# Développement de méthodes chimiométriques pour le traitement des données massives

Maxime Metz

# THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

**En Génie des procédés**

**École doctorale École doctorale GAIA –Biodiversité, Agriculture, Alimentation, Environnement, Terre, Eau Portée par l'Université de Montpellier**

**Unité de recherche UMR ITAP –Information, Technologie, Analyse environnementale, Procédés agricoles**

# Développement de méthodes chimiométriques pour le traitement des données massives


**Présentée par Maxime Metz**

**Le 26 Novembre 2021**

**Sous la direction de Matthieu Lesnoff et Jean-Michel Roger**


**Devant le jury composé de**

| | |
|---|---|
| Douglas Rutledge, Professeur émérite, AgroParisTech, France | Président |
| Fédérico Marini, Professeur, Université de Rome, Italie | Rapporteur |
| Marina Cocchi, Professeure associée, Université de Modène et de Reggio d'Émilie, Italie | Rapporteur |
| Florent Masseglia, Directeur de recherche, INRIA, France | Examinateur |
| Gilbert Saporta, Professeur émérite, CNAM, France | Examinateur |
| Jean-Michel Roger, Ingénieur de recherche, INRAE-ITAP, France | Co-directeur |
| **Invités** | |
| Matthieu Lesnoff, Chercheur, CIRAD, France | Co-directeur/ Invité |
| Reza Akbarinia, Chargé de Recherche, INRIA, France | Invité |

UNIVERSITÉ DE MONTPELLIER

# Remerciements

Ces trois années de thèse furent très enrichissantes pour moi, tant sur le plan personnel que professionnel. Ces quelques lignes ne me permettent pas de remercier autant que je le souhaiterais l'ensemble des personnes qui ont participé à ma thèse de près ou de loin. Cependant, je vais faire de mon mieux pour n'oublier personne.

Tout d'abord, je vais commencer par ceux qui m'ont supervisé durant 3 années de thèse, merci à vous Matthieu et Jean-Michel. Grâce à vous j'ai pu progresser, sur ma rigueur scientifique, en chimiométrie et surtout dans la connaissance des différents vignobles de l'Hérault. Merci à vous deux, par moment ce fut long et exténuant, mais grâce à vous je n'ai pas perdu le nord (même si le nord c'est surfait). Je tiens également à remercier Reza et Florent, merci de m'avoir accordé autant de temps pour discuter science. Merci à toi Reza, ta bonne humeur perpétuelle et ton optimisme ont rendu très agréable notre collaboration !

Je remercie L'UMR ITAP de m'avoir accueillie, merci à l'ancien DU, Tewfki pour les discussions autour du JLL. Merci également aux membres de l'équipe COMIC dont : Ryad, Daniel, Daphné, Maxime, Anis, Aldrig, Florent, Sylvia, Belal, Shérif, Arnaud, Héloïse, Véronique. Ryad, merci à toi, tu as été plus qu'un chef d'équipe pour nous, merci pour ton aide et tes conseils ! Daphné, merci à toi pour ta bienveillance, ta rigueur et ta bonne humeur (et pour TOUTES les corrections que tu as pu me faire pour la thèse). Maxime, le plus COMIC de tous ! Merci pour tes blagues. Merci Anis, Aldrig pour les craquages le midi. Merci à Silvia, Florent pour les soirées et surtout leur aide pendant ma thèse. Merci à Véronique pour les conseils sur la rédaction de mon mémoire et les repas à la cantine. Ah Belal ! Mon 'khouya', avant de te connaître je ne savais pas ce que c'était que la fierté des DZ, on en a partagé des choses ensemble ! Prends soin de toi mec, bon courage pour ta thèse. PS : vas-y molo sur les croissants ! Ah Daniel ! Merci pour tes blagues douteuses (tu rigolais vraiment ? haha), est-ce qu'on est vraiment trop jeune ?

Merci aux collègues de l'IFV pour les repas, leur gentillesse et leurs accueils.

Merci Anice et Eva pour les pauses café, les discussions, l'entraide, la moto, etc. Merci aux gars de l'atelier, Gérard, JF, David, Augustin, et tous les autres, merci pour les repas, les blagues douteuses, les coups de mains pour la mécanique, les pauses café, la pêche, etc, vous faites partis des permanents qui nous permettent de nous sentir bien au sein de l'unité.

Merci à l'équipe ChemHouse, je suis désolé je ne peux pas tous vous citer... Mais

merci pour les collaborations et les discussions. Merci à toi Alessandra pour m'avoir aidé pour mon premier papier, reviens quand tu veux !

Merci à mes collègues de Verdansk : Jimmy, Thomas, Ines, Jojo, Alex et tous les autres, grâce à vous le confinement a été un peu moins dur pour moi.

Merci à toute ma famille, Mélanie, merci à toi, pour la patience et le soutien dont tu as fait preuve pendant toute la durée de cette thèse. Notre couple était un socle solide à l'épanouissement de ce projet. Merci Mathieu, tu es un exemple de combativité pour moi. Papa, Maman, merci à vous, merci pour votre aide, merci d'être et d'avoir été là pour moi. Merci à mes grands-parents, vous êtes et avez été pour moi des modèles, c'est grâce à vous que j'en suis là aujourd'hui.

# Préface

Cette thèse est présentée dans le but d'obtenir le grade de docteur de l'école doctorale GAIA de l'université de Montpellier et de Montpellier SupAgro/l'institut Agro.

Ces travaux de thèse résultent d'une collaboration entre 3 instituts : l'INRAE (l'Institut national de recherche pour l'agriculture, l'alimentation et l'environnement), le CIRAD (Centre de coopération internationale en recherche agronomique pour le développement) et l'INRIA (Institut national de recherche en informatique et en automatique). Cette thèse a été financée par l'institut de convergence #DigitAg et par l'INRAE. Trois laboratoires ont été concernés par ce sujet de thèse : l'unité mixte de recherche ITAP (technologies et méthodes pour l'agriculture de demain), l'unité mixte de recherche SELMET (Systèmes d'élevages méditerranéens et tropicaux) et le LIRMM (Laboratoire d'informatique, de robotique et de microélectronique de Montpellier).

Ces travaux de recherches ont été co-dirigés par Dr. Matthieu Lesnoff (SELMET, CIRAD) et Dr. Jean-Michel Roger (ITAP, INRAE) et ont été co-supervisés par Dr. Reza Akbarinia (LIRMM,INRIA), Dr. Florent Masseglia (LIRMM, INRIA), et Dr. Florent Abdelghafour (ITAP, INRAE).

Le projet de thèse a duré 3 ans, d'octobre 2018 à novembre 2021.

Cette thèse est composée de six chapitres : le chapitre 1 introduit l'état de l'art et les questions de recherches de la thèse. Les chapitres 2, 3 et 4 présentent les contributions principales de la thèse. Le chapitre 6 présente quant à lui les conclusions générales et les perspectives de la thèse. Dans le sixième chapitre, les publications scientifiques associées aux travaux de thèse sont présentées.

# Résumé (Français)

L'analyse des données chimiques, communément appelée chimiométrie, est utilisée en agronomie pour répondre à diverses questions telles que l'étude des sols, des fourrages ou le phénotypage. Aujourd'hui, une grande quantité de données peut être générée et les chimiométriciens doivent être capables de les analyser. Les outils habituels ne sont pas encore capables de traiter efficacement ces données. Des outils dans le domaine du big-data ont été développés afin de permettre de traiter des bases de données volumineuses. Ces outils n'ont pas encore été évalués pour la chimiométrie. L'objectif de cette thèse est donc d'étudier le traitement de données massives pour la chimiométrie. Pour ce faire, trois axes de recherche ont été étudiés. Le premier axe de recherche consiste à étudier comment permettre le traitement de données massives par des méthodes locales. Les méthodes locales calibrent un modèle par individu à prédire sur ses plus proches voisins. Le deuxième axe de recherche consiste à étudier la pertinence d'un individu au sein d'un modèle local. Le troisième axe consiste à combiner les idées développées dans les deux premiers axes pour rendre les méthodes performantes pour la chimiométrie. Pour répondre au premier axe, une nouvelle méthode nommée parSketch-PLS a été étudiée et développée. Pour aborder le deuxième axe, une méthode appelée RoBoost-PLSR a été développée. Pour étudier le troisième axe, deux prémices de méthodes ont été proposées. Les résultats associés à ces développements ont mis en évidence l'intérêt d'adapter les outils de traitement de données massives à la chimiométrie. En effet, les outils utilisés pour le traitement des données massives ne reposent pas forcément sur les mêmes connaissances que les outils développés pour la chimiométrie. Cela peut donc conduire à une diminution de la capacité prédictive des méthodes chimiométriques. Cette thèse met donc en avant l'intérêt de rapprocher ces deux domaines afin de proposer un ensemble de méthodes et d'outils de traitement de données massives chimique.

# Abstract

Chemical data analysis, also known as chemometrics, is widely used in agronomy to address various issues such as soil, forage or phenotyping studies. Today, a large amount of data can be generated and chemometricians must be able to analyse them. However, their usual tools are not able to process this amount of data efficiently. Tools originated from big data field have been developed to process large databases but have not yet been evaluated for chemometrics. The objective of this thesis is therefore to study massive data processing for chemometrics. To this end, three research axis were investigated. The first research axis is to study how to enable massive data processing by local methods. Local methods calibrate a model per individual to be predicted on its nearest neighbours. The second research axis is to study the relevance of an individual within a local model. The third research axis consists in combining the ideas developed in the first two axis to propose efficient methods for chemometrics. To address the first research focus, a new method called parSketch-PLS was studied and developed. To address the second focus, a method called RoBoost-PLSR was developed. To address the third focus, two premise methods were proposed. The results associated with these developments highlighted the interest in adapting massive data processing tools to chemometrics. Indeed, tools used for massive data processing do not necessarily rely on the same knowledge as tools developed for chemometrics. This can degrade the predictive capacity of chemometrics. This thesis therefore highlights the interest in bringing these two fields together in order to propose a set of methods and tools for processing massive chemical data.

# Table des matières

# Introduction

*Chapitre* 1

## Sommaire



## 1.1 La chimiométrie, l'agriculture numérique et les big-data

L'agriculture doit faire face aujourd'hui à de nouveaux enjeux comme la remise en question de l'usage des produits phytosanitaires, l'alimentation de proximité, la sécurité alimentaire tout au long des filières ou encore la transition agroécologique [1]. Ces défis prennent également en compte les nouvelles exigences des consommateurs et des politiques publiques qui imposent des contraintes supplémentaires aux systèmes

de production. Ainsi, l'agriculture connaît de fortes transformations pour s'adapter et répondre conjointement à ces différents défis. Ces transformations concernent principalement les aspects techniques de l'agriculture, mais il s'agit avant tout de mieux comprendre les systèmes d'exploitation agricole dans leur ensemble. A ce titre, de nombreuses technologies de l'information (capteurs, réseaux intelligents, outils de la science de la donnée, voire automatisme et robotique) sont désormais utilisées en agriculture. Cette convergence entre ces nouvelles technologies numériques et l'agriculture est appelée agriculture numérique.

Le développement de l'agriculture numérique est la raison d'un changement important dans la création de connaissances en agriculture. Comme pour un grand nombre de domaines, la connaissance était créée à partir de variables définies par des experts scientifiques et étayées par des données réelles [2]. Cela était dû en grande partie à l'utilisation de moyens de mesures et d'expérimentations coûteux qui rendaient difficile la réalisation de mesures en grande quantité. Il était donc nécessaire d'obtenir le maximum d'informations possibles avec le minimum d'expériences. L'expert devait sélectionner le minimum de variables afin de réduire au maximum les coûts. Désormais, il est possible d'acquérir rapidement une quantité considérable de données et donc d'obtenir un grand nombre de variables pour analyser les expérimentations.

Le rôle des experts scientifiques dans la stratégie de développement de la connaissance est alors redéfini. Par conséquent, le moteur du développement de la connaissance devient en premier lieu l'acquisition d'un grand nombre de variables, qui sont analysées et exploitées afin d'extraire les informations les plus pertinentes. L'acquisition d'un grand nombre de variables permet d'étudier des phénomènes potentiellement plus globaux. L'analyse de données, les statistiques et notamment l'utilisation des outils de la chimiométrie (analyse des données chimiques) en agriculture mettent en avant la possibilité d'extraire des informations pertinentes à partir d'un grand nombre de variables. En agriculture numérique, la chimiométrie a été très souvent utilisée avec la spectroscopie PIR (proche infrarouge) [3]. La chimiométrie a permis à partir de variables faiblement prédictives mais présentes en grand nombre de répondre à des problématiques complexes. La chimiométrie utilisée pour analyser des données issues de capteurs à faible coût permet d'obtenir des solutions adaptées aux besoins de l'agriculture, c'est-à-dire des moyens de mesure non destructifs, rapides et low cost. Par exemple, la chimiométrie a été utilisée pour différentes applications comme la prédiction de la maturité des baies de raisin [4] ou encore la prédiction des caractéristiques du sol [5]. Cependant, de nouveaux outils doivent être créés afin de répondre à une complexité grandissante du milieu, des capteurs ou encore des échantillons à analyser. Pour cela, de nombreuses méthodes d'analyse en chimiométrie ont été développées pour adresser différentes problématiques comme la

robustesse des modèles face aux perturbations des capteurs [6] ou bien des stratégies d'analyses des sources de variabilités [7].
En agriculture numérique, il y a désormais une volonté de comprendre des phénomènes de plus en plus complexes et globaux. L'hypothèse faite est qu'il faut acquérir une masse de données considérable afin de représenter au mieux les différents phénomènes étudiés, leurs variabilités et leurs interactions. Pour acquérir des masses de données considérables, différents capteurs ont été développés. C'est notamment le cas des capteurs frugaux, qui par leur portabilité et leur débit, permettent une collecte de données massives à faible coût. D'autres outils comme l'imagerie hyperspectrale sont des sujets de recherche et d'applications prolifiques notamment dans le développement de plateformes de phénotypages [8; 9]. Cependant, la massification des données entraîne naturellement une obsolescence de certains outils de l'agriculture numérique, notamment de certaines méthodes d'analyse de données développées en chimiométrie.
Les nouvelles méthodes associées au traitement des données massives se retrouvent dans un domaine appelé "big-data". Le big-data est défini comme une famille d'outils permettant de traiter des données variées et volumineuses. Ces outils doivent répondre aux trois "V", respectivement associés au volume, à la vélocité et à la variété. Ces outils doivent pouvoir traiter des données volumineuses avec une efficience calculatoire optimisée et être versatiles à la diversité des sources et des natures des données exploitées (*ex :* : spectres, images, vidéos). Dans [10] les six « V » du Big Data agricole sont définis par :

- Le «Volume» : la taille des données à traiter,
- La «Variété» : les sources extrêmement variées des données,
- La «Vélocité» : le flux de données constant à analyser,
- La «Véracité» et la «Validité» : avoir des outils technologiques et mathématiques toujours pertinents,
- La «Visualisation» : permettant de conserver une représentation des données,
- La «Visibilité» : liée à la compréhension des phénomènes sous-jacents.

Les nouveaux défis de la chimiométrie sont donc de développer des méthodes permettant de prendre en compte ces six "V" du big-data.

## 1.2 Contexte de la thèse

Aujourd'hui la chimiométrie fait essentiellement face à deux problématiques issues du big-data qui sont la variété des données et le volume de données. La variété est notamment traitée en chimiométrie par des approches du type multi-blocs ou multi-voies

[11–13]. Cependant, le traitement de base de données massives (beaucoup d'échantillons) est peu étudié en chimiométrie. Dans ce contexte nous proposons de nous intéresser à la problématique du traitement de gros volumes de données par des méthodes dites locales. Ces méthodes sont très utilisées en chimiométrie et intéressantes notamment pour le traitement de données spectrales. Dans cette thèse, il est proposé d'évaluer les approches locales dans un cadre prédictif (i.e. prédire une/des variables $\mathbf{Y}$ à partir de variables $\mathbf{X}$) lorsque le nombre d'échantillons ou d'individus est trop grand pour permettre aux approches locales d'être utilisables. Dans la suite de cette thèse, on parlera de données massives lorsque le nombre d'individus est très grand.

### 1.2.1 Les données massives

L'un des principaux défis rencontrés dans le traitement de données massives découle naturellement de la quantité de données, qui induit un temps de calcul qui peut être rédhibitoire. Dans ces conditions, même les opérations les plus élémentaires peuvent devenir coûteuses, voire limitantes. Les données massives sont souvent présentées comme des quantités de données de l'ordre du Téra ($10^{12}$). Toutefois, les données massives ne se restreignent pas uniquement à ce cas. Il est également possible de parler de données massives, si le nombre de données ne permet pas d'utiliser les infrastructures et algorithmes classiques [14]. En effet, il est possible de faire face à un nombre moins conséquent de données mais qui nécessitent l'utilisation d'algorithmes bien plus complexes. Même si ce nombre de données n'atteint pas le téra, le chimiométricien ne pourra pas utiliser l'algorithme car il ne sera pas en mesure de l'appliquer. Comme précisé dans [15], plusieurs paramètres doivent être considérés pour parler de traitement de données :

— Le volume de données : il peut être un frein pour la plupart des algorithmes utilisés pour traiter les données. Par exemple, pour traiter des données avec plusieurs millions voire milliards d'individus, il est nécessaire de mettre à disposition des serveurs de calcul pour réaliser le stockage et le traitement de ces dernières.

— La complexité algorithmique : certains algorithmes de traitement de données ne seront plus utilisables en calcul centralisé à partir de quelques milliers d'échantillons, alors que d'autres sont opérationnels avec plusieurs millions.

— La puissance de calcul matérielle et/ou logicielle : certains ordinateurs par leur architecture limitent la capacité de calcul. Mais également, certains logiciels ne permettent pas de tirer parti de l'architecture de l'ordinateur.

Cela signifie donc que l'on parle de données massives quand il n'est plus possible de traiter les données avec les outils et algorithmes usuels de la chimiométrie. Dans

[16] les différents challenges associés au traitement des données massives sont mis en évidence. Les outils développés en chimiométrie devront donc également répondre à trois challenges détaillés ci-dessous.

### La complexité algorithmique

Les algorithmes utilisés en chimiométrie sont tous plus ou moins dépendants du nombre d'individus à traiter. Le nombre d'individus va agir sur deux points importants des algorithmes utilisés en chimiométrie : d'une part les temps de calcul et d'autre part la mémoire utilisée pour réaliser ce calcul. En effet, certains algorithmes voient leurs temps de calcul évoluer de manière linéaire, quadratique ou même logarithmique par rapport au nombre d'individus traités. De même, les algorithmes nécessitent de disposer de ressources mémoires adaptées. Il est donc nécessaire de développer de nouveaux algorithmes offrant une faible complexité temporelle (réduction du temps de calcul) et spatiale (diminution du stockage).

### Stockage et modularité

La plupart des algorithmes reposent sur l'hypothèse que les données sont analysables dans leur totalité et déployées entièrement en mémoire. Cependant, dans certains cas il n'est pas possible de charger la totalité des données dans la mémoire disponible. De nombreux développements ont été réalisés pour produire des outils permettant de traiter les données par morceaux. C'est par exemple le cas du package "biglmm" [17] développé sur R à cette fin. En mettant à jour un modèle linéaire, il est possible de traiter une base de données par morceaux sans perdre de l'information. Une solution est donc d'utiliser des outils basés sur le paradigme "Mapreduce" [18]. Cette approche permet de traiter en parallèle (sur plusieurs noeuds) de grands ensembles de données. Le traitement des données en parallèle consiste en l'exécution simultanée d'une même tâche afin de pouvoir être répartie entre plusieurs processeurs en vue de traiter plus rapidement les données. Des outils dédiés au stockage et au traitement de données volumineuses tels que "Spark" [19], permettent de partitionner les données et de réaliser des calculs en parallèles. Malgré l'efficience de Spark, peu d'algorithmes sont compatibles avec ce type d'approches. Par conséquent, les nouveaux algorithmes, développés en chimiométrie spécifiquement pour le traitement des données massives, doivent être pensés pour présenter une faible complexité ainsi qu'être adaptés aux architectures distribuées.

**Non-linéarité**

La masse de données peut également créer de nouvelles contraintes liées à la nature même des méthodes d'analyse des données. L'accroissement des bases de données résulte bien souvent d'une augmentation de la variabilité des données, à tel point que dans certains cas, les méthodes linéaires usuellement employées ne sont plus pertinentes dans le contexte des données massives. Par exemple, il est possible de combiner des mesures réalisées sur différents types d'échantillons afin de prédire un même paramètre physico-chimique. C'est notamment le cas d'applications comme la prédiction des propriétés physico-chimiques du sol, pour lesquelles des bases de données particulièrement conséquentes ont été produites [20; 21]. Dans ce cas, les données étant issues de différentes régions géographiques sont donc très variables et introduisent des phénomènes non-linéaires. Pour traiter les phénomènes non-linéaires, des méthodes non-linéaires devront donc être employées. Bien que ce problème ne soit pas exclusif aux grands ensembles de données, on peut s'attendre à ce que les non-linéarités soient plus récurrentes, voire exacerbées dans ce contexte.

### 1.2.2 Les méthodes PLS locales

La chimiométrie propose un large panel d'outils pour l'analyse et l'interprétation des données spectroscopiques. Un des objectifs de ces outils est d'associer l'information spectrale (portée par la matrice $\mathbf{X}$) à des propriétés physico-chimiques (portées par la matrice $\mathbf{Y}$) dans le but de réaliser des prédictions. Les prédictions peuvent concerner des variables qualitatives (*ex :* appartenance à un génotype, une espèce, etc) ou bien des variables quantitatives (ex : taux de protéines, taux de sucre, etc...). Parmi eux, une méthode de référence, la régression des moindres carrés partiels (PLSR), permet de réaliser des modèles prédictifs très efficients lorsqu'il existe une relation linéaire entre $\mathbf{X}$ et $\mathbf{Y}$. De par la diversité des applications dans le domaine, il est courant d'être confronté à des données issues de l'agrégation de mesures réalisées sur des échantillons de différentes natures. Cette agrégation introduit souvent des relations non-linéaires entre $\mathbf{X}$ et $\mathbf{Y}$. Ces relations non-linéaires peuvent altérer significativement la qualité des prédictions. Une solution pour répondre à cette problématique est l'utilisation de méthodes PLS locales. Les méthodes PLS locales ont pour objectif de calibrer un modèle PLS sur les plus proches voisins d'un individu à prédire. La définition des plus proches voisins est réalisée à l'aide d'un critère de dissimilarité : plus les individus de la base de données sont similaires à l'individu à prédire, plus ils impactent la calibration du modèle PLS local. Afin d'intégrer la notion de dissimilarité dans le calcul d'un modèle PLS, les individus se voient attribuer un poids entre 0 et 1. Si un individu de la base de données a un poids de 0, il n'est pas

utilisé pour la calibration du modèle PLS. Au contraire, si un individu a un poids de 1, il est totalement pris en compte dans la calibration du modèle.

Les méthodes PLS locales sont utilisées dans un grand nombre d'applications [22–28], et particulièrement dans le contexte de l'agriculture numérique [29–32]. Parmi ces applications, on retrouve : la prédiction de propriétés du sol [33; 34], des ressources fourragères [35], ou bien la prédiction de la qualité nutritionnelle du régime alimentaire des bovins [36].
Les méthodes PLS locales s'appuient sur une approche initialement développée dans [37]. Cette approche avait pour objectif de modéliser une relation non-linéaire en réalisant une série de modèles linéaires. Cette approche a été développée uniquement dans le cadre où une seule variable explicative était utilisée pour modéliser une réponse. Dans [37] une double stratégie de pondération a été intégrée. En effet, les plus proches voisins d'un individu dans la base de données sont utilisés pour calibrer un modèle linéaire (pondération discrète 0 ou 1) puis les plus proches voisins de l'individu sont pondérés en fonction de leurs mesures d'aberrance (pondération continue entre 0 et 1). Cette méthode est très intéressante car elle permet à partir de modèles linéaires d'avoir un méta-modèle non-linéaire. Cependant, cette approche ne peut pas s'appliquer au cas des données spectrales car elle n'a été développée que pour la régression univariée (une variable explicative). Pour résoudre cette problématiques, différentes méthodes ont été développées pour la calibration de modèles linéaires locaux en grandes dimensions, notamment avec la Locally Weighted Regression (LWR) [38; 39]. Plus particulièrement, [39] propose de réaliser une analyse en composantes principales (PCA) sur l'ensemble des données. Sur la base des scores de cette PCA, un voisinage est déterminé et des modèles locaux de régression sur la base de ces scores sont calibrés. Par la suite, la méthode LOCAL[40] a été développée. Cette méthode propose de sélectionner un voisinage d'un nouvel individu puis de calibrer un modèle PLS sur ce voisinage. Par ailleurs, la Locally Weighted Partial Least Squared Regression (LWPLSR) [41–43] a été développée afin d'introduire les notions de distances au sein d'un voisinage directement dans le modèle PLS par le biais d'une pondération des individus. Contrairement à la méthode LOCAL, les poids des individus ne sont pas 0 ou 1 mais sont des poids compris entre 0 et 1. Cette pondération permet donc de ne pas supprimer totalement les individus dans la calibration. Cette méthode a été développée dans le cas de la régression ainsi que de la classification [44].

Les deux stratégies principalement utilisées dans les méthodes PLS locales peuvent donc se résumer en deux types de stratégies : KNN-PLS (dont fait partie la méthode LOCAL) et LWPLS. KNN(k-nearest neighbors) définit la famille de méthodes de recherche des plus proches voisins. Dans ces deux cas, le voisinage est pris en compte.

Dans le premier cas, une pondération discrète des individus de la base de données est réalisée, c'est-à-dire que seuls les plus proches voisins sont utilisés pour la calibration. Dans le deuxième cas, tous les individus de la base de données sont utilisés pour la calibration mais avec un poids dans la calibration dépendant de la distance à l'individu à prédire.

Étant donné leurs avantages, ces méthodes ont été particulièrement déclinées et évaluées. Un des points les plus étudiés dans ces approches est le critère de similarité choisi pour estimer le voisinage d'un individu. En effet, si le critère de similarité n'est pas pertinent, il se peut que les voisins estimés ne permettent pas de calibrer le modèle le plus adapté pour prédire le nouvel individu. De nombreuses stratégies ont donc été développées, notamment dans le cadre des stratégies KNN-PLS et LWPLS, afin d'estimer les voisins les plus pertinents pour un individu à prédire [43; 45–48]. La stratégie LWPLS est très utilisée lorsque le nombre d'individus dans la base de données est faible. Dans cette approche, tous les individus de la base de données sont potentiellement utilisés pour estimer le modèle. Chaque individu est pondéré en fonction de sa similarité avec l'individu à prédire. De ce fait, le coût calculatoire est très élevé. Pour résoudre ce problème, [49] propose de combiner les stratégies KNN-PLS et LWPLS, c'est-à-dire qu'un voisinage est défini, puis un modèle PLS pondéré est calculé sur ce voisinage. Cette approche permet d'obtenir des temps de calcul réduits et des performances de prédictions élevées.

### 1.2.3 Les méthodes PLS locales pour le traitement des données massives

Les méthodes PLS locales font partie des candidats potentiellement intéressants pour le traitement des données massives. En effet, les méthodes PLS locales consistent avant tout à s'intéresser à des sous-ensembles définis spécifiquement pour modéliser au mieux chacun des individus à prédire. On nomme ces sous-ensembles "voisinages". Définir un voisinage, c'est également déterminer les individus *a priori* pertinents dans la base de calibration pour modéliser un individu à prédire. Pour définir les voisinages, l'approche la plus répandue est celle des KNN. Elle permet de réaliser un filtrage des données et potentiellement d'écarter du modèle des données aberrantes ou fortement bruitées. Selon les algorithmes employés, le KNN peut être massivement parallélisé. Les méthodes PLS locales traitent donc certains paradigmes de non-linéarité et de variabilité associés aux données massives. Malgré des récents développements comme dans [49], les méthodes PLS locales ne permettent pas de résoudre les problématiques associées aux données massives. En effet, les méthodes PLS locales utilisent pour la plupart l'algorithme "force-brute" pour la recherche de voisins. Cet algorithme consiste

à calculer des dissimilarités entre l'individu à prédire et tous les individus de la base de données de calibration. Les dissimilarités les plus faibles permettent de définir les voisins de l'individu à prédire. De plus, l'algorithme "force-brute" permet une recherche dite exacte c'est-à-dire que tous les individus de la base de données ont été comparés à l'individu à prédire. De ce fait, l'ordre défini à l'aide du critère de similarité est exact. Cela signifie dans d'autres termes qu'il n'y a aucun risque qu'un individu similaire selon un critère défini soit considéré comme loin. Cependant, cet algorithme impose de calculer pour chaque modèle et/ou prédiction, autant de distances qu'il y a d'individus dans la base de données. L'approche "force-brute" est donc peu efficiente d'un point de vue calculatoire. Lorsque le nombre d'individus de la base de calibration est très grand, les temps de calcul peuvent être rédhibitoires ou rendre la méthode inopérante par manque de mémoire. Il est donc nécessaire de trouver de nouveaux algorithmes permettant une recherche de voisins dans des bases de données massives.

D'autre part, bien que l'algorithme de recherche des plus proches voisins soit exact, il ne permet pas toujours de sélectionner les voisins les plus pertinents. S'il y a présence de voisins nuisibles à la calibration, la performance de prédiction de la méthode PLS locale se verra donc diminuée. En effet, la PLS n'est pas robuste aux individus qui ne partagent pas les mêmes distributions que les autres [50]. La calibration de la PLS avec ces individus peut rendre le modèle globalement peu prédictif. De plus, si le voisinage sélectionné pour un échantillon est trop variable et que la relation entre les variables explicatives et les variables à expliquer est non-linéaire, le voisinage sélectionné ne permet pas de construire un modèle PLS performant. Cela signifie donc que certains individus peuvent être pertinents au sein du voisinage et nuisibles au sein d'un autre voisinage. Lors du traitement de données massives ce type de problématique peut se retrouver voire être exacerbé.

Dans ce contexte, l'utilisation de données massives pour établir des modèles de prédiction posent deux problèmes. D'une part, le problème lié aux limites calculatoires atteintes par les méthodes locales, et d'autre part, le problème lié à la définition des voisins pertinents.

## 1.3 Objectifs et questions scientifiques de la thèse

Comme mentionné précédemment, les méthodes PLS locales possèdent des propriétés intéressantes pour le traitement de grands ensembles de données mais souffrent encore de plusieurs limites dues à des verrous méthodologiques. Dans cette thèse, trois sujets

vont être étudiés :

- Les algorithmes KNN-PLS existants utilisent l'algorithme force-brute pour la détermination du voisinage. Cet algorithme présente des temps de calcul rédhibitoires dans le cas de données massives. Il est donc nécessaire d'étudier des algorithmes permettant une recherche rapide de voisins dans un contexte de données massives.
- Au sein du voisinage défini, il est possible que certains individus ne soient pas réellement pertinents et qu'ils détériorent l'étalonnage. Cette situation peut être exacerbée lors de l'utilisation d'algorithmes différents de force-brute. Il est donc nécessaire d'étudier la pertinence des voisins pour la calibration d'un modèle PLS.
- Les méthodes PLS locales développées en chimiométrie intègrent des connaissances métier et permettent d'obtenir de très bonnes performances de prédictions. Il est donc intéressant d'étudier la combinaison des connaissances de chimiométrie avec les algorithmes développés pour le traitement des base de données massives.

Associée à ces trois problématiques, les questions scientifiques suivantes sont posées dans cette thèse :

— Comment mettre à profit des paradigmes du "big-data" pour améliorer les modèles PLS locaux actuels ?

— Comment estimer la pertinence d'un individu par rapport à un modèle PLS ?

— Comment associer les paradigmes de la chimiométrie et du big-data ?

Pour répondre à ces questions, une méthode issue du big-data a été étudiée dans le cadre de la stratégie KNN-PLS. Cette méthode a pour avantage d'être massivement parallélisable et donc de répondre déjà à la plupart des problématiques liées au traitement des données massives. Dans un second temps, les méthodes robustes ont été étudiées pour définir la pertinence des individus. Cette étude a permis de développer une méthode permettant de traiter des données aberrantes. Pour finir, des outils de chimiométrie ont été combinés aux stratégies utilisées dans la méthode issue du big-data.

