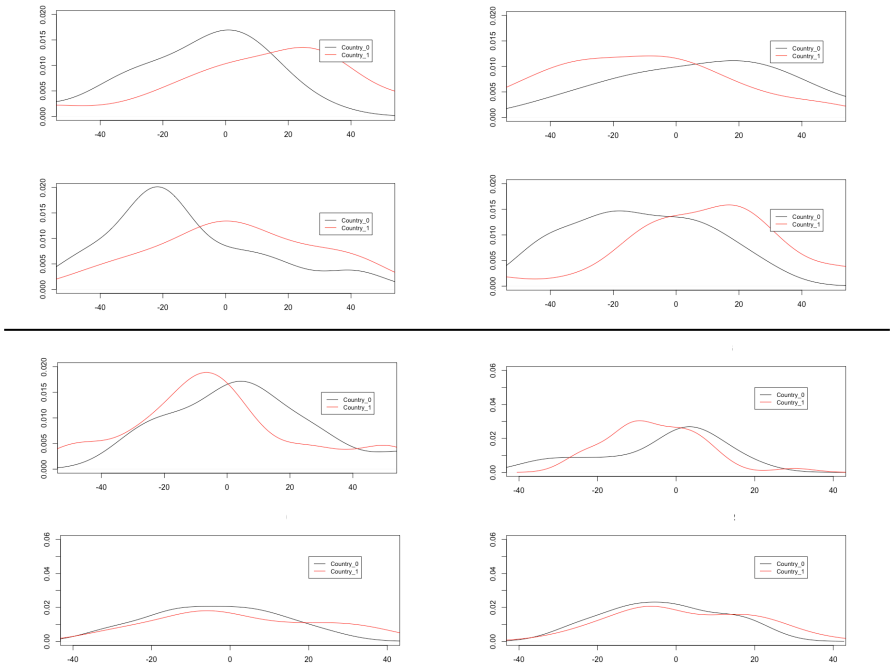

Figure 5.3 – Conditional distributions of first unfair and fair principal components with respect to the country

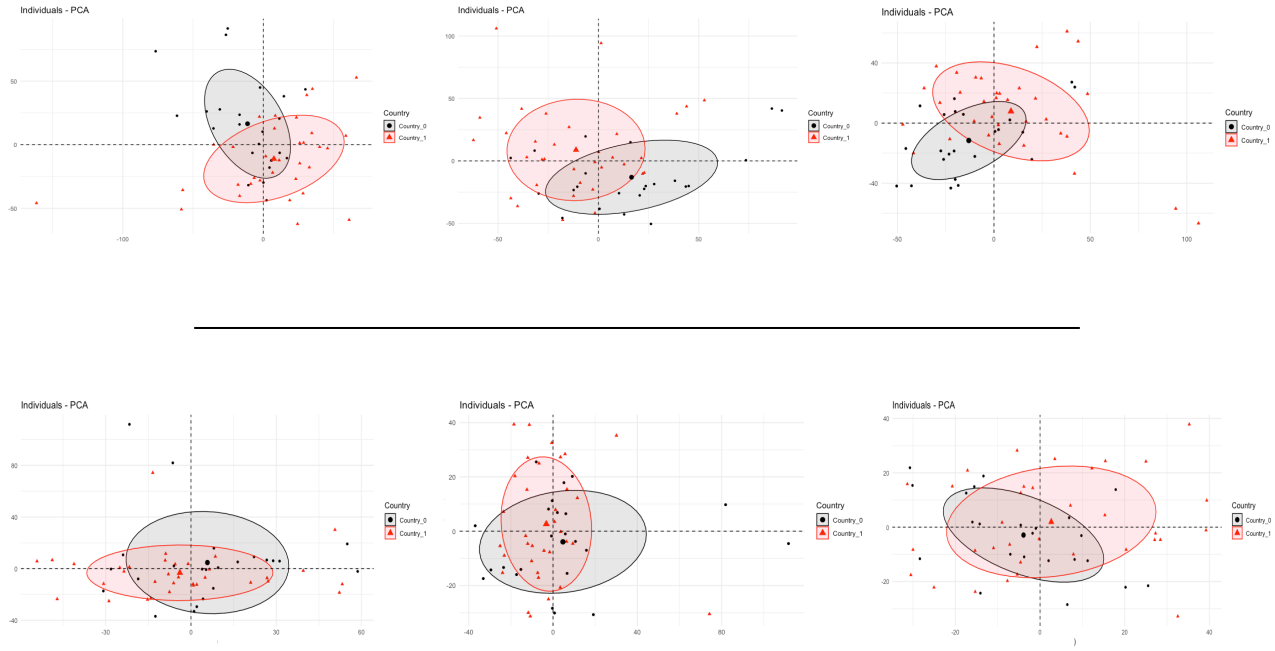

Figure 5.4 – Individuals factor map for the first unfair and fair principal components

Results of fair PLS based on correlation measure

The results of the fair PLS experiments can be visualized in Figure 5.5 representing the Mean Squared Error, the number of components removed after thresholding and the associated explained variance according to the threshold values. We get almost the same graphical trends than in fair PCA. The optimal threshold, marked in a red line, minimizes the MSE value with about 30 fair latent variables, explaining 50% of the variance of the original data and predicting the outcome with a MSE around 0.80. Note that standard PLS prediction accuracy on this dataset is already relatively low. We can deduce that it does not seem to be suitable for prediction.

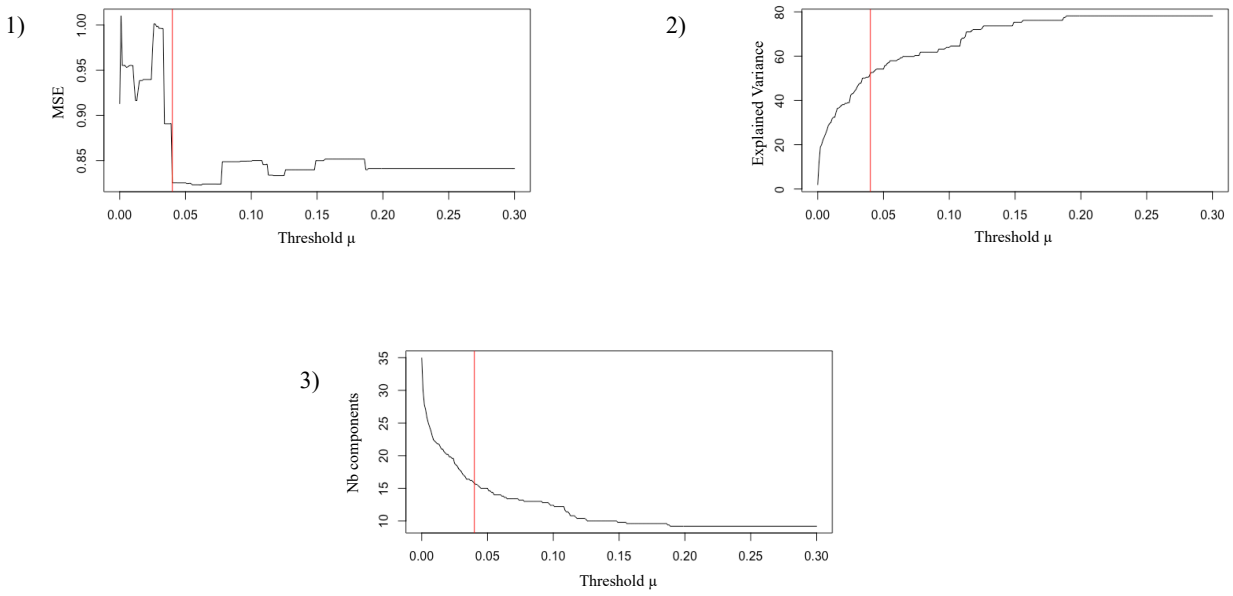

Figure 5.5 – 1) Mean Squared Error of the fair predictions, 2) Total explained variance of the fair latent variables and 3) Number of variables removed after thresholding, depending on the different threshold values.

3.4.3 Liver fibrosis dataset

Liver fibrosis is closely related to diabetes, obesity and cardiovascular diseases. It occurs when healthy tissues of the liver become too scarred to work well. Recent studies proved that gut microbiota can help understanding the development of such metabolic diseases. In this context, through a statistical analysis, our objective is to understand the pathological impact of gut microbiota on the development of liver fibrosis.

Fibrosis dataset includes a 82×411 microbial data count matrix from the liver of 82 patients of which 411 bacteria are retained for this analysis after preprocessing and filtering. This dataset stems

from the fusion of two sets of analysis. The first set is derived from a Romanian cohort whereas the second one collected clinical and microbial information on Italian, Spanish and Austrian patients. From a primary analysis, we realized that the fusion of both cohort thereby introduced a bias in the study.