## 1.4 Contributions

Cette thèse a conduit à différents développements méthodologiques en chimiométrie. Ces développements sont principalement associés aux deux premières questions scientifiques. Dans cette thèse, trois méthodes principales ont été développées : une méthode de simulation de données spectrales, une méthode KNN-PLS de traitement des données massives (parSketch-PLS) et une méthode PLS robuste pour la régression (Roboost-PLSR). Ces trois contributions principales sont discutées dans les paragraphes ci-dessous. Ces travaux de thèse ont mené à neuf publications dont sept sont associées

à la thèse (numérotées de 1 à 7). Dans la suite de cette thèse les citations associées à ces articles scientifiques seront en orange, afin de faciliter la lecture.

1. Matthieu Lesnoff, Maxime Metz, and Jean-Michel Roger. Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data. Journal of Chemometrics, 34(5) :e3209, 2020. ISSN 1099-128X. doi: 10.1002/cem.3209. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.3209. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.3209
2. Maxime Metz, Alessandra Biancolillo, Matthieu Lesnoff, and Jean-Michel Roger. A note on spectral data simulation. Chemometrics and Intelligent Laboratory Systems, 200 :103979, May 2020. ISSN 0169-7439. doi: 10.1016/j.chemolab.2020.103979. URL http://www.sciencedirect.com/science/article/pii/S0169743919306720
3. Maxime Metz, Matthieu Lesnoff, Florent Abdelghafour, Reza Akbarinia, Florent Masseglia, and Jean-Michel Roger. A "big-data" algorithm for KNN-PLS. Chemometrics and Intelligent Laboratory Systems, 203 :104076, August 2020. ISSN 0169-7439. doi: 10.1016/j.chemolab.2020.104076. URL http://www.sciencedirect.com/science/article/pii/S0169743920301908
4. Maxime Metz, Florent Abdelghafour, Jean-Michel Roger, and Matthieu Lesnoff. A novel robust pls regression method inspired from boosting principles : Roboost-plsr. Analytica Chimica Acta, 1179 :338823, 2021. ISSN 0003-2670. doi: https://doi.org/10.1016/j.aca.2021.338823. URL https://www.sciencedirect.com/science/article/pii/S0003267021006498
5. Maxime Ryckewaert, Maxime Metz, Daphné Héran, Pierre George, Bruno Grèzes-Besset, Reza Akbarinia, Jean-Michel Roger, and Ryad Bendoula. Massive spectral data analysis for plant breeding using parsketch-plsda method : Discrimination of sunflower genotypes. Biosystems Engineering, 210 :69–77, 2021. ISSN 1537-5110. doi: https://doi.org/10.1016/j.biosystemseng.2021.08.005. URL https://www.sciencedirect.com/science/article/pii/S1537511021001896
6. Maxime Metz, Maxime Ryckewaert, Silvia Mas Garcia, Ryad Bendoula, Matthieu Lesnoff, and Jean-Michel Roger. Roboost-pls2-r : An extension of roboost-plsr method for multi-responses. Chemometrics and Intelligent Laboratory Systems, XXXX(under revision)
7. Aldrig Courand, Maxime Metz, Daphné Héran, Carolen Feilhes, Fanny Prezman, Eric Serrano, Ryad Bendoula, and Maxime Ryckewaert. Evaluation of a robust

regression method (roboost-plsr) to predict biochemical variables for agronomic applications : case study of grape berry maturity monitoring. Chemometrics and Intelligent Laboratory Systems, XXXX(under revision)

Antoine Laborde, Benoît Jaillais, Jean-Michel Roger, Maxime Metz, Delphine Jouan-Rimbaud Bouveresse, Luc Eveleigh, and Christophe Cordella. Subpixel detection of peanut in wheat flour using a matched subspace detector algorithm and near-infrared hyperspectral imaging. Talanta, 216 :120993, August 2020. ISSN 0039-9140. doi: 10.1016/j.talanta.2020.120993. URL http://www.sciencedirect.com/science/article/pii/S0039914020302848

Julien Petit, Nassim Ait-Mouheb, Sílvia Mas García, Maxime Metz, Bruno Molle, and Ryad Bendoula. Potential of visible/near infrared spectroscopy coupled with chemometric methods for discriminating and estimating the thickness of clogging in drip-irrigation. Biosystems Engineering, 209 :246–255, 2021. ISSN 1537-5110. doi: https://doi.org/10.1016/j.biosystemseng.2021.07.013. URL https://www.sciencedirect.com/science/article/pii/S1537511021001719

### 1.4.1 Méthode parSketch-PLS

parSketch-PLS est une méthode développée pour traiter les données massives en chimiométrie. Cette approche a été évaluée dans la thèse et à mené à une publication : [53]. Elle se base sur la stratégie KNN-PLS, c'est-à-dire qu'un voisinage d'un point est sélectionné puis un modèle PLS est calibré sur ce voisinage pour prédire cet individu. Comme expliqué dans le paragraphe 1.2.3, la limite de la stratégie KNN-PLS est que l'algorithme force-brute utilisé habituellement ne permet pas de réaliser des recherches de plus proches voisins lorsque le nombre d'individus qui constitue la base de données est très grand.

Dans parSketch-PLS, l'algorithme force-brute est remplacé par parSketch [60] afin de lever cette limitation. L'objectif de ce travail était de s'intéresser au potentiel de parSketch-PLS ainsi qu'à ses capacités prédictives. En effet, parSketch est une méthode approximative, ce qui signifie que l'estimation de la distance entre l'individu à prédire et les individus de la base de données peut être faussée.

L'idée de ces travaux est donc de comprendre cette méthode et de l'intégrer dans une stratégie KNN-PLS, afin de permettre le traitement des données massives en chimiométrie. Pour cela, deux études ont été menées, une première étude visant à mettre en avant le potentiel des approches de recherches de voisins pour les données massives [53] , la deuxième montrant une application de cette approche dans le cadre

du phénotypage [55] . Dans cette thèse, il a été constaté que parSketch-PLS est une approche permettant de lever le verrou associé au temps de calcul du traitement des données massives. Cependant, cette méthode ne permet pas d'atteindre d'aussi bonnes capacités de prédiction que les approches KNN-PLS classiques car les voisins renvoyés par parSketch ne sont pas toujours pertinents.

### 1.4.2 Méthode RoBoost-PLSR

Pour évaluer si les voisins renvoyés par parSketch étaient pertinents, les méthodes PLS robustes ont été utilisées. Cependant, les méthodes robustes existantes dans la littérature ne sont pas nécessairement les plus pertinentes pour traiter des problématiques associées au données spectrales. Pour cette raison, une nouvelle méthode robuste intitulée RoBoost-PLSR a été développée dans cette thèse et évaluée dans [54] . Cette méthode propose de définir des pondérations des individus spécifiques à chaque variable latente. Cette approche a été développée et étendue à la prédiction de réponses multiples (PLS2) dans [56] . Pour finir, la méthode RoBoost-PLSR a été appliquée dans le cas de la prédiction de taux de sucre dans les baies de raisin [57] .

### 1.4.3 Simulation de données

Afin de faciliter l'étude des différentes méthodes utilisées dans cette thèse, un cadre de simulation permettant de produire des données où les sources de variabilités sont connues, maîtrisées et quantifiées a été proposé. Pour cela, une approche de simulation des données spectrales a été décrite dans [52] . L'hypothèse principale dans cette approche est que les données spectrales peuvent se résumer à travers deux sous-espaces, l'espace nuisible et l'espace utile. Cette idée a été développée dans [61]. Cette approche permet de simuler un grand nombre de situations qui peuvent représenter des problématiques réelles et ce, grâce à un cadre méthodologique relativement simple. Dans le cadre de la thèse, cette méthodologie a permis d'appréhender différentes notions associées à la robustesse développée dans [54; 58].

## 1.5 Plan de la thèse

Suivant ce chapitre 1 d'introduction, quatre autres chapitres seront développés dans cette thèse.
Le chapitre 2 étudie la première question scientifique de cette thèse, à savoir "Comment mettre à profit des paradigmes du “big-data” pour améliorer les modèles PLS locaux actuels ?". Il est divisé en quatre sections principales. Dans la section 2.1, les motivations

des travaux associés à l'indexation de données spectrales sont mises en avant. Ensuite, dans la section 2.2, la stratégie d'indexation est décrite puis des méthodes simples d'indexation sont utilisées à titre d'exemple. Dans la section 2.3, la méthode parSketch-PLS développée durant la thèse est discutée. Une description détaillée de la méthode parSketch est également fournie puis des résumés des articles produits et associés à cette thèse sont présentés. Pour finir, les perspectives de parSketch-PLS sont discutées en section 2.4.

Dans le chapitre 3, la deuxième question scientifique de la thèse "Comment estimer la pertinence d'un individu par rapport à un modèle PLS ?" est étudiée. Comme pour le chapitre 2, ce chapitre est divisé en quatre sections. Dans la section 3.1, les motivations des travaux associés à la pertinence d'un point dans un modèle sont discutées. Puis, les méthodes de régressions robustes existant dans la littérature sont discutées dans la section 3.2. La section 3.3 présente une nouvelle méthode de régression PLS robuste développée durant cette thèse. Enfin, les perspectives de ce chapitre sont discutées en section 3.4.

Le chapitre 4 étudie la troisième question scientifique de la thèse "Comment associer les paradigmes de la chimiométrie et du big-data ?". Il est aussi divisé en quatre sections. Les motivations des travaux associés à la combinaison des outils d'indexation et de la chimiométrie sont discutées dans la section 4.1. Il est ensuite proposé dans la section 4.2 d'orienter l'indexation de manière à obtenir des voisinages plus pertinents pour la calibration de modèle PLS. La section 4.3 propose de combiner les méthodes de robustesse et d'indexation. Enfin, les perspectives de ce chapitre sont discutées en section 4.3.4.

Pour finir, le chapitre 5 conclut sur les travaux de thèse puis discute des perspectives de ces travaux.

# Bibliographie

[1] CEMA (European Agricultural Machinery). Digital farming : what does it really mean ? and what is the vision of europe's farm machinery industry for digital farming ? 2017. URL https://www.cema-agri.org/images/publications/position-papers/CEMA_Digital_Farming_-_Agriculture_4.0__13_02_2017_0.pdf.

[2] Stéphane Mallat. L'apprentissage face à la malédiction de la grande dimension, 2017. URL https://www.college-de-france.fr/site/stephane-mallat/course-2017-2018.htm.

[3] Véronique Bellon-Maurel. Application de la spectroscopie proche infrarouge au contrôle en ligne de la qualité des fruits et légumes. These de doctorat, Toulouse, INPT, January 1992. URL https://www.theses.fr/1992INPT011G.

[4] V Geraudie, J M Roger, J L Ferrandis, J M Gialis, P Barbe, V Bellon Maurel, and R Pellenc. A revolutionary device for predicting grape maturity based on NIR spectrometry. page 9, 2009.

[5] Véronique Bellon-Maurel, E. Fernandez-Ahumada, B. Palagos, J. M. Roger, and A. Mcbratney. Prediction of soil attributes by NIR spectroscopy. A critical review of chemometric indicators commonly used for assessing the quality of the prediction. Trends in Analytical Chemistry, 29(9) :1073, 2010. doi: 10.1016/j.trac.2010.05.006. URL https://hal.inrae.fr/hal-02593648.

[6] Jean-Michel Roger. Développements chimiométriques pour améliorer la robustesse des mesures spectrométriques appliquées aux agro-procédés. page 65, 2005.

[7] Maxime Ryckewaert, Nathalie Gorretta, Fabienne Henriot, Federico Marini, and Jean-Michel Roger. Reduction of repeatability error for analysis of variance-Simultaneous Component Analysis (REP-ASCA) : Application to NIR spectroscopy on coffee sample. Analytica Chimica Acta, 1101 :23–31, March 2020. ISSN 0003-2670. doi: 10.1016/j.aca.2019.12.024. URL https://www.sciencedirect.com/science/article/pii/S0003267019314606.

[8] Aakash Chawade, Joost van Ham, Hanna Blomquist, Oscar Bagge, Erik Alexandersson, and Rodomiro Ortiz. High-throughput field-phenotyping tools for plant breeding and precision agriculture. Agronomy, 9(5) :258, 2019. Publisher : Multidisciplinary Digital Publishing Institute.

[9] Nadia Shakoor, Scott Lee, and Todd C. Mockler. High throughput phenotyping to accelerate crop breeding and monitoring of diseases in the field. Current opinion in plant biology, 38 :184–192, 2017. Publisher : Elsevier.

[10] Véronique Bellon-Maurel, Pascal Neveu, and Alexandre Termier. Le Big Data en agriculture. Annales des Mines, page 5, 2018. URL http://annales.org/enjeux-numeriques/2018/en-2018-02/EN-2018-06-16.pdf.

[11] Kristian Hovde Liland, Tormod Naes, and Ulf G. Indahl. ROSA-a fast extension of partial least squares regression for multiblock data analysis : ROSA - a fast extension of PLSR for Multiblock Data Analysis. Journal of Chemometrics, 30 (11) :651–662, November 2016. ISSN 08869383. doi: 10.1002/cem.2824. URL https://onlinelibrary.wiley.com/doi/10.1002/cem.2824.

[12] Alessandra Biancolillo, Ingrid Måge, and Tormod Næs. Combining SO-PLS and linear discriminant analysis for multi-block classification. Chemometrics and Intelligent Laboratory Systems, 141 :58–67, February 2015. ISSN 01697439. doi: 10.1016/j.chemolab.2014.12.001. URL https://linkinghub.elsevier.com/retrieve/pii/S0169743914002470.

[13] Jean-Michel Roger, Alessandra Biancolillo, and Federico Marini. Sequential preprocessing through ORThogonalization (SPORT) and its application to near infrared spectroscopy. Chemometrics and Intelligent Laboratory Systems, 199 : 103975, April 2020. ISSN 01697439. doi: 10.1016/j.chemolab.2020.103975. URL https://linkinghub.elsevier.com/retrieve/pii/S0169743919308135.

[14] Amir Gandomi and Murtaza Haider. Beyond the hype : Big data concepts, methods, and analytics. International Journal of Information Management, 35(2) :137–144, April 2015. ISSN 02684012. doi: 10.1016/j.ijinfomgt.2014.10.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0268401214001066.

[15] Stéphane Tufféry. Big data, machine learning et apprentissage profond. 2019. ISBN 978-2-7108-1188-6. OCLC : 1135947642.

[16] Alexandra L'Heureux, Katarina Grolinger, Hany F. Elyamany, and Miriam A. M. Capretz. Machine Learning With Big Data : Challenges and Approaches. IEEE Access, 5 :7776–7797, 2017. ISSN 2169-3536. doi: 10.1109/ACCESS.2017.2696365. URL https://ieeexplore.ieee.org/document/7906512/.

[17] Thomas Lumley. Bounded Memory Linear and Generalized Linear Models [R package biglmm version 0.9-2], May 2020. URL https://CRAN.R-project.org/package=biglmm. Publisher : Comprehensive R Archive Network (CRAN).

[18] Jeffrey Shafer, Scott Rixner, and Alan L Cox. The Hadoop distributed filesystem : Balancing portability and performance. In 2010 IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS), pages 122–133, White Plains, NY, March 2010. IEEE. ISBN 978-1-4244-6023-6. doi: 10.1109/ISPASS.2010.5452045. URL http://ieeexplore.ieee.org/document/5452045/.

[19] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark : Cluster Computing with Working Sets. page 7, 2010.

[20] R. A. Viscarra Rossel, T. Behrens, E. Ben-Dor, D. J. Brown, J. A. M. Demattê, K. D. Shepherd, Z. Shi, B. Stenberg, A. Stevens, V. Adamchuk, H. Aïchi, B. G. Barthès, H. M. Bartholomeus, A. D. Bayer, M. Bernoux, K. Böttcher, L. Brodský, C. W. Du, A. Chappell, Y. Fouad, V. Genot, C. Gomez, S. Grunwald, A. Gubler, C. Guerrero, C. B. Hedley, M. Knadel, H. J. M. Morrás, M. Nocita, L. Ramirez-Lopez, P. Roudier, E. M. Rufasto Campos, P. Sanborn, V. M. Sellitto, K. A. Sudduth, B. G. Rawlins, C. Walter, L. A. Winowiecki, S. Y. Hong, and W. Ji. A global spectral library to characterize the world's soil. Earth-Science Reviews, 155 :198–230, April 2016. ISSN 0012-8252. doi: 10.1016/j.earscirev.2016.01.012. URL https://www.sciencedirect.com/science/article/pii/S0012825216300113.

[21] A. Orgiazzi, C. Ballabio, P. Panagos, A. Jones, and O. Fernández-Ugalde. LUCAS Soil, the largest expandable soil dataset for Europe : a review. European Journal of Soil Science, 69(1) :140–153, 2018. ISSN 1365-2389. doi: 10.1111/ejss.12499. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/ejss.12499. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1111/ejss.12499.

[22] C Ariza-Nieto, OL Mayorga, B Mojica, D Parra, and G Afanador-Tellez. Use of LOCAL algorithm with near infrared spectroscopy in forage resources for grazing systems in Colombia. Journal of Near Infrared Spectroscopy, 26(1) :44–52, February 2018. ISSN 0967-0335. doi: 10.1177/0967033517746900. URL https://doi.org/10.1177/0967033517746900.

[23] Donato Andueza, Fabienne Picard, Dominique Dozias, and Jocelyne Aufrère. Fecal Near-Infrared Reflectance Spectroscopy Prediction of the Feed Value of Temperate Forages for Ruminants and Some Parameters of the Chemical Composition of Feces : Efficiency of Four Calibration Strategies :. Applied Spectroscopy, June 2017. doi: 10.1177/0003702817712740. URL https://journals.sagepub.com/doi/10.1177/0003702817712740.

[24] Benoit Igne and Charles R. Hurburgh. Local chemometrics for samples and

variables : optimizing calibration and standardization processes. Journal of Chemometrics, 24(2) :75–86, 2010. ISSN 1099-128X. doi: 10.1002/cem.1274. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.1274.

[25] I. I. F. E. Barton, J. S. Shenk, M. O. Westerhaus, and D. B. Funk. The Development of near Infrared Wheat Quality Models by Locally Weighted Regressions :. Journal of Near Infrared Spectroscopy, February 2017. doi: 10.1255/jnirs.280. URL https://journals.sagepub.com/doi/10.1255/jnirs.280.

[26] G. Sinnaeve, P. Dardenne, and R. Agneessens. Global or Local ? A Choice for NIR Calibrations in Analyses of Forage Quality. Journal of Near Infrared Spectroscopy, 2(3) :163–175, June 1994. ISSN 0967-0335. doi: 10.1255/jnirs.43. URL https://doi.org/10.1255/jnirs.43.

[27] D. Pérez-Marín, A. Garrido-Varo, and J. E. Guerrero. Implementation of LOCAL algorithm with near-infrared spectroscopy for compliance assurance in compound feedingstuffs. Applied Spectroscopy, 59(1) :69–77, January 2005. ISSN 0003-7028. doi: 10.1366/0003702052940585.

[28] D. Pérez-Marín, A. Garrido-Varo, and J. E. Guerrero. Non-linear regression methods in NIRS quantitative analysis. Talanta, 72(1) :28–42, April 2007. ISSN 1873-3573. doi: 10.1016/j.talanta.2006.10.036.

[29] Pierre Dardenne, George Sinnaeve, and Vincent Baeten. Multivariate Calibration and Chemometrics for near Infrared Spectroscopy : Which Method ? Journal of Near Infrared Spectroscopy, 8(4) :229–237, October 2000. ISSN 0967-0335. doi: 10.1255/jnirs.283. URL https://doi.org/10.1255/jnirs.283.

[30] Juan Antonio Fernández Pierna and Pierre Dardenne. Soil parameter quantification by NIRS as a Chemometric challenge at 'Chimiométrie 2006'. Chemometrics and Intelligent Laboratory Systems, 91(1) :94–98, March 2008. ISSN 0169-7439. doi: 10.1016/j.chemolab.2007.06.007. URL http://www.sciencedirect.com/science/article/pii/S0169743907001190.

[31] E. Fernández-Ahumada, T. Fearn, A. Gómez, P. Vallesquino, J. E. Guerrero, D. Pérez-Marín, and A. Garrido-Varo. Reducing NIR prediction errors with nonlinear methods and large populations of intact compound feedstuffs. Measurement Science and Technology, 19(8) :085601, 2008. ISSN 1361-6501. URL https://www.academia.edu/4325266/Reducing_NIR_prediction_errors_with_nonlinear_methods_and_large_populations_of_intact_compound_feedstuffs.

[32] Paolo Berzaghi, John S. Shenk, and Mark O. Westerhaus. LOCAL Prediction with near Infrared Multi-Product Databases. Journal of Near Infrared Spectroscopy, 8 (1) :1–9, January 2000. ISSN 0967-0335. doi: 10.1255/jnirs.258. URL https://doi.org/10.1255/jnirs.258.

[33] F. Gogé, R. Joffre, C. Jolivet, I. Ross, and L. Ranjard. Optimization criteria in sample selection step of local regression for quantitative analysis of large soil NIRS database. Chemometrics and Intelligent Laboratory Systems, 110(1) :168–176, January 2012. ISSN 0169-7439. doi: 10.1016/j.chemolab.2011.11.003. URL http://www.sciencedirect.com/science/article/pii/S0169743911002310.

[34] Michel Rabenarivo, Lydie Chapuis-Lardy, Didier Brunet, Jean-Luc Chotte, Lilia Rabeharisoa, and Bernard G. Barthès. Comparing near and Mid-Infrared Reflectance Spectroscopy for Determining Properties of Malagasy Soils, Using Global or LOCAL Calibration. Journal of Near Infrared Spectroscopy, 21(6) :495–509, December 2013. ISSN 0967-0335. doi: 10.1255/jnirs.1080. URL https://doi.org/10.1255/jnirs.1080. Publisher : SAGE Publications Ltd STM.

[35] C Ariza-Nieto, OL Mayorga, B Mojica, D Parra, and G Afanador-Tellez. Use of local algorithm with near infrared spectroscopy in forage resources for grazing systems in colombia. Journal of Near Infrared Spectroscopy, 26(1) :44–52, 2018. doi: 10.1177/0967033517746900. URL https://doi.org/10.1177/0967033517746900.

[36] H. Tran, P. Salgado, E. Tillard, P. Dardenne, X.T. Nguyen, and P. Lecomte. "global" and "local" predictions of dairy diet nutritional quality using near infrared reflectance spectroscopy. Journal of Dairy Science, 93(10) :4961–4975, 2010. ISSN 0022-0302. doi: https://doi.org/10.3168/jds.2008-1893. URL https://www.sciencedirect.com/science/article/pii/S002203021000528X.

[37] William S. Cleveland and Susan J. Devlin. Locally Weighted Regression : An Approach to Regression Analysis by Local Fitting. Journal of the American Statistical Association, 83(403) :596–610, September 1988. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.1988.10478639. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1988.10478639.

[38] Tormod. Naes, Tomas. Isaksson, and Bruce. Kowalski. Locally weighted regression and scatter correction for near-infrared reflectance data. Analytical Chemistry, 62 (7) :664–673, April 1990. ISSN 0003-2700, 1520-6882. doi: 10.1021/ac00206a003. URL https://pubs.acs.org/doi/abs/10.1021/ac00206a003.

[39] Tormod Næs and Tomas Isaksson. Locally Weighted Regression in Diffuse Near-Infrared Transmittance Spectroscopy. Applied Spectroscopy, 46(1) :34–43, January 1992. URL https://www.osapublishing.org/as/abstract.cfm?uri=as-46-1-34.

[40] John S. Shenk, Mark O. Westerhaus, and Paolo Berzaghi. Investigation of a local calibration procedure for near infrared instruments. Journal of Near Infrared Spectroscopy, 5(4) :223–232, 1997. doi: 10.1255/jnirs.115. URL https://doi.org/10.1255/jnirs.115.

[41] Stefan Schaal, Christopher G. Atkeson, and Sethu Vijayakumar. Scalable Techniques from Nonparametric Statistics for Real Time Robot Learning. Applied Intelligence, 17(1) :49–60, July 2002. ISSN 1573-7497. doi: 10.1023/A:1015727715131. URL https://doi.org/10.1023/A:1015727715131.

[42] E. Sicard and R. Sabatier. Theoretical framework for local pls1 regression, and application to a rainfall data set. Computational Statistics & Data Analysis, 51 (2) :1393–1410, 2006. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2006.05.002. URL https://www.sciencedirect.com/science/article/pii/S0167947306001368.

[43] Sanghong Kim, Manabu Kano, Hiroshi Nakagawa, and Shinji Hasebe. Estimation of active pharmaceutical ingredients content using locally weighted partial least squares and statistical wavelength selection. International Journal of Pharmaceutics, 421 (2) :269–274, 2011. ISSN 0378-5173. doi: https://doi.org/10.1016/j.ijpharm.2011.10.007. URL https://www.sciencedirect.com/science/article/pii/S0378517311009021.

[44] Marta Bevilacqua and Federico Marini. Local classification : Locally weighted–partial least squares-discriminant analysis (LW–PLS-DA). Analytica Chimica Acta, 838 :20–30, August 2014. ISSN 0003-2670. doi: 10.1016/j.aca.2014.05.057. URL http://www.sciencedirect.com/science/article/pii/S0003267014007557.

[45] Koji Hazama and Manabu Kano. Covariance-based locally weighted partial least squares for high-performance adaptive modeling. Chemometrics and Intelligent Laboratory Systems, 146 :55–62, August 2015. ISSN 0169-7439. doi: 10.1016/j.chemolab.2015.05.007. URL http://www.sciencedirect.com/science/article/pii/S0169743915001203.

[46] Guanghui Shen, Matthieu Lesnoff, Vincent Baeten, Pierre Dardenne, Fabrice Davrieux, Hernan Ceballos, John Belalcazar, Dominique Dufour, Zengling Yang, Lujia Han, and Juan Antonio Fernández Pierna. Local partial least squares based on global PLS scores. Journal of Chemometrics, 33(5) :e3117, 2019. ISSN 1099-128X. doi: 10.1002/cem.3117. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.3117.

[47] Tom Fearn and Anthony M.C. Davies. Locally-Biased Regression. Journal of Near Infrared Spectroscopy, 11(6) :467–478, December 2003. ISSN 0967-0335. doi: 10.1255/jnirs.397. URL https://doi.org/10.1255/jnirs.397.

[48] Xinmin Zhang, Manabu Kano, and Zhihuan Song. Optimal Weighting Distance-Based Similarity for Locally Weighted PLS Modeling. Industrial & Engineering Chemistry Research, 59(25) :11552–11558, June 2020. ISSN 0888-5885, 1520-5045. doi: 10.1021/acs.iecr.9b06847. URL https://pubs.acs.org/doi/10.1021/acs.iecr.9b06847.

[49] Matthieu Lesnoff, Maxime Metz, and Jean-Michel Roger. Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data. Journal of Chemometrics, 34(5), May 2020. ISSN 0886-9383, 1099-128X. doi: 10.1002/cem.3209. URL https://onlinelibrary.wiley.com/doi/10.1002/cem.3209.

[50] Peter Filzmoser, Sven Serneels, Ricardo Maronna, and Christophe Croux. Robust multivariate methods in Chemometrics. arXiv :2006.01617 [stat], pages 393–430, 2020. doi: 10.1016/B978-0-12-409547-2.14642-6. arXiv : 2006.01617.

[51] Matthieu Lesnoff, Maxime Metz, and Jean-Michel Roger. Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data. Journal of Chemometrics, 34(5) :e3209, 2020. ISSN 1099-128X. doi: 10.1002/cem.3209. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.3209. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.3209.

[52] Maxime Metz, Alessandra Biancolillo, Matthieu Lesnoff, and Jean-Michel Roger. A note on spectral data simulation. Chemometrics and Intelligent Laboratory Systems, 200 :103979, May 2020. ISSN 0169-7439. doi: 10.1016/j.chemolab.2020.103979. URL http://www.sciencedirect.com/science/article/pii/S0169743919306720.

[53] Maxime Metz, Matthieu Lesnoff, Florent Abdelghafour, Reza Akbarinia, Florent Masseglia, and Jean-Michel Roger. A “big-data” algorithm for KNN-PLS.

Chemometrics and Intelligent Laboratory Systems, 203 :104076, August 2020. ISSN 0169-7439. doi: 10.1016/j.chemolab.2020.104076. URL http://www.sciencedirect.com/science/article/pii/S0169743920301908.

[54] Maxime Metz, Florent Abdelghafour, Jean-Michel Roger, and Matthieu Lesnoff. A novel robust pls regression method inspired from boosting principles : Roboost-plsr. Analytica Chimica Acta, 1179 :338823, 2021. ISSN 0003-2670. doi: https://doi.org/10.1016/j.aca.2021.338823. URL https://www.sciencedirect.com/science/article/pii/S0003267021006498.

[55] Maxime Ryckewaert, Maxime Metz, Daphné Héran, Pierre George, Bruno Grèzes-Besset, Reza Akbarinia, Jean-Michel Roger, and Ryad Bendoula. Massive spectral data analysis for plant breeding using parsketch-plsda method : Discrimination of sunflower genotypes. Biosystems Engineering, 210 :69–77, 2021. ISSN 1537-5110. doi: https://doi.org/10.1016/j.biosystemseng.2021.08.005. URL https://www.sciencedirect.com/science/article/pii/S1537511021001896.

[56] Maxime Metz, Maxime Ryckewaert, Silvia Mas Garcia, Ryad Bendoula, Matthieu Lesnoff, and Jean-Michel Roger. Roboost-pls2-r : An extension of roboost-plsr method for multi-responses. Chemometrics and Intelligent Laboratory Systems, XXXX(under revision).

[57] Aldrig Courand, Maxime Metz, Daphné Héran, Carolen Feilhes, Fanny Prezman, Eric Serrano, Ryad Bendoula, and Maxime Ryckewaert. Evaluation of a robust regression method (roboost-plsr) to predict biochemical variables for agronomic applications : case study of grape berry maturity monitoring. Chemometrics and Intelligent Laboratory Systems, XXXX(under revision).

[58] Antoine Laborde, Benoît Jaillais, Jean-Michel Roger, Maxime Metz, Delphine Jouan-Rimbaud Bouveresse, Luc Eveleigh, and Christophe Cordella. Subpixel detection of peanut in wheat flour using a matched subspace detector algorithm and near-infrared hyperspectral imaging. Talanta, 216 :120993, August 2020. ISSN 0039-9140. doi: 10.1016/j.talanta.2020.120993. URL http://www.sciencedirect.com/science/article/pii/S0039914020302848.

[59] Julien Petit, Nassim Ait-Mouheb, Sílvia Mas García, Maxime Metz, Bruno Molle, and Ryad Bendoula. Potential of visible/near infrared spectroscopy coupled with chemometric methods for discriminating and estimating the thickness of clogging in drip-irrigation. Biosystems Engineering, 209 :246–255, 2021. ISSN 1537-5110.

doi: https://doi.org/10.1016/j.biosystemseng.2021.07.013. URL https://www.sciencedirect.com/science/article/pii/S1537511021001719.

[60] Oleksandra Levchenko, Djamel-Edine Yagoubi, Reza Akbarinia, Florent Masseglia, Boyan Kolev, and Dennis Shasha. Spark-parSketch : A Massively Distributed Indexing of Time Series Datasets. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pages 1951–1954, Torino Italy, October 2018. ACM. ISBN 978-1-4503-6014-2. doi: 10.1145/3269206.3269226. URL https://dl.acm.org/doi/10.1145/3269206.3269226.

[61] Jean-Michel Roger and Jean-Claude Boulet. A review of orthogonal projections for calibration. Journal of Chemometrics, 32(9) :e3045, September 2018. ISSN 08869383. doi: 10.1002/cem.3045. URL http://doi.wiley.com/10.1002/cem.3045.

# parSketch-PLS : une nouvelle méthode prédictive pour les données massives

*Chapitre* 2

## Sommaire



## 2.1 Motivation et vue d'ensemble des travaux

Dans ce chapitre, la méthode parSketch [1] issue du big-data, est étudiée dans le cadre de la stratégie KNN-PLS en chimiométrie. Comme discuté précédemment, les méthodes de type KNN-PLS utilisent pour la plupart l'algorithme force-brute. Une solution proposée dans cette thèse est de remplacer l'algorithme force-brute par un algorithme de recherche rapide de voisinages. Pour cela, des méthodes d'indexation massivement parallélisables sont étudiées, permettant une recherche rapide des plus proches voisins d'un individu à prédire. parSketch [1] fait partie de ces méthodes. parSketch est éprouvée pour son efficience calculatoire [1], notamment par rapport à des méthodes telles que force-brute. Cependant, elle n'avait pas été étudiée pour la calibration de modèles prédictifs mais pour son coût calculatoire.

Dans le cadre des travaux de thèse, cette méthode a été combinée à la PLS afin d'évaluer le potentiel prédictif d'une nouvelle méthode locale pour le traitement de données massives. Dans ce contexte, parSketch-PLS a été développée pour répondre à des problèmes de régression ou de classification. Cette approche a été étudiée d'un

point de vue méthodologique. Plus précisément, l'étude a porté sur son comportement dans la recherche de voisins en fonction des paramètres de parSketch appliqués [2] . Elle a aussi été mise en oeuvre dans un cas pratique [3] , pour un problème de classification. Dans ces deux articles, les temps et les coûts calculatoires ne sont pas discutés car ils ont déjà été étudiés, ce sont les performances de prédiction et la pertinence des modèles établis qui sont évaluées.
Ce chapitre se donne pour objectif d'accompagner le lecteur dans la compréhension des deux travaux [2; 3] associés à parSketch. Ainsi, les méthodes d'indexation sont introduites afin de positionner parSketch et de mieux retranscrire ses propriétés. Par la suite, les principes et le fonctionnement de parSketch sont décrits en détails. Enfin, les deux articles correspondants sont discutés pour mettre en évidence les perspectives de cette méthode.

## 2.2 Les méthodes d'indexation

Les méthodes d'indexation ont pour objectif d'organiser les données afin que la procédure de recherche des plus proches voisins soit la plus rapide possible. Pour cela, un index est donc créé pour regrouper en sous-ensemble les individus de la base de données. Un index est donc la structure chargée d'ordonner et de trier les données afin de pouvoir retrouver plus rapidement les plus proches voisins. Par la suite, au lieu de réaliser une recherche de plus proches voisins sur l'ensemble de la base de données, il est possible d'effectuer la recherche sur un sous-ensemble. Ceci permet donc de réduire considérablement les temps de calculs associés à la recherche des plus proches voisins.
L'indexation de base de données, bien que très intéressante pour la recherche de plus proches voisins, peut s'avérer coûteuse en temps de calcul en fonction de son usage. Contrairement à l'algorithme force-brute, les méthodes d'indexation peuvent se résumer en deux étapes, une étape de construction de l'index et une étape de recherche de voisins. La complexité algorithmique de l'étape de construction de l'index est plus grande que celle de force-brute. Néanmoins, quand l'index est construit un gain de temps considérable pour la recherche des plus proches voisins peut être obtenu. Lorsque l'on souhaite utiliser un index, il est nécessaire d'évaluer l'usage de cet index. Si un grand nombre de recherches des plus proches voisins est réalisé, il est très utile de construire un index.

Au vu de l'intérêt pratique d'indexer des bases de données, un grand nombre de méthodes d'indexation ont été développées. Ces méthodes peuvent se regrouper en deux grands types de familles : les méthodes arborescentes et les méthodes de hachage.

Les méthodes arborescentes permettent de construire un arbre, *i.e.* une structure hiérarchique qui regroupe les individus de la base de données dans des feuilles. La recherche des plus proches voisins d'un individu consiste à parcourir l'arbre et à déterminer la feuille qui contient les voisins les plus proches de l'individu considéré. Parmi les méthodes arborescentes, on retrouve des méthodes très populaires comme la recherche par arbre-B (B-Tree) [4], ou par arbre-R [5].

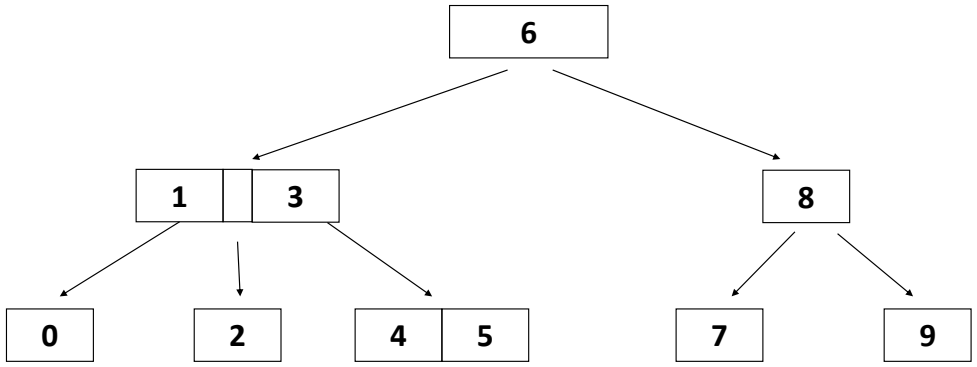



FIGURE 2.1 – Schéma illustrant un B-tree

La figure 2.1 présente un exemple d'indexation par arbre-B. Chaque feuille de l'arbre (les valeurs $\{0\,;2\,;4\,;5\,;7\,;9\}$) correspond à une valeur d'un individu de la base de données. Les autres valeurs présentes dans l'arbre sont appelées noeuds. Pour accéder à une de ces feuilles, il faut parcourir l'arbre c'est-à-dire tester si la valeur d'un individu est supérieure, inférieure, ou entre la/les valeurs présentes dans les noeuds. Si la valeur de l'individu qui parcourt l'arbre est supérieure à la valeur du noeud, il se positionnera dans le prochain noeud à droite. On peut comprendre que dans ce cas-ci, pour retrouver les plus proches voisins d'un échantillon, il suffit de parcourir l'arbre. Ceci permet donc en parcourant deux noeuds (en réalisant deux ou trois comparaisons) de retrouver les plus proches voisins plutôt que devoir réaliser dans cet exemple six comparaisons.

Les méthodes de hachage quant à elles regroupent directement les individus en paquets à l'aide d'une fonction de hachage. Chacun des paquets d'individus se voit attribuer une clé. Pour pouvoir rechercher les plus proches voisins d'un individu il faudra donc uniquement calculer la clé du nouvel individu pour obtenir ses plus proches voisins.

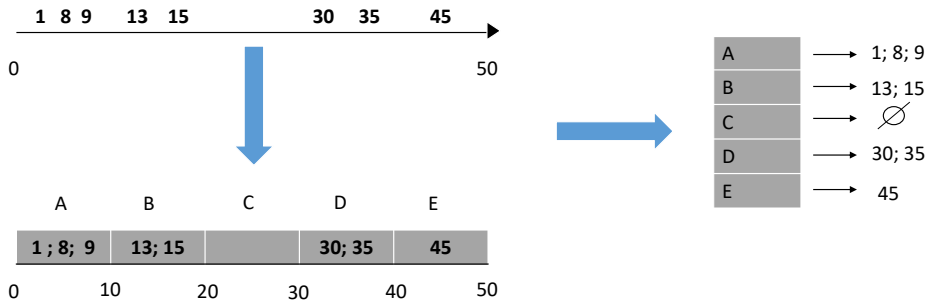



FIGURE 2.2 – Schéma illustrant un hachage par grille 1D

La figure 2.2 présente un exemple d'indexation par hachage par grille à une dimension (1D). Chaque valeur présente sur l'axe en haut à gauche du schéma, allant de 0 à 50, correspond à un individu de la base de données. Pour réaliser des paquets, une grille 1D de cinq segments, nommés $\{A ; B ; C ; D ; E\}$, est construite à partir des données. Chaque lettre correspondant à un segment va donc devenir une clé associée aux valeurs des individus contenus dans ce segment. Par exemple, la clé A est associée aux individus qui possèdent les valeurs $\{1 ; 8 ; 9\}$. Pour réaliser la recherche de voisins dans ce contexte, il faudra obtenir un numéro de cellule dans la grille d'un nouvel individu. Les individus de la base de données possédant le même numéro de cellule que le nouvel individu sont les plus proches voisins.

Ceci permet donc de réaliser une recherche très rapide plutôt que de réaliser huit comparaisons dans cet exemple.

Toutes ces approches sont très efficientes quand le nombre de dimensions (*i.e.* de variables décrivant les individus) dans la base de données est relativement faible ($<$ 5-10 variables typiquement). Lorsque le nombre de variables devient grand, ces approches ne sont plus applicables [6]. Pour répondre à cette problématique, des méthodes spécifiques ont été développées, notamment pour le traitement de séries temporelles [7–11]. La démarche la plus utilisée pour l'indexation des séries temporelles est d'opérer une réduction de dimension permettant d'appliquer par la suite des approches d'indexations classiques.

Ces méthodes s'avèrent pertinentes dans le contexte de la chimiométrie. En effet, tout

comme les données spectrales, les séries temporelles sont de très grandes dimensions et présentent de fortes corrélations entre leurs variables. Ces méthodes ont été développées pour réaliser la recherche de voisins en grandes dimensions mais ne sont pas parrallélisables. Lorsque l'index est construit, ces méthodes permettent un gain de temps considérable par rapport à l'algorithme force-brute. Toutefois, puisque les algorithmes associés à ces méthodes sont uniquement développés pour du calcul centralisé, les temps de calcul de ces approches augmentent fortement avec le nombre d'individus.

Pour passer outre ces limites d'échelles, deux méthodes permettant de distribuer massivement les calculs ont été récemment développées : DpiSAX [12] et parSketch [1]. La méthode DpiSAX combine une réduction de dimension par le biais de la méthode Symbolic Aggregate approXimation (SAX) [13] et une méthode d'indexation arborescente. DpiSAX est inspirée de iSAX [10] et propose un algorithme massivement parallélisable. DpiSAX détermine des groupes de données (feuilles). Les plus proches voisins d'un individu sont donc ceux qui appartiennent à une même feuille.

Cette méthode s'approche fortement des méthodes de clustering et donc serait particulièrement adaptée aux méthodes clustered-PLS [14; 15]. Ces méthodes ont pour objectif de former des groupes d'individus puis de calculer un modèle PLS sur chacun de ces groupes, ceci conduisant à un ensemble de modèles PLS. Par la suite, un nouvel individu est associé à un groupe puis à un modèle.

La méthode parSketch quant à elle combine une réduction de dimension par projection sur vecteurs aléatoires [16] avec une méthode d'indexation par hachage. La particularité de parSketch est qu'elle ne construit pas un index pour la base de données mais un méta-index en créant des grilles 2D calculées sur les vecteurs aléatoires. Cela signifie que les groupes créés par chaque index ne regroupent pas les individus de la base de données pour l'ensemble des grilles mais pour chaque grille. Les plus proches voisins d'un point sont ceux qui ont le plus de cellules en commun (case dans une grille). Cela signifie donc qu'il est peu probable de retrouver le même voisinage pour chaque point contrairement à la méthode DpiSAX. En pratique, parSketch, de par sa conception, est plus adaptée aux méthodes KNN-PLS, car parSketch détermine des voisinages spécifiques à l'individu à prédire et pas prédéfini comme pour DpiSAX. Dans cette thèse, les travaux se sont donc focalisés sur la méthode parSketch et son intérêt.

## 2.3 parSketch-PLS

La méthode parSketch-PLS combine le principe des approches de type KNN-PLS avec la méthode parSketch dédiée à la recherche de voisins. Cette nouvelle méthode permet donc de rechercher rapidement les voisins d'un individu, puis de calibrer un modèle de

régression ou de discrimination à partir de ses voisins.

Dans cette thèse, parSketch-PLS s'est montrée efficace pour définir un voisinage dans un but prédictif [2; 3]. Dans les deux cas précédemment cités, la méthode a été évaluée dans le cadre de classifications. En effet, il est plus simple à l'heure actuelle d'obtenir des données massives annotées en classes que des données destinées à la régression où il faut disposer de mesures de références. Plus de détails sur la méthode parSketch et son application sont donnés dans [2; 3].

### 2.3.1 Présentation de parSketch

La méthode parSketch va maintenant être présentée en détail dans cette section. Deux étapes principales composent parSketch. La première étape est une étape d'indexation de la base de données qui combine une réduction de dimension puis une méthode d'indexation. La deuxième étape quant à elle est une étape de recherche des plus proches voisins. La méthode parSketch possède trois paramètres qui vont influencer sur les voisinages renvoyés : le nombre de vecteurs aléatoires (noté $v$), le nombre de segments (noté $s$) et le nombre minimal de cellules en commun (noté $m$). Le réglage des paramètres de parSketch est fondamental à la fois pour définir un voisinage suffisant pour l'estimation d'un modèle mais aussi pour que le voisinage constitue un sous-ensemble cohérent permettant des performances de prédictions élevées et permettant également des temps de calcul raisonnables. Chaque paramètre et leurs impacts sont discutés dans la suite de cette section.

Les prochaines sous-sections décrivent plus précisément chaque étape de parSketch. De plus, son intérêt méthodologique est discuté en fin de section.

#### Étape 1.1 : réduction de dimension

L'opération de réduction de dimension est basée sur la génération de vecteurs aléatoires suivie de la projection de chaque individu de la base de données sur ces vecteurs. Les vecteurs lignes résultants sont appelés 'sketches'. Les sketches sont donc calculés comme suit :

$$\mathbf{T} = \mathbf{XP} \tag{2.1}$$

Où $\mathbf{P}$ est une matrice de taille $(p,v)$ contenant des valeurs selon une sélection aléatoire. $\mathbf{X}$ et $\mathbf{T}$ les matrices des spectres et des sketches. Cette stratégie de réduction de dimension repose sur le lemme de Johnson-Lindenstrauss [16]. Ce lemme énonce que des individus représentés par un grand nombre de variables peuvent être plongés dans un espace de plus petite dimension avec une distorsion très faible. En d'autres termes,

les individus peuvent être plongés dans un espace de plus petite dimension sans modifier fortement leur ordonnancement.

L'étape de réduction de dimension de parSketch possède différentes propriétés :

— La réduction de dimension par vecteurs aléatoires réalise des combinaisons linéaires aléatoires des variables à partir de $\mathbf{P}$ générée avec n'importe quelles distributions symétriques. Ces combinaisons, malgré leur nature aléatoire, permettent de préserver de façon approchée les distances Euclidiennes entre individus. Cette réduction de dimension peut donc être utilisée pour réaliser une approximation des distances Euclidiennes lors de la recherche des plus proches voisins. Cependant, pour obtenir une approximation de bonne qualité, il est nécessaire de projeter les individus sur un espace de dimension suffisamment grande. Cela signifie donc que plus la dimension des sketches ($v$) est grande plus l'approximation sera bonne [16]. En contrepartie, les temps de calculs seront plus conséquents. La dimension $v$ de $\mathbf{P}$ est donc un paramètre de parSketch qu'il faut ajuster pour obtenir le meilleur compromis entre temps de calcul et qualité d'approximation.

— Cette approche de réduction de dimension se révèle particulièrement intéressante dans le contexte des big-data car la réduction de dimension est réalisée en une seule opération matricielle. En effet, contrairement aux approches usuelles utilisées en chimiométrie, elle ne nécessite pas d'estimer un modèle initial (ACP, PLS, etc) pour opérer une réduction de dimension. C'est un avantage conséquent par rapport à la PLS ou l'ACP, pour lesquelles il est nécessaire de considérer (et donc charger en mémoire) dans un premier temps l'ensemble de la base de données.

### Étape 1.2 : construction des grilles

Une fois la réduction de dimension de la base de données effectuée, l'étape d'indexation peut débuter. La construction des grilles représente l'étape d'indexation à proprement parlé. C'est avec les données compressées (*i.e.* les sketches) qu'est construit l'index. Cette opération est réalisée à l'aide d'une méthode de hachage par grille.

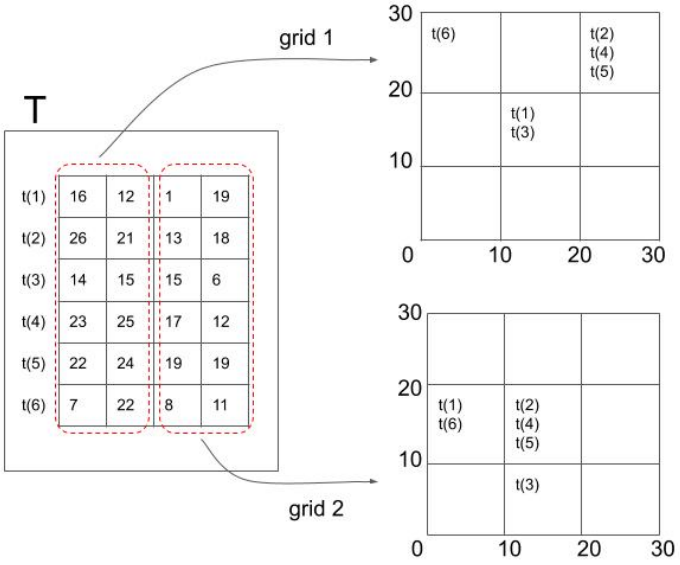



FIGURE 2.3 – Exemple de création de grille avec un nombre de segments s $= 3$.

La figure 2.3 présente un exemple de processus de création des grilles à partir de sketches de 4 dimensions. Dans cet exemple, des grilles de deux dimensions sont construites à partir de paires adjacentes de variables. Pour construire les grilles, chaque axe est segmenté en $s$ segments. Plus le nombre de segments est grand plus il y aura de cellules dans chaque grille. Les individus dans la base de données sont associés à une cellule pour chaque grille. Par exemple, on peut observer que la grille 1 est créée à l'aide des deux premières variables de $\mathbf{T}$. Dans cette grille, les individus $t(1)$ à $t(6)$ sont positionnés dans les cellules. A chaque cellule est associé un numéro, ici les cellules sont numérotées de 1 à 9 en commençant en haut et de gauche à droite. Par exemple, l'individu $t(6)$ est dans la cellule 1 (axe des abscisses : 0-10, axe des ordonnées : 20-30), la clé de cet individu est donc 1. Dans la grille 2 l'individu $t(6)$ est dans la cellule 4 (axe des abscisses : 0-10, axe des ordonnées : 10-20). L'individu $t(6)$ n'est pas dans la même cellule pour chaque grille. Ceci est dû à la projection sur vecteurs aléatoires qui fait que chaque individu aura une position différente à chaque grille.

L'étape de création des grilles de parSketch possède différentes propriétés :

— Dans parSketch le plus souvent des grilles 2-D sont construites car l'utilisation de grilles peut s'avérer complexe dans des espaces de grande dimension. En effet, lorsque le nombre de dimensions est grand et que chaque variable est subdivisée en un nombre fixe de segments, le nombre de cellules constituant la grille augmente.

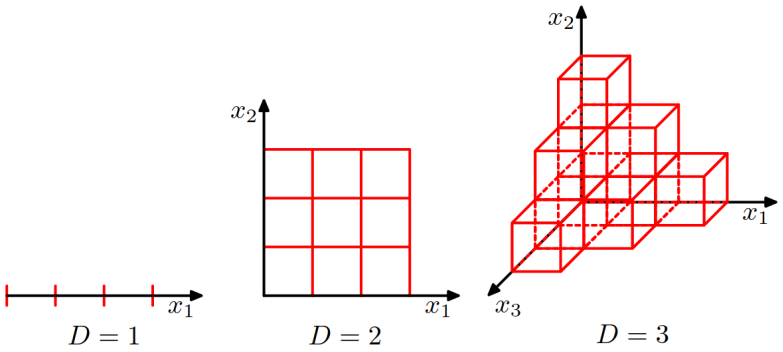



FIGURE 2.4 – Un exemple de construction de grilles 1-D, 2-D, 3-D à trois segments. Le nombre de cellules représenté dans chaque grille est : 3, 9, 27 (image issue de [17])

La figure 2.4 montre que lorsque le nombre de dimensions augmente, le nombre de cellules nécessaire pour définir l'espace avec s segments par dimension est de $s^D$, avec $D$ le nombre de dimensions. Par conséquent si le nombre de variables est grand chaque individu de la base de données peut se retrouver seul dans l'ensemble des cellules. Cela signifie donc que par la suite il sera difficile de retrouver les plus proches voisins d'un individu avec des grilles de très grande dimension. Ce phénomène est une conséquence directe du fléau de la dimensionnalité [17]. C'est donc pour cette raison que parSketch construit un ensemble de grilles 2-D et non une grille de plus grande dimension. Cette stratégie se retrouve également dans les approches Locality Sensitive Hashing (LSH) [18] qui combinent différentes fonctions de hachage pour construire une méta-fonction.

— La construction d'un ensemble de grilles est facilitée par la méthode de réduction de dimension. En effet, les variables résultantes de la projection sur vecteurs aléatoires n'ont pas d'ordre et peuvent donc être combinées, interverties, sans aucune attention particulière.

— La création d'un ensemble de grilles présente deux avantages. Premièrement, il est possible d'indexer en parallèle les individus car chaque grille peut être estimée indépendamment des autres. Deuxièmement, la compression par un ensemble de grilles à deux dimensions permet de réduire une nouvelle fois l'espace de recherche en divisant par 2 (la dimension de la grille) la dimension des sketches. En effet, l'espace $2D$ va être résumé en un vecteur à une seule dimension avec le numéro de cellule auquel est associé chaque individu.

### Étape 2 : recherche des plus proches voisins

Pour rechercher les plus proches voisins d'un nouvel individu, ce dernier est projeté sur les vecteurs aléatoires $\mathbf{P}$. Puis le sketch de ce nouvel individu est utilisé pour connaître son numéro de cellule pour chaque grille.

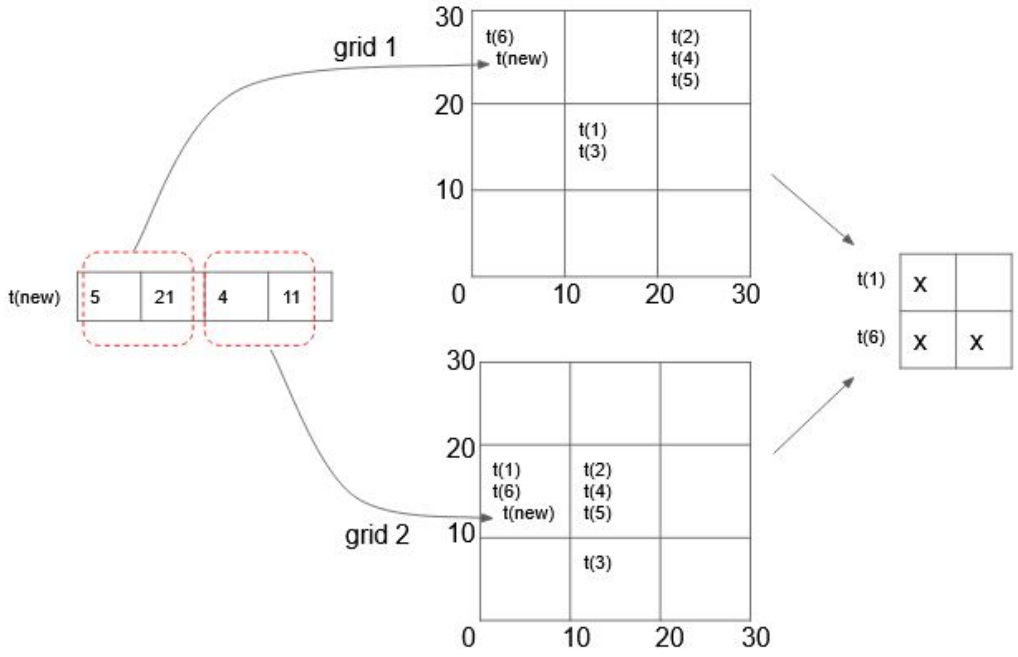



FIGURE 2.5 – Exemple de recherche des voisins d'un nouvel individu

La figure 2.5 montre un exemple de procédure de recherche des plus proches voisins d'un nouvel individu. Tout d'abord, le sketch d'un nouvel individu, appelé $t(new)$, est calculé par projection sur les vecteurs aléatoires prédéfinis. Avec le sketch de l'individu à prédire le numéro de cellule pour chaque grille est calculé. Dans l'exemple donné en figure 2.5, l'individu $t(new)$ est dans la même cellule que l'individu $t(6)$ pour la grille 1, et dans la même cellule que $t(1)$ et $t(6)$ pour la grille 2. Enfin, un comptage des individus présents dans la même cellule que $t(new)$ sur l'ensemble des grilles permet de déterminer quels sont les voisins potentiels les plus proches. Afin de définir les potentiels plus proches voisins dans parSketch, le pourcentage de cellules en commun avec l'individu à prédire est le paramètre de la méthode noté : $m$. Dans la figure 2.5 on peut compter que l'individu $t(1)$ est dans la même cellule que $t(new)$ pour une grille sur deux (soit $50\%$) alors que l'individu $t(6)$ se trouve dans la même cellule que $t(new)$ dans les deux grilles (soit $100\%$). Dans cet exemple, l'individu $t(6)$ est le potentiel plus proche voisin de $t(new)$.

L'étape de recherche des plus proches voisins par grille de parSketch possède différentes propriétés :

— Le fait de faire la recherche de plus proches voisins dans un grand nombre de grilles permet une parallélisation massive de la recherche. En effet, il est possible de positionner l'individu à prédire dans chaque grille indépendamment les unes des autres.

— Il est possible de quantifier la proximité entre les individus de la base de données et l'individu à prédire en connaissant le nombre de cellules dans lesquelles les individus de la base de données sont communs avec l'individu à prédire.

### 2.3.2 parSketch-PLSDA

parSketch-PLSDA est une méthode développée dans [2; 3] qui consiste à déterminer le voisinage d'un individu en appliquant parSketch puis à prédire cet individu en estimant un modèle d'analyse discriminante PLS (PLSDA) sur le voisinage.

#### Etude de parSketch-PLSDA

Dans [2], la méthode parSketch-PLSDA a été comparée à la méthode BF-PLSDA (brut-force-PLSDA), dans le cadre de la discrimination de génotypes du blé.

Pour la méthode BF-PLSDA, une PLS globale sur l'ensemble du jeu de données a été réalisée. Ensuite, un voisinage a été déterminé à partir des scores PLS. Enfin à partir de chaque voisinage une PLS-DA a été calibrée. La comparaison entre parSketch-PLSDA, BF-PLSDA et PLS-DA, est réalisée à partir d'une base de données comportant 360 000 spectres. Cela permet de se placer à la limite en terme de quantité de données de ce que la méthode BF-PLSDA est capable de traiter. Grâce à cette étude, il a été observé que parSketch-PLSDA permet d'être bien plus prédictif que PLSDA et d'approcher la méthode BF-PLSDA. La méthode parSketch-PLSDA offre un bon compromis entre des performances de classification et un coût calculatoire raisonnable en comparaison de BF-PLSDA.

Dans cet article, il a également été observé que les paramètres de parSketch n'influent pas uniquement sur le voisinage mais également sur la capacité prédictive de la méthode. Un mauvais paramétrage peut donner lieu à des sous-ensembles mal définis, auquel cas les capacités prédictives de la méthode deviennent très faibles. Lorsque les voisinages sont trop grands, la calibration d'un modèle linéaire PLS local n'est plus aussi efficace. De plus, le coût calculatoire d'une PLS est directement lié au nombre d'individus traités. Des voisinages trop grands mitigent donc les avantages de parSketch en terme de gain calculatoire. Cette limite est régulièrement rencontrée lorsque l'on souhaite renvoyer un

voisinage pour tous les nouveaux individus, même ceux qui sont extrêmes. En effet, si même des individus extrêmes doivent avoir un voisinage suffisamment grand pour être prédits, il faudra utiliser des paramètres peu stricts. Ces paramètres peu stricts vont donc influencer les voisinages renvoyés pour les individus typiques en renvoyant un grand nombre de voisins.

Dans cet article, il a été également observé que le paramétrage de la méthode parSketch peut s'avérer complexe. En effet, si l'on souhaite intégrer le paramétrage de parSketch dans un processus de cross-validation les temps de calculs deviennent rapidement rédhibitoires entre la construction de l'index et le calcul des modèles PLS-DA. La solution proposée dans [2] est d'observer la distribution des voisinages par le biais de boxplots. Les performances de prédiction de parSketch-PLSDA pourraient donc être améliorées en optimisant la sélection des paramètres de parSketch dans un but prédictif.

D'après cet article, il est possible de formuler l'hypothèse que parSketch-PLSDA est une méthode pertinente et efficace pour remplacer la méthode BF-PLSDA lorsque les données sont trop volumineuses. De plus, ces travaux mettent en perspective l'intérêt de développer encore cette méthode pour la chimiométrie. Deux aspects principaux semblent particulièrement intéressants : la gestion de la taille des voisinages renvoyés et l'étude des critères de similarités (métriques) utilisés pour déterminer et ordonner les voisins entre eux. En effet, dans cet article, la méthode force-brute permet facilement de modifier le critère de similarité tandis que parSketch ne permet que d'approcher les distances Euclidiennes.

### Application de parSketch-PLSDA

Dans [3], la méthode parSketch-PLSDA est appliquée à un problème de discrimination de variétés de tournesol sur la base d'images hyperspectrales. Dans ce contexte, la quantité de données devient très vite limitante pour la méthode BF-PLSDA. ParSketch-PLSDA est comparée en terme de performances à PLSDA qui constitue la méthode de référence dans cet article. Pour ce faire, une base de données spectrales a été constituée à partir d'un plan d'expérience contenant quatre variétés de tournesol. Cette base se divise en un ensemble d'individus de calibration contenant 650 000 spectres et un ensemble d'individus de test indépendant de l'ensemble de calibration de 14 000 spectres. Les résultats obtenus pour PLSDA et parSketch-PLSDA sont encourageants et confirment l'intérêt de la spectroscopie VIS-NIR pour la discrimination des variétés de tournesol. Pour cette problématique, parSketch-PLSDA surpasse la méthode de référence en terme de qualité de prédiction avec une précision en erreur de classification de $10\%$ supérieure. Ces résultats confirment par l'application l'intérêt de la stratégie parSketch-PLSDA pour l'exploitation de données spectrales massives.

## 2.4 Conclusion et perspectives

La méthode parSketch-PLS a démontrée sa pertinence pour le traitement des données massives en chimiométrie. Elle permet d'appliquer une stratégie de type KNN-PLS dans un contexte big-data. Dans [2; 3], l'intérêt d'adapter une approche issue du big-data pour la chimiométrie a été mis en avant. parSketch-PLSDA permet de réduire les temps de calculs pour la recherche de voisinages, tout en conservant des qualités de prédictions satisfaisantes. Néanmoins, certaines limitations doivent être étudiées et franchies afin d'offrir aux chimiométriciens des outils versatiles, fiables et performants :

— La méthode parSketch-PLS propose uniquement de définir des voisinages en fonction des distances Euclidiennes. Cependant, il serait intéressant de développer parSketch pour utiliser d'autres métriques. En effet, la distance Euclidienne n'est pas toujours la plus intéressante pour traiter des données spectrales et d'autres métriques plus spécifiques à ces données peuvent être utilisées. Par exemple, [19] propose d'estimer un voisinage à partir des scores d'une PLS globale. Cette stratégie permet d'orienter la sélection des plus proches voisins à partir de critères considérant à la fois X et Y. Il semble donc judicieux d'évaluer ce même type de démarche en y combinant une opération d'indexation.

— Il est assez difficile de paramétrer le nombre de voisins renvoyés par parSketch. En effet, le nombre de voisins renvoyés par parSketch dépend de deux critères, la topologie de la base de données et les paramètres de la méthode. La taille des voisinages est directement associée à la distance entre les voisins et les individus à prédire. En fonction de la position de ces individus dans la base de données la taille des voisinages va donc fortement varier. Les paramètres de parSketch quant à eux vont affecter la taille des voisinages mais ne permettront pas de choisir cette taille comme le fait l'algorithme force-brute. En pratique, c'est *a posteriori* qu'on peut établir la relation entre les paramètres et la taille des voisinages, car il y a une dépendance entre les paramètres de parSketch et la topologie de la base de données.

— Les grilles construites dans parSketch-PLSDA ne tiennent pas compte de la distribution des individus. Pour un jeu de paramètres donné, la segmentation réalisée est régulière et déterminée par les bornes (*i.e.* valeurs minimales et maximales) des sketches. Dans les faits, on observe donc moins de voisins pour les individus extrêmes que pour ceux plus proches des propriétés médianes. Il semble donc pertinent de construire les grilles à partir d'une segmentation non plus uniforme mais considérant la distribution des individus dans la base de données.

— La méthode parSketch-PLS renvoie uniquement des voisins potentiels et non pas les plus proches voisins. Il serait intéressant d'appliquer en premier lieu parSketch, puis utiliser l'algorithme force-brute. Réaliser une telle approche permettrait d'intégrer des valeurs de paramètres de parSketch plus strictes pour renvoyer moins de voisins par individu à prédire. En effet, les individus extrêmes sans voisins retournés par parSketch peuvent être traités par force-brute. Ceci permettrait de calibrer des modèles avec des voisinages plus faibles et exacts. Cela mènerait donc à des performances de prédictions plus élevées et potentiellement un temps de calcul plus faible.

— Il serait intéressant de remplacer parSketch par des méthodes comme DpiSAX et de comparer leurs capacités prédictives. DpiSAX permet de regrouper les individus de la base de données de manière indépendante du point à prédire (contrairement à parSketch). Cette approche pourrait notamment être pertinente pour remplacer les approches de clustering utilisées dans des méthodes de type clustered PLS.