Let X be the microbial data count matrix, Y and S the following variables:

$$\forall i \in \{1, \ldots, n\}\,,\; Y_i = \begin{cases} 0 \text{ if patient } i \text{ is in the early onset of liver fibrosis} \\ 1 \text{ if patient } i \text{ is in a more advanced stage of liver fibrosis} \end{cases}$$

$$\forall i \in \{1, \ldots, n\}\,,\; S_i = \begin{cases} 0 \text{ if patient } i \text{ comes from Romania} \\ 1 \text{ otherwise.} \end{cases}$$

To overcome this bias, we apply algorithms 6 and 7 to identify, from the microbial dataset, country-insensitive principal components that are linearly linked with the binary fibrosis variable.

3.4.4 Results

This section is dedicated to experimental studies conducted on liver fibrosis microbial dataset to assess numerical performances of fair conditional PCA/PLS algorithms where the clinical outcome to predict is a binary variable.

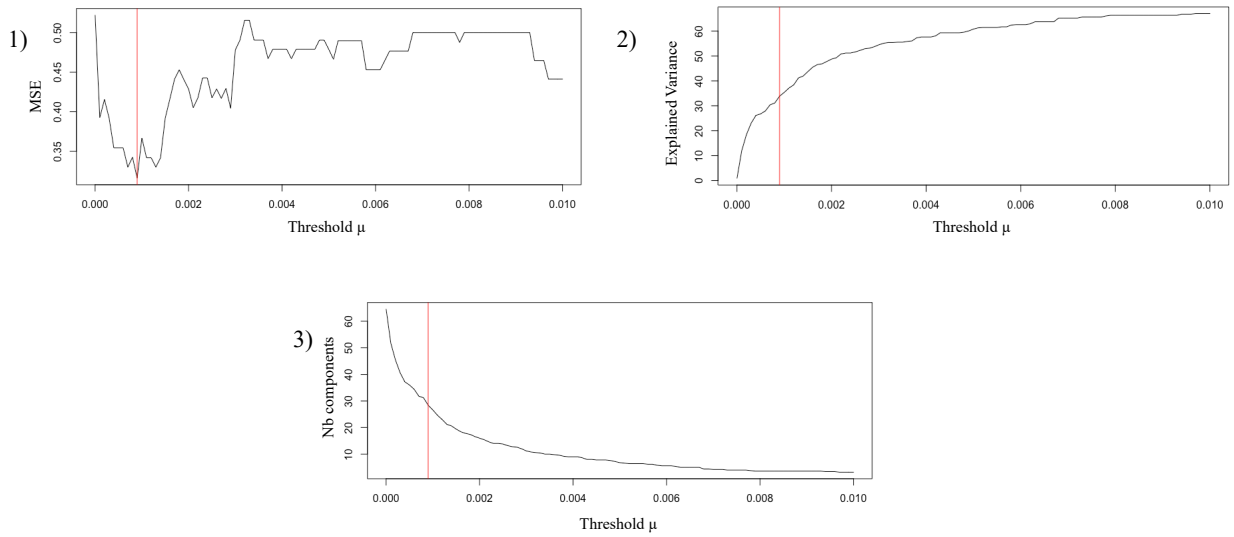

Figure 5.6 – 1) Mean Squared Error of the fair predictions, 2) Total explained variance of the fair principal components and 3) Number of components removed after thresholding, depending on the different threshold values.

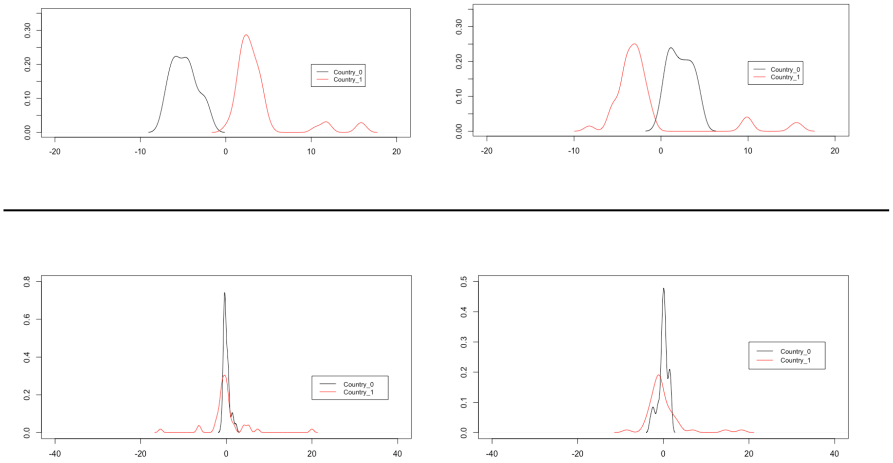

Figure 5.7 – Conditional distributions of the first two unfair and fair principal components with respect to the country

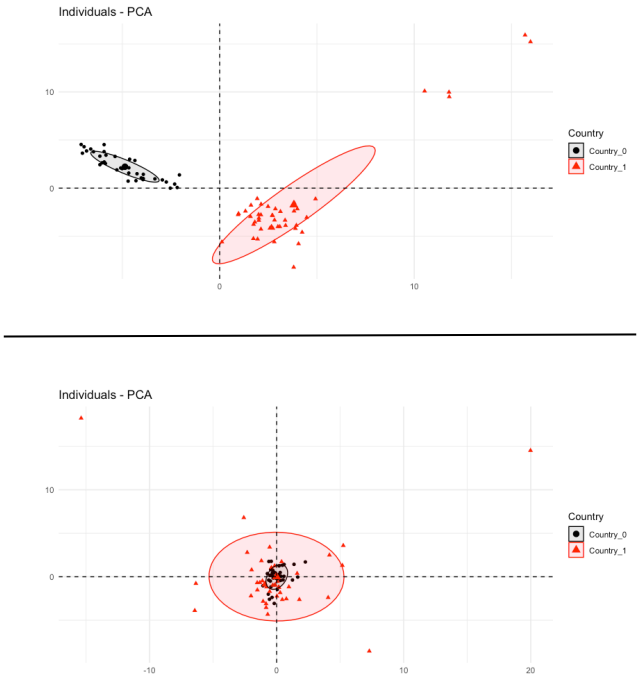

Figure 5.8 – Individuals factor map for the first two unfair and fair principal components

Results of fair PCA based on correlation measure

The results of the fair PCA experiments can be visualized in Figure 5.6 representing the Mean Squared Error, the number of components removed after thresholding and the associated explained variance according to the threshold values. These figures reflect that the MSE decreases until a certain threshold is reached. Hence, the optimal threshold that guarantees good predictions of the outcome and good representativity of the variabilities is the one that minimizes the MSE, represented with a red line. About 15 fair principal components explaining 30% of the variance of the original data and predicting fibrosis with a MSE around 0.3 were selected. Then, to assess the independence between the selected components and the sensitive variable, we compared the plots of Figure 5.7 representing the marginal distributions conditioned to the country variable of the first two unfair and fair selected principal components. Note a separation between the marginals of the unfair components whereas this effect is attenuated or nonexistant for the fair components. We can observe the same tendency on the individuals factor maps (see Figure 5.8) where the ellipses, representative of the patient nationality, overlap on each other when the individuals are projected on the fair principal components. Consequently, the selected principal components are effectively independent from the countries.

Results of fair PLS based on correlation measure

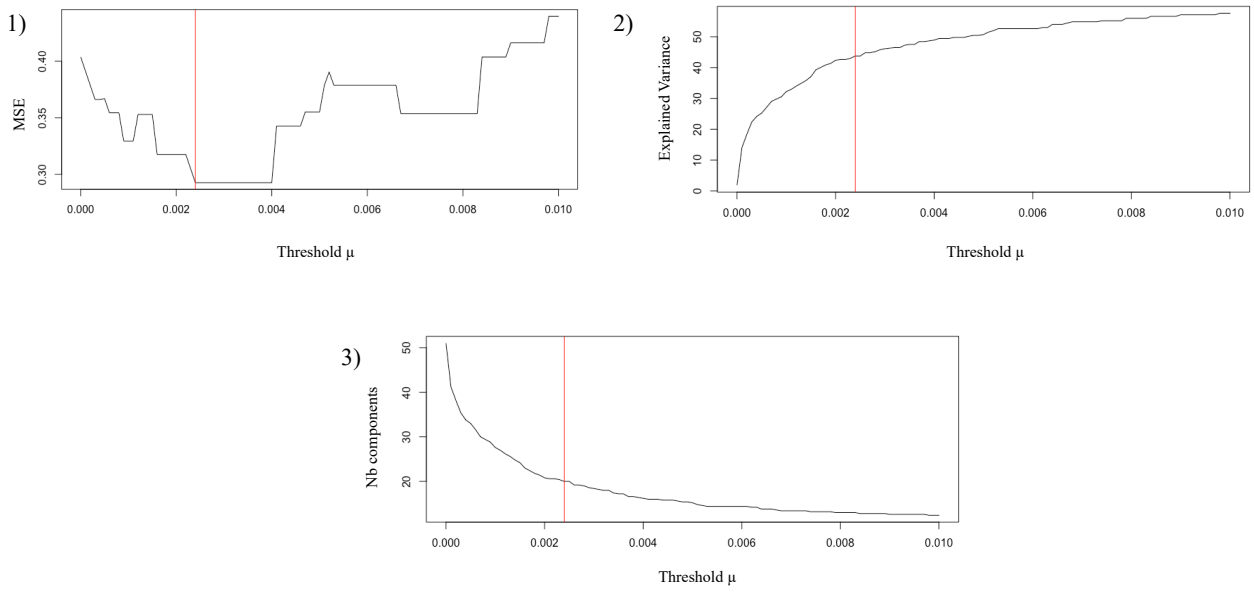

Figure 5.9 – 1) Mean Squared Error of the fair predictions, 2) Total explained variance of the fair latent variables and 3) Number of variables removed after thresholding, depending on the different threshold values.