— Il se peut que certains voisins renvoyés par parSketch ne soient pas pertinents pour la calibration de modèles PLSR. En effet, comme le voisinage renvoyé par parSketch est approximatif et dépend uniquement de la distance Euclidienne entre spectres, des plus proches voisins non pertinents peuvent être générés. Ces voisins non pertinents vont potentiellement réduire les capacités prédictives de la méthode.

# Bibliographie

[1] Oleksandra Levchenko, Djamel-Edine Yagoubi, Reza Akbarinia, Florent Masseglia, Boyan Kolev, and Dennis Shasha. Spark-parSketch : A Massively Distributed Indexing of Time Series Datasets. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pages 1951–1954, Torino Italy, October 2018. ACM. ISBN 978-1-4503-6014-2. doi: 10.1145/3269206.3269226. URL https://dl.acm.org/doi/10.1145/3269206.3269226.

[2] Maxime Metz, Matthieu Lesnoff, Florent Abdelghafour, Reza Akbarinia, Florent Masseglia, and Jean-Michel Roger. A "big-data" algorithm for KNN-PLS. Chemometrics and Intelligent Laboratory Systems, 203 :104076, August 2020. ISSN 0169-7439. doi: 10.1016/j.chemolab.2020.104076. URL http://www.sciencedirect.com/science/article/pii/S0169743920301908.

[3] Maxime Ryckewaert, Maxime Metz, Daphné Héran, Pierre George, Bruno Grèzes-Besset, Reza Akbarinia, Jean-Michel Roger, and Ryad Bendoula. Massive spectral data analysis for plant breeding using parsketch-plsda method : Discrimination of sunflower genotypes. Biosystems Engineering, 210 :69–77, 2021. ISSN 1537-5110. doi: https://doi.org/10.1016/j.biosystemseng.2021.08.005. URL https://www.sciencedirect.com/science/article/pii/S1537511021001896.

[4] R Bayer. Organization and Maintenance of Large Ordered Indices. page 35, 1970.

[5] Antonin Guttman. R-trees : a dynamic index structure for spatial searching. In Proceedings of the 1984 ACM SIGMOD international conference on Management of data - SIGMOD '84, page 47, Boston, Massachusetts, 1984. ACM Press. ISBN 978-0-89791-128-3. doi: 10.1145/602259.602266. URL http://portal.acm.org/citation.cfm?doid=602259.602266.

[6] Djamel-Edine Yagoubi. Indexing and analysis of very large masses of time series. page 147, 2018.

[7] Ira Assent, Ralph Krieger, Farzad Afschari, and Thomas Seidl. The TS-tree : efficient time series search and retrieval. In Proceedings of the 11th international conference on Extending database technology Advances in database technology - EDBT '08, page 252, Nantes, France, 2008. ACM Press. ISBN 978-1-59593-926-5. doi: 10.1145/1353343.1353376. URL http://portal.acm.org/citation.cfm?doid=1353343.1353376.

[8] Yuhan Cai and Raymond Ng. Indexing spatio-temporal trajectories with Chebyshev polynomials. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, SIGMOD '04, pages 599–610, New York, NY, USA, June 2004. Association for Computing Machinery. ISBN 978-1-58113-859-7. doi: 10.1145/1007568.1007636. URL https://doi.org/10.1145/1007568.1007636.

[9] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. ACM SIGMOD Record, 23(2) :419–429, June 1994. ISSN 0163-5808. doi: 10.1145/191843.191925. URL https://dl.acm.org/doi/10.1145/191843.191925.

[10] Jin Shieh and Eamonn J. Keogh. iSAX : indexing and mining terabyte sized time series. In KDD, 2008. doi: 10.1145/1401890.1401966.

[11] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn Keogh. iSAX 2.0 : Indexing and Mining One Billion Time Series. In 2010 IEEE International Conference on Data Mining, pages 58–67, Sydney, Australia, December 2010. IEEE. ISBN 978-1-4244-9131-5. doi: 10.1109/ICDM.2010.124. URL http://ieeexplore.ieee.org/document/5693959/.

[12] Djamel Edine Yagoubi, Reza Akbarinia, Florent Masseglia, and Themis Palpanas. DPiSAX : Massively Distributed Partitioned iSAX. In 2017 IEEE International Conference on Data Mining (ICDM), pages 1135–1140, New Orleans, LA, November 2017. IEEE. ISBN 978-1-5386-3835-4. doi: 10.1109/ICDM.2017.151. URL http://ieeexplore.ieee.org/document/8215614/.

[13] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery - DMKD '03, page 2, San Diego, California, 2003. ACM Press. doi: 10.1145/882082.882086. URL http://portal.acm.org/citation.cfm?doid=882082.882086.

[14] Jan M Kriegl, Lennart Eriksson, Thomas Arnhold, Bernd Beck, Erik Johansson, and Thomas Fox. Multivariate modeling of cytochrome p450 3a4 inhibition. European journal of pharmaceutical sciences : official journal of the European Federation for Pharmaceutical Sciences, 24(5) :451—463, April 2005. ISSN 0928-0987. doi: 10.1016/j.ejps.2004.12.009. URL https://doi.org/10.1016/j.ejps.2004.12.009.

[15] Cristian Preda and Gilbert Saporta. PLS Approach for Clusterwise Linear Regression on Functional Data. In David Banks, Frederick R. McMorris, Phipps Arabie, and Wolfgang Gaul, editors, Classification, Clustering, and Data Mining Applications, pages 167–176. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-22014-5 978-3-642-17103-1. doi: 10.1007/978-3-642-17103-1_17. URL http://link.springer.com/10.1007/978-3-642-17103-1_17.

[16] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In Richard Beals, Anatole Beck, Alexandra Bellow, and Arshag Hajian, editors, Contemporary Mathematics, volume 26, pages 189–206. American Mathematical Society, Providence, Rhode Island, 1984. ISBN 978-0-8218-5030-5 978-0-8218-7611-4. doi: 10.1090/conm/026/737400. URL http://www.ams.org/conm/026/.

[17] Christopher M. Bishop. Pattern recognition and machine learning. Information science and statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.

[18] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors : towards removing the curse of dimensionality. In Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC '98, pages 604–613, Dallas, Texas, United States, 1998. ACM Press. ISBN 978-0-89791-962-3. doi: 10.1145/276698.276876. URL http://portal.acm.org/citation.cfm?doid=276698.276876.

[19] Matthieu Lesnoff, Maxime Metz, and Jean-Michel Roger. Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data. Journal of Chemometrics, 34(5) :e3209, 2020. ISSN 1099-128X. doi: 10.1002/cem.3209. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.3209. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.3209.

# RoBoost-PLSR : une nouvelle méthode de régression PLS robuste

*Chapitre* 3

## Sommaire



## 3.1 Motivations et vue d'ensemble des travaux

Dans le chapitre précédent, il a été observé que parSketch renvoie des voisins potentiellement non pertinents pour la calibration de modèles PLS. Ce chapitre de thèse a donc pour objectif d'étudier la pertinence des individus pour la calibration de modèles PLS. Dans cette thèse, il est considéré qu'un individu est pertinent pour un modèle, s'il contribue à la bonne estimation de la majorité des individus. Les méthodes de régression robustes s'appuient sur cette notion en définissant la pertinence d'un individu pour la calibration. En pratique, un poids est attribué à chaque individu de la base de calibration.

Ces poids ont pour objectif de représenter la pertinence des individus. Les stratégies de pondération dans les méthodes robustes sont encore aujourd'hui très étudiées [1]. Afin d'obtenir des critères de pertinence plus appropriés aux données spectrales, une nouvelle approche a été développée dans cette thèse, RoBoost-PLSR, et fait l'objet de ce chapitre.

Cette méthode a tout d'abord été étudiée dans un cadre prédictif dans [2] pour la régression PLS1 (Y à une dimension c'est-à-dire que l'on cherche à prédire qu'une seule variable). Puis, cette méthode a été également développée en version PLS2 (Y multivarié) dans [3] . Enfin, dans [4] la méthode a été appliquée dans un contexte agronomique pour la prédiction du taux de sucre dans les baies de raisin.

Ce chapitre a pour objectif d'accompagner le lecteur dans la compréhension des trois travaux associés à RoBoost-PLS mais également d'apporter des informations complémentaires sur ce cadre RoBoost-PLSR. Dans ce chapitre, les méthodes robustes seront introduites et RoBoost-PLSR sera positionnée parmi elles. Ensuite, RoBoost-PLSR sera expliquée en détail. Par la suite, les trois articles correspondant au cadre RoBoost-PLSR seront discutés. Pour finir, les perspectives de ce chapitre seront présentées.

## 3.2 Les méthodes de régression linéaires robustes et les données aberrantes

Les données aberrantes sont des individus qui diffèrent fortement d'une population d'individus. La présence de données aberrantes est nuisible pour l'estimation de la relation entre les variables explicatives $\mathbf{X}$ et les valeurs de référence $\mathbf{Y}$. L'origine des données aberrantes en chimiométrie peut s'expliquer par des phénomènes très divers qui peuvent apparaître dans les matrices $\mathbf{X}$ et $\mathbf{Y}$ comme une erreur de mesure due à un défaut d'instrumentation, de la variation de l'environnement de mesure ou bien encore d'erreurs d'étiquetages. La plupart des méthodes conventionnelles de la chimiométrie sont sensibles aux valeurs aberrantes. Pour faire face à ces problèmes, un grand nombre de méthodes robustes ont été développées en chimiométrie. L'objectif d'une méthode robuste est donc de réduire ou d'éliminer l'effet des données aberrantes et de permettre aux autres points de déterminer principalement les résultats.

### 3.2.1 La régression linéaire et les données aberrantes

La régression linéaire est un outil pertinent pour l'analyse de données chimiques. Elle permet d'illustrer, dans un cas unidimensionnel, l'influence de différentes données atypiques auxquelles les modèles linéaires sont sensibles [5] : les points leviers et les

points verticaux.

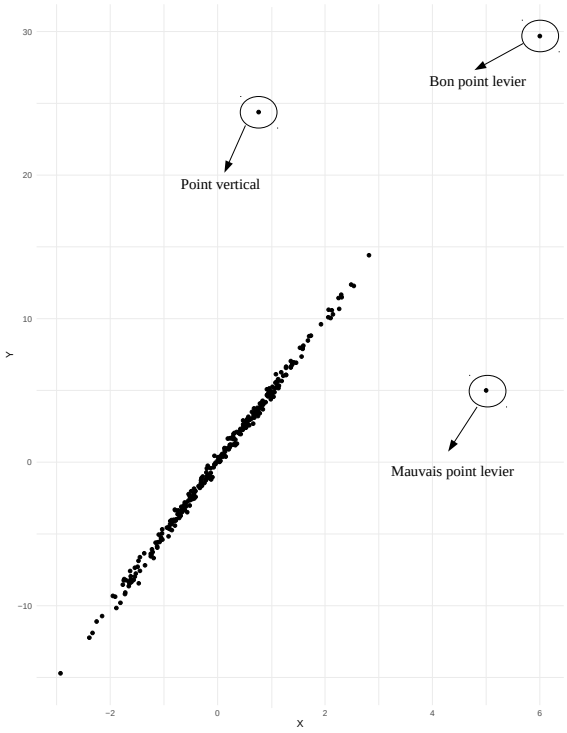



FIGURE 3.1 – Différents types de données atypiques

La figure 3.1 présente les différents types de données atypiques. Les points leviers sont des individus qui s'éloignent fortement du centre des données sur l'axe X. Cet éloignement impacte fortement la calibration d'un modèle linéaire. Cependant, les points leviers, s'ils sont bons, permettent une meilleure estimation du modèle linéaire. Les points verticaux quant à eux ne sont pas éloignés selon l'axe X mais selon l'axe Y. Ils ont une influence modérée sur la calibration du modèle linéaire. En effet, étant donné qu'ils sont proches du centre des données sur X, ils influent peu sur la calibration du modèle. Les mauvais points leviers sont des échantillons qui s'écartent du centre des données selon l'axe des X et Y. Ils perturbent la calibration d'un modèle linéaire.

Ces phénomènes s'appliquent par extension aux cas multidimensionnels. Ils ont été particulièrement étudiés dans le cadre de la régression multiple et par conséquent de nombreuses méthodes de régressions ont été développées [6], notamment les M-estimateurs [7] et la régression LST [8]. Cependant, ces méthodes sont pertinentes uniquement lorsque $\mathbf{X}$ est de petite dimension.

### 3.2.2 La régression PLS et les données aberrantes

La régression des moindres carrés partiels (PLSR) [9], est un outil très utilisé en chimiométrie. La PLSR est particulièrement pertinente pour le traitement de données

spectrales puisqu'elle permet de traiter des données de grande dimension, même lorsque le nombre d'échantillons est plus faible que le nombre de variables, ce qui est un cas courant en chimiométrie.

Cette méthode est particulièrement performante lorsque la relation entre les variables explicatives et la ou les variables à expliquer est linéaire. Cependant l'estimation de cette relation linéaire peut être perturbée par la présence de données aberrantes [10]. Pour gérer la présence des données aberrantes, un grand nombre de méthodes a été développé [11–23]. Toutefois, en grande dimension, l'estimation de la pertinence d'un point au sein d'un modèle de calibration n'est pas triviale. Parmi toutes ces méthodes, deux références sont particulièrement appliquées en chimiométrie : la méthode RSIMPLS [24] et la méthode PRM (Partial Robust M-regression) [25].

La méthode RSIMPLS propose d'estimer de façon robuste les matrices de cross-covariance $\mathbf{C_{xy}}$ et de covariance $\mathbf{C_x}$ dans l'algorithme SIMPLS [26]. Pour cela, une ROBPCA [27] est calculée sur la matrice $\mathbf{Z}$ résultante de la concaténation de $\mathbf{X}$ et de $\mathbf{Y}$. Ensuite, RSIMPLS calcule la distance résiduelle robuste au modèle ROBPCA pour calculer un modèle linéaire pondéré entre les scores PLS robuste $\mathbf{T}$ et $\mathbf{Y}$.

La méthode PRM quant à elle se base sur les M-estimateurs afin de réaliser un modèle robuste. Cela signifie donc qu'au lieu de minimiser les moindres carrés :

$$\hat{\beta}_{LS} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i\beta)^2 \tag{3.1}$$

PRM consiste à minimiser :

$$\hat{\beta}_{M} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} p(y_i - x_i\beta) \tag{3.2}$$

Où $p(.)$ désigne la fonction du M-estimateur, $\beta$ les coefficients de régression entre $\mathbf{X}$ et $\mathbf{Y}$, $x_i$ un individu de $\mathbf{X}$, $y_i$ un individu de $\mathbf{Y}$.

Pour cela, PRM consiste à estimer un poids à partir de mesures : le levier et les résidus Y. Dans PRM, le levier est défini par la distance Euclidienne au centre médian des données. Le poids résultant est une valeur comprise entre 0 et 1. Ces critères sont utilisés lors de la calibration d'un modèle PLSR défini pour un nombre de variables latentes déterminé. Cette démarche est itérée jusqu'à ce que le modèle converge, c'est-à-dire, que les variations de poids des individus au sein du modèle sont tellement faibles qu'ils impactent très peu la calibration.

Ces deux approches de référence sont performantes mais présentent quelques limitations. Dans le cadre de RSIMPLS, il est nécessaire de réaliser un premier modèle ROBPCA et d'estimer des paramètres pertinents avant d'estimer le modèle robuste. De

plus, l'estimation de matrices de covariance robustes peut être très coûteuse en temps de calcul. Pour PRM, la détection des données aberrantes se fait à l'aide du modèle PLS avec un nombre de variables latentes prédéfini. D'un point de vue pratique, cela peut entraîner des opérations de paramétrage lourdes car il faut réaliser une calibration pour chaque modèle avec chaque paramètre de pondération. Par ailleurs, dans certains cas, les critères de pertinence utilisés par PRM, notamment pour l'estimation des points leviers peuvent s'avérer insuffisants. En effet, comme le critère est une distance Euclidienne dans l'espace des scores PLS, seules les premières variables latentes contribuent à l'estimation des points leviers, puisque l'amplitude des scores PLS varie pour chaque variable latente.

## 3.3 RoBoost-PLSR

### 3.3.1 Notations

Les caractères gras majuscules seront utilisés pour les matrices, *e.g.* $\mathbf{X}$ ; les petits caractères gras pour les vecteurs colonnes, *e.g.* $\mathbf{x}_j$ désignera la $j^{ième}$ colonne de $\mathbf{X}$ ; les vecteurs lignes seront désignés par la notation transposée, *e.g.* $\mathbf{x}_i^\top$ désignera la $i^{ième}$ ligne de $\mathbf{X}$ ; les caractères en italique seront utilisés pour les scalaires, *e.g.* éléments de matrice $x_{ij}$ ou indices $i$. Les scalaires constants seront désignés par des caractères en italique, *e.* nombre d'échantillons $n$. $\mathbb{1}$ représentera un vecteur colonne de uns, de dimension appropriée. $med$ définit la médiane. $\mathbf{X}$ et $\mathbf{Y}$ sont les matrices des spectres et des réponses. $g$ est la fonction de poids. $\mathbf{D}$ est la matrice des poids des échantillons où la diagonale de la matrice est le poids des échantillons et les autres termes sont nuls.

### 3.3.2 Présentation de la méthode

RoBoost-PLSR est une nouvelle méthode robuste développée dans le cadre de cette thèse. Une première version de RoBoot-PLSR a été proposée pour la PLS1 dans [2] . Cet article met en avant l'intérêt de la méthode pour le traitement des données spectrales en présence de données aberrantes de natures variées. Dans cet article, RoBoost-PLSR a été comparée à la méthode PRM et à la méthode PLSR avec et sans données aberrantes. Pour réaliser ces comparaisons, trois jeux de données ont été simulés et un jeu de données réelles a été utilisé. Il a été observé que la méthode RoBoost-PLSR permettait d'approcher les résultats de la PLSR sans données aberrantes. Cela signifie donc que l'effet des données aberrantes sur les modèles RoBoost-PLSR était presque nul.

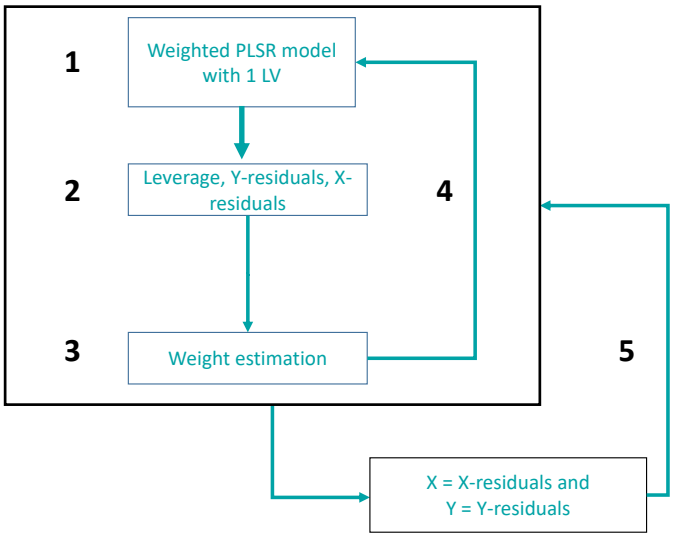



FIGURE 3.2 – Schéma représentant l'algorithme de RoBoost-PLSR

La figure 3.2 représente la stratégie de calibration de la méthode :

1. Un modèle PLSR pondéré à une variable latente est calculé
2. Les mesures associées aux points leviers, aux résidus $\mathbf{X}$ et $\mathbf{Y}$ sont calculés.
3. Les critères précédents sont convertis en poids et ces poids sont multipliés entre eux pour donner un poids pour chaque individu
4. Cette opération est répétée jusqu'à convergence du modèle
5. Les résidus $\mathbf{X}$ et $\mathbf{Y}$ sont ensuite utilisés pour réaliser le futur modèle à une variable latente

   L'ensemble de ces opérations est réalisé jusqu'au nombre de variables latentes choisi

La méthode RoBoost-PLSR se distingue à plusieurs niveaux des stratégies habituelles utilisées dans les méthodes robustes et permet dans certains cas des gains de performance ou bien des gains de temps.

Premièrement, elle permet comme la PLS, d'estimer tous les modèles de 1 à $K$ variables latentes choisi. Ceci est très intéressant car les méthodes comme PRM ne le permettent pas. En effet, il est nécessaire de calibrer un modèle robuste pour chaque nombre de variables latentes. Cette limitation peut être rédhibitoire lors de l'étape de paramétrage (e.g. nombre de variables latentes, etc) car les temps de calcul nécessaires pour réaliser cette opération sont trop conséquents. RoBoost-PLSR quand à elle permet de calculer tous les modèles de 1 à $K$ variables latentes à partir d'un modèle à $K$ variables latentes.

Deuxièmement, l'utilisation des résidus $X$ comme critère permet de mieux définir la pertinence des individus. En effet, certains individus ne sont ni leviers ni verticaux mais peuvent tout de même posséder une distance orthogonale au modèle élevé.

Troisièmement, la méthode possède aussi des propriétés qui facilitent l'estimation des points leviers. Les détails associés aux critères de pondération et à l'algorithme sont détaillés précisément dans [2; 3] . L'algorithme le plus abouti a été proposé dans [3] :

**Algorithm** RoBoost-PLSR pour $K$ LV

Pour un nombre défini de $K$ variables latentes, l'algorithme se déroule comme décrit ci-dessous :

1: Étape d'initialisation

$$k = 1$$

$$\mathbf{D} = diag(d_1, d_2, ..., d_n) \text{ with } d_i = \frac{1}{n}$$

2: Centrer les données :

$$\mathbf{X}_k = \mathbf{X} - \mathbb{1}\mathbb{1}^\mathsf{T}\mathbf{D}\mathbf{X}$$

$$\mathbf{Y}_k = \mathbf{Y} - \mathbb{1}\mathbb{1}^\mathsf{T}\mathbf{D}\mathbf{Y}$$

3: Définir $\mathbf{u}_k$ comme une colonne arbitraire de $\mathbf{Y}$

4: Calculer une variable latente pondérée NIPALS :

$$\mathbf{w}_k = \frac{\mathbf{X}_k^\mathsf{T}\mathbf{D}\mathbf{u}_k}{||\mathbf{X}_k^\mathsf{T}\mathbf{D}\mathbf{u}_k||}$$

$$\mathbf{t}_k = \mathbf{X}_k\mathbf{w}_k$$

$$\mathbf{p}_k = \frac{\mathbf{X}_k^\mathsf{T}\mathbf{D}\mathbf{t}_k}{\mathbf{t}_k^\mathsf{T}\mathbf{D}\mathbf{t}_k}$$

$$\mathbf{q}_k = \frac{\mathbf{Y}_k^\mathsf{T}\mathbf{D}\mathbf{t}_k}{\mathbf{t}_\mathbf{k}^\mathsf{T}\mathbf{D}\mathbf{t}_k}$$

$$c_k = \frac{\mathbf{u}_k^\mathsf{T}\mathbf{D}\mathbf{t}_k}{\mathbf{t}_\mathbf{k}^\mathsf{T}\mathbf{D}\mathbf{t}_k}$$

5: Calculer $(\mathbf{F})$, $(\mathbf{E})$, $(\mathbf{l})$ :

$$\mathbf{E} = \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^\top$$

$$\mathbf{F} = \mathbf{Y}_k - \mathbf{t}_k \mathbf{q}_k$$

$$\mathbf{l} = \mathbf{t}_k$$

6: Mettre à jour les poids :

$$d_i = \frac{1}{n} \times g(||\mathbf{e}_i||, \alpha) \times \prod_{j=1}^{m} g(f_{ij}, \beta), \times g(l_i, \gamma)$$

7: Retourner (étape (2) pour k $= 1$, autrement retourner à l'étape (4) ) jusqu'à convergence de $c$'s.

8: Étape de déflation

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^\top$$

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k - \mathbf{t}_k \mathbf{q}_k$$

$$\mathbf{u}_{k+1} = \mathbf{Y}_k \mathbf{q}_k$$

$k = k + 1 \rightarrow$ ensuite aller à l'étape $(4)$

Les coefficients de régression résultant pour $K$ variables latentes sont estimés comme suit :

$$\mathbf{B} = \mathbf{R}\mathbf{c}^\top$$

Avec $\mathbf{R}$ :

$$\mathbf{R} = \mathbf{W}(\mathbf{P}^\top \mathbf{W})^{-1}$$

### 3.3.3 Résidus de X

En chimiométrie, les résidus de $X$ sont utilisés pour réaliser des graphiques de détection d'outliers. Les méthodes PLS robustes quant à elles utilisent très peu les résidus de $X$ pour attribuer automatiquement un poids aux individus. Pourtant, il est fréquent que des individus se différencient des autres de part leurs résidus $X$, particulièrement dans le cadre des données PIR (proche infrarouge), où la mesure est très facilement perturbée par les conditions ou variations de l'environnement de mesure. Dans les méthodes PLS robustes, ce sont principalement deux critères d'aberrance qui sont cités : les points leviers et les points verticaux. Cependant, il est possible d'introduire un critère supplémentaire : les points horizontaux. En combinant les trois critères : levier, points verticaux et horizontaux, RoBoost-PLSR propose de définir un nouvel estimateur de la pertinence des individus pour la calibration de modèle PLS.

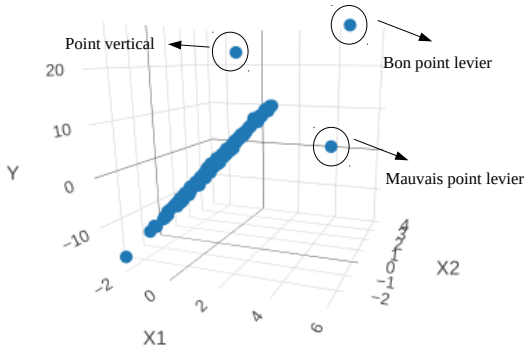



FIGURE 3.3 – Visualisation d'un point vertical et de bons et mauvais points leviers

La figure 3.3 présente Y en fonction de x1 et x2. Ces données ont été simulées pour mettre en avant les différents points atypiques possibles. La matrice $\mathbf{X}$ a été simulée avec 10 variables. Le vecteur $y$ est corrélé à la première variable de $\mathbf{X}$ ($x1$). Dans ce jeu de données, quatre individus ont été simulés pour être atypiques :

— Un individu avec une valeur de x1 extrême, c'est un bon point levier

— Un individu avec un bruit fort sur Y, c'est un point vertical

— Un individu avec un bruit fort sur Y et une valeur extrême sur x1, c'est un mauvais point levier

— Un individu avec des signatures spectrales différentes des autres individus, c'est un point horizontal

Sur la figure 3.3 il est possible d'observer les points leviers et le point vertical. Cependant, le point horizontal n'est pas observable car les signatures spectrales utilisées

ont une faible contribution dans la simulation du point horizontal. Cet exemple met en évidence qu'il n'est pas possible de prendre en compte facilement ce point avec des critères utilisés dans les méthodes PLS robustes comme le levier ou les résidus de $\mathbf{Y}$.

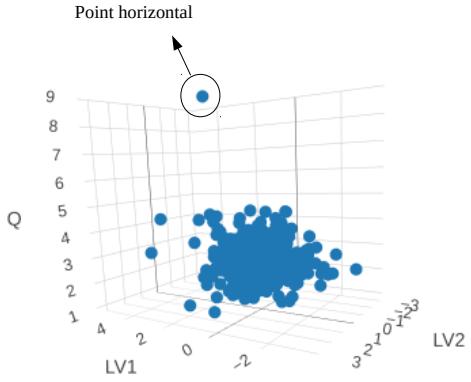



FIGURE 3.4 – Visualisation d'un point horizontal

La figure 3.4 présente le graphique des deux premières variables latentes d'un modèle PLS calibré sur les données simulées, en fonction des résidus de $\mathbf{X}$ (notés $Q$). Sur ce graphique, il est observé qu'un individu de la base de données se distingue des autres. Cet individu est celui qui a été simulé avec des signatures spectrales différentes. Il est également possible d'observer que les différents points atypiques classiques ne sont pas visibles par le biais des résidus de $\mathbf{X}$. Ce graphique met donc en évidence l'intérêt de combiner les critères classiques utilisés dans les méthodes PLS robustes (levier et point vertical) avec le critère des points horizontaux utilisé en chimiométrie.

Dans le cadre de RoBoost-PLSR, il est proposé d'intégrer les résidus de $X$ dans le calcul des poids des individus afin de prendre en compte un maximum de critères.

### 3.3.4 Points leviers

L'estimation des points leviers peut être complexe dans le contexte de la régression PLS. La démarche classique utilisée en chimiométrie est le calcul d'une distance sur scores PLS. Les deux distances principalement utilisées en chimiométrie pour définir la mesure du levier sont la distance Euclidienne et la distance de Mahalanobis au centre du modèle.

Ces deux stratégies peuvent ne pas être optimales et rendre difficile la mesure du levier. En effet, la distance Euclidienne sur scores PLS va favoriser les variables latentes qui ont la plus forte amplitude de scores. Cependant l'amplitude des scores n'est pas associée aux capacités prédictives de la méthode.

La distance de Mahalanobis au centre des données, permet de supprimer l'effet de l'amplitude des scores sur le calcul de la distance Euclidienne. Cependant, cette stratégie peut représenter une problématique dans le cas où le nombre de variables latentes est grand (5-10 variables typiquement) car les points leviers ne seront pas toujours identifiables. En effet, plus le nombre de dimensions augmente, plus la différence entre les individus par rapport au centre du modèle diminue. Ceci est dû au fléau de la dimensionnalité [28]. Ce résultat s'explique par le fait que l'augmentation du nombre de variables rend de plus en plus différents les individus entre eux et éloignés du centre du modèle. Ceci entraîne donc que des individus possédant des données aberrantes peuvent être masqués par l'effet du nombre de variables qui les décrivent.

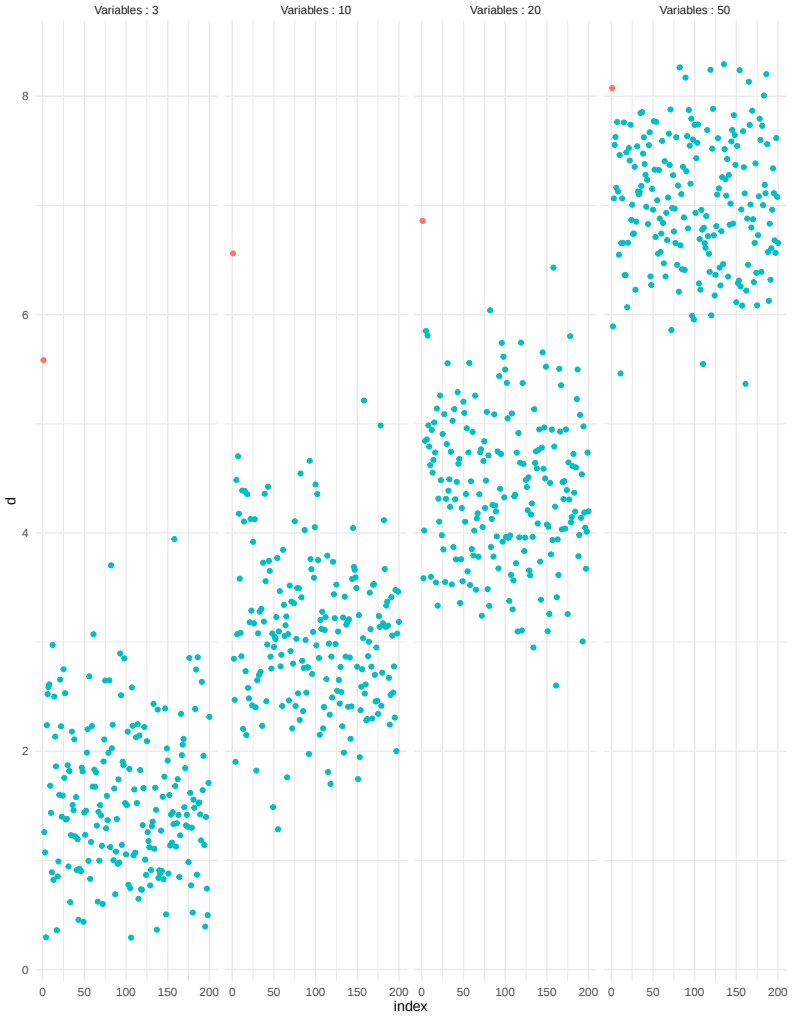



FIGURE 3.5 – Impact de l'estimation de la distance de Mahalanobis $d$, au centre du modèle en fonction du nombre de variables, pour un ensemble de 200 individus

La figure 3.5 montre l'influence du nombre de variables sur l'évolution des distances de Mahalanobis au centre des données. Cette démonstration a été réalisée à partir d'un

jeu de données simulé. Les variables du jeu de données sont générées aléatoirement à partir de distributions gaussiennes. Dans ce jeu de données, un individu a une valeur extrême (en rouge dans le graphique) pour la première variable. Il est possible d'observer sur la figure 3.5 que lorsque le nombre de variables augmente, l'individu atypique en rouge se confond avec les autres individus. En effet, lorsque la dimension augmente tous les individus deviennent à la fois très différents les uns des autres mais également très similaires vis-à-vis du centre des données.