The results of the fair PLS experiments can be visualized in Figure 5.9 representing the Mean Squared Error, the number of components removed after thresholding and the associated explained variance according to the threshold. We get almost the same graphical trends than in fair PCA. The minimal MSE value is reached with a threshold in the range of 0.025 where about 30 fair latent variables are selected, explaining around 40% of the variance of the original data and predicting the outcome with a MSE lower than 0.30. Note that this fair PLS strategy seems to be better suited to this dataset than the PCA one as the selected country insensitive components represent almost half of the dataset variability and guarantee a low MSE. However, when comparing unfair and fair PCA/PLS procedures, we can observe a loss of accuracy, in terms of MSE, in the order of 0.1.

4 Conclusion

In this chapter, we examined two specific biomedical datasets which first exploratory analysis reflected a bias. To remove the underlying adverse effects, we proposed two fair learning approaches based on standard dimension reduction techniques (PCA and PLS) that enable to select components that restore the variability of the original data while limiting the bias effect. Fairness is imposed as a threshold to control the independence between the PCA and PLS components and the sensitive variable causing bias. In order to select the optimal number of unbiased components, we evaluated the performances of each algorithm in terms of Mean Squared Error. In fact, in such biomedical context, we aim to identify relevant variables or groups of interacted variables involved in the development of a disease. Through numerical experiments and comparisons, we show the effectiveness of the fair PLS strategy based on correlation measure and identify a set of latent variables related to the indicator of the disease but decorrelated with the sensitive variable.

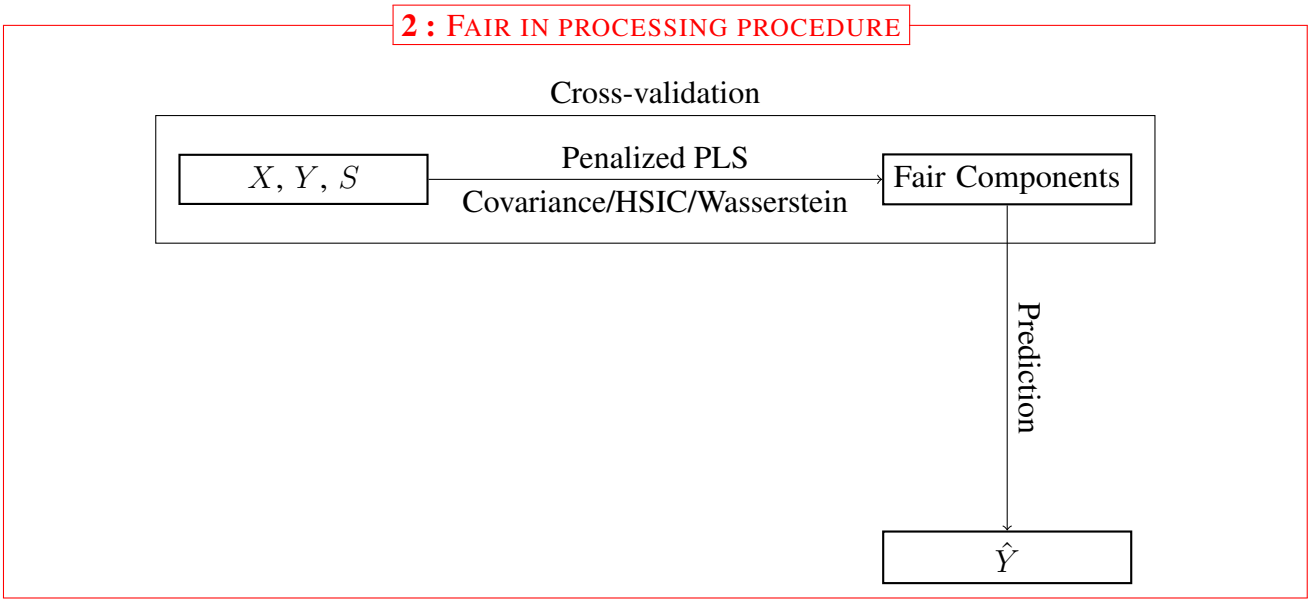

Figure 5.10 – Fair in processing procedure

From a methodological perspective, an interesting research direction would be to impose fairness as an in-processing step of the standard PLS technique. Our goal would be to directly create fair components by penalizing PLS optimization problem using independence criteria between the conditional distributions such as covariance, Hilbert-Schmidt Independence Criterion (HSIC) (Mooij et al., 2009) or Wasserstein-2 distance (Villani, 2009) (see Figure 5.10). The primary developments of these strategies are detailed hereafter (Section 5).

5 Further developments

In this section, we provide in-processing fair approaches based on standard PLS method.

5.1 PLS under fair constraint based on covariance measure

The first fair approach aims to directly create fair components by penalizing PLS objective function (Equation (5.2)) using a covariance measure. It is somehow an extension of Zafar et al. (2019) which also introduced a fair covariance measure in a constraint-based framework but incorporated into a classifier formulation.

Let S be the sensitive variable and T the latent variable scores matrix output of PLS algorithm. Then, the covariance between T and S can be written as:

$$\begin{aligned}\text{Cov}(T, S) &= \mathbb{E}[TS] - \mathbb{E}[T]\mathbb{E}[S] \\ &= p\,\mathbb{E}[T|S=1] - p\,\mathbb{E}[T],\end{aligned}$$

where $\mathbb{E}[S] = p$ and

$$\begin{aligned}\mathbb{E}[T] &= \mathbb{E}[\mathbb{E}[T|S]] \\ &= p\,\mathbb{E}[T|S=1] + (1-p)\,\mathbb{E}[T|S=0].\end{aligned}$$

Hence,

$$\begin{aligned}\text{Cov}(T, S) &= p\,\mathbb{E}[T|S=1] - p^2\,\mathbb{E}[T|S=1] - p(1-p)\,\mathbb{E}[T|S=0] \\ &= p\,(\mathbb{E}[T|S=1](1-p) - (1-p)\,\mathbb{E}[T|S=0]) \\ &= p(1-p)\,(\mathbb{E}[T|S=1] - \mathbb{E}[T|S=0]).\end{aligned}$$

We recall that we aim to find an optimal subspace where the latent variables are independent from the sensitive feature S, i.e. we are looking for T such that $\mathbb{E}[T|S=1] - \mathbb{E}[T|S=0]$ is very small, and such that each T is a good predictor of the outcome. Hence, we must find a trade-off between

fairness, i.e. low covariance coefficients and good predictions. We are therefore looking for principal factors $(w_1, \ldots, w_p)$ defined as:

$$\forall h \in \{1, \ldots, p\} \;\; w_h = \underset{\|w\|=1}{\arg\max}, \; \{f_{\mathrm{PLS}}(w) - \; \lambda \, \mathrm{Cov}(Xw, S)\}, \tag{5.3}$$

where λ is a positive penalty parameter.

5.2 PLS under fair constraint based on Hilbert-Schmidt Independance Criterion

In this subsection, we introduce another fair approach based on Hilbert Schmidt independence criterion. Suppose $(\mathcal{X}, \mathcal{Y})$ are drawn from distribution $\mathbb{P}_{\mathcal{X},\mathcal{Y}}$, where $\mathcal{X} \in \mathbb{R}^p$ is a p-dimensional explanatory variable and $\mathcal{Y} \in \mathbb{R}$ is a response variable. Denote by $\forall i \in \{1, \ldots, n\}, \; X_i := (X_i^1, \ldots, X_i^p)$ a p-dimensional vector of variables.

Hilbert Schmidt Independence Criterion

Hilbert Schmidt Independence Criterion (HSIC) measures allow to simultaneously take into account many forms of dependence between two random variables $\mathcal{X}$ and $\mathcal{Y}$ relying on Reproducing Kernel Hilbert Spaces (RKHS). The reader can refer to Aronszajn (1950) for a complete bibliography on RKHS spaces.

Definition 7 *Let $\mathcal{X}$ an arbitrary set and $\mathcal{H}$ be a Hilbert space of real-valued functions on $\mathcal{X}$ with a scalar product $<,>_{\mathcal{H}}$. The Hilbert space $\mathcal{H}$ is said to be a RKHS if, for all X_i in $\mathcal{X}$ the application $h \mapsto h(X_i) \forall h \in \mathcal{H}$ is a continuous linear form.*

The particularity and interest of RKHS spaces is the Riesz representation theorem. This representation consists in associating to each value of the starting set, a function in the RKHS. Each element is then represented by a random functional variable belonging to a space having good properties.