[28] montre que lorsque le nombre de dimensions est très grand, tous les individus se placent autour d'une hypersphère où le centre de la sphère représente le centre des données. La détection des points leviers dans ce contexte est donc complexe. Il est possible de retrouver cette même problématique lors du calcul de la distance de Mahalanobis sur les scores d'une PLS. En effet, plus le nombre de variables latentes est grand, plus la détection des points leviers est complexe. C'est en cela que définir des données aberrantes à partir d'un modèle PLS avec un grand nombre de variables latentes peut être difficile et certains individus atypiques ne peuvent pas être détectés. Dans le cadre RoBoost-PLSR, il est proposé d'aborder ce problème en réalisant ces estimations variable latente par variable latente c'est-à-dire dans des espaces à une dimension.

### 3.3.5 RoBoost-PLS2-R

RoBoost-PLS2-R est une extension de RoBoost-PLSR pour les cas où $\mathbf{Y}$ est multivarié [3] . Le développement de cette extension nécessite de lever deux verrous méthodologiques liés au critère de convergence du modèle ainsi qu'aux données aberrantes dans le cas où $\mathbf{Y}$ est multidimensionnel. En effet, dans ce cas, $q$ est utilisé pour mesurer la convergence du modèle :

$$q = \frac{\mathbf{y}^\top \mathbf{D}\mathbf{t}}{\mathbf{t}^\top \mathbf{D}\mathbf{t}} \tag{3.3}$$

Cependant, lorsque $\mathbf{Y}$ est multidimensionnel $q$ n'est plus un scalaire mais un vecteur. La solution proposée est alors d'estimer $c$ :

$$c = \frac{\mathbf{u}^\top \mathbf{D}\mathbf{t}}{\mathbf{t}^\top \mathbf{D}\mathbf{t}} \tag{3.4}$$

Où $T$ sont les X-scores et $u$ sont les Y-scores tels que :

$$u = \mathbf{Y}c. \tag{3.5}$$

Ainsi le critère proposé est un scalaire et donc l'estimation de la convergence devient alors simplifiée.

L'estimation des poids est également complexifiée dans le contexte $\mathbf{Y}$multidimensionnel, puisque les résidus $\mathbf{Y}$ sont alors également multidimensionnels. De ce fait, il est important de tenir compte des variances respectives de chaque coordonnée de $\mathbf{Y}$. Dans le cas de RoBoost-PLS2-R, il est proposé de réaliser un produit des poids estimés sur chaque variable de $\mathbf{Y}$. Cette stratégie est intéressante car elle permet de ne pas tenir compte de la variabilité de chaque $\mathbf{Y}$. Cependant, cette stratégie pourrait potentiellement être mise en défaut quand le nombre de variables Y est très grand. Ce nouvel algorithme et ses propriétés sont détaillés dans [3] .

### 3.3.6 Interprétabilité de RoBoost-PLSR

L'algorithme proposé dans les premiers travaux de cette thèse [2] ne permet pas d'estimer les coefficients de régression ni la matrice de rotation $R$ qui permet de calculer directement les scores. Or, la popularité de la PLS s'explique en partie par son interprétabilité. C'est pourquoi, [3] propose de réaliser un seul centrage pondéré à l'estimation du premier modèle (*i.e.* la première variable latente) plutôt que de mettre à jour le centre à chaque modèle comme initialement proposé. Les centrages successifs entraînent un biais qui ne sont pas pris en compte dans le calcul des coefficients de régression. Ainsi, cette procédure simplifiée permet d'estimer la matrice de rotation par :

$$R = W(P^\top W)^{-1} \tag{3.6}$$

En découle les coefficients de régression suivants :

$$B = RC^\top \tag{3.7}$$

RoBoost-PLSR permet ainsi l'observation et l'interprétation des coefficients de régression.

### 3.3.7 Paramétrage de RoBoost-PLSR

Le paramétrage des méthodes robustes PLS n'est pas trivial. Les méthodes PLS robustes sont développées pour résister à la présence de données aberrantes lors de la calibration du modèle. Cependant, pour tester ou valider la qualité de calibration d'un modèle il faut des indicateurs appropriés. Les indicateurs classiques ($R^2$,$RMSE$, etc.) vont chercher à minimiser l'erreur de prédiction du modèle en prenant en compte au même niveau chaque individu du jeu de validation. Or, il est possible d'avoir des données aberrantes dans le jeu de test. Si le modèle PLS robuste élimine l'effet des

données aberrantes, ces dernières vont être mal prédites. Dans ce cas-ci, les indicateurs classiques mettront en évidence que le modèle possède des capacités prédictives faibles car ces indicateurs vont être biaisés par les données aberrantes.

Dans cette thèse, une stratégie de réglage des paramètres du modèle par validation croisée a été proposée [4] . Cette stratégie permet de définir les paramètres d'un modèle tout en utilisant la totalité du jeu de données pour valider les modèles calibrés. Pour permettre de réaliser une cross-validation, des indicateurs robustes ont été développés. Dans [1] un nouvel indicateur robuste de la RMSECV ( root-mean-square error of cross-validation) est proposé. Il consiste à écarter les individus possédant les résidus de $\mathbf{Y}$ les plus forts du calcul de la RMSECV. En réalisant cela, les individus mal prédits vis-à-vis de $\mathbf{Y}$ sont définis aberrants. Ce procédé ne s'est pas révélé efficace dans le cadre de RoBoost-PLSR. Comme observé dans les sections précédentes, tous les types de points atypiques ne seraient pas forcément pris en compte. La solution proposée dans [4] est de calculer la pertinence des individus du jeu de validation en utilisant les mêmes critères (résidus X, point levier, point vertical) que pour établir la pertinence des individus de calibration. Par la suite, comme dans [1] un pourcentage des échantillons est exclu du calcul des différents indicateurs pour les rendre robustes.

## 3.4 Perspectives de RoBoost-PLS

Bien que la méthode RoBoost-PLSR s'est prouvée à la fois pertinente et performante [2–4] , elle nécessite encore plusieurs études complémentaires sur les points développés ci-après.

Premièrement, une étude complète concernant les fonctions de poids et leur optimisation devrait être menée, afin de mieux ajuster les modèles. En effet, le paramétrage de RoBoost-PLSR peut être une tâche fastidieuse (trois paramètres doivent être optimisés pour chaque variable latente). Dans les travaux réalisés durant la thèse, et afin de faciliter la compréhension, les constantes ont été fixées pour toutes les variables latentes. Cependant, il serait bien plus judicieux de définir des constantes spécifiques, optimisées pour chaque variable latente.

Deuxièmement, le calcul du centre des données (lors du centrage) et l'estimation des poids des échantillons peuvent être corrompus par la présence de données aberrantes. Dans RoBoost-PLSR, les données sont centrées sur la moyenne. La moyenne n'est pas robuste et peut donc fournir des valeurs de départ biaisées se répercutant dans la suite de l'algorithme. Une solution potentielle consisterait à remplacer ces estimateurs par des alternatives robustes (*e.g.* localisation multivariée robuste). Quant à la fonction de poids "bisquares", elle s'appuie sur la médiane, qui peut se trouver en dehors de l'enveloppe

convexe des données. Il serait pertinent de considérer d'autres fonctions de poids qui prennent en compte ces aspects.

Troisièmement, les méthodes multivariées robustes ont prouvé leur qualité et leur fiabilité prédictive pour des problématiques de classification [29]. Il serait alors intéressant d'adapter le formalisme de la méthode Roboost-PLS2-R pour les cas de variables catégorielles et ainsi proposer une méthode discriminante robuste.

Quatrièmement, le nouvel algorithme proposé dans [3] permet désormais l'estimation de coefficients de régression. Il serait intéressant d'étudier ces coefficients pour évaluer le comportement de la méthode en dehors des capacités de prédiction. Il serait également possible de comparer les coefficient de regression de RoBoost-PLSR par rapport à PLSR sans les données aberrantes.

Cinquièmement, il serait possible d'utiliser RoBoost-PLSR dans des cas d'applications où les données sont d'une nature différente des données spectrales. Cela permettrait une ouverture intéressante vers d'autres disciplines de la chimie analytique comme la métabolomique.

Enfin, la présence de données aberrantes dans les contextes multi-tableaux serait intéressante à étudier. La définition d'outliers dans un contexte multi-tableau n'est pas trivial car des tableaux pourraient contenir des données aberrantes tandis que d'autres non.

# Bibliographie

[1] Peter Filzmoser, Sven Serneels, Ricardo Maronna, and Christophe Croux. Robust multivariate methods in Chemometrics. arXiv :2006.01617 [stat], pages 393–430, 2020. doi: 10.1016/B978-0-12-409547-2.14642-6. arXiv : 2006.01617.

[2] Maxime Metz, Florent Abdelghafour, Jean-Michel Roger, and Matthieu Lesnoff. A novel robust pls regression method inspired from boosting principles : Roboost-plsr. Analytica Chimica Acta, 1179 :338823, 2021. ISSN 0003-2670. doi: https://doi.org/10.1016/j.aca.2021.338823. URL https://www.sciencedirect.com/science/article/pii/S0003267021006498.

[3] Maxime Metz, Maxime Ryckewaert, Silvia Mas Garcia, Ryad Bendoula, Matthieu Lesnoff, and Jean-Michel Roger. Roboost-pls2-r : An extension of roboost-plsr method for multi-responses. Chemometrics and Intelligent Laboratory Systems, XXXX(under revision).

[4] Aldrig Courand, Maxime Metz, Daphné Héran, Carolen Feilhes, Fanny Prezman, Eric Serrano, Ryad Bendoula, and Maxime Ryckewaert. Evaluation of a robust regression method (roboost-plsr) to predict biochemical variables for agronomic applications : case study of grape berry maturity monitoring. Chemometrics and Intelligent Laboratory Systems, XXXX(under revision).

[5] S. Frosch Møller, J. von Frese, and R. Bro. Robust methods for multivariate data analysis. Journal of Chemometrics, 19(10) :549–563, 2005. ISSN 1099-128X. doi: 10.1002/cem.962. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.962. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.962.

[6] Peter J. Rousseeuw and Annick M. Leroy. Robust Regression and Outlier Detection : Rousseeuw/Robust Regression & Outlier Detection. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, USA, October 1987. ISBN 978-0-471-85233-9 978-0-471-72538-1. doi: 10.1002/0471725382. URL http://doi.wiley.com/10.1002/0471725382.

[7] D. Q. F. de Menezes, D. M. Prata, A. R. Secchi, and J. C. Pinto. A review on robust M-estimators for regression analysis. Computers & Chemical Engineering, 147 : 107254, 2021. ISSN 0098-1354. doi: https://doi.org/10.1016/j.compchemeng.2021.107254. URL https://www.sciencedirect.com/science/article/pii/S0098135421000326.

[8] Peter J. Rousseeuw. Least Median of Squares Regression. Journal of the American Statistical Association, 79(388) :871–880, December 1984. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.1984.10477105. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1984.10477105.

[9] S. Wold, H. Martens, and H. Wold. The multivariate calibration problem in chemistry solved by the pls method. In Bo Kågström and Axel Ruhe, editors, Matrix Pencils, pages 286–293, Berlin, Heidelberg, 1983. Springer Berlin Heidelberg. ISBN 978-3-540-39447-1.

[10] Sven Serneels, Christophe Croux, and Pierre J. Van Espen. Influence properties of partial least squares regression. Chemometrics and Intelligent Laboratory Systems, 71(1) :13–20, 2004. ISSN 0169-7439. doi: https://doi.org/10.1016/j.chemolab.2003.10.009.

[11] P. Filzmoser, S. Höppner, I. Ortner, S. Serneels, and T. Verdonck. Cellwise robust M regression. Computational Statistics & Data Analysis, 147 :106944, July 2020. ISSN 0167-9473. doi: 10.1016/j.csda.2020.106944.

[12] M.I. Griep, I.N. Wakeling, P. Vankeerberghen, and D.L. Massart. Comparison of semirobust and robust partial least squares procedures. Chemometrics and Intelligent Laboratory Systems, 29(1) :37–50, July 1995. ISSN 01697439. doi: 10.1016/0169-7439(95)80078-N.

[13] Ivana Stanimirova, Sven Serneels, Pierre J. Van Espen, and Beata Walczak. How to construct a multiple regression model for data with missing elements and outlying objects. Analytica Chimica Acta, 581(2) :324–332, January 2007. ISSN 0003-2670. doi: 10.1016/j.aca.2006.08.014.

[14] Randy J. Pell. Multiple outlier detection for multivariate calibration using robust statistical techniques. Chemometrics and Intelligent Laboratory Systems, 52(1) : 87–104, August 2000. ISSN 0169-7439. doi: 10.1016/S0169-7439(00)00082-4.

[15] Juan A. Gil and Rosario Romera. On robust partial least squares (PLS) methods. Journal of Chemometrics, 12(6) :365–378, 1998. ISSN 1099-128X. doi: 10.1002/(SICI)1099-128X(199811/12)12:6<365::AID-CEM519>3.0.CO;2-G.

[16] Sukru Acitas, Peter Filzmoser, and Birdal Senoglu. A new partial robust adaptive modified maximum likelihood estimator. Chemometrics and Intelligent Laboratory Systems, 204 :104068, September 2020. ISSN 01697439. doi: 10.1016/j.chemolab.2020.104068.

[17] Javier González, Daniel Peña, and Rosario Romera. A robust partial least squares regression method with applications. Journal of Chemometrics, 23(2) :78–90, 2009. ISSN 1099-128X. doi: 10.1002/cem.1195.

[18] I. N. Wakeling and H. J. H. Macfie. A robust PLS procedure. Journal of Chemometrics, 6(4) :189–198, July 1992. ISSN 0886-9383, 1099-128X. doi: 10.1002/cem.1180060404.

[19] Jiangtao Peng, Silong Peng, and Yong Hu. Partial least squares and random sample consensus in outlier detection. Analytica Chimica Acta, 719 :24–29, March 2012. ISSN 00032670. doi: 10.1016/j.aca.2011.12.058.

[20] Peter Filzmoser, Ricardo Maronna, and Mark Werner. Outlier identification in high dimensions. Computational Statistics & Data Analysis, 52(3) :1694–1711, January 2008. ISSN 0167-9473. doi: 10.1016/j.csda.2007.05.018.

[21] M. Hubert and K. Vanden Branden. Robust methods for partial least squares regression. Journal of Chemometrics, 17(10) :537–549, 2003. ISSN 1099-128X. doi: 10.1002/cem.822.

[22] Uwe Kruger, Yan Zhou, Xun Wang, David Rooney, and Jillian Thompson. Robust partial least squares regression : Part II, new algorithm and benchmark studies. Journal of Chemometrics, 22(1) :14–22, 2008. ISSN 1099-128X. doi: 10.1002/cem.1095. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.1095.

[23] Irene Hoffmann, Sven Serneels, Peter Filzmoser, and Christophe Croux. Sparse partial robust M regression. Chemometrics and Intelligent Laboratory Systems, 149 : 50–59, December 2015. ISSN 0169-7439. doi: 10.1016/j.chemolab.2015.09.019.

[24] M. Hubert and K. Vanden Branden. Robust methods for partial least squares regression. Journal of Chemometrics, 17(10) :537–549, October 2003. ISSN 0886-9383, 1099-128X. doi: 10.1002/cem.822.

[25] Sven Serneels, Christophe Croux, Peter Filzmoser, and Pierre J. Van Espen. Partial robust M-regression. Chemometrics and Intelligent Laboratory Systems, 79(1-2) : 55–64, October 2005. ISSN 01697439. doi: 10.1016/j.chemolab.2005.04.007.

[26] Martin Andersson. A comparison of nine PLS1 algorithms. Journal of Chemometrics, 23(10) :518–529, October 2009. ISSN 08869383, 1099128X. doi: 10.1002/cem.1248.

[27] Mia Hubert, Peter J Rousseeuw, and Karlien Vanden Branden. ROBPCA : A New Approach to Robust Principal Component Analysis. Technometrics, 47(1) :64–79, February 2005. ISSN 0040-1706, 1537-2723. doi: 10.1198/004017004000000563. URL http://www.tandfonline.com/doi/abs/10.1198/004017004000000563.

[28] Christopher M. Bishop. Pattern recognition and machine learning. Information science and statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.

[29] Irene Hoffmann, Peter Filzmoser, Sven Serneels, and Kurt Varmuza. Sparse and robust PLS for binary classification. Journal of Chemometrics, 30(4) :153–162, 2016. ISSN 1099-128X. doi: 10.1002/cem.2775.

# De nouvelles méthodes pour les données massives en chimiométrie

*Chapitre* 4

## Sommaire

## 4.1 Motivations et vue d'ensemble des travaux

Ces travaux ont pour objectif de combiner des paradigmes du big-data avec des paradigmes de la chimiométrie. La méthode parSketch-PLS est une nouvelle approche permettant d'obtenir de premiers résultats très satisfaisants en terme de performances de prédiction et de temps de calcul [1] . Cependant, cette méthode se heurte à quelques limitations. Parmi ces limitations, deux problématiques principales associées à la définition du voisinage et à la pertinence des voisins dans parSketch-PLS vont être discutées.

Premièrement, la définition d'un voisinage est discutée. En effet, si un ensemble de voisinages n'est pas bien défini pour la prédiction des individus, cela signifie alors que la métrique utilisée n'est pas la bonne. Dans le cadre de parSketch-PLS seule la distance Euclidienne est utilisée pour définir les voisinages des individus à prédire. Pour répondre à cette première problématique, il est proposé d'étudier une approche d'indexation qui permet d'utiliser une autre métrique. La prochaine section propose une démarche pour la réalisation d'indexations orientées. Pour cela, la PLS est combinée à la méthode d'indexation par grilles. L'objectif de ce travail est d'évaluer, dans le contexte des méthodes locales, l'intérêt de développer une indexation orientée.

Deuxièmement, la pertinence d'un voisin est discutée. En effet, certains individus de la base de données peuvent être définis comme voisins d'un point mais ne sont pas pertinents pour la calibration d'un modèle PLS permettant de prédire l'individu concerné. Ces voisins non pertinents vont donc potentiellement nuire à la calibration du modèle ce qui va engendrer des performances de prédiction faibles. Dans ce chapitre, il est proposé de combiner une méthode robuste avec parSketch afin d'étudier si cette démarche limite l'impact des voisins nuisibles à la calibration de modèle PLS.

## 4.2 Indexation orientée

### 4.2.1 Introduction

La définition des voisinages dans les méthodes locales est une étape clé. En effet, si les voisinages sélectionnés par la méthode PLS locale ne sont pas bien définis ils ne permettront pas d'obtenir de bonnes capacités prédictives. Cette bonne définition du voisinage est indépendante de la méthode KNN mais dépendante de la métrique choisie pour définir le voisinage. On peut observer dans [2] que la métrique n'a pas uniquement comme objectif de définir un groupe pertinent mais bien un ordonnancement des données

pertinent pour la construction d'un modèle local. La définition des voisins d'un individu en grande dimension n'est pas triviale car une infinité d'ordonnancements dans les données est possible. De nombreuses stratégies ont donc été proposées pour définir de nouvelles métriques [3–7]. Ces développements ont pour but de fournir un large panel de métriques pouvant répondre à un grand nombre de problématiques.

Les outils d'indexation sont très utiles dans le cadre des méthodes KNN-PLS car ils permettent de traiter des bases de données que l'algorithme usuel force-brute ne permet pas [1] . Cependant, les outils tels que parSketch ont pour objectif de retourner des voisinages selon une métrique spécifique. Dans le cas de parSketch, c'est la distance Euclidienne qui est approchée.

Dans le cas des méthodes KNN-PLS, il est nécessaire de développer de nouveaux algorithmes d'indexation qui utilisent des métriques utilisées couramment en chimiométrie. Les métriques utilisées en chimiométrie ont pour avantage de se baser sur des connaissances de la base de données à traiter et permettent d'obtenir des résultats très satisfaisants. Par exemple, une des métriques très utilisée en chimiométrie est la distance de Mahalanobis calculée à partir de scores PLS. Pour répondre à cette problématique, nous proposons de remplacer la méthode de réduction de dimension utilisée dans parSketch par la méthode PLS. Cette nouvelle stratégie s'appelle PLSgrid. Par la suite PLSgrid a été appliquée sur un jeu de données utilisé dans [1] et les performances de prédiction et le comportement de la méthode ont été comparés à parSketch-PLSDA.

### 4.2.2 PLSgrid

La méthode PLSgrid consiste à réaliser une réduction de dimension par PLS puis à construire une grille 1D sur chaque variable latente. La recherche de voisins sera réalisée de la même manière que la recherche de voisins dans parSketch. Un comptage du nombre de cellules en commun va être réalisé puis un seuil va être défini pour sélectionner les voisins. Cette démarche possède certaines propriétés qui nécessitent d'être discutées. Premièrement, la réduction de dimension par PLS a pour objectif d'orienter l'indexation. Cela signifie que les voisins ne sont pas uniquement définis à l'aide de $\mathbf{X}$ mais également de $\mathbf{Y}$. En effet, les sous-espaces définis par la PLS seront reliés à la matrice des réponses $\mathbf{Y}$. Deuxièmement, les grilles construites dans PLSgrid sont 1-D. En effet, ici, comme chaque variable est spécifique et n'est pas une combinaison linéaire aléatoire des variables de départ, il est difficile de pouvoir les combiner en un espace 2-D sans a priori. Troisièmement, cette démarche permet de se rapprocher d'une distance de Mahalanobis sur scores PLS. En effet, lorsqu'une grille 1-D est appliquée sur les variables latentes, la plage de variations des scores n'est pas prise en compte.

Ceci peut être assimilé à la distance de Mahalanobis qui permet de ne pas tenir compte de la variance des scores pour chaque variable latente. Quatrièmement, cette méthode possède un coût calculatoire plus conséquent que la méthode parSketch. En effet, la PLS nécessite une suite d'opérations matricielles et le coût calculatoire de cette dernière est bien plus grand que le coût calculatoire de la réduction par projection sur vecteurs aléatoires.

### 4.2.3 Matériels et méthodes

#### Données

Pour cette évaluation, le jeu de données utilisé dans [1] a été utilisé. Ce jeu de données est un jeu de données pour la classification créé à l'aide d'images hyperspectrales. La base de données initiale était constituée de 360 000 spectres de réflectance de feuilles de blé. Les individus mesurés appartenaient à quatre classes, correspondant à quatre génotypes différents. Les spectres ont été acquis à l'aide d'une caméra hyperspectrale, sur 256 longueurs d'onde allant de 410 à 1000 nm. Une image hyperspectrale a été acquise pour chaque classe. Chaque classe contenait 90 000 spectres issus d'une image. Pour chaque image, et donc pour chaque classe, 100 individus tests ont été sélectionnés avec la méthode Kennard-Stone appliquée sur les coordonnées des pixels dans l'image. Les individus spatialement voisins des individus tests (dans un carré de 7*7 pixels) ont été retirés de la base de données et tous les individus restants ont été placés dans la base d'étalonnage. La base de données résultante était constituée d'un jeu test de 400 individus et un jeu de calibration de 354 426 individus. Les échantillons de validation et de calibration étaient les mêmes que ceux utilisés dans [1]. Les résultats obtenus dans cette étude sont donc comparables aux résultats obtenus dans [1].

#### Méthodes de prédiction

Dans cette étude, trois méthodes de prédiction sont comparées : la méthode BF-PLSDA, parSketch-PLSDA (résultats présents dans [1]) et une nouvelle stratégie PLSgrid-PLSDA.

Pour la méthode PLSgrid, les trois paramètres $v$ (nombre de variables latentes), $s$ (nombre de segments) et $m$ (pourcentage minimal de cellules en commun), ont pris différentes valeurs : $\{5, 10, 20, 30\}$ pour $v$, $\{5, 7, 9, 11, 15\}$ pour $s$ et $\{50, 70, 90\}$ pour $m$. Pour ce troisième modèle, trois critères d'évaluation des paramètres de PLSgrid ont été sélectionnés. Premièrement, le nombre de voisins prédictibles a été étudié, c'est-à-dire le nombre de points tests qui possédaient plus de trente voisins. Deuxièmement, la distribution des voisinages a été observée. Troisièmement, l'erreur de

prédiction de PLSgrid-PLSDA a été observée sur quatre combinaisons des paramètres $v$, $s$, $m$.

### 4.2.4 Résultats et discussions

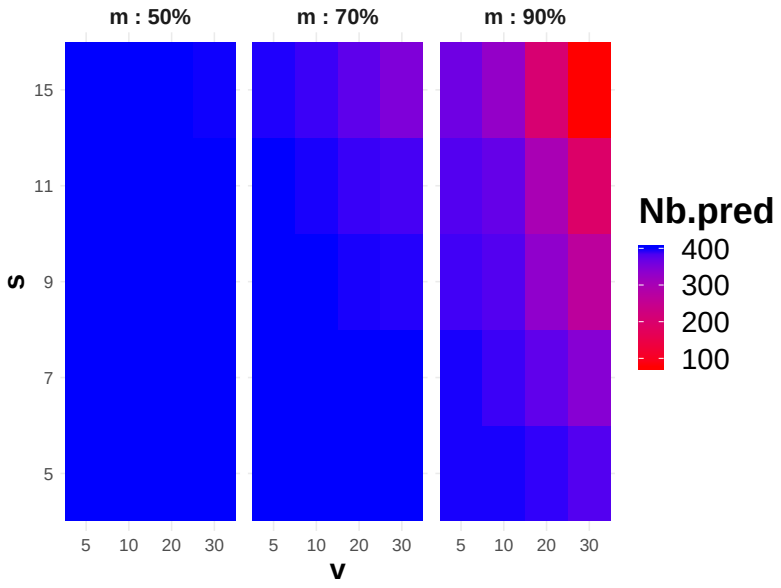



FIGURE 4.1 – Heatmap du nombre de points prédictibles trouvés par parSketch, en fonction des trois paramètres $v$, $s$, $m$ . La figure est divisée en trois graphiques, chaque graphique correspond à une valeur du paramètre $m$.

La figure 4.1 montre que le nombre de points prédictibles diminue lorsque la valeur de $m$ augmente. Le paramètre $m$ permet de sélectionner les points les plus souvent présents dans la même case que l'individu à prédire. La méthode PLSgrid construit des grilles sur des scores PLS. Il est très peu probable d'obtenir un grand nombre de voisins si le seuil $m$ est trop élevé car tous les individus de la base de données peuvent se distinguer par certains de leurs scores.

D'autre part, la figure 4.1 montre que le nombre de points prédictibles diminue lorsque les valeurs de $s$ et $v$ augmentent. Le paramètre $s$ définit le nombre de cellules dans chaque grille. Quand le nombre de cases de chaque grille augmente, les cases sont plus petites et donc contiennent moins de points. Le nombre de variables latentes $v$ impacte également le nombre de points prédictibles car plus il sera grand plus le nombre de cellules minimales en commun entre les individus de calibration et de test doit être grand.

Les paramètres $s$ et $m$ ont donc un impact fort sur le nombre de points prédictibles alors que $v$ a un impact faible (indirect) sur le nombre de points prédictibles. Le

comportement observé du nombre de points prédictibles en fonction des paramètres de PLSgrid est similaire au comportement de parSketch dans [1]. Il est donc possible de conclure que ce comportement est dû à la méthode d'indexation et non à la méthode de réduction de dimension.

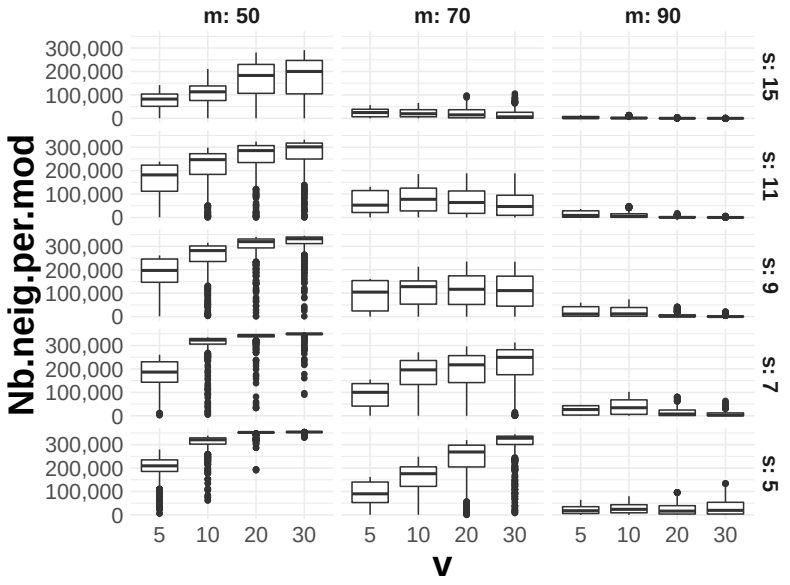



FIGURE 4.2 – Distribution du nombre de voisins par échantillon de test en fonction des paramètres de PLSgrid ($v$, $m$, $s$).

La figure 4.2 montre que lorsque $v$ augmente (pour $m = 50{,}70$), le nombre médian de voisins augmente et l'écart interquartile des voisinages diminue. Cette observation peut se faire lorsque le nombre minimal de grilles $m$ et le nombre de segments $s$ ont des valeurs faibles. En effet ici, contrairement à parSketch, $v$ possède un impact fort sur les voisinages renvoyés. Contrairement aux sketchs qui sont générés sans a priori, les scores-PLS le sont, cela signifie donc que chaque nouvelle dimension utilisée dans PLSgrid est spécifique et ajoute une variable indépendante des autres. Dans ce contexte, cette stratégie peut également souffrir du fléau de la dimensionnalité [8] car ici chaque nouvelle variable est indépendante de l'autre contrairement aux variables des sketchs qui cherchent chacune à approcher la distance Euclidienne entre individus.

De plus, la figure 4.2 montre que lorsque $s$ augmente, le nombre médian de voisins par point à prédire diminue. En effet, $s$ définit le nombre de cases de chaque grille, plus $s$ aura une forte valeur plus il y aura de cases et donc moins il y aura de points dans chaque case. Quand $s$ augmente, l'écart interquartile des voisinages diminue excepté pour $s = 15$. En effet, plus la segmentation est forte moins il y aura une influence de la structure de la position des points à prédire dans la base de données.

Enfin, la figure 4.2 montre que lorsque $m$ augmente, le nombre médian de voisins par

point diminuent. $m$ est un seuil, plus la valeur de $m$ sera élevée plus les points renvoyés par parSketch seront similaires. Donc, si la valeur de $m$ est élevée, moins de voisins seront retournés par parSketch.

Pour conclure, contrairement à parSketch, $v$ possède un impact fort sur la taille des voisinages renvoyés. Dans un cas extrême ($v$=30 , $m$=50, $s$=5), l'ensemble de la base de données est renvoyé. Ici, il est donc possible d'observer que $v$ est un paramètre primordial à faire varier car il impacte fortement les voisinages renvoyés. Les deux autres paramètres associés à la méthode de hachage utilisée dans PLSgrid se comportent de la même manière que les paramètres $m$ et $s$ de parSketch.

Afin de tester PLSgrid, quatre combinaisons de paramètres de PLSgrid ont été choisies pour la calibration de PLSgrid-PLSDA (voir table 4.1).

TABLE 4.1 – Les différents choix de combinaisons des paramètres PLSgrid

| Combinaison | $m$ | $v$ | $s$ |
|---|---|---|---|
| 1 | 70 | 5 | 7 |
| 2 | 70 | 5 | 9 |
| 3 | 70 | 10 | 9 |
| 4 | 70 | 5 | 11 |

Ces combinaisons de paramètres ont été choisies car les voisinages obtenus sont réduits et tous les points sont prédictibles.

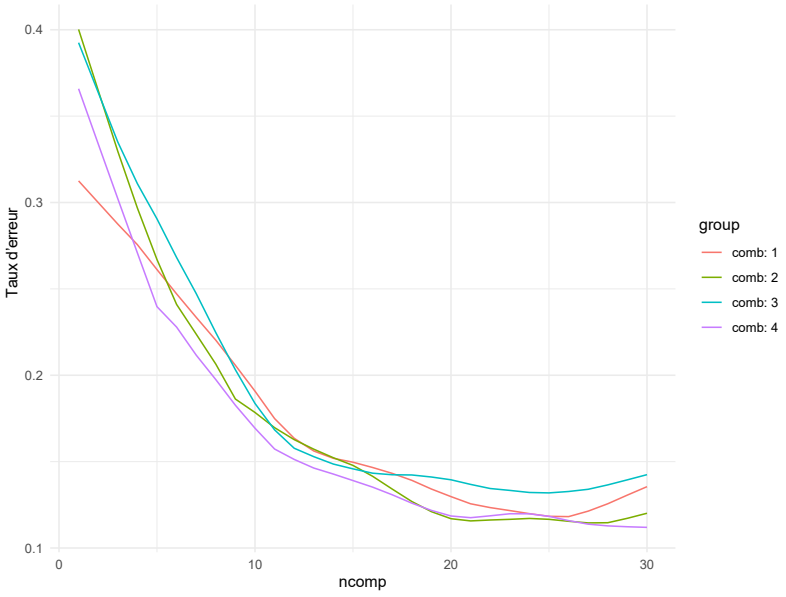



FIGURE 4.3 – Erreur de classification de la méthode : PLSgrid-PLSDA en fonction du nombre de variables latentes PLS, pour les quatre combinaisons de paramètres choisies

Sur la figure 4.3, les courbes d'erreur de PLSgrid-PLSDA sont toutes très proches et fournissent un optimal d'approximativement 11,5% d'erreur sauf pour la troisième combinaison de paramètres qui est moins performante et avoisine une performance de prédiction optimale de 13%. Ce résultat peut s'expliquer par une variation de $v$ qui entraîne un voisinage basé sur une métrique différente. En effet, les trois autres combinaisons sont basées sur une PLS à 5 variables latentes tandis que la troisième combinaison utilise 10 variables latentes pour définir le voisinage.

Cette approche dans son formalisme actuel ne permet pas d'obtenir d'aussi bons résultats que la méthode parSketch-PLSDA qui avait un résultat optimal proche de 10% [1]. De plus, cette approche utilise une stratégie de réduction de dimension plus coûteuse en temps de calcul et donc il est difficile de justifier le développement d'une telle approche dans sa forme actuelle. Au vu des différents résultats, la méthode d'indexation par grilles ne semble pas être la plus pertinente pour être utilisée sur des scores-PLS. D'autres outils comme les arbres-R pourraient substituer les grilles et permettre de mieux approcher les voisinages. Néanmoins, les résultats de cette approche mettent en avant la pertinence d'utiliser la PLS comme outil de réduction de dimension combiné à des approches d'indexations. En effet, cette approche permet tout de même une meilleure performance de prédiction que la PLSDA globale qui avait un résultat proche de 24% d'erreur.

### 4.2.5 Conclusion

Cette étude montre que PLSgrid-PLSDA approche les résultats de parSketch-PLSDA sur l'exemple traité (discrimination de génotypes de feuilles de blé par imagerie hyperspectrale). La méthode PLSgrid-PLSDA n'est pas en mesure d'obtenir d'aussi bons résultats de classification que la méthode parSketch-PLSDA mais permet d'obtenir de meilleurs résultats qu'une PLSDA globale. Il est possible de formuler l'hypothèse que la PLS est une alternative potentielle à la projection sur vecteurs aléatoires car elle permettrait d'orienter le voisinage sélectionné. Cela signifie donc que par cette stratégie il est possible de représenter une autre métrique que les distances Euclidiennes sur spectres. Cependant, cette approche ne permet pas d'obtenir dans ce cas des performances de prédiction équivalentes à la méthode BF-PLSDA. De plus, contrairement à la méthode BF-PLSDA qui utilise force-brute, l'approche proposée ne permet pas d'estimer des voisins sur un très grand nombre de métriques.

Pour développer cette stratégie d'indexation orientée, différents développements seraient pertinents. Premièrement, la PLS pourrait être combinée avec d'autres méthodes d'indexation comme les arbres R. En effet, contrairement à la réduction sur vecteurs aléatoires, les variables latentes possèdent un ordre d'importance et donc appliquer des

méthodes d'indexation par grilles sur des variables latentes peut s'avérer moins pertinent. Deuxièmement, il serait également possible de combiner la méthode PLSgrid-PLSDA avec la méthode force-brute afin de réduire le voisinage et sélectionner un voisinage plus pertinent. Troisièmement, il serait pertinent d'utiliser des outils d'approximation du voisinage permettant d'approcher n'importe quelle métrique comme les méthodes iDistance [9], M-index [10] ou top-k [11]. Ces méthodes se basent sur la construction de points pivots permettant d'utiliser n'importe quelle métrique. Enfin, il serait intéressant d'évaluer des méthodes de régression et de classification issues du machine learning basées sur des méthodes arborescentes comme les méthodes CART (Classification And Regression Tree) [12].

## 4.3 parSketch-RoBoost-PLSR

### 4.3.1 Introduction

Malgré un grand nombre de critères de similarité proposé dans la littérature, certains voisins peuvent ne pas être pertinents au sein d'un voisinage. En effet, il est possible que certains paramètres des méthodes PLS locales définis pour l'ensemble des échantillons à prédire, ne correspondent pas à chaque individu. Par exemple, dans le cas des méthodes BF-PLS, le nombre de voisins renvoyés pour faire la calibration est un paramètre à ajuster. Cependant, pour certains individus à prédire la taille optimale du voisinage n'est pas la même que pour les autres individus à prédire. Cela peut donc entraîner de mauvaises prédictions pour certains individus. Dans [2] il est proposé de réaliser des modèles linaires robustes afin de limiter ou bien supprimer l'impact de certains voisins non pertinents sur les modèles calibrés. Avec le traitement de masses de données et l'utilisation de stratégies comme parSketch, ce type d'approche devient de plus en plus pertinent. En effet, parSketch retourne un voisinage approximatif et également très grand [1]. Il est donc possible que certains voisins ne soient pas pertinents voire même nuisibles pour la calibration de modèles PLS. Une solution permettant de limiter voire supprimer l'impact de potentiels voisins nuisibles sur la calibration est de combiner parSketch avec une méthode PLS robuste.

Dans cette étude, la méthode RoBoost-PLSR proposée dans [13] a été combinée à la méthode parSketch. Afin d'évaluer cette combinaison, parSketch-RoBoost-PLSR a été comparée avec parSketch-PLSR et BF-PLSR. Pour réaliser cette comparaison, les méthodes ont été appliquées et étudiées à l'aide d'un jeu de données simulé.

### 4.3.2 Matériels et méthodes

#### Données

Pour évaluer les performances de parSketch-RoBoost-PLSR par rapport à parSketch-PLSR et BF-PLSR, une simulation a été effectuée. La simulation représentait une base de données de 100 000 échantillons. Les 100 000 échantillons ont été générés selon le cadre proposé par [14]. En utilisant cette approche, la matrice des variables explicatives $\mathbf{X}$ a été calculée comme suit :

$$\mathbf{X} = \mathbf{T_u}\mathbf{P_u} + \mathbf{T_d}\mathbf{P_d} + \mathbf{E} \tag{4.1}$$

Et la relation $f$ entre $\mathbf{T_u}$ et $\mathbf{y}$ par :

$$\mathbf{y} = f(\mathbf{T_u}) + \mathbf{F} \tag{4.2}$$

Où $\mathbf{P_u}$ et $\mathbf{P_d}$ sont des signatures spectrales et $\mathbf{T_u}$ et $\mathbf{T_d}$ leurs contributions associées. Les matrices $\mathbf{E}$ et $\mathbf{F}$ sont définies comme des bruits gaussiens de $\mathbf{X}$ et $\mathbf{y}$, respectivement. L'indice $u$ est associé à l'espace utile et $d$ à l'espace nuisible.

Parmi ces échantillons, cinq groupes de données distincts de même taille sont représentés. Chacun de ses groupes partage des signatures spectrales communes qui sont les spectres purs de l'eau, du glucose et de l'éthanol plus le spectre d'interaction eau éthanol. Les signatures spectrales utilisées pour les simulations étaient les signatures spectrales de l'eau, de l'éthanol et du glucose estimées dans [14]. Les cinq groupes se distinguent spectralement par vingt autres signatures spectrales simulées, différentes pour chaque groupe. De plus, pour chaque groupe, les matrices $\mathbf{T_u}$ et $\mathbf{T_d}$ ont été simulées à partir de lois normales repliées (F-n distributions) ayant des paramètres différents. Pour finir, la relation entre les données spectrales et la mesure de référence (la fonction $f(.)$) est différente au sein de chaque groupe. Les paramètres de simulation sont résumés dans le tableau 4.2.

TABLE 4.2 – Les choix des simulations pour les cinq groupes de données (G1 à G5)

| | G1 | G2 | G3 | G4 | G5 |
|---|---|---|---|---|---|
| $\mathbf{P}_u$ | Pure spectrum of water | | | | |
| $\mathbf{T}_u$ | F-n distribution | F-n distribution | F-n distribution | F-n distribution | F-n distribution |
| $\mathbf{P}_d$ | Pure spectrum of glucose<br>Pure spectrum of ethanol<br>Spectrum of water-ethanol Interaction | | | | |
| | 20 artificial spectra | 20 artificial spectra | 20 artificial spectra | 20 artificial spectra | 20 artificial spectra |
| $\mathbf{T}_d$ | F-n distribution<br>F-n distribution<br>Product between $T_{water}$ and $T_{ethanol}$ | | | | |
| | F-n distribution | F-n distribution | F-n distribution | F-n distribution | F-n distribution |
| $\mathbf{E}$ | Gaussian distribution | Gaussian distribution | Gaussian distribution | Gaussian distribution | Gaussian distribution |
| $f$ | $Y_{g1} = 0.2 * T_{eau}$ | $Y_{g2} = 4 * T_{eau}$ | $Y_{g3} = 20 * T_{eau}$ | $Y_{g4} = 10 * T_{eau}$ | $Y_{g5} = 1 * T_{eau}$ |
| $\mathbf{F}$ | Gaussian distribution | Gaussian distribution | Gaussian distribution | Gaussian distribution | Gaussian distribution |

F-n distribution : Folded normal distribution

### Stratégie d'évaluation

Les méthodes BF-PLSR, parSketch-PLSR et parSketch-RoBoost-PLSR ont été comparées à l'aide d'un même jeu de test de 300 individus. La méthode de référence est la méthode BF-PLSR. Cette méthode consiste à sélectionner les k plus proches voisins des individus de validation puis de calculer un modèle PLSR pour chaque point à prédire. Ces voisins ont été définis par la distance Euclidienne entre l'individu à prédire et les individus de la base de données. Pour parSketch-PLSR et parSketch-RoBoost-PLSR, les mêmes paramètres de parSketch ont été fixés. Pour les deux méthodes, les trois paramètres de parSketch ($v$, $s$, $m$), ont pris différentes valeurs : $v$ : {20,30,50}, $s$ : {7,8,9,10,11}, $m$ : {50,70,80}, respectivement. Pour paramétrer les deux méthodes, la distribution des voisinages a été observée. Pour parSketch-RoBoost-PLSR des paramètres de RoBoost-PLSR ont été varié selon la table 4.3. Les paramètres $\alpha$, $\beta$, $\gamma$ sont les paramètres de la fonction de poids bisquare utilisée dans [13; 15; 16].

TABLE 4.3 – Les différents choix de combinaison des paramètres RoBoost-PLSR

| Combinaison | $\alpha$ | $\beta$ | $\gamma$ |
|---|---|---|---|
| 1 | Inf | 6 | Inf |
| 2 | Inf | 4 | Inf |
| 3 | Inf | 8 | Inf |
| 4 | Inf | 6 | 6 |
| 5 | 6 | 6 | Inf |
| 6 | 4 | 6 | Inf |

### 4.3.3 Résultats et discussions

#### Visualisation des données

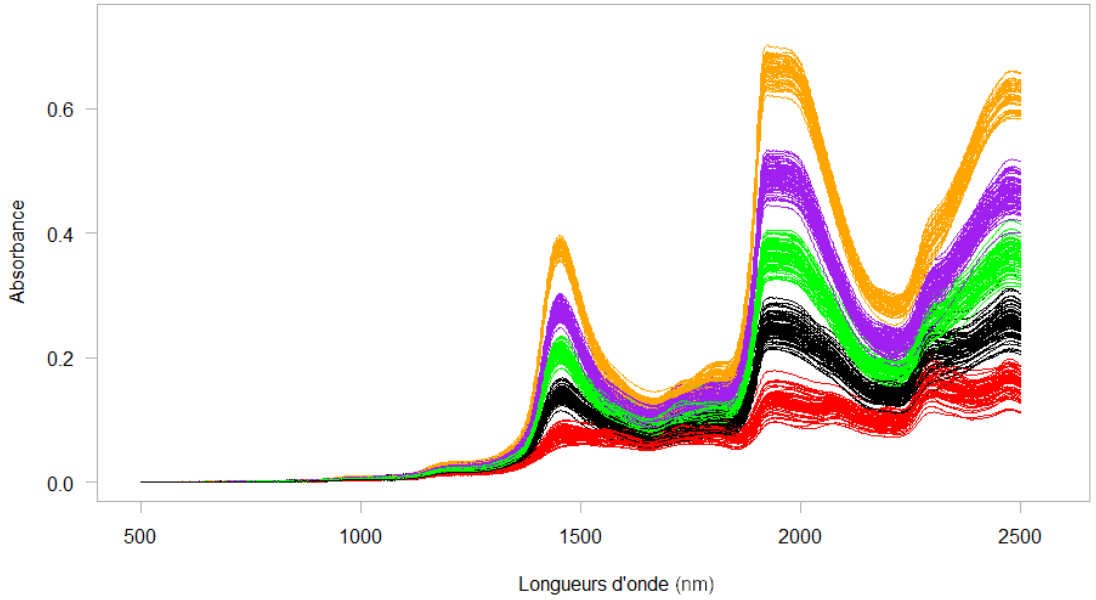



FIGURE 4.4 – Spectres simulés appartenant aux cinq groupes de la base de données

La figure 4.4 montre des spectres d'absorbance tirés aléatoirement dans la base de données. Il est observé que les spectres possèdent une forme principales correspondant au spectre de l'eau. En effet, chaque groupe peut se distinguer par sa proportion en eau. Dans ce cas-ci, il est alors possible de formuler l'hypothèse que la distance Euclidienne sera une bonne métrique pour permettre de sélectionner un voisinage pertinent pour la calibration de modèles PLSR. Cela signifie donc que parSketch sera également adapté à cette problématique.

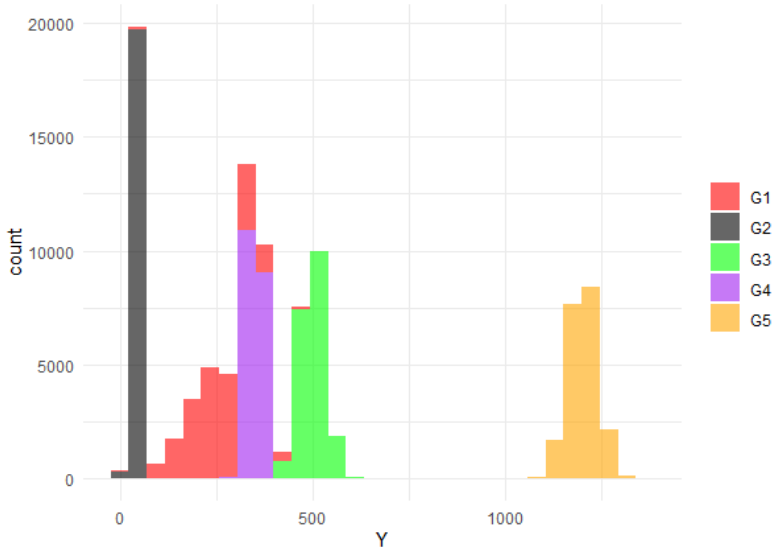



FIGURE 4.5 – Histogramme de $\mathbf{y}$

La figure 4.5 montre l'histogramme des valeurs de $\mathbf{y}$. Il est possible d'observer les cinq groupes de données avec les groupes G2, G3, G4 qui possèdent des distributions qui se confondent. Le groupe G2 est confondu avec une partie du groupe G1. Le groupe G5 à une distribution de valeurs très différente des autres groupes. Au vu de la distribution des valeurs de $\mathbf{y}$ et des observations réalisées sur les spectres, l'utilisation de méthodes PLS locales est justifiée.

### Evaluation de la méthode de référence BF-PLSR

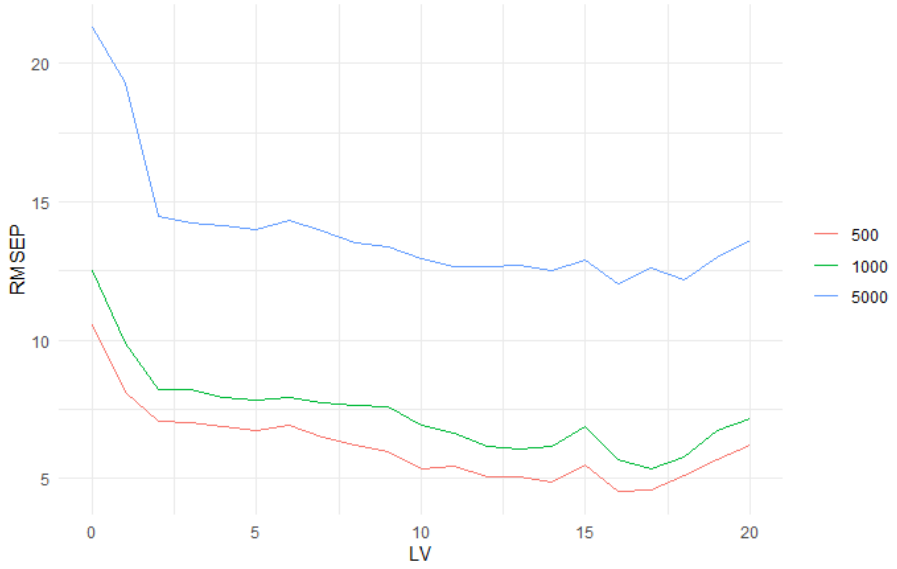



FIGURE 4.6 – Erreur de prédiction de la méthode BF-PLSR : avec un nombre de voisins par point défini à 500, 1000, 5000, en fonction du nombre de variables latentes PLS

La figure 4.6 présente l'erreur de prédiction en fonction du nombre de variables latentes de la méthode BF-PLSR pour 500, 1000 et 5000 voisins. Il est possible d'observer que la BF-PLSR à 16 variables latentes pour 500 voisins est la plus performante. Malgré le fait que chaque groupe de données est représenté par 20 000 individus, la méthode BF-PLSR performe au mieux avec 500 voisins. Cela signifie donc qu'il est possible que certains individus de test se situent aux frontières de chaque groupe et qu'alors il est difficile de définir un grand nombre de voisins pertinents pour l'ensemble des individus tests.

Comme observé dans les dernières études, parSketch renvoie généralement un voisinage très grand si l'on souhaite obtenir un nombre minimal de voisins pour chaque individu test. Il est donc également possible de formuler l'hypothèse que la méthode parSketch-PLSR performera moins bien que la méthode BF-PLSR car la méthode parSketch-PLSR sera contrainte de calibrer les modèles PLSR sur de grands ensembles de voisins.

### Paramétrage de parSketch

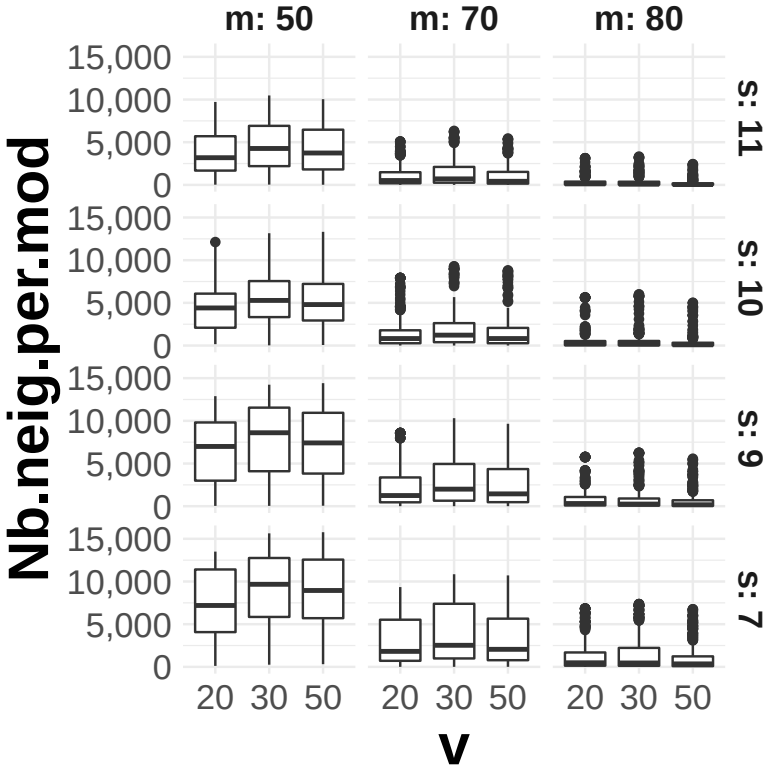



FIGURE 4.7 – Distribution du nombre de voisins par échantillon de test en fonction des paramètres de parSketch ($v$, $m$, $s$).

La figure 4.7 montre l'impact des valeurs des paramètres de parSketch sur la taille des voisinages de chaque individu à prédire. Il est possible de tirer les même conclusions de l'impact des paramètres de parSketch sur la taille des voisinages que pour les différentes études [1; 17]. Cela signifie que le nombre de segments $s$ et le % minimal de grilles $m$ impactent fortement la taille des voisinages. Comme pour les précédentes études, il est choisi de sélectionner la combinaison de paramètres de parSketch qui permet d'obtenir un voisinage de taille réduite tout en pouvant prédire encore l'ensemble des individus. Par la suite, la combinaison de paramètres $v = 30$, $s = 11$, $m = 50$ a été choisie. Cette combinaison permettait de renvoyer un nombre médian de 3500 voisins par individu à prédire.

### Évaluation de parSketch-PLSR et parSketch-RoBoost-PLSR

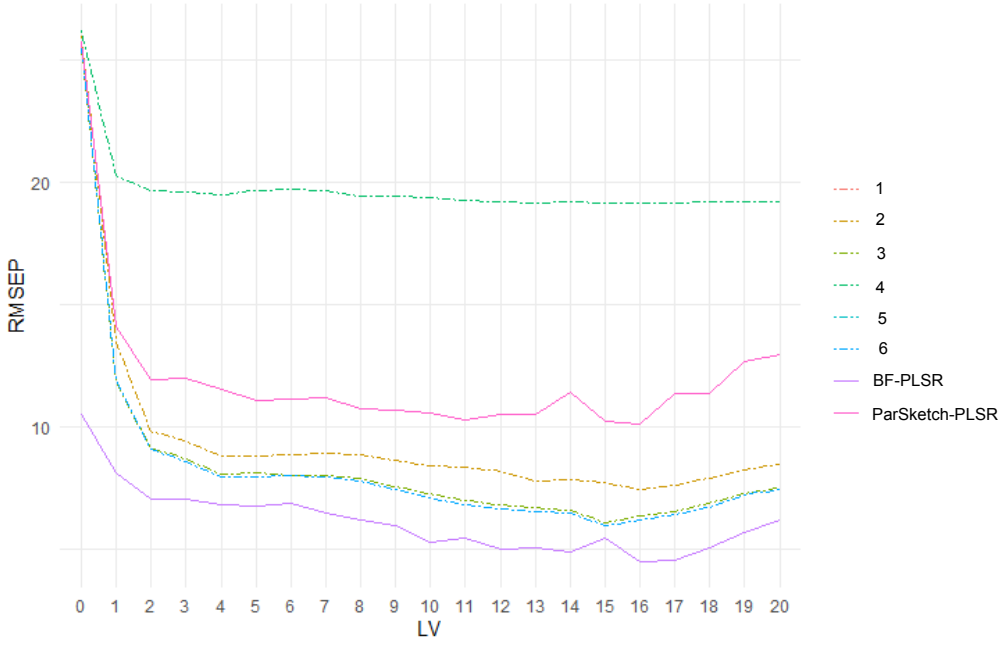



FIGURE 4.8 – Erreur de prédiction des méthodes : parSketch-RoBoost-PLSR, parSketch-PLSR et BF-PLSR (avec k = 500). Pour parSketch-RoBoost-PLSR six combinaisons de paramètres ont été testées (numérotées de 1 à 6 et détaillées en table 4.3).

La figure 4.8 montre la RMSEP en fonction du nombre de variables latentes pour les trois méthodes. Il est possible d'observer que la méthode parSketch-PLSR possède une RMSEP proche de 10 pour 16 LV. Pour la méthode BF-PLSR, la RMSEP la plus faible est inférieure à 5 pour 16 LV. Cela signifie donc que la méthode parSketch-PLSR n'a pas sélectionnée uniquement les voisins pertinents pour la calibration des modèles PLSR.

Il es également possible d'observer que les performances de la méthode parSketch-RoBoost-PLSR pour les six combinaisons se situent pour la plupart entre BF-PLSR et parSketch-PLSR. Seule la combinaison 4 possède des capacités prédictives moins bonnes que parsketch-PLSR, ceci peut être dû à l'utilisation de valeurs des paramètres trop strictes. Pour finir, on observe que les performances de prédiction de parSketch-RoBoost-PLSR pour les combinaisons $\{1,2,3,5,6\}$ sont meilleures que les performances de prédiction de parSketch-PLSR.

En conclusion, on observe que lorsque certains voisins ne sont pas pertinents pour la calibration de modèles PLSR, il est intéressant d'utiliser une approche robuste pour pondérer certains voisins et permettre une meilleure capacité prédictive de la méthode.

### 4.3.4 Conclusion

Dans cette étude comparative il a été montré que certains voisins peuvent être nuisibles à la calibration de modèles PLS bien que la métrique choisie soit pertinente. Pour résoudre cette problématique, une solution envisageable est d'utiliser une approche robuste pour réduire voire éliminer l'impact de ces voisins nuisibles sur les capacités prédictives de l'approche. Cette étude a été réalisée dans un contexte où la taille de la base de données était constituée d'un grand nombre d'individus. Ceci justifie donc pleinement l'utilisation des outils comme parSketch afin de sélectionner les plus proches voisins. Dans cet exemple parSketch et RoBoost-PLSR sont combinés. Pour traiter des bases de données plus classiques (non massives) en chimiométrie, il est également possible d'utiliser RoBoost-PLSR pour améliorer les approches comme BF-PLSR.

Bien que cette approche se montre très pertinente et que la méthode RoBoost-PLSR permet de réduire voire éliminer l'effet nuisible de certains voisins sur la qualité prédictive de parSketch-RoBoost-PLSR, différents développements ou études supplémentaires devront être réalisés afin d'utiliser cet outil dans un contexte big-data. Une des problématiques majeure est que les méthodes robustes telles que RoBoost-PLSR ont un coût calculatoire supérieur à la méthode PLSR. Ceci peut donc entraîner une perte d'efficacité de la méthode si les voisinages renvoyés sont trop grands. Pour résoudre cette problématique, différentes stratégies sont possibles.

Premièrement, il est possible comme pour les approches de boosting, de sous échantillonner les voisinages renvoyés par parSketch. En effet, si le voisinage renvoyé est très grand, les temps de calcul de la méthode RoBoost-PLSR seront rédhibitoires. Sous échantillonner permet de réduire les temps de calcul mais peut diminuer la qualité prédictive de la méthode.

Deuxièmement, il serait également pertinent de développer une approche

massivement parallélisable de RoBoost-PLSR afin de permettre de traiter de grands ensembles de données rapidement. En effet, le calcul matriciel est fortement parallélisable et cette parallélisation est rendue très puissante grâce à du calcul intensif sur GPU (graphics processing unit).

Troisièmement, il serait intéressant de combiner des approches d'indexation basées sur clustering telles qu'iSAX avec RoBoost-PLSR. En effet, une des problématiques principales dans l'utilisation de parSketch-PLSR est le fait qu'il y ait un modèle calibré par individu à prédire. Cette stratégie peut être rédhibitoire si les modèles à calculer possèdent un coût calculatoire élevé comme RoBoost-PLSR. Dans ce contexte, des outils comme iSAX qui permettent des approches de clustering, permettent de ne pas avoir à calculer un modèle par individu à prédire. En effet, iSAX structure les individus de la base de données en cluster, puis l'individu à prédire va être positionné dans un cluster. Dans ce contexte, les modèles peuvent être calibrés avant la recherche des plus proches voisins.

# Bibliographie

[1] Maxime Metz, Matthieu Lesnoff, Florent Abdelghafour, Reza Akbarinia, Florent Masseglia, and Jean-Michel Roger. A "big-data" algorithm for KNN-PLS. Chemometrics and Intelligent Laboratory Systems, 203 :104076, August 2020. ISSN 0169-7439. doi: 10.1016/j.chemolab.2020.104076. URL http://www.sciencedirect.com/science/article/pii/S0169743920301908.

[2] William S. Cleveland. LOWESS : A Program for Smoothing Scatterplots by Robust Locally Weighted Regression. The American Statistician, 35(1) :54, February 1981. ISSN 00031305. doi: 10.2307/2683591. URL https://www.jstor.org/stable/2683591?origin=crossref.

[3] Sanghong Kim, Manabu Kano, Hiroshi Nakagawa, and Shinji Hasebe. Estimation of active pharmaceutical ingredients content using locally weighted partial least squares and statistical wavelength selection. International Journal of Pharmaceutics, 421 (2) :269–274, 2011. ISSN 0378-5173. doi: https://doi.org/10.1016/j.ijpharm.2011.10.007. URL https://www.sciencedirect.com/science/article/pii/S0378517311009021.

[4] Koji Hazama and Manabu Kano. Covariance-based locally weighted partial least squares for high-performance adaptive modeling. Chemometrics and Intelligent Laboratory Systems, 146 :55–62, August 2015. ISSN 0169-7439.

doi: 10.1016/j.chemolab.2015.05.007. URL http://www.sciencedirect.com/science/article/pii/S0169743915001203.

[5] Guanghui Shen, Matthieu Lesnoff, Vincent Baeten, Pierre Dardenne, Fabrice Davrieux, Hernan Ceballos, John Belalcazar, Dominique Dufour, Zengling Yang, Lujia Han, and Juan Antonio Fernández Pierna. Local partial least squares based on global PLS scores. Journal of Chemometrics, 33(5) :e3117, 2019. ISSN 1099-128X. doi: 10.1002/cem.3117. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.3117.

[6] Tom Fearn and Anthony M.C. Davies. Locally-Biased Regression. Journal of Near Infrared Spectroscopy, 11(6) :467–478, December 2003. ISSN 0967-0335. doi: 10.1255/jnirs.397. URL https://doi.org/10.1255/jnirs.397.

[7] Xinmin Zhang, Manabu Kano, and Zhihuan Song. Optimal Weighting Distance-Based Similarity for Locally Weighted PLS Modeling. Industrial & Engineering Chemistry Research, 59(25) :11552–11558, June 2020. ISSN 0888-5885, 1520-5045. doi: 10.1021/acs.iecr.9b06847. URL https://pubs.acs.org/doi/10.1021/acs.iecr.9b06847.

[8] Christopher M. Bishop. Pattern recognition and machine learning. Information science and statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.

[9] H. V. Jagadish, Beng Chin Ooi, Kian-Lee Tan, Cui Yu, and Rui Zhang. iDistance : An adaptive B $^{+}$ -tree based indexing method for nearest neighbor search. ACM Transactions on Database Systems, 30(2) :364–397, June 2005. ISSN 0362-5915, 1557-4644. doi: 10.1145/1071610.1071612. URL https://dl.acm.org/doi/10.1145/1071610.1071612.

[10] David Novak, Michal Batko, and Pavel Zezula. Metric Index : An efficient and scalable solution for precise and approximate similarity search. Information Systems, 36(4) :721–733, June 2011. ISSN 0306-4379. doi: 10.1016/j.is.2010.10.002. URL https://www.sciencedirect.com/science/article/pii/S0306437910001109.

[11] Reza Akbarinia, Esther Pacitti, and Patrick Valduriez. Best Position Algorithms for Top-k Queries. page 495. ACM, August 2007. URL https://hal.inria.fr/inria-00378836.

[12] Leo Breiman, editor. Classification and regression trees. Chapman & Hall/CRC, Boca Raton, Fla., 1. crc press repr edition, 1998. ISBN 978-0-412-04841-8.