Proposition 7 *Let $\mathcal{H}$ be a RKHS space associated to a set $\mathcal{X}$ and with a scalar product denoted by $<,>_{\mathcal{H}}$. Then, for all X_i in $\mathcal{X}$ there is a unique ϕ in $\mathcal{H}$ such that $h(X_i) =< h, \phi >_{\mathcal{H}}$, for all $h \in \mathcal{H}$.*

The HSIC characteristics depend entirely on the choice of this RKHS. This choice consists in associating a mapping function

$$\begin{aligned} \phi : \mathcal{X} &\longrightarrow \mathcal{H} \\ X_i &\longmapsto \phi(X_i). \end{aligned}$$

that assigns to each element of the domain a representative functional ϕ in the RKHS and a scalar product that defines the nature of the relationships between the representatives and so between the elements of the domain. The application that defines this scalar product is called kernel and is defined as follows:

$$\begin{aligned} K : \mathcal{X} \times \mathcal{X} &\longrightarrow \mathcal{X} \\ X_i, X_j &\longmapsto K(X_i, X_j). \end{aligned}$$

satisfying for all $X_i, X_j \in \mathcal{X}$,

$$K(X_i, X_j) =< \phi(X_i), \phi(X_j) >_{\mathcal{H}} .$$

This independence criterion, proposed by (Gretton et al., 2005), is measured by computing the Hilbert-Schmidt norm of the cross covariance operator associated with their RKHSs. In order to make HSIC a practical criterion for testing independence, it can be approximited given finite number of observations as:

$$\text{HSIC}(\mathbb{P}_{(\mathcal{X},\mathcal{Y})}, \mathcal{H}, \mathcal{G}) = \frac{1}{(n-1)^2}\text{Trace}(KHLH),$$

where K and L are the kernels of X and Y and H a centering matrix verifying $(I_n - \frac{1}{n}1_{n,n})$ where $1_{n,n}$ is a matrix which all coefficients are equal to 1 and I_n the identity matrix of size n. Based on this result, in order to maximize the dependence between two kernels we need to increase the value of this empirical estimate.

Kernel based PLS algorithm

In many situations, non linear transformations are required to successfully apply learning algorithms. Consider a non linear transformation of $X_i \in \mathcal{X} \subset \mathbb{R}^p$ into a feature space $\mathcal{H}$. Using the straightfoward link between a RKHS and $\mathcal{H}$, PLS has been extended into a non linear kernel form (Rosipal and Trejo, 2001). This extension implies the construction of a PLS model in $\mathcal{H}$.

Denote Φ as the $(n \times M)$ matrix of mapped $\mathcal{X}$-space data $\phi(X_i)$ into an M-dimensional feature space $\mathcal{H}$. Instead of an explicit mapping of the data, we can directly use the following matrix using the kernel trick Schölkopf et al. (1998):

$$K = \Phi\Phi^T,$$

where K represents the $(n \times n)$ kernel Gram matrix of the cross dot products between all input data points $\{\phi(X_i)\}_{1 \leq i \leq n}$: $K_{ij} = K(X_i, X_j)$, where $K(.,.)$ is a selected kernel function.

Motivated by the theory of RKHS, Rosipal and Trejo (2001) derived the PLS algorithm for the (nonlinear) kernel PLS model whose purpose is to find principal factors $(w_1, \ldots, w_p)$, solutions of the following optimization problems:

$$\forall h \in \{1, \ldots, p\}\,,\ w_h = \underset{\|w\|=1}{\arg\max}\, \text{Cov}(\phi(X)w, Y), \tag{5.4}$$

PLS under fair constraint based on Hilbert-Schmidt Independance Criterion

Several authors have considered the framework of penalized regression either to achieve fairness or to provide guarantees for independence (Pérez-Suay et al., 2017). In this subsection, we describe how HSIC can be used in Partial Least Square technique as an independence criterion between the latent variables and the sensitive variable S.

Let T_h be the latent variables created by standard PLS method and defined as $T^h = Xw_h,\ \forall h \in \{1, \ldots, p\}$. Fair PLS adresses the problem of finding the subspace Xw such that the dependence between the projected data Xw and the outcome is maximized but the dependence between the projected data and the outcome is minimized. In order to measure this independence, we use the Hilbert-Schmidt Independence Criterion.

Hence, we need to minimize HSIC$(\mathbb{P}_{(Xw,S)}, \mathcal{H}_K, \mathcal{H}_L) = \frac{1}{(n-1)^2} tr(KHLH)$, where K and L are the kernels associated to Xw and S. Let us consider linear kernels i.e. $\forall x, x' \in \mathcal{X},\ K(x, x') = x^T x'$. Thus, we need to find the following principal factors $(w_1, \ldots, w_p)$:

$$\forall h \in \{1, \ldots, p\},\ w_h = \underset{\|w\|=1}{\arg\min}\ \text{tr}(w^T X^T HLHXw).$$

The optimal solutions to this optimization problem is $U := [v_1, \ldots, v_p]$ where p denotes the final number of latent variables created, $v_1, \ldots, v_p$ are the eigenvectors associated to the k smallest eigenvalues $\lambda_1, \ldots, \lambda_p$ obtained by eigen decomposing the symmetric and real matrix $Q = X^T HLHX$.

This measure is added as a regularization term to the equation (5.2). Hence, we look for principal factors $w_1, \ldots, w_p$ defined as:

$$\forall h \in \{1, \ldots, p\},\ w_h = \underset{\|w\|=1}{\arg\max}\ \text{Cov}(Xw, Y)\ -\ \mu\, \text{tr}(w^T X^T HLHXw), \tag{5.5}$$

where $\mu > 0$ is the regularization parameter.

5.3 PLS through Wasserstein-2 distance

In this subsection, we introduce a third in-processing fair approach based on Wasserstein-2 distance. We sought to impose independence between PLS features with respect to the protected variable S i.e. that either the distribution of the features, or its conditional distribution does not depend on S. Hence, fairness quantification can be implemented using distance between conditional distributions. The Wasserstein distance appears as an appropriate tool for comparing probability distributions. We refer to Villani (2009) and references there in for more details on Wasserstein's distance.

Our last regularization strategy consists in ensuring that the Wasserstein-2 distance between the conditional distributions $\mu_{Xw|S=0}$ and $\mu_{Xw|S=1}$ remain close to each other. Hence, we need to find principal factors $(w_1, \ldots, w_p)$ defined as:

$$\forall h \in \{1, \ldots, p\},\ w_h = \underset{\|w\|=1}{\arg\max}\ \text{Cov}(Xw, Y)\ -\ \mu\, W_2^2(\mu_{Xw|S=0}, \mu_{Xw|S=1}), \tag{5.6}$$

where $\mu > 0$ is a regularization parameter.

In Risser et al. (2019), such a regularization term is approximated by calculating the gradient of the Wasserstein distance between the distributions of the two groups.

Chapter 6

Conclusion

1 Synthesis

This thesis, raised from the MATHOBIOMIX project, is devoted to the study and the development of statistical methods and cluster-based algorithms of specific biomedical datasets. In collaboration with the Institute for the Study of Cardiovascular and Metabolic Diseases (I2MC), this project fits specifically within the framework of liver based diseases which we strive to understand the biological foundations using genomic and metagenomic databases.

Chapter II and Chapter III result in the development of graph clustering algorithms dedicated to the detection of densely connected variables in complex and respectively noisy networks, charcateristic of real biological systems. The associated scripts give rise to the development of two R packages.

In Chapter IV, as part of a medical research project, we conducted a statistical study on metagenomic datasets characterizing patient in their early development of liver fibrosis disease, which confirms the need for specific tools due to the nature of the features under consideration. An adapted strategy combining a variety of exploratory, clustering and predictive methods are thus proposed to reveal the main biological interactions implied in the development of the disease.

While the datasets, submitted to statistical studies, highlighted undesirable effects on standard methods, we developed, in Chapter V, fair learning approaches to limit the bias. This study could be extended in the future in order to incorporate differently the limiting factor and eventually, to find the best model improving standard learning procedures under bias.

2 Scientific production

2.1 Published papers

— C. Champion, A.C. Brunet, R. Burcelin, J.M. Loubes, L. Risser (2021). Detection of Representative Variables in Complex Systems with Interpretable Rules Using Core-Clusters. **Algorithms**, 14(2), 66,

— M. Minty, T. Canceill, S. Lê, P. Dubois, O. Amestoy, P. Loubieres, J. Christensen, C. Champion, V. Azalbert, E. Grasset, S. Hardy, J.M. Loubes, J.P. Mallet, F. Tercé, J.N. Vergnes, R. Burcelin, M. Serino, F. Diemer, V. Blasco-Baque (2018). Oral health and microbiota in professional rugby players a case-control study. **Journal of Dentistry**, 79:53–60.

2.2 Submitted papers

— C. Champion, M. Champion, M. Blazère, R. Burcelin, J.M. Loubes (2021). ℓ_1-spectral clustering algorithm: a robust spectral clustering using Lasso regularization. (hal-03095805).
— C. Champion, R.M. Neagoe, M. Effernberger, D. T. Sala, F. Servant, J.E. Christensen, M. Arnoriaga-Rodriguez, J. Amar, B. Lelouvier, F. Gamboa, H. Tilg, M. Federici, J.M. Fernandez-Real, J.M. Loubes and R. Burcelin (2021). Human liver microbiota modeling strategy at the early onset of fibrosis.
— C. Thomas, M. Minty, T. Canceill, P. Loubieres, V. Azalbert, F. Terce, C. Champion, R. Burcelin, P. Barthet, S. Laurencin-Dalicieux, V. Blasco-Baque (2021). Obesity drives an oral microbiota signature of female patients with periodontitis: a pilot study.

2.3 R packages

— R package CoreClustering, Available on sourceforge [1],
— R package l1spectral, Available on CRAN and GitHub [2].

1. `https://es.sourceforge.net/projects/core-clustering/`
2. `https://github.com/championcamille/l1-SpectralClustering`

Bibliography

Abbe, E. (2017). Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531.

Agarwal, P., Har-Peled, S., and Varadarajan, K. (2005). Geometric approximation via coresets. *Combinatorial and Computational Geometry, MSRI. University Press*, pages 1–30.

Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422.

Amar, J., Chabo, C., Waget, A., Klopp, P., Vachoux, C., Bermudez-Humaran, L., Smirnova, N., Berge, M., Sulpice, T., Lahtinen, S., Ouwehand, A., Langella, P., Rautonen, N., Sansonetti, P., and Burcelin, R. (2011a). Intestinal mucosal adherence and translocation of commensal bacteria at the early onset of type 2 diabetes: molecular mechanisms and probiotic treatment. *EMBO Mol Med*, 3(9):559–572.

Amar, J., Serino, M., Lange, C., Chabo, C., Iacovoni, J., Mondot, S., Lepage, P., Klopp, C., Mariette, J., Bouchez, O., Perez, L., Courtney, M., Marre, M., Klopp, P., Lantieri, O., Dore, J., Charles, M., Balkau, B., and Burcelin, R. (2011b). Involvement of tissue bacteria in the onset of diabetes in humans: evidence for a concept. *Diabetologia*, 54(12):3055–3061.

Aronszajn, N. (1950). Theory of reproducing kernels. *transactions of the American mathematical society*, 68:337–404.

Bachem, O., Lucic, M., and Lattanzi, S. (2018). One-shot coresets:the case of k-clustering. In *Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, pages 784–792.

Baharan, M., Kaidi, C., and Jure, L. (2020). Coresets for robust training of deep neural networks against noisy labels. In *Neural Information Processing Systems (NeurIPS 2020)*.

Balmer, M., Slack, E., de Gottardi, A., Lawson, M., Hapfelmeier, S., Miele, L., Grieco, A., Van Vlierberghe, H., Fahrner, R., Patuto, N., Bernsmeier, C., Ronchi, F., Wyss, M., Stroka, D., Dickgreber, N., Heim, M., McCoy, K., and Macpherson, A. (2014). The liver may act as a firewall mediating mutualism between the host and its gut commensal microbiota. *Sci Transl Med*, 6(237):237ra266.

Barker, M. and Rayens, W. (2003). Partial least squares for discrimination. *Journal of Chemometrics*, 17:166–173.

Batagelj, V. and Zaversnik, M. (2011). Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification*, 5:129–145.

Baykal, C., Liebenwein, L., Gilitschenski, I., Feldman, D., and Rus, D. (2019). Data-dependent coresets for compressing neural networks with applications to generalization bounds. In *International Conference on Learning Representations*.

Bellemo, V., Lim, G., Rim, T., Tan, G., Cheung, C., Sadda, S., He, M., Tufail, A., Lee, M., Hsu, W., and D.Ting (2019). Artificial intelligence screening for diabetic retinopathy: the real-world emerging application. *Current Diabetes Reports*, 19:9–72.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 57(1):289–300.

Berg, R., Wommack, E., and Deitch, E. (1988). Immunosuppression and intestinal bacterial overgrowth synergistically promote bacterial translocation. *Arch Surg*, 123(11):1359–1364.

Bieghs, V. and Trautwein, C. (2014). Innate immune signaling and gut-liver interactions in non-alcoholic fatty liver disease. *Hepatobiliary Surg Nutr*, 3(6):377–385.

Bishop, C. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Blazère, M., Gamboa, F., and Loubes, J. (2014). Pls: a new statistical insight through the prism of orthogonal polynomials. *arXiv preprint arXiv:1405.5900*.

Blease, C., Kaptchuk, T., Bernstein, M., Mandl, K., Halamka, J., and DesRoches, C. (2019). Artificial intelligence and the future of primary care: Exploratory qualitative study of uk general practitioners' views. *J Med Internet Res*, 21(3).

Blondel, V., Guillaume, J., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, (10):10008.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D. (2006). Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308.

Bojchevski, A., Matkovic, Y., and Günnemann, S. (2017). Robust spectral clustering for noisy data: Modeling sparse corruptions improves latent embeddings. In *KDD17: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 737–746.

Brady, M. (1985). Artificial intelligence and robotics. *Artificial Intelligence*, 26(1):79 – 121.

Brandl, K. and Schnabl, B. (2017). Intestinal microbiota and non-alcoholic steatohepatitis. *Curr Opin Gastroenterol*, 33(3):128–133.

Bray, R. and Curtis, J. (1957). An ordination of the upland forest communities of southern wisconsin. *Ecological Monographs*, 27(4):325–349.

Brunet, A., Azais, J., Loubes, J., Amar, J., and Burcelin, R. (2016). A new gene co-expression network analysis based on core structure detection (csd). (abs/1607.01516).

Brunet, A., Loubes, J., Azais, J., and Courtney, M. (2015). Method of identification of a relationship between biological elements.

Buolamwini, J. and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91.

Burcelin, R., Serino, M., Chabo, C., Garidou, L., Pomie, C., Courtney, M., Amar, J., and Bouloumie, A. (2013). Metagenome and metabolism: the tissue microbiota hypothesis. *Diabetes Obes Metab*, 15 Suppl 3:61–70.

Cani, P., Amar, J., Iglesias, M., Poggi, M., Knauf, C., Bastelica, D., Neyrinck, A., Fava, F., Tuohy, K., Chabo, C., Waget, A., Delmee, E., Cousin, B., Sulpice, T., Chamontin, B., Ferrieres, J., Tanti, J., Gibson, G., Casteilla, L., Delzenne, N., Alessi, M., and Burcelin, R. (2007). Metabolic endotoxemia initiates obesity and insulin resistance. *Diabetes*, 56(7):1761–1772.

Cao, K. L., Boitard, S., and Besse, P. (2011). Sparse pls discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems. *BMC bioinformatics*, 1:12–253.

Challen, R., Denny, J., Pitt, M., Gompels, L., Edwards, T., and Tsaneva-Atanasova, K. (2019). Artificial intelligence, bias and clinical safety. *BMJ Quality & Safety*, 28(3):231–237.

Champion, C., Brunet, A., Burcelin, R., Loubes, J., and Risser, L. (2021a). Detection of representative variables in complex systems with interpretable rules using core-clusters. *Algorithms*, 14:66–66.

Champion, C., Brunet, A.-C., Loubes, J.-M., and Risser, L. (2018). Coreclust: a new package for a robust and scalableanalysis of complex data. (hal-01799117).

Champion, C., Champion, M., Blazère, M., and Loubes, J.-M. (2021b). $\ell 1$-spectral clustering algorithm: a robust spectralclustering using lasso regularization. (hal-03095805).

Chao, A. (1984). Nonparametric estimation of the number of classes in a population. *Scandinavian Journal of Statistics*, 11(4):265–270.

Chen, J., adn P. Castaldi, Y. C., Cho, M., Hobbs, B., and Dy, J. (2018). Crowdclustering with partitions labels. In *Proceedings of Machine Learning Research*, volume 84, pages 1127–1136.

Claici, S., Genevay, A., and Solomon, J. (2020). Wasserstein measure coresets. (arXiv:1805.07412).

Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2001). *Introduction to Algorithms*. MIT Press and McGraw-Hill.

Cox, M. and Cox, T. (2008). *Multidimensional Scaling*. Springer, Berlin, Heidelberg.

Culhane, A., Schwarz, I., Sultana, R., Picard, K., Picard, S., Lu, T., Franklin, K., French, S., Papenhausen, G., Corell, M., and Quackenbush, J. (2010). GeneSigDB, a curated database of gene expression signatures. *Nucleic Acids Research*, 38:D716–D725.