[13] Maxime Metz, Florent Abdelghafour, Jean-Michel Roger, and Matthieu Lesnoff. A novel robust pls regression method inspired from boosting principles : Roboost-plsr. Analytica Chimica Acta, 1179 :338823, 2021. ISSN 0003-2670. doi: https://doi.org/10.1016/j.aca.2021.338823. URL https://www.sciencedirect.com/science/article/pii/S0003267021006498.

[14] Maxime Metz, Alessandra Biancolillo, Matthieu Lesnoff, and Jean-Michel Roger. A note on spectral data simulation. Chemometrics and Intelligent Laboratory Systems, 200 :103979, May 2020. ISSN 0169-7439. doi: 10.1016/j.chemolab.2020.103979. URL http://www.sciencedirect.com/science/article/pii/S0169743919306720.

[15] Maxime Metz, Maxime Ryckewaert, Silvia Mas Garcia, Ryad Bendoula, Matthieu Lesnoff, and Jean-Michel Roger. Roboost-pls2-r : An extension of roboost-plsr method for multi-responses. Chemometrics and Intelligent Laboratory Systems, XXXX(under revision).

[16] Courand Aldrig, Metz Maxime, Héran Daphné, Feilhes Carolen, Prezman Fanny, Serrano Eric, Bendoula Ryad, and Maxime Ryckewaert. Evaluation of a robust regression method (roboost-plsr) to predict biochemical variables for agronomic applications : case study of grape berry maturity monitoring. Chemometrics and Intelligent Laboratory Systems, XXXX(under revision).

[17] Maxime Ryckewaert, Maxime Metz, Daphné Héran, Pierre George, Bruno Grèzes-Besset, Reza Akbarinia, Jean-Michel Roger, and Ryad Bendoula. Massive spectral data analysis for plant breeding using parsketch-plsda method : Discrimination of sunflower genotypes. Biosystems Engineering, 210 :69–77, 2021. ISSN 1537-5110. doi: https://doi.org/10.1016/j.biosystemseng.2021.08.005. URL https://www.sciencedirect.com/science/article/pii/S1537511021001896.

# Chapitre 5 Conclusion générale

## 5.1 Résumé et perspectives des travaux

Aujourd'hui, une grande quantité de données peut être générée en agronomie. Les chimiométriciens doivent désormais être en mesure d'analyser ces bases de données. Les outils usuels de la chimiométrie ne sont pas encore opérationnels. Une solution envisagée pour répondre à cette problématique est la parallélisation et l'utilisation d'outils à coût calculatoire faible. L'objectif de cette thèse était donc d'introduire le traitement des données massives en chimiométrie. Pour ce faire, trois axes de recherches associés au traitement des données massives pour la chimiométrie ont été étudiés.

Le premier axe de recherche consistait à étudier comment permettre le traitement des données massives par les méthodes PLS locales. Actuellement, les méthodes locales ne passent pas à l'échelle, c'est-à-dire qu'elles ne sont pas en mesure de traiter de grands ensembles de données. En effet, le coût calculatoire de l'algorithme force-brute utilisé en chimiométrie pour calculer les plus proches voisins d'un point à prédire, est très fortement dépendant du nombre d'individus présents dans la base de données.

Dans cette thèse il a donc été proposé d'étudier une méthode massivement parallélisable pour le calcul de voisinages : parSketch. Par le fait que cette méthode soit massivement parallélisable, elle est en mesure de répondre à des problématiques de traitement de données massives. Pour étudier cette méthode, elle a été intégrée au sein de l'approche locale et a donné lieu à une nouvelle méthode : parSketch-PLS. Cette méthode consiste au remplacement de l'algorithme force-brute par parSketch. Il a été observé lors de ce remplacement que parSketch pouvait être une alternative à force-brute car cette méthode permettait de retourner des plus proches voisins rapidement. La méthode parSketch-PLS a également montré son utilité dans un cadre applicatif pour une problématique de phénotypage. Cependant quelques limites ont pu être observées. Premièrement, la méthode parSketch approche uniquement la distance Euclidienne.

Cette distance n'est pas toujours la plus performante pour les données chimiques. Deuxièmement, le paramétrage de parSketch peut s'avérer complexe, d'autant plus lors de la prédiction d'individus extrêmes. Ceci peut entraîner des performances de prédiction réduites car les voisins renvoyés ne seront pas tous pertinents pour la calibration d'un modèle PLS local.

Le deuxième axe de recherche consistait à étudier la pertinence d'un individu au sein d'un modèle local. En effet, certains individus peuvent être non pertinents voire nuisibles à la construction de modèles PLS locaux. Cependant, cette problématique peut également se poser pour la PLS globale. En effet, dans cette thèse la problématique était de définir la pertinence des individus pour la calibration d'un modèle PLS. Une solution envisageable est de définir la pertinence d'un individu par des méthodes dites robustes. Ces méthodes ont pour objectif de définir des critères d'aberrance d'un individu à un ensemble de données puis de limiter voire éliminer l'impact de cet individu sur le calcul d'un modèle. Cependant, la définition de ces critères pour les données de grandes dimensions (avec beaucoup de variables) est difficile et pour certains types de données aberrantes les méthodes existantes ne sont pas adaptées. Dans cette thèse, il a donc été proposé une nouvelle méthode PLSR robuste pour évaluer la pertinence des individus : RoBoost-PLSR. Cette méthode a tout d'abord été présentée puis évaluée. Pour finir, la méthode RoBoost-PLSR a été appliquée pour la prédiction de taux de sucre dans les bais de raisin en présence de données aberrantes. Cette méthode a prouvé son efficacité sur certaines problématiques de régression. Cependant, il serait intéressant de la faire évoluer pour des problématiques de classifications. Il serait également intéressant d'évaluer plus en détail les limites de RoBoost-PLSR.

Le troisième axe quant à lui, consistait à combiner les outils de chimiométrie avec ceux du big-data. En effet, dans le premier chapitre il a été montré que la méthode parSketch-PLS n'atteignait pas d'aussi bonnes performances que celles obtenues avec la méthode BF-PLS. Bien que cette méthode de traitement de bases de données massives soit efficiente, elle souffre de certaines limites pour le traitement de données chimiques. En effet, deux problématiques peuvent être associées à l'utilisation de méthodes comme parSketch en chimiométrie.

La première est que la pertinence du voisinage renvoyé dépend de la métrique utilisée pour définir les voisins. En effet, en chimiométrie certaines métriques ont été développées et permettent d'obtenir un voisinage plus pertinent en fonction de la base de données traitée. Comme dit précédemment, parSketch sélectionne les plus proches voisins en fonction de la distance Euclidienne alors que cette métrique n'est pas toujours en mesure

de retourner les voisinages les plus pertinents pour la calibration de modèles PLS. Pour répondre à cette problématique il a été proposé d'utiliser une stratégie d'indexation orientée. Dans ce contexte, une nouvelle méthode a été proposée : PLSgrid. Cette méthode combinait la PLS et la méthode d'indexation par grille. Cette approche peut être pertinente mais la méthode d'indexation par grille doit être remplacée par une autre méthode d'indexation plus adaptée aux variables latentes PLS. Cependant cette étude a mis en évidence qu'il est possible et potentiellement intéressant de combiner les méthodes de chimiométrie et les approches d'indexation pour obtenir des méthodes rapides et avec de bonnes capacités prédictives.

La deuxième problématique est que les voisins renvoyés par parSketch peuvent ne pas être tous pertinents. En effet, comme pour les méthodes locales classiques, malgré une bonne métrique, tous les voisins ne sont pas nécessairement pertinents. Cet effet est d'autant plus important pour les méthodes comme parSketch qui vont renvoyer pour certains individus à prédire un très grand nombre de voisins. Dans cette thèse, il est proposé de combiner parSketch et RoBoost-PLSR pour résoudre cette problématique. Il a été mis en évidence à l'aide d'un jeu de données simulées que cette stratégie peut être appropriée. Cependant, certaines limites sont discutées dans cette étude. En effet, la principale limite est le coût calculatoire associé à la méthode RoBoost-PLSR, plus élevé que pour la PLSR. Il serait donc avantageux de développer une alternative plus efficiente de RoBoost-PLSR.

Pour conclure, dans cette thèse l'intérêt de combiner les connaissances issues du traitement de big-data avec les connaissances de la chimiométrie a été mis en évidence. En effet, les connaissances métiers de la chimiométrie permettent de développer des outils adaptés aux problématiques du traitement des données chimiques et les outils du big-data permettent de traiter les données chimiques massives. Dans de futurs travaux il sera donc intéressant de joindre ces deux domaines afin de développer des outils big-data pour le traitement de données chimiques.

## 5.2 Perspectives générales

Les études réalisés dans cette thèse ont montré que l'association des outils big-data avec ceux de la chimiométrie pouvait être grandement utile pour répondre aux problématiques actuelles de traitement de données massives. Cependant, les études réalisées doivent encore être développées pour offrir à la communauté de chimiométrie des outils et méthodes pour traiter de grands ensembles de données. Pour ce faire, des réflexions diverses doivent être portées autour de la question du traitement de gros

ensembles de données.

Premièrement, il serait intéressant d'évaluer d'autres outils utilisés dans le domaine du big-data. En effet, des outils permettant de visualiser de très grands ensembles de données pourraient permettre de réaliser des analyses plus poussées.

Deuxièmement, de grands ensembles de données n'impliquent pas toujours de meilleures performances. Par exemple, le bruit présent dans une base de données spectrales peut avoir un impact fort sur les outils d'analyses. Dans de nombreux domaines, le big-data est également associé au smart data. Le smart data signifie que les données sont transformées en nouvelles données plus pertinentes et en plus petit nombre. Il serait intéressant d'étudier cette voie pour le traitement des données massives en chimiométrie.

Troisièmement, aujourd'hui, le traitement de base des données massives est réservé exclusivement à quelques outils de traitement de données. En effet, les algorithmes développés en chimiométrie permettent rarement d'utiliser les logiciels et infrastructures de calculs utilisés par les outils de traitement de données massives. Cependant, comme les outils classiques de la chimiométrie se basent sur du calcul matriciel, ils sont massivement parallélisables notamment sur GPU. Il serait donc intéressant pour les chimiométriciens de disposer d'un plus large panel de méthodes parallélisées afin de choisir les outils les plus performants et pas uniquement les outils disponibles.

Pour finir, l'augmentation de la quantité de données, l'utilisation d'outils complexes et de moyens de calcul onéreux posent la question de l'impact de nos stratégies de modélisation sur l'environnement. En effet, il serait nécessaire de remettre en question des pratiques et usages d'outils permettant de traiter des données massives comme les outils du deep learning qui peuvent avoir un impact environnemental fort. En effet, ces outils nécessitent des moyens de calcul plus conséquents que nos outils usuels (PLS, KNN-PLS, etc). L'utilisation de telles techniques peut être justifiée par la complexité des données traitées, mais une étude de cette complexité devrait être réalisée avant d'effectuer des traitements à coût calculatoire très élevé. Dans de futures études, il serait intéressant d'intégrer l'impact environnemental de nos stratégies de modélisations dans les choix des méthodes d'analyse.

# Communications scientifiques

# Chapitre 6

## Article 1 : Comparison of locallyweighted PLS strategies for regression and discrimination on agronomic NIR data

Référence :

Matthieu Lesnoff, Maxime Metz, and Jean-Michel Roger. Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data. Journal of Chemometrics, 34(5) :e3209, 2020. ISSN 1099-128X. doi:10.1002/cem.3209. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.3209.

Received: 17 April 2019 | Revised: 22 August 2019 | Accepted: 26 November 2019

DOI: 10.1002/cem.3209

RESEARCH ARTICLE

Journal of CHEMOMETRICS WILEY

# Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data

**Matthieu Lesnoff**[1,2,4] | **Maxime Metz**[3,4] | **Jean-Michel Roger**[3,4]

[1]CIRAD, UMR SELMET, Montpellier, France

[2]SELMET, Univ Montpellier, CIRAD, INRA, Montpellier SupAgro, Montpellier, France

[3]ITAP, Montpellier SupAgro, Irstea, Univ Montpellier, Montpellier, France

[4]ChemHouse Research Group, Montpellier, France

**Correspondence**
Matthieu Lesnoff, Selmet Joint Research Unit (Tropical and Mediterranean Animal Production Systems), CIRAD, TA C-112/A—Campus international de Baillarguet—34398 Montpellier Cedex 5, France.
Email: matthieu.lesnoff@cirad.fr

**Abstract**

In multivariate calibrations, locally weighted partial least squared regression (LWPLSR) is an efficient prediction method when heterogeneity of data generates nonlinear relations (curvatures and clustering) between the response and the explicative variables. This is frequent in agronomic data sets that gather materials of different natures or origins. LWPLSR is a particular case of weighted PLSR (WPLSR; ie, a statistical weight different from the standard $1/n$ is given to each of the $n$ calibration observations for calculating the PLS scores/loadings and the predictions). In LWPLSR, the weights depend from the dissimilarity (which has to be defined and calculated) to the new observation to predict. This article compares two strategies of LWPLSR: ($a$) "LW": the usual strategy where, for each new observation to predict, a WPLSR is applied to the $n$ calibration observations (ie, entire calibration set) vs ($b$) "KNN-LW": a number of $k$ nearest neighbors to the observation to predict are preliminary selected in the training set and WPLSR is applied only to this selected KNN set. On three illustrating agronomic data sets (quantitative and discrimination predictions), both strategies overpassed the standard PLSR. LW and KNN-LW had close prediction performances, but KNN-LW was much faster in computation time. KNN-LW strategy is therefore recommended for large data sets. The article also presents a new algorithm for WPLSR, on the basis of the "improved kernel #1" algorithm, which is competitor and in general faster to the already published weighted PLS nonlinear iterative partial least squares (NIPALS).

**KEYWORDS**

discrimination, locally weighted calibration, near-infrared spectroscopy, partial least squares, regression

## 1 | INTRODUCTION

Near-infrared spectroscopy (NIRS) is a fast and nondestructive analytical method for predicting chemical compositions, currently used in many agronomic contexts.[1] The partial least squared regression (PLSR)[2-5] is very efficient for NIRS predictions when the relationship between the spectral information and the response is linear. Nevertheless, agronomic databases (eg, in soils, feed, or food researches) usually aggregate samples of different natures or origins. This heterogeneity generates nonlinearity in the data (curvatures and/or clustering) that can alter significantly the predictions.[6-9] Among other strategies, the locally weighted PLSR (LWPLSR),[10-12] which is a particular case of weighted PLSR

(WPLSR), can turn out such problem. The principle of WPLSR is to define a statistical weight $d$ for each of the $n$ calibration observations and to incorporate them into the PLSR algorithm (in the standard PLSR, $d = 1/n$). The method is in two steps: ($a$) calculation of PLS scores by maximizing weighted covariances and ($b$) prediction by weighted least squared (WLS) regression on the scores. Weighting in PLSR can target robustness (eg, Hubert and Vanden Branden[13]) or, which is the focus of this article, locally weighted predictions. In LWPLSR, the weights $d$ are calculated from the dissimilarities (eg, Euclidean or Mahalanobis distances) of the calibration observations to the new observation to predict: closer is a calibration observation to the new observation, higher is its weight (and therefore importance) in the calculations. For nonlinear data, in the same principle as for the well-known locally weighted regression (LWR),[14,15] taking into account locality in PLSR generally decreases the prediction biases. A LWPLSR nonlinear iterative partial least squares (NIPALS) algorithm has been described in Schaal et al[10] for the PLS1 case, ie, where the response to be predicted is a single variable (vector $\boldsymbol{y}$). Sicard[16] and Sicard and Sabatier[11] derived a direct expression of the weighted PLS coefficients. Then LWPLSR NIPALS algorithms have been described for the PLS2 case,[12,17,18] ie, where the (multi)response to be predicted is composed of several variables (matrix $\boldsymbol{Y}$).

At our knowledge, all the studies using LWPLSR consisted in fitting, for each new observation to predict, a WPLSR using the entire set of the $n$ calibration observations. This strategy can be very time-consuming if $n$ is large (eg, $n \geq 1000$). An alternative strategy, proposed in this article, is to do a preselection of $k$ nearest neighbors of the observation to predict (KNN selection) and then only apply LWPLSR to the $k$ neighbors. The first objective of this article was to compare these two strategies (LWPLSR without vs with preliminary KNN selection) in terms of efficiency and computation time. Three agronomic data sets were used as illustration. The second objective of this article was to propose a faster algorithm than NIPALS for LWPLSR. This proposed algorithm is a weighted version of the PLS "improved kernel #1" algorithm.[19]

## 2 | MATERIAL AND METHODS

### 2.1 | Data sets

The three study data sets (PROT, DMD, and BARL, respectively) are summarized in Table 1. PROT and DMD were used in a regression objective and BARL in a discrimination objective (the discriminant LWPLSR method is detailed in a next section). The dependent response for each of the three data sets was unidimensional. In PROT, the response was the protein content in samples collected from different feed or food products. In DMD, the response was the in vitro dry matter enzymatic digestibility of tropical forages. In BARL, the (binary) response was the absence vs presence of barley in commercial compound feeds. For all data sets, NIR spectra were collected on Foss instrument models 5000 or 6500 in the spectral range 1100 to 2498 nm (2 nm intervals). A preprocessing was applied to the spectra, consisting in a Savitzky-Golay 2nd derivation (polynomial of order 3, and window of 21 spectral points for PROT and BARL and of 11 spectral points for DMD) followed by a standard normal variate (SNV). A preliminary principal component analysis (PCA) was implemented on the preprocessed spectra for describing the heterogeneity in the data sets (Figure 1). All the data sets are multiproduct but with different structural patterns. High clustering was observed for PROT and BARL because of the very different natures of products in the data sets. The clustering was less apparent for DMD because of a continuum in the types of plants.

**TABLE 1** The three data sets (PROT, DMD, and BARL) used for methods comparisons

| Data Set | N | Response $y$ | Type of Material | Source |
|---|---|---|---|---|
| PROT | 4075 | Protein content (mean = 31.9; min = 2.8; max = 76.6, sd = 20.3) | Animal feed, rapeseed, corn gluten, grass silage, soya, wheat, milk powder, maize, sun flower | CRA-W, Belgium |
| DMD | 1148 | In vitro dry matter enzymatic digestibility (mean = 52.2; min = 9.9; max = 95.0, sd = 16.8) | Cereal, grass, legume, tree, grassland mixing forages (mainly from tropical drylands) | CIRAD, France |
| BARL | 7515 | Absence/presence of barley (proportion of samples with barley = 0.76) | Commercial compound feeds with variable formulations and animal destinations | ETSIAM,[20] Univ Cordoba, Spain |

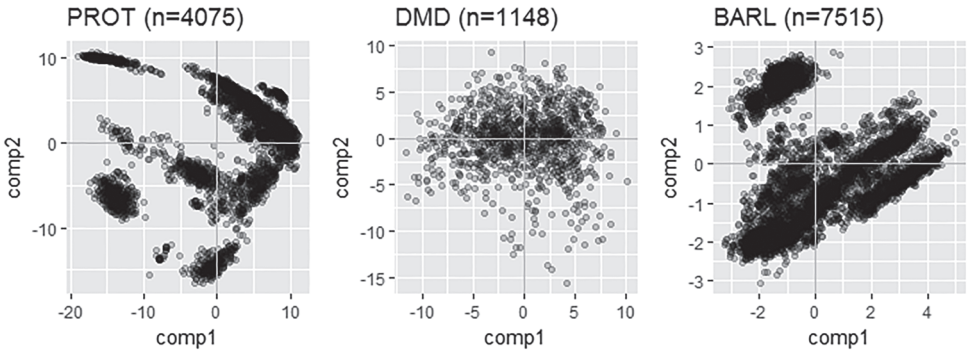



**FIGURE 1** Principal component analysis (PCA) score plots (components 1 and 2) of the preprocessed spectra for the three data sets (PROT, DMD, and BARL)

## 2.2 | Weighted PLSR

### 2.2.1 | Standard PLSR

Let note $\boldsymbol{X}$ a spectral matrix of size $n \times p$ and $\boldsymbol{Y}$ a multiresponse matrix of size $n \times q$:

$$\boldsymbol{X} = \left[\boldsymbol{x_1}\, \boldsymbol{x_2} ... \boldsymbol{x_p}\right] \text{ with } \boldsymbol{x_j} = \left[x_{1j}, x_{2j}, ..., x_{nj}\right]' \quad j = 1, ..., p$$

$$\boldsymbol{Y} = \left[\boldsymbol{y_1}\, \boldsymbol{y_2} ... \boldsymbol{y_q}\right] \text{ with } \boldsymbol{y_j} = \left[y_{1j}, y_{2j}, ..., y_{nj}\right]' \quad j = 1, ..., q$$

where $n$ represents the number of observations. We note $\boldsymbol{t_a}$ and $\boldsymbol{u_a}$ the $X$- and $Y$-PLS scores (size $n \times 1$), respectively, with $a = 1, ..., n_{comp}$, where $n_{comp}$ is the maximum number of scores considered for factorizing $\boldsymbol{X}$ and $\boldsymbol{Y}$. The standard PLS NIPALS[2,21-23] assumes a same statistical weight, $d_i = 1/n$ ($i = 1, ..., n$), for the $n$ calibration observations. Matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ are centered from the usual column means: $\mathbf{1_n}'\boldsymbol{X} = 0$ and $\mathbf{1_n}'\boldsymbol{Y} = 0$ (where $\mathbf{1_n}$ is the $n \times 1$ unity vector). Scores $\boldsymbol{t_a}$ and $\boldsymbol{u_a}$ ($n \times 1$ vectors) maximize the squared value of the empirical covariance $Cov(\boldsymbol{t_a}, \boldsymbol{u_a}) = 1/n * \boldsymbol{t_a}'\boldsymbol{u_a}$ and are constrained to be $\boldsymbol{I}$-orthogonal: $\boldsymbol{t_a}'\boldsymbol{t_k} = \boldsymbol{u_a}'\boldsymbol{u_k} = 0$ for $a \neq k$. In PLSR, the $Y$-predictions for the calibration observations (matrix $\hat{\boldsymbol{Y}}$ of size $n \times q$) are solutions of the ordinary least squared regression of $\boldsymbol{Y}$ on the $X$-score matrix $\boldsymbol{T}$. For a given $a$, and noting $\boldsymbol{B}$ the matrix of the score regression coefficients (size $a \times q$), the predictions for the calibration observations are $\hat{\boldsymbol{Y}} = \boldsymbol{T}\,\hat{\boldsymbol{B}}$, where $\boldsymbol{T} = [\boldsymbol{t_1}\boldsymbol{t_2} ... \boldsymbol{t_a}]$ and $\hat{\boldsymbol{B}} = (\boldsymbol{T}'\boldsymbol{T})^{-1}\boldsymbol{T}'\boldsymbol{Y}$. For a new observation to predict, say $\boldsymbol{x_{(new)}}$ (vector $p \times 1$), the $Y$-predictions are $\hat{\boldsymbol{y}}_{(\boldsymbol{new})}{}' = \boldsymbol{t_{(new)}}'\,\hat{\boldsymbol{B}}$, where $\boldsymbol{t_{(new)}}$ (vector $a \times 1$) is the score vector obtained by projection of $\boldsymbol{x_{(new)}}$ on the space of columns of $\boldsymbol{T}$.

### 2.2.2 | Weighted PLSR

WPLSR generalizes PLSR by allocating different statistical weights $d_i$ ($i = 1, ..., n$) to the calibration observations. Let us note $\boldsymbol{D} = diag(d_1, d_2, ..., d_n)$ a diagonal $n \times n$ weight matrix (in this article, weights $d_i$ are assumed to be normalized to sum to 1, ie, $trace(\boldsymbol{D}) = 1$). $\boldsymbol{X}$ and $\boldsymbol{Y}$ are now centered from the weighted (instead of simple) column means: $\mathbf{1_n}'\boldsymbol{D}\boldsymbol{X} = 0$ and $\mathbf{1_n}'\boldsymbol{D}\boldsymbol{Y} = 0$. The covariance that is maximized is $Cov(\boldsymbol{t_a}, \boldsymbol{u_a}) = \boldsymbol{t_a}'\boldsymbol{D}\boldsymbol{u_a}$ and, in the weighted NIPALS, scores are constrained to be $\boldsymbol{D}$-orthogonal: $\boldsymbol{t_a}'\boldsymbol{D}\boldsymbol{t_k} = \boldsymbol{u_a}'\boldsymbol{D}\boldsymbol{u_k} = 0$ for $a \neq k$. The $Y$-predictions for the calibration observations are solutions of the WLS regression of $\boldsymbol{Y}$ on $\boldsymbol{T}$. For a given $a$, $\hat{\boldsymbol{Y}} = \boldsymbol{T}\,\hat{\boldsymbol{B}}$, where $\hat{\boldsymbol{B}} = (\boldsymbol{T}'\boldsymbol{D}\boldsymbol{T})^{-1}\boldsymbol{T}'\boldsymbol{D}\boldsymbol{Y}$. The prediction for a new observation is obtained as for standard PLSR.

### 2.2.3 | The weighted improved kernel #1 algorithm

WPLSR NIPALS algorithms have been detailed by Schaal et al[10] for a unidimensional response (PLSR1) and by Kim et al,[12] Hazama and Kano,[24] and Zhang et al[18] for multiresponses (PLSR2). NIPALS has good properties such as stability and the ability to manage missing value.[3,23] Nevertheless, it requires (at each round $a$ of the PLSR calculations) the deflation of $n * p$ $X$-components (and optionally $n * q$ $Y$-components). This can become time-consuming when $n$ is high.

For standard PLS, Dayal and McGregor[19] proposed an algorithm giving the same result as NIPALS, referred as the "improved kernel algorithm #1," which deflates the kernel matrix $\boldsymbol{X}'\boldsymbol{Y}$ (instead of $\boldsymbol{X}$). This algorithm is stable[25] and faster than NIPALS, specifically when $(n * p) > (p * q)$. For the present study, we developed a weighted version of the improved kernel algorithm #1, for PLSR1 or PLSR2. It gives the same results as the WPLSR NIPALS algorithms discussed above. An implementation using the R software language[26] is presented in the Appendix (as well as a Matlab transcription).

## 2.3 | Locally weighted PLSR

The characteristic of LWPLSR (within all WPLSR approaches) is that the statistical weights $d_i$ $(i = 1, ..., n)$ are calculated from the dissimilarities (eg, distances) between the calibration observations and the observation to predict. Let us note $\boldsymbol{\delta} = [\delta_1, \delta_2, ..., \delta_n]'$ these dissimilarities. In LWR methods,[14] the weights are calculated as a decreasing function of the dissimilarities. In this study, the weight function was defined as $f(\delta_i^*) = \exp(-\delta_i^*/(h * \mathrm{sd}[\boldsymbol{\delta}^*]))$,[12] where $\delta_i^* = \delta_i/\max\{\delta_i, i = 1, ..., n\}$ represents a normalized dissimilarity[15,27] and $h$ a scalar defining the shape of $f(\delta^*)$. The normalized weights were $d_i^* = f(\delta_i^*)/\max\{f(\delta_i^*), i = 1, ..., n\}$ giving, after a last normalization to sum to 1, the final weights $d_i$ used in the WPLSR algorithm. Parameter $h$ determines the shape of the weight function $f$. Low $h$ values make $f$ sharper, which gives more importance to the closest neighbors of the new observation (Figure 2). The case $h = \infty$ is the unweighted situation. Many alternative weight functions (eg, bicube or tricube functions[15]) can be defined. The form used in this article allows an easy optimization (by varying a single shape factor) of $f$. In LWPLSR, it is important to note that since $\boldsymbol{\delta}$ varies, a different model is fitted for each new observation to predict.

## 2.4 | Discrimination

For standard (ie, nonlocally weighted) context, a simple approach for partial least squares discriminant analysis (PLSDA) is to transform the unidimensional response $\boldsymbol{y}$ (containing $q$ classes) to a $n \times q$ multiresponse matrix $\boldsymbol{Y}_{\mathbf{dummy}}$ of $q$ 0-1 dummy variables (where 1 denotes the given class and 0 the other classes), compress the $X$-data with a PLS2 on $(\boldsymbol{X}, \boldsymbol{Y}_{\mathbf{dummy}})$, and then implement a discriminant analysis (DA) on the PLS2 scores and vector $\boldsymbol{y}$. A frequent DA method[28-31] is the linear discriminant analysis (LDA), but any alternative DA can be used. We followed the same overall approach: a dummy matrix $\boldsymbol{Y}_{\mathbf{dummy}}$ was built from the class variable $\boldsymbol{y}$, and LWPLSR (using PLS2) was implemented on $(\boldsymbol{X}, \boldsymbol{Y}_{\mathbf{dummy}})$. For simplicity (but also since it gave better discrimination results on our data than LDA or quadratic DA), the DA was as follows: for each new observation to predict $\boldsymbol{x}_{(new)}$, the predicted class corresponded to the column of the LWPLSR prediction $\hat{\boldsymbol{y}}_{dummy(new)}{}'$ (size $1 \times q$) having the highest value.

## 2.5 | Compared strategies

Two LWPLSR strategies were compared in this article. For each new observation to predict, they are as follows:

- LW: the LWPLSR is run taking into account for all the $n$ calibration observations. This is the usual LWPLSR as reported in the literature.

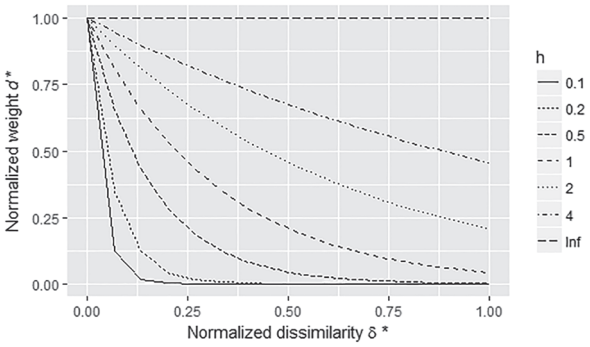



**FIGURE 2** Normalized dissimilarity ($\delta^*$) and weight ($d^*$)

- KNN-LW (this is the strategy proposed in this article): $k$ nearest neighbors of the observation to predict are preliminary selected in the calibration data set, and the LWPLSR is only run on these $k$ nearest neighbors.

In addition, LW and KNN-LW were compared with two other common strategies: a global (ie, implemented on the $n$ calibration observations) PLSR and the "local PLSR" (LP) in which a standard PLSR is implemented on a preliminary selection of $k$ nearest neighbors (this is a special case of KNN-LW where the weights are set to 1/-$k$).

LW, KNN-LW, and LP require calculating the dissimilarity measure $\delta$. In this article, $\delta$ was the Mahalanobis distance computed from a number $n_{compdis}$ of scores of a global PLSR. After preliminary data exploration, $n_{compdis}$ was set to 15 for PROT and DMD and 30 for BARL. Many other dissimilarity measures could be considered (eg, based on Euclidean distances or correlations[32,33]).

## 2.6 | Model optimization and prediction

The process of model optimization and predictions was the same for the three data sets of Table 1. A test set (VAL) representing 20% of the total data set was built (from random sampling) and separated for the other part (CAL) of the data. A $K$-fold random cross-validation (CV), with $K = 5$, was implemented on CAL for optimizing the parameters of the models (LW, KNN-LW, LP, and PLSR). Then the optimized models were used for predicting VAL, and the predictions were compared with the observed data for computing predictive error rates. The error rates were the root mean squared errors (RMSEP) for PROT and DMD, and the proportion of misclassified observations (ERRP) for BARL. For checking the stability of the results, this procedure was repeated three times independently, corresponding to a simplified double CV.[34]

In the CV, the parameters to optimize were the number of latent components ($n_{comp}$), the number of selected nearest neighbors ($k$), and the weighting shape factor ($h$), depending on the models:

- PLSR: $n_{comp}$
- LW: $n_{comp}$, $h$
- KNN-LW: $n_{comp}$, $k$, $h$
- KNN-L: $n_{comp}$, $k$

The parameter values evaluated were $n_{comp} = (1, 2, ..., 40)$, $k = (50, 100, 200, ..., 600)$, and $h = (0.1, 0.2, ..., 1, 2, 3, 4)$. For each model, all the combinations of the considered values were explored. For preventing overfitting in PLSR, as well as in LW, KNN-LW, and KNN-L for each $h$, $k \times h$, and $k$, respectively, the number of components $n_{comp}$ was firstly selected using a heuristic criterion. This criterion was $R = 1 - r(a + 1)/r(a)$,[10,35,36] where $r$ is the error rate (RMSECV or ERRCV) and $a$ the PLS model dimension. $R$ represents the relative gain in efficiency after a new dimension is added in the model. The iterations continue until $R$ becomes lower than a threshold value $\alpha$ (set to 1% in this article). Then, values (for the selected $n_{comp}$) of $h$, $k \times h$, and $k$, respectively, showing the lowest CV error rate were selected and used for calculating the error rates on VAL (RMSEP and ERRP).