Cysouw, M. (2018). R function cor.sparse.

Davidson, E. and Levin, M. (2005). Gene regulatory networks. *Proceedings of the National Academy of Sciences*, 102(14):4935–4935.

DeCamp, M. and Lindvall, C. (2020). Latent bias and the implementation of artificial intelligence in medicine. *Journal of the American Medical Informatics Association*, 27(12):2020–2023.

Denou, E., Lolmede, K., Garidou, L., Pomie, C., Chabo, C., Lau, T., Fullerton, M., Nigro, G., Zakaroff-Girard, A., Luche, E., Garret, C., Serino, M., Amar, J., Courtney, M., Cavallari, J., Henriksbo, B., Barra, N., Duggan, K. F. J. M. B., O'Neill, H., Lee, A., Sansonetti, P., Ashkar, A., Khan, W., Surette, M., Bouloumie, A., Steinberg, G., Burcelin, R., and Schertzer, J. (2015). Defective nod2 peptidoglycan sensing promotes diet-induced inflammation, dysbiosis, and insulin resistance. *EMBO molecular medicine*, 7(3):259–274.

Dias, R. and Torkamani, A. (2019). Artificial intelligence in clinical and genomic diagnostics. *Genome Medicine*, 11(1):1–12.

DiMaira, G., Pastore, M., and Marra, F. (2018). Liver fibrosis in the context of non-alcoholic steatohepatitis: the role of adipokines. *Minerva gastroenterologica e dietologica*, 64(1):39–50.

Dirican, C. (2015). The impacts of robotics, artificial intelligence on business and economics. *Procedia - Social and Behavioral Sciences*, 195:564 – 573.

Doherty, N., Kartasheva, A., and Phillips, R. (2012). Information effect of entry into credit ratings market: The case of insurers' ratings. *Journal of Financial Economics*, 106(2):308–330.

Douglas, G., Maffei, V., Zaneveld, J., Yurgel, S., Brown, J., Taylor, C., Huttenhower, C., and Langille, M. (2020). Picrust2 for prediction of metagenome functions. *Nat Biotechnol*, 38(6):685–688.

Dunphy, P., Phillips, P., and Brodie, A. (1971). Separation and identification of menaquinones from microorganisms. *J Lipid Res*, 12(4):442–449.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM.

Easly, D. and Kleinberg, J. (2010). *Networks, Crows and Markets: Reasoning About a Highly Connected World*. Cambridge University Press.

Escudie, F., Auer, L., Bernard, M., Mariadassou, M., Cauquil, L., Vidal, K., Maman, S., Hernandez-Raquet, G., Combes, S., and Pascal, G. (2018). Frogs: Find, rapidly, otus with galaxy solution. *Bioinformatics*, 34(8):1287– 1294.

Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A density- based algorithm for discovering clusters in large spatial databases with noise. In *KDD96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231.

Esteva, A., Kuprel, B., Novoa, R., Ko, J., Swetter, S., Blau, H., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118.

Farrand, S. and Taber, H. (1973). Physiological effects of menaquinone deficiency in bacillus subtilis. *J Bacteriol*, 115(3):1035–1044.

Fisher, R. (1925). *Statistical methods for research workers*. Edinburgh Oliver & Boyd.

Gadat, S., Gavra, I., and Risser, L. (2018). How to calculate the barycenter of a weighted graph. *Informs*, 43(4).

Garidou, L., Pomie, C., Klopp, P., Waget, A., Charpentier, J., Aloulou, M., Giry, A., Serino, M., Stenman, L., Lahtinen, S., Dray, C., Iacovoni, J., Courtney, M., Collet, X., Amar, J., Servant, F., Lelouvier, B., Valet, P., Eberl, G., Fazilleau, N., Douin-Echinard, V., Heymes, C., and Burcelin, R. (2015). The gut microbiota regulates intestinal CD4 T cells expressing RORgammat and controls metabolic disease. *Cell Metab*, 22(1):100–112.

Gary, C. and Bennetts, R. (1996). Analysis of frequency count data using the negative binomial distribution. *Ecology*, 77(8):2549–2557.

Giatsidis, C., Thilikos, F. M. D., and Vazirgiannis, M. (2014). Corecluster: A degeneracy based graph clustering framework. In *Proceeding of the Twenty-Eight AAAI Conference on Artificial Intelligence*, pages 44–50, Québec City, QC, Canada.

Gillen, S., Jung, C., Kearns, M., and Roth, A. (2018). Online learning with an unknown fairness metric. *Neural Information Processing Systems (NeurIPS)*.

Girvan, M. and Newman, E. (2002). Community structure in social and biology networks. In *Proceedings of the national academy of sciences*, volume 99, pages 7821–7826.

gouic, T. L., Loubes, J., and Rigollet, P. (2020). Projection to fairness in statistical learning. *arXiv e-prints*, (arXiv:2005.11720 [cs.LG]).

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In Jain, S., Simon, H., and Tomita, E., editors, *Algorithmic Learning Theory*, pages 63–77. Springer.

Gusikhin, O., Rychtyckyj, N., and Filev, D. (2007). Intelligent systems in the automotive industry: applications and trends. *Knowledge and Information Systems*, 12(2):147 – 168.

Hagen, L. and Kahng, A. (1992). New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085.

Hamet, P. and Tremblay, J. (2017). Artificial intelligence in medicine. *Metabolism*, 69:36 – 40.

Handcock, M. and Gile, K. (2010). Modeling social networks form sampled data. *The Annals of Applied Statistics*, 4(1):5–25.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New Yok Inc.

Hendrickson, B. and Leland, R. (1995). An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM Journal on Scientific Computing*, 16:452–469.

Hess, S., Duivesteijn, W., Honysz, P., and Morik, K. (2019). The SpectACl of non convex clustering: a spectral approach to density-based clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3788–3795.

Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L., and Aerts, H. (2018). Artificial intelligence in radiology. *Nature Reviews Cancer*, 18:500–510.

Hoyles, L., Fernandez-Real, J., Federici, M., Serino, M., Abbott, J., Charpentier, J., Heymes, C., Luque, J., Anthony, E., Barton, R., Chilloux, J., Myridakis, A., Martinez-Gili, L., Moreno-Navarrete, J., Benhamed, F., Azalbert, V., Blasco-Baque, V., Puig, J., Xifra, G., Ricart, W., Tomlinson, C., Woodbridge, M., Cardellini, M., Davato, F., Cardolini, I., Porzio, O., Gentileschi, P., Lopez, F., Foufelle, F., Butcher, S., Holmes, E., Nicholson, J., Postic, C., Burcelin, R., and Dumas, M. (2018). Molecular phenomics and metagenomics of hepatic steatosis in non-diabetic obese women. *Nat Med*, 24(7):1070–1080.

Hu, T. (1961). The maximum capacity route problem. *Operations research*, 9(6):898–900.

Iwata-Reuyl, D. (2003). Biosynthesis of the 7-deazaguanosine hypermodified nucleosides of transfer rna. *Bioorg Chem*, 31(1):24–43.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Société Vaudoise des Sciences Naturelle*, 37(142):547–579.

Jeong, H., B. Tombor, R. A., Oltvai, Z., and Barabasi, A. (2000). The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654.

Johndrow, J. and Lum, K. (2019). An algorithm for removing sensitive information: application to race-independent recidivism prediction. *The Annals of Applied Statistics*, 13(1):189–220.

Jolliffe, I. and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of The Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065).

Joseph, A. and Yu, B. (2016). Impact of regularization on spectral clustering. *The Annals of Statistics*, 44(4):1765–1791.

Joseph, M., Kearns, M., Morgenstern, J., Neel, S., and Roth, A. (2016). Fairness in learning: Classic and contextual bandits. *Neural Information Processing Systems (NIPS)*.

Joseph, M., Kearns, M., Morgenstern, J., Neel, S., and Roth, A. (2018). Meritocratic fairness for infinite and contextual bandits. *AAAI / ACM Conference on Artificial Intelligence*.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. (2012). Fairness-aware classifier with prejudice remover regularizer. In Flach, P., Bie, T. D., and Cristianini, N., editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7524, pages 35–50. Springer.

Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., and Tanabe, M. (2012). Kegg for integration and interpretation of large- scale molecular data sets. *Nucleic Acids Res*, 40 (Database issue):D109–114.

Kawashima, A., Kanazawa, T., Kidani, Y., Yoshida, T., Hirata, M., Nishida, K., Nojima, S., Yamamoto, Y., Kato, T., Hatano, K., Ujike, T., Nagahara, A., Fujita, K., Moritomo-Okazawa, A., Iwahori, K., Uemura, M., Imamura, R., Ohkura, N., Morii, E., Sakaguchi, S., Wada, H., and Nonomura, N. (2020). Tumour grade significantly correlates with total dysfunction of tumour tissue-infiltrating lymphocytes in renal cell carcinoma. *Scientific Reports*, 10(1):6220.

Kearns, M., Roth, A., and Sharifi-Malvajerdi, S. (2019). Average individual fairness: Algorithms, generaliza- tion and experiments. *Neural Information Processing Systems (NeurIPS)*.

Kelly, C., Karthikesalingam, A., Suleyman, M., Corrado, G., and King, D. (2019). Key challenges for delivering clinical impact with artificial intelligence. *BMC medicine*, 17(1):195.

Kleindessner, M., Samadi, S., Awasthi, P., and Morgenstern, J. (2019). Guarantees for spectral clustering with fairness constraints. *International Conference on Machine Learning (ICML)*.

Kruskal, J. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proceedings of the American Mathematical Society*, volume 7, pages 48–50.

Kruskal, J. (1964). Multidimensional scaling by optimizing goodness of fit to nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

Lachenbruch, P. and Mickey, M. (1968). Estimation of error rates in discriminant analysis. *Technometrics*, 10(1):1–11.

Lara, N. D. and Bonald, T. (2020). Spectral embedding of regularized block models. (arXiv:1912.10903 [cs.LG]).

Latino, L., Caroff, M., and Pourcel, C. (2007). Fine structure analysis of lipopolysaccharides in bacteriophage-resistant pseudomonas aeruginosa pao1 mutants. . *Microbiology*, 163(6):848–855.

Le Cao, K., Rossouw, D., Robert-Granie, C., and Besse, P. (2008). A sparse pls for variable selection when integrating omics data. *Statistical applications in genetics and molecular biology*, 7(1):Article 35.

Lelouvier, B., Servant, F., Paisse, S., Brunet, A., Benyahya, S., Serino, M., Valle, C., Ortiz, M., Puig, J., Courtney, M., Federici, M., Fernandez-Real, J., Burcelin, R., and Amar, J. (2016). Changes in blood microbiota profiles associated with liver fibrosis in obese patients: A pilot analysis. *Hepatology (Baltimore, Md)*, 64(6):2015–2027.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R., Tang, J., and Liu, H. (2018). Feature selection: A data perspective. *ACM Comput. Surv.*, 50(6).

Li, X., Kao, B., Zaochung, R., and Dawei, Y. (2019). Spectral clustering in heterogeneous information networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4221–4228.

Liang, H., Tsui, B., Ni, H., Valentim, C., Baxter, S., Cai, G. L. W., Kermany, D., Sun, X., Chen, J., He, L., Zhu, J., Tian, P., Shao, H., Zheng, L., Hou, R., Hewett, S., Li, G., Liang, P., Zang, X., and H.Xia (2019). Evaluation and accurate diagnoses of pediatric diseases using artificial intelligence. *Nature*, 25(3):433–438.

Liberzon, A., Birger., C., Thorvaldsdottir, H., Ghandi, M., Mesirov, J., and Tamayo, P. (2015). The molecular signatures database (MSigDB) hallmark gene set collection. *Cell systems*, 1(6):417–425.

Linden, A. and Mäntyniemi, S. (2011). Using the binomial distribution to model overdispersion in ecological count. *Ecology*, 92(7):1414–1421.

Liu, Z. and Barahona, M. (2020). Graph-based data clustering via multiscale community detection. *Applied Network Science*, 5(3).

Lloyd, S. (1982). Least square quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

Loomba, R., Seguritan, V., Li, W., Long, T., Klitgord, N., Bhatt, A., Dulai, P., Caussy, C., Bettencourt, R., Highlander, S., Jones, M., Sirlin, C., Schnabl, B., Brinkac, L., Schork, N., Chen, C., Brenner, D., Biggs, W., Yooseph, S., Venter, J., and Nelson, K. (2017). Gut microbiome-based metagenomic signature for non-invasive detection of advanced fibrosis in human non-alcoholic fatty liver disease. *Cell Metab*, 25(5):1054–1062 e1055.

Luche, E., Cousin, B., Garidou, L., Serino, M., Waget, A., Barreau, C., Andre, M., Valet, P., Courtney, M., Casteilla, L., and Burcelin, R. (2013). Metabolic endotoxemia directly increases the proliferation of adipocyte precursors at the onset of metabolic diseases through a cd14-dependent mechanism. *Mol Metab*, 2(3):281–291.

Luckow, A., Cook, M., Ashcraft, N., Weill, E., Djerekarov, E., and Vorster, B. (2016). Deep learning in the automotive industry: Applications and tools. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3759–3768.

Lumaka, A., Cosemans, N., Mampasi, A. L., Mubungu, G., Mvuama, N., Lubala, T., Mbuyi-Musanzayi, S., Breckpot, J., Holvoet, M., de RAVEL, T., Buggenhout, G. V., Peeters, H., Donnai, D.,

Mutesa, L., Verloes, A., Tshilobo, P. L., and Devriendt, K. (2017). Facial dysmorphism is influenced by ethnic background of the patient and of the evaluator. *Clinical genetics*, 92(2):166–171.

Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

MacQueen, B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, CA, USA. University of California Press.

McArthur, D. and Bishary, M. L. M. (2005). The roles of artificial intelligence in education: Current progress and future prospects. *Journal of Educational Technology*, 1(4):42–80.

McCarty, R., Somogyi, A., Lin, G .and Jacobsen, N., and Bandarian, V. (2009). The deazapurine biosynthetic pathway revealed: in vitro enzymatic synthesis of preq(0) from guanosine 5'-triphosphate in four steps. *Biochemistry*, 48(18):3847–3852.

McKinney, S., Sieniek, M., Godbole, V., Godwin, J., Antropova, N., Ashrafian, H., Back, T., Chesus, M., Corrado, G., Darzi, A., Etemadi, M., Garcia-Vicente, F., Gilbert, F., Halling-Brown, M., Hassabis, D., Jansen, S., Karthikesalingam, A., Kelly, C., King, D., Ledsam, J., and Shetty, S. (2020). International evaluation of an ai system for breast cancer screening. *Nature*, 577:89–94.

Me, R. (2015). Non-alcoholic fatty liver disease: a systematic review. *JAMA*, 313(22):2263–2273.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chlovskii, D., and Alon, U. (2002). Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827.

Minty, M., Canceill, T., Lê, S., Dubois, P., Amestoy, O., Loubieres, P., Christensen, J., Champion, C., Azalbert, V., Grasset, E., Hardy, S., Loubes, J., Mallet, J., Tercé, F., Vergnes, J., Burcelin, R., Serino, M., Diemer, F., and Blasco-Baque, V. (2018). Oral health and microbiota in professional rugby players a case-control study. *Journal of Dentistry*, 79:53–60.

Miura, K., Yang, L., van Rooijen, N., Brenner, D., Ohnishi, H., and Seki, E. (2013). Toll-like receptor 2 and palmitic acid cooperatively contribute to the development of non-alcoholic steatohepatitis through inflammasome activation in mice. *Hepatology*, 57(2):577–589.

Mooij, J., Janzing, D., Peters, J., and Schölkopf, B. (2009). Regression by dependence minimization and its application to causal inference in additive noise models. In Danyluk, A. P., Bottou, L., and Littman, M. L., editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 745–752.

Morik, K. (2010). *Medicine: Applications of Machine Learning*. Springer US.

Nelson, G. (2019). Bias in artificial intelligence. *North Carolina medical journal*, 80(4):220–222.

Newman, M. (2009). *Networks: An Introduction*. Oxford University Press, Oxford, UK.

Newman, M. and Girvan, M. (2004). Finding and evaluating community ctructure in networks. *Physical review E*, 69(2):026–113.

Ng, A., Jordan, M., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *In Advances in neural information processing systems*, pages 849–856.

Olfat, M. and Aswani, A. (2019). Convex formulation for fair principal component analysis. *AAAI / ACM Conference on Artificial Intelligence*.

Parikh, R., Teeple, S., and Navathe, A. (2019). Addressing bias in artificial intelligence in health care. *Jama*, 322(24):2377–2378.

Paulson, J., Stine, O., Bravo, H., and Pop, M. (2013). Differential abundance analysis for microbial marker-gene surveys. *Nature methods*, 10:1200–1202.

Pearson, K. (1901). Liii. on lines and planes of closest fit to system of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Peche, S. and Perchet, V. (2020). Robustness of community detection to random geometric perturbations. In *Proceedings of the 34th Conference on Neural Information Processing Systems*.

Pelleg, D. and Baras, D. (2007). K-means with large and noisy constraint sets. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 674–682. Springer Berlin Heidelberg.

Pérez-Suay, A., Laparra, V., Mateo-García, G., noz Marí, J. M., Gómez-Chova, L., and Camps-Valls, G. (2017). Fair kernel learning. In Ceci, M., Hollmén, J., Todorovski, L., Vens, C., and Dzeroski, S., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 339–355. Springer.

Pierantonelli, I., Rychlicki, C., Agostinelli, L., Giordano, D., Gaggini, M., Fraumene, C., Saponaro, C., Manghina, V., Sartini, L., Pinto, E. M. C., Buzzigoli, E., Trozzi, L., Giordano, A., Marzioni, M., Minicis, S., Uzzau, S., Cinti, S., Gastaldelli, A., and Svegliati-Baroni, G. (2017). Lack of nlrp3-inflammasome leads to gut-liver axis derangement, gut dysbiosis and a worsened phenotype in a mouse model of nafld. *Scientific reports*, 7(1):12200.

Pisarchik, A., Maksimenko, V., and Hramov, A. (2019). From novel technology to novel applications: Comment on "an integrated brain-machine interface platform with thousands of channels" by elon musk and neuralink. *J Med Internet Res*, 21(10).

Pollack, M. (1960). The maximum capacity through a network. *Operations research*, 8:733–736.

Pomie, C., Blasco-Baque, V., Klopp, P., Nicolas, S., Waget, A., Loubieres, P., Azalbert, V., Puel, A., Lopez, F., Dray, C., Valet, P., Lelouvier, B., Servant, F., Courtney, M., Amar, J., Burcelin, R., and Garidou, L. (2016). Triggering the adaptive immune system with commensal gut bacteria protects against insulin resistance and dysglycemia. *Mol Metab*, 5(6):392–403.

Pothen, A. (1997). Graph partitioning algorithms with applications to scientific computing. *Parallel Numerical Algorithms*, 4:323–368.

Qi, Y., Zhang, Y., Peng, Z., Wang, L., Wang, K., Feng, D., He, J., and Zheng, J. (2018). SERPINH1 overexpression in clear cell renal cell carcinoma: association with poor clinical outcome and its potential as a novel prognostic marker. *Journal of Cellular Molecular Medicine*, 22(2):1224–1235.

R., Berk, Heidari, H., Jabbari, S., Kearns, M., and Roth, A. (2018). Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*.

Randall, K. (1998). *Cilk: Efficient Multithreaded Computing*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Risser, L., Vincenot, Q., Couellan, N., and Loubes, J. (2019). Using wasserstein-2 regularization to ensure fair decisions with neural-network classifiers. *arXiv preprint arXiv:1908.05783*.

Roh, Y. and Seki, E. (2013). Toll-like receptors in alcoholic liver disease, non-alcoholic steatohepatitis and carcinogenesis. *J Gastroenterol Hepatol 28 Suppl*, 1:38–42.

Rohart, F., Gautier, B., Singh, A., and Cao, K. L. (2017). mixomics: An r package for 'omics feature selection and multiple data integration. *PLoS Comput Biol.*, 13(11).

Rosipal, R. and Trejo, L. (2001). Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 2:97–123.

Saeys, Y., Inza, I., and Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

Samadi, S., Tantipongpipat, U., J.H.Morgenstern, and Vempala, M. S. S. (2018). The price of fair pca: One extra dimension. In *Advances in Neural Information Processing Systems 31*, pages 10976–10987.

Schierwagen, R., Alvarez-Silva, C., Madsen, M., Kolbe, C., Meyer, C., Thomas, D., Uschner, F., Magdaleno, F., Jansen, C., Pohlmann, A., Praktiknjo, M., Hischebeth, G., Molitor, E., Latz, E., Lelouvier, B., Trebicka, J., and Arumugam, M. (2019). Circulating microbiome in blood of different circulatory compartments. *Gut*, 68:578–580.

Schölkopf, B., Smola, A., and Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1310.

Segata, N., Izard, J., Waldron, L., Gevers, D., Miropolsky, L., Garrett, W., and Huttenhower, C. (2011). Metagenomic biomarker discovery and explanation. *Genome biology*, 12(6):R60.

Seidman, S. (1983). Network structure and minimum degree. *Social Networks*, 5(3):269–287.

Shannon, C. (1948). A mathematical theory of communication. *The Bess System technical Journal*, 27(3):379–423 and 623–656.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Sidey-Gibbons, J. (2019). Machine learning in medicine: a practical introduction. *BMC Medical Research Methodology*, 19.

Siegel, R. (2014). Race-conscious but race-neutral: The constitutionality of disparate impact in the roberts court. *Ala. L. Rev.*, 66:653.

Simpson, E. (1949). Measurement of diversity. *Nature*, 163:688.

Smith, S. (1997). The integration of communications networks in the intelligent building. *Automation in Construction*, 6(5):511 – 527.

Sookoian, S., Salatino, A., Castano, G., Landa, M., Fijalkowky, C., Garaycoechea, M., and Pirola, C. (2020). Intrahepatic bacterial metataxonomic signature in non-alcoholic fatty liver disease. *Gut*, 69:1483–1491.

Spellman, P., Sherlock, G., Zhang, M., Vishwanath, I., Eisen, K. A. M., Brown, P., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell-cycle-regulated genes of yeast saccharomyces cerevisiae by microarray hybridization. *Molecular biology of the cell*, 9(12):3273–3297.

Steele, J. (2002). Minimal spanning trees for graphs with random edge lengths. In Springer, editor, *Mathematics and Computer Science II*, pages 223–245.

Stephan, L. and Massoulié, L. (2019). Robustness of spectral methods for community detection. In Beygelzimer, A. and Hsu, D., editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2831–2860, Phoenix, USA. PMLR.

Tang, W. and Khoshgoftaar, T. M. (2004). Noise identification with the k-means algorithm. In *16th IEEE International Conference on Tools with Artificial Intelligence*, pages 373–378.

Tarjan, R. (1972). Depth first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160.

Tat, E., Bhatt, D., and Rabbat, M. (2020). Addressing bias: artificial intelligence in cardiovascular medicine. *The Lancet Digital Health*, 2(12):e635–e636.

Tay, R., Richardson, E., and Toh, H. (2020). Revisiting the role of CD4+ T cells in cancer immunotherapy - new insights into old paradigms. *Cancer Gene Therapy*, pages https://doi.org/10.1038/s41417–020–0183–x.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Methodological*, 58(1):267–288.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Trippi, R. and Turban, E. (1992). *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance*. McGraw-Hill, Inc.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Verges, B., Duvillard, L., Lagrost, L., Vachoux, C., Garret, C., Bouyer, K., Courtney, M., Pomie, C., and Burcelin, R. (2014). Changes in lipoprotein kinetics associated with type 2 diabetes affect the distribution of lipopolysaccharides among lipoproteins. *The Journal of clinical endocrinology and metabolism*, 99(7):E1245–1253.

Villani, C. (2009). *Optimal transport: old and new*. Springer Verlag.

Wang, L., Zhang, Y., Zhang, Y., Xu, X., and Cao, S. (2017). Prescription function prediction using topic model and multilabel classifiers. *Evid Based Complement Alternat Med*, 2017:8279109.

Wang, X. and Davidson, I. (2010). Flexible constrained spectral clustering. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 563–572. Association for Computing Machinery.

Ward, J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the AmericanStatistical Association*, 58(301):236–244.

Weirauch, M. (2011). Gene coexpression networks for the analysis of dna microarray data. *Applied statstics for network biology: methods in systems biology*, 1:215–250.

Whittaker, R. (1972). Evolution and measurement of species diversity. *Taxon*, 21(2-3):213–251.

Wold, H. (1966). *Multivariate analysis*. Academic Press.

Wold, H. (1975). Soft modelling by latent variables: The non-linear iterative partial least squares (nipals) approach. *Journal of Applied Probability*, 12:117–142.

Wold, S., Ruhe, H., Wold, H., and III, W. D. (1984). The collinearity porblem in linear regression. the partieal least squares (pls) approach to generalize inverse. *SIAM Journal of Scientific and Statistical Computations*, 5:735–743.

Wu, C., Ioannidis, S., Szaier, M., Li, X., Kaeli, D., and Dy, J. (2018). Iterative spectral method for alternative clustering. In *Proceedings of Machine Learning Research*, volume 84, pages 115–123.

Xiao, Y. and Meierhofer, D. (2019). Glutathione metabolism in renal cell carcinoma progression and implications for therapies. *International Journal of Molecular Sciences*, 20(15):3672.

Yu, K., Beam, A., and Kohane, I. (2018). Artificial intelligence in healthcare. *Nature biomedical engineering*, 2:719 – 731.

Yu, L. and Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, pages 856–863.

Zafar, M., Valera, I., M.Gomez-Rodriguez, and Gummadi, K. (2019). Fairness contraints: A flexinle approach for fair classification. *Journal of Machine Learning Research*, 20:1–42.

Zan, B. and Noon, C. (1998). Shortest path algorithms: An evaluation using real road networks. *Transportation Science*, 32(1):65–73.

Zelnik-Manor, L. and Perona, P. (2005). Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, volume 17, pages 1601–1608.

Zeng, D., Chen, H., Lusch, R., and Li, S. (2010). Social media analytics and intelligence. *IEEE Intelligent Systems*, 25(6):13–16.

Zhang, Y. and Rohe, K. (2018). Understanding regularized spectral clustering via graph conductance. *In Advances in Neural Information Processing Systems*, pages 10631–10640.

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., and Liu, H. (2010). Advancing feature selection research. In *ASU feature selection repository*, pages 1–28.