# 3 | RESULTS

## 3.1 | Prediction error rates

Prediction error rates RMSEP and ERRP for the three data sets (and the three replications of the test samples) are presented in Table 2. Globally, the ranking of the methods did not change between the three replications, indicating a good stability. In average, all the methods involving locally calculations (LWPLSR, KNN-LW, and KNN-L) had lower prediction error rates than PLSR (for PROT, DMD, and BARL, the best of the locally methods was 49%, 22%, and 77% more efficient than PLSR, respectively). Within these locally methods, the weighting (LW and KNN-LW vs KNN-L) decreased the average error rates. The highest gains in efficiency were observed for DMD and BARL: for instance, average RMSEP for DMD was 4.57 for KNN-LW vs 5.19 for KNN-L, and average ERRP for BARL was 3.7% vs 6.9%, respectively. These gains were less important for PROT (eg, average RMSEP was 0.72 for KNN-LW vs 0.76 for KNN-L). This was probably in relation with the high clustering of PROT: in such a situation, weighting the observations within the neighborhood

**TABLE 2** Prediction error rates for the three data sets (PROT, DMD, and BARL)

| **(a) PROT** | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **VAL1** | | | | **VAL2** | | | | **VAL3** | | | | **Mean** |
| | **h** | **k** | **$n_{comp}$** | **RMSEP** | **h** | **k** | **$n_{comp}$** | **RMSEP** | **h** | **k** | **$n_{comp}$** | **RMSEP** | **RMSEP** |
| PLSR | – | – | 17 | 1.45 | – | – | 19 | 1.38 | – | – | 17 | 1.42 | 1.41 |
| KNN-L | – | 100 | 9 | 0.77 | – | 100 | 9 | 0.77 | – | 100 | 9 | 0.76 | 0.76 |
| LW | 0.4 | – | 12 | 0.75 | 0.5 | – | 15 | 0.72 | 0.4 | – | 12 | 0.80 | 0.75 |
| KNN-LW | 0.9 | 200 | 11 | 0.70 | 2.0 | 200 | 11 | 0.70 | 1.0 | 200 | 10 | 0.75 | 0.72 |
| **(b) DMD** | | | | | | | | | | | | | |
| **Method** | **VAL1** | | | | **VAL2** | | | | **VAL3** | | | | **Mean** |
| | ***h*** | ***k*** | ***$n_{comp}$*** | **RMSEP** | ***h*** | ***k*** | ***$n_{comp}$*** | **RMSEP** | ***h*** | ***k*** | ***$n_{comp}$*** | **RMSEP** | **RMSEP** |
| PLSR | – | – | 12 | 5.37 | – | – | 6 | 6.65 | – | – | 12 | 5.14 | 5.68 |
| KNN-L | – | 50 | 5 | 4.95 | – | 50 | 4 | 5.58 | – | 50 | 4 | 5.03 | 5.19 |
| LW | 0.4 | – | 7 | 4.67 | 0.4 | – | 8 | 4.69 | 0.5 | – | 6 | 4.24 | 4.53 |
| KNN-LW | 0.8 | 600 | 7 | 4.73 | 0.6 | 600 | 8 | 4.78 | 0.8 | 600 | 6 | 4.19 | 4.57 |
| **(c) BARL** | | | | | | | | | | | | | |
| **Method** | **VAL1** | | | | **VAL2** | | | | **VAL3** | | | | **Mean** |
| | *h* | *k* | *$n_{comp}$* | ERRP | *h* | *k* | *$n_{comp}$* | ERRP | *h* | *k* | *$n_{comp}$* | ERRP | ERRP |
| PLSR | – | – | 14 | 0.147 | – | – | 11 | 0.172 | – | – | 14 | 0.173 | 0.164 |
| KNN-L | – | 100 | 11 | 0.077 | – | 50 | 8 | 0.064 | – | 50 | 5 | 0.068 | 0.069 |
| LW | 0.5 | – | 10 | 0.041 | 0.3 | – | 7 | 0.035 | 0.4 | – | 9 | 0.036 | 0.037 |
| KNN-LW | 1.0 | 200 | 7 | 0.043 | 1.0 | 400 | 9 | 0.032 | 0.8 | 300 | 8 | 0.035 | 0.037 |

can become less beneficial (main of the variability is modelled by the preliminary KNN selection). Finally, LW and KNN-LW returned very similar average error rates for the three data sets.

## 3.2 | Computation times

Another important output was that KNN-LW was much faster than LW (Figure 3). Depending on the data sets, the ratio of the computation times of LW vs KNN-LW went from 4.7 to 28.4 for $k = 50$ neighbors and from 1.4 to 7.7 for $k = 600$ neighbors. Concerning the WPLSR algorithms (Figure 4), the improved kernel #1 algorithm was faster than NIPALS (for instance, for $k = 300$ neighbors, the computation time ratio NIPALS vs Kernel#1 went from 1.7 to 3.2 depending the data sets) except for the small neighborhood size ($k = 50$) for which NIPALS was slightly faster (ratios going from 0.81 to 0.90).

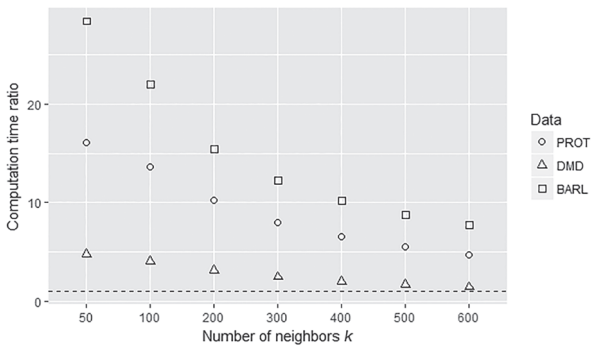



**FIGURE 3** Computation time ratio "LW/KNN-LW" (averages over VAL1, VAL2, and VAL3)

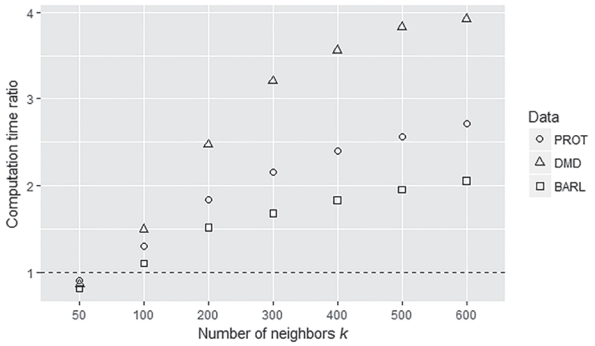



**FIGURE 4** Computation time ratio "NIPALS/improved kernel #1 algorithms" for KNN-LW (averages over VAL1, VAL2, and VAL3)

## 4 | DISCUSSION AND CONCLUSION

At our knowledge, the first application of LWRs on latent variables for NIR data was proposed by Naes et al,[27] with the "LWR" algorithm. The first step of LWR is to compute global PCA scores on the calibration data set and, for a given new observation to predict, to select (from Mahalanobis distances calculated from the global scores) $k$ nearest neighbors of this new observations. The second step is to regress, on the $k$ neighbors and by WLS (with weights depending on the distances), the response on the global scores. A slight variant of LWR recalculates the PCA scores locally (ie, over the $k$ neighbors) before implementing the WLS regression.[37] LWR can however be considered as an incomplete weighted method since only the regression step is weighted (PCA scores are calculated without weighting). Another well-known local algorithm, referred as "LOCAL,"[32] selects the neighbors from correlation dissimilarities and predicts the response from standard PLSR, therefore without weighting, on the $k$ neighbors. For discrimination, an "LWPLSDA" algorithm was presented by Bevilacqua and Marini,[38] consisting in a direct weighting of the data.[39] After a KNN selection, the PLSDA is implemented on $\boldsymbol{D} * \boldsymbol{X}$ (instead of $\mathbf{X}$ in the standard PLSDA). The method provided efficient prediction results in the study cases reported by the authors. Nevertheless, for PLS, this direct weighting does not insure optimality for the weighted covariance $Cov(\boldsymbol{t_a}, \boldsymbol{u_a}) = \boldsymbol{t_a}'\boldsymbol{D}\boldsymbol{u_a}$. More recently, Song et al[40] also used a local PLSDA algorithm for discrimination but still with no weighting of the neighbors. In contrast with these algorithms, the two LWPLSR methods compared in this article (LW and KNN-LW) have the advantage to integrate the statistical weights of the observations in the both steps of the PLSR: the computation of the scores and loadings matrices and the regression coefficients. In this way, these methods could be considered as a more consistent approach for the objective to get locally dependent predictions.

For the three study data sets, LWPLSR (LW and KNN-LW) improved the predictions compared with the standard PLSR and the unweighted local method (KNN-L). LW and KNN-LW showed close prediction performances. Nevertheless, the preliminary selection of neighbors in KNN-LW before the weighting enabled to decrease, considerably in some cases, the computation times. In practice, KNN-LW can therefore be generally recommended at the expense of LW. For strategies looking forward decreasing the computation times, the weighted version of the improved kernel #1 proposed in this article is a performant alternative to the weighted NIPALS especially when the neighborhood size $k$ increases.

## ACKNOWLEDGEMENTS

Vincent Baeten and Pierre Dardenne (CRA-W, Agronomic Research Centre of Wallonia, Belgium), Dolores Pérez-Marí and Ana Garrido-Varo (ETSIAM, Universidad de Córdoba, Spain), and Laurent Bonnal (CIRAD, France) are thanked for having provided the data sets illustrating this article. We thank Belal Gaci for helping to the Matlab transcription of our original R function of the proposed WPLSR algorithm. Finally, we are grateful to two anonymous reviewers for their constructive remarks that improved this article.

## ORCID

*Matthieu Lesnoff* https://orcid.org/0000-0002-5205-9763
*Jean-Michel Roger* https://orcid.org/0000-0003-2123-5266

## REFERENCES

1. Shen G, Lesnoff M, Baeten V, et al. Local partial least squares based on global PLS scores. *J Chemometr*. 2019;33(5):e3117. https://doi.org/10.1002/cem.3117

2. Wold H. Nonlinear iterative partial least squares (NIPALS) modeling: some current developments. In: Krishnaiah PR, ed. *Multivariate Analysis II. Wright State University, Dayton, Ohio, USA. June 19–24, 1972*. New York: Academic Press; 1973:383-407.
3. Wold S, Sjöström M, Eriksson L. PLS-regression: a basic tool of chemometrics. *Chemom Intel Lab Syst*. 2001;58(2):109-130. https://doi.org/10.1016/S0169-7439(01)00155-1
4. de Jong S. SIMPLS: an alternative approach to partial least squares regression. *Chemom Intel Lab Syst*. 1993;18(3):251-263. https://doi.org/10.1016/0169-7439(93)85002-X
5. Tenenhaus M. *La Régression PLS: Théorie et Pratique*. Paris: Editions Technip; 1998.
6. Dardenne P, Sinnaeve G, Baeten V. Multivariate calibration and chemometrics for near infrared spectroscopy: which method? *J Infrared Spectrosc*. 2000;8(4):229-237.
7. Clairotte M, Grinand C, Kouakoua E, et al. National calibration of soil organic carbon concentration using diffuse infrared reflectance spectroscopy. *Geoderma*. 2016;276:41-52. https://doi.org/10.1016/j.geoderma.2016.04.021
8. Davrieux F, Dufour D, Dardenne P, et al. LOCAL regression algorithm improves near infrared spectroscopy predictions when the target constituent evolves in breeding populations. *J Infrared Spectrosc*. 2016;24(2):109-117. https://doi.org/10.1255/jnirs.1213
9. Tran H, Salgado P, Tillard E, Dardenne P, Nguyen XT, Lecomte P. "Global" and "local" predictions of dairy diet nutritional quality using near infrared reflectance spectroscopy. *J Dairy Sci*. 2010;93(10):4961-4975. https://doi.org/10.3168/jds.2008-1893
10. Schaal S, Atkeson CG, Vijayakumar S. Scalable techniques from nonparametric statistics for real time robot learning. *Appl Intell*. 2002;17(1):49-60. https://doi.org/10.1023/A:1015727715131
11. Sicard E, Sabatier R. Theoretical framework for local PLS1 regression, and application to a rainfall data set. *Comput Stat Data Anal*. 2006;51(2):1393-1410. https://doi.org/10.1016/j.csda.2006.05.002
12. Kim S, Kano M, Nakagawa H, Hasebe S. Estimation of active pharmaceutical ingredients content using locally weighted partial least squares and statistical wavelength selection. *Int J Pharm*. 2011;421(2):269-274. https://doi.org/10.1016/j.ijpharm.2011.10.007
13. Hubert M, Vanden Branden K. Robust methods for partial least squares regression. *J Chemometr*. 2003;17:537-549. https://doi.org/10.1002/cem.822
14. Cleveland WS. Robust locally weighted regression and smoothing scatterplots. *J Am Stat Assoc*. 1979;74(368):829. https://doi.org/10.2307/2286407
15. Cleveland WS, Devlin SJ. Locally weighted regression: an approach to regression analysis by local fitting. *J Am Stat Assoc*. 1988;83(403):596-610. https://doi.org/10.1080/01621459.1988.10478639
16. Sicard E. Choix de composantes optimales pour l'analyse spatiale et la modélisation: application aux pluies mensuelles du Nordeste brésilien. 2004.
17. Yoshizaki R, Kano M, Tanabe S, Miyano T. Process parameter optimization based on LW-PLS in pharmaceutical granulation process**This work was partially supported by Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (C) 24560940. *IFAC-Pap*. 2015;48(8):303-308. https://doi.org/10.1016/j.ifacol.2015.08.198
18. Zhang X, Kano M, Li Y. Locally weighted kernel partial least squares regression based on sparse nonlinear features for virtual sensing of nonlinear time-varying processes. *Comput Chem Eng*. 2017;104:164-171. https://doi.org/10.1016/j.compchemeng.2017.04.014
19. Dayal BS, MacGregor JF. Improved PLS algorithms. *J Chemometr*. 1997;11(1):73-85. https://doi.org/10.1002/(SICI)1099-128X(199701)11:1<73::AID-CEM435>3.0.CO;2-#
20. Pérez-Marín D, Fearn T, Guerrero JE, Garrido-Varo A. Improving NIRS predictions of ingredient composition in compound feedingstuffs using Bayesian non-parametric calibrations. *Chemom Intel Lab Syst*. 2012;110(1):108-112. https://doi.org/10.1016/j.chemolab.2011.10.007
21. Manne R. Analysis of two partial-least-squares algorithms for multivariate calibration. *Chemom Intel Lab Syst*. 1987;2(1-3):187-197. https://doi.org/10.1016/0169-7439(87)80096-5
22. Höskuldsson A. PLS regression methods. *J Chemometr*. 1988;2(3):211-228. https://doi.org/10.1002/cem.1180020306
23. Bastien P. Régression PLS et données censurées. 2008.
24. Hazama K, Kano M. Covariance-based locally weighted partial least squares for high-performance adaptive modeling. *Chemom Intel Lab Syst*. 2015;146:55-62. https://doi.org/10.1016/j.chemolab.2015.05.007
25. Andersson M. A comparison of nine PLS1 algorithms. *J Chemometr*. 2009;23(10):518-529. https://doi.org/10.1002/cem.1248
26. R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing; 2018. http://www.R-project.org.
27. Naes T, Isaksson T, Kowalski B. Locally weighted regression and scatter correction for near-infrared reflectance data. *Anal Chem*. 1990;62(7):664-673.
28. Ståhle L, Wold S. Partial least squares analysis with cross-validation for the two-class problem: a Monte Carlo study. *J Chemometr*. 1987;1(3):185-196. https://doi.org/10.1002/cem.1180010306
29. Vong R, Geladi P, Wold S, Esbensen K. Source contributions to ambient aerosol calculated by discriminant partial least squares regression (PLS). *J Chemometr*. 1988;2(4):281-296. https://doi.org/10.1002/cem.1180020406
30. Kemsley EK. Discriminant analysis of high-dimensional data: a comparison of principal components analysis and partial least squares data reduction methods. *Chemom Intel Lab Syst*. 1996;33(1):47-61. https://doi.org/10.1016/0169-7439(95)00090-9
31. Barker M, Rayens W. Partial least squares for discrimination. *J Chemometr*. 2003;17(3):166-173. https://doi.org/10.1002/cem.785
32. Shenk J, Westerhaus M, Berzaghi P. Investigation of a LOCAL calibration procedure for near infrared instruments. *J Infrared Spectrosc*. 1997;5(1):223. https://doi.org/10.1255/jnirs.115

33. Centner V, Massart DL. Optimization in locally weighted regression. *Anal Chem*. 1998;70(19):4206-4211. https://doi.org/10.1021/ac980208r
34. Filzmoser P, Liebmann B, Varmuza K. Repeated double cross validation. *J Chemometr*. 2009;23(4):160-171. https://doi.org/10.1002/cem.1225
35. Wold S. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*. 1978;20(4):397-405. https://doi.org/10.2307/1267639
36. Andries JPM, Vander Heyden Y, Buydens LMC. Improved variable reduction in partial least squares modelling based on predictive-property-ranked variables and adaptation of partial least squares complexity. *Anal Chim Acta*. 2011;705(1-2):292-305. https://doi.org/10.1016/j.aca.2011.06.037
37. Aastveit AH, Marum P. Near-infrared reflectance spectroscopy: different strategies for local calibrations in analyses of forage quality. *Appl Spectrosc*. 1993;47(4):463-469. https://doi.org/10.1366/0003702934334912
38. Bevilacqua M, Marini F. Local classification: locally weighted–partial least squares-discriminant analysis (LW–PLS-DA). *Anal Chim Acta*. 2014;838:20-30. https://doi.org/10.1016/j.aca.2014.05.057
39. Atkeson CG, Moore AW, Schaal S. Locally weighted learning for control. *Artif Intell Rev*. 1997;11(1):75-113. https://doi.org/10.1023/A:1006511328852
40. Song W, Wang H, Maguire P, Nibouche O. Local partial least square classifier in high dimensionality classification. *Neurocomputing*. 2017;234:126-136. https://doi.org/10.1016/j.neucom.2016.12.053


**How to cite this article:** Lesnoff M, Metz M, Roger J-M. Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data. *Journal of Chemometrics*. 2020;34:e3209. https://doi.org/10.1002/cem.3209

## APPENDIX A.

Implementation of the weighted version of the PLS improved kernel #1 algorithm. See Dayal and McGregor[19] for the standard (ie, unweighted) version.

a. R version. The function is part of the R package *rnirs* available at GitHub (https://github.com/mlesnoff/rnirs) or by request to the first author of this article.

```
function(X, Y, ncomp, weights) {

### Notations

## Inputs

# X = spectral matrix (n × p)
# Y = response matrix (n × q)
# ncomp = nb. PLS components (integer)
# weights = weights vector (n × 1)



## Outputs

# T = X-score matrix (n × ncomp)
# P = X-loadings matrix (p × ncomp)
# C = Y-loading weights matrix (q × ncomp) (C = B', where B is the b-coefficients matrix)
# W = X-loading weights matrix (p × ncomp)
```

```
# R = projection matrix (p × a)

### Initialization of dimensions n, p, q

n <- nrow(X)
zp <- ncol(X)
zq <- ncol(Y)

### Normalization of the weights to sum to 1

d <- weights/sum(weights)

### Centering of X and Y

Xd <- d * X # d * X = D %*% X

xmeans <- colSums(Xd)

ymeans <- colSums(Y * d)

X <- scale(X, center = xmeans, scale = FALSE)
Y <- scale(Y, center = ymeans, scale = FALSE)

### Initialization of T, P, C, W, R

nam <- paste("comp", 1:ncomp, sep = "")
T <- matrix(nrow = n, ncol = ncomp, dimnames = list(row.names(X), nam))
R <- W <- P <- matrix(nrow = zp, ncol = ncomp, dimnames = list(colnames(X), nam))
C <- matrix(nrow = zq, ncol = ncomp, dimnames = list(colnames(Y), nam))
TT <- vector(length = ncomp)

### Initialization of the kernel matrix X'Y

XY <- crossprod(Xd, Y) # = (D * X)' * Y = X' * D * Y

### Calculations

for(a in 1:ncomp) {

if(zq == 1) w <- XY # PLSR1
else { # PLSR2
z <- svd(t(XY), nu = 1, nv = 0)
w <- XY %*% z$u
}

w <- w/sqrt(sum(w * w))

r <- w
if(a > 1)
for(j in 1:(a - 1)) r <- r - sum(P[, j] * w) * R[, j]

t <- X %*% r
```

```
tt <- sum(t * d * t)

p <- crossprod(Xd, t)/tt

c <- crossprod(XY, r)/tt

XY <- XY − tcrossprod(p, c) * tt # Deflation step

T[, a] <- t
P[, a] <- p
W[, a] <- w
R[, a] <- r
C[, a] <- c

TT[a] <- tt

}

### Outputs

list(T = T, W = W, P = P, C = C, R = R, TT = TT,
xmeans = xmeans, ymeans = ymeans, weights = d)

}
```

a. Matlab version.

```
function mod = pls_kernelw(X, Y, ncomp, weights)

%%% Notations

%% Inputs

% X = spectral matrix (n × p)
% Y = response matrix (n × q)
% ncomp = nb. PLS components (integer)
% weights = weights vector (n × 1)

%% Outputs

% T = X-score matrix (n × ncomp)
% P = X-loadings matrix (p × ncomp)
% C = Y-loading weights matrix (q × ncomp) (C = B?, where B is the b-coefficients matrix)
% W = X-loading weights matrix (p × ncomp)
% R = projection matrix (p × a)

%%% Initialization of T, P, C, W, R, TT

W = [] ;
R = [] ;
```

```
P = [] ;
C = [] ;
T = [] ;
TT = [] ;

%%% Initialization of dimension q

zq = size(Y,2) ;

%%% Normalization of the weights to sum to 1

weights = diag (weights) ;
d = weights/trace (weights) ;

%%% Centering of X and Y

Xd = (X' * d)' ;
xmeans = sum (Xd) ;
ymeans = sum(d * Y) ;
X = X - repmat (xmeans,size(X,1),1) ;
Y = Y - repmat (ymeans,size(Y,1),1) ;

%%% Initialization of the kernel matrix XY

XY = Xd' * Y ;

%%% Calculations

for a = 1:ncomp

 if zq == 1
 w = XY ;
 else
 [U V D] = svd (XY') ;
 w = XY * U(:,1) ;
 end

w = w/sqrt(w'*w) ;
r = w ;

 for j = 1:a - 1,
 r = r - (P(:,j)' * w) * R(:,j) ;
 end

t = X * r ;
tt = t' * d * t ;
p = (Xd' * t)/tt ;
c = (r' * XY)'/tt ;

XY = XY - (p * c') * tt ;

W = [W w] ;
P = [P p] ;
```

```
T = [T t] ;
TT = [TT tt] ;
C = [C c] ;
R = [R r] ;

end

%%% Outputs

mod. P = P; mod. T = T ;
mod. C = C; mod. R = R ;
mod.TT = TT ;
mod.xmeans = xmeans ;
mod.ymeans = ymeans ;
mod.weights = d ;
```

# Chapitre 6

## Article 2 : A note on spectral data simulation

Référence :

Maxime Metz, Alessandra Biancolillo, Matthieu Lesnoff, and Jean-Michel Roger. A note on spectral data simulation. Chemometrics and Intelligent Laboratory Systems, 200 :103979, May 2020. ISSN 0169-7439. doi:10.1016/j.chemolab.2020.103979. URL http://www.sciencedirect.com/science/article/pii/S0169743919306720

# A Note on Spectral Data Simulation

Maxime Metz [1,2], Alessandra Biancolillo [1], Matthieu Lesnoff [2,3] , Jean-Michel Roger [1,2]

[1]ITAP, Montpellier SupAgro, Irstea, Univ Montpellier, Montpellier, France
[2]ChemHouse Research Group, Montpellier, France
[3]CIRAD,UMR SELMET, Univ Montpellier, F-34398 Montpellier, France

## Corresponding author

Maxime Metz
**Email:** maxime.metz@irstea.fr
**Postal address:** 361 Rue Jean François Breton, 34196 Montpellier

## Keywords

Simulation of spectral data, subspaces, near infrared spectroscopy, partial least squares regression

# Abstract

In chemometrics, it is common to simulate data to test new methods. However, it is difficult to find an article that only discusses the spectral data simulation in a global context. Most of the time, the simulation is performed specifically for one method. In this article, a global approach is proposed to simulate spectra and a relation with one or more responses (qualitative or quantitative). This method of simulation is based on the basic principles of chemometrics and allows a simple and fast simulation of data. This will be highlighted by three examples.

# 1. Introduction

One of the main steps in developing chemometric methods is testing them on different databases. This enables the study of method characteristics. However, it is difficult to find data that perfectly illustrates the strengths and weaknesses of novel approaches. Simulation is one way to obtain relevant data. It permits to manage precisely the characteristics of the data, as its structure or its signal to noise ratio.

The data simulation subject has already been widely discussed [2]. However, in chemometrics, spectral data simulation has only been addressed in specific cases and not in a generic context. For instance, a method based on the concept of relevant PCA components [7] has recently been proposed, but only for simulating a linear relation between **X** and **Y** [11]. Other specific simulations were carried out for method studies. For example, in [5] the author simulates 3-way data to analyze a linear multi-way method. To achieve this

goal, the author simulates spectra by multiplying Gaussian elution profiles with experimental spectra. To finish the author adds different Gaussian noises to the data. In [6] the author simulates data to compare linear regression methods. Spectral data simulation is performed using a multivariate normal distribution. In [8], the author performs a simulation using a combination of loadings from a PCA. In [4], the author simulates data using a PLS model. Other approaches have also been developed, for instance in a multi-block context [3] using scores and loadings to simulate different blocks of spectra. In a large number of cases, the authors seek to reproduce most easily a situation close to reality. Additionally, in some cases the rationale behind the choices of the implemented parameters of the simulation methods is difficult to understand. Therefore, we consider that a generic approach for simulating data could facilitate the method testing processings.

Performing a data simulation close to reality in chemometrics is not a straightforward task. Approaches and concepts for understanding the composition of a spectral database have recently been discussed . In particular, Roger and Boulet [9] proposed that a spectrum is composed of a *useful* and a *detrimental* part, related to a useful space and a detrimental space. The useful space is linked to information sought (e.g. the concentration of pure compounds). The detrimental space can is spanned by the other pure compounds but also by other sources of variability possibly present (e.g. temperature, scattering, etc.). Handling the detrimental information present in data is one of the main difficulties that chemometrics must overcome. Therefore, when simulating data for method tests, it is necessary to take into account spurious sources of variability in order to avoid biased interpretations.

The purpose of this paper is therefore to provide a generic simulation frame accounting for the characteristics and realities of spectrometric data. The first part presents the general principle of this generic frame. The second part provides several illustrations of data simulation in 3 different contexts, addressing the aspects of regression, robustness and topology

# 2. Theory

## 2.1. The composition of a spectral database

A generalization of the construction of a set of spectra is proposed in Eq.1:

$$\mathbf{X} = \mathbf{X_u} + \mathbf{X_d} + \mathbf{E} \quad (1)$$

$\mathbf{X_u}$ represents the part of the spectra belonging to the useful space, $\mathbf{X_d}$ the one belonging to the detrimental space and **E** the noise (e.g., Gaussian noise).
To build these spaces, it is assumed that each of them is spanned by several specific structures. These structures contribute more or less to the final spectrum depending on their concentration. As a result, the set of spectra can be detailed as follows:

$$\mathbf{X_u = T_u \cdot P_u'} \quad \textbf{(2)}$$

and

$$\mathbf{X_d = T_d \cdot P_d'} \quad \textbf{(3)}$$

Where $\mathbf{T_u}$ are the scores of the loadings $\mathbf{P_u}$ for the *useful* part, and $\mathbf{T_d}$ are the scores of the loadings $\mathbf{P_d}$ for the *detrimental* part.
If some **Y** must be predicted from **X**, the decomposition carried out in Eq.2 allows us to link **X** to **Y** by:

$$\mathbf{Y = \mathit{f}(T_u) + F} \quad \textbf{(4)}$$

**Y** is the reference value, ***f*** is the function that links the contributions of the column structures from $\mathbf{X_u}$ to **Y**. **F** is a noise (e.g: Gaussian).

## 2.2. Simulation of a database

Following the approach outlined in section 3.1, the simulation can be summarized by the generic frame:

$$\mathbf{X = T_u.P_u + T_d.P_d + E} \quad \textbf{(5)}$$

and

$$\mathbf{Y = \mathit{f}(T_u) + F} \quad \textbf{(6)}$$

and five types of parameters can be modified in order to obtain the desired database (see Table 1)

| **Parameter** | **Type** | **Impact** |
|---|---|---|
| **T** | Scores | Topology |
| **P** | Loadings | Correlation among spectral variables |
| **E** | Noise | Signal to noise ratio |
| *f* | Function | Relation between **X** and **Y** |
| **F** | Noise | Fitting error |

**Table 1: Type and impact of the different parameters in the simulation**

Each parameter has a different impact on the outcome of the simulation.
**T** scores are associated with the database topology. Moreover, they allow us to modify the appearance of the spectra. Indeed, the scores correspond to the contributions of each loading in the spectral database. Scores can be defined with parametric methods, using a priori distribution (Gaussian, etc.), or with nonparametric methods from an existing database (inverse CDF, etc.). In spectral data, it is common to have interactions. To simulate interactions it is possible to link scores to other scores or to create new scores related to an interaction spectrum.

The loadings **P** define the correlation between the spectral variables and they correspond to specific *spectral signatures*. **P** can be obtained in different ways. First, they can be real measured spectra of pure compounds. Second, they can be provided as the loadings of a PCA [14] or a PLS [13] model on an existing database. Third, they can be provided as the demixed spectra, using methods like independent component analysis (ICA) [10], Multivariate curve resolution alternating least squares (MCR-ALS) [1], classical least squares (CLS) [12]. Eventually, it is possible to simulate them as combination of functions (e.g., Gaussian mixture).

The noise **E** makes it possible to modify the signal-to-noise ratio of the simulated spectra. **E** can be obtained by simulation (e.g: Gaussian noise) or by measurements (e.g: spectral measurements in the dark).

The relation ***f*** can be either theoretical or estimated from a real database (eg: PLSR model).

Noise **F** represents the noise of the reference measurement. Like **E,** it can be obtained by simulation (e.g: Gaussian noise) or by measurements (e.g: repetitions).

# 3. Material and methods

## 3.1. Simulations

Twenty-two (22) different mixtures of glucose, water, and ethanol were prepared in controlled proportions. The spectra of these mixtures were measured by a Jasco V-670 spectrometer working on the spectral range 500-2500 nm. CLS applied on these data provided 4 chemical loadings made up of the estimated spectra of water, glucose, ethanol and water-ethanol interaction. In addition, artificial loadings were simulated by combinations of random Gaussian curves. Three series of 3 simulations were carried out using these loadings, each series illustrating a different strategy. Each simulation produced a set of 1000 spectra (**X**) and 1000 responses (**Y**) following the workflow below:

- Distribute the chemical and artificial loadings between the detrimental ($\mathbf{P_d}$) and the useful ($\mathbf{P_u}$) spaces
- Build the score matrices ($\mathbf{T_u}$ and $\mathbf{T_d}$) following predefined distributions
- Build the noise matrix **E** following a normal multivariate distribution N(0,sigma), with sigma = 10 % of the total signal range.
- Build **X** by the equation **(5)**
- Build **Y** from $\mathbf{T_u}$ according to the following relation: $\mathbf{Y} = 1000*\mathbf{T_u} + \mathbf{F}$, with **F** following a normal distribution N(0,sigma), with sigma = 15 % of the standard deviation of **Y**

The scores **T** were simulated independently except for the scores corresponding to the interaction spectrum, which were obtained by multiplying the scores related to ethanol and water. The score values associated with pure spectra were set higher than other scores. In order to respect the additivity of the absorbances, the $\mathbf{T_u}$ and $\mathbf{T_d}$ scores cannot have a negative value. Consequently, to simulate only positive values a folded-normal distribution was used. The noise **E** added to the spectral data was spherical. For the first two simulations, several PLS models were applied on simulated data, varying the calibration and test sets. The calibration set and the test set accounted for 80% and 20% of the data, respectively.

## 3.2. Simulation 1

In simulation 1, The dimension of $\mathbf{P_d}$ was varied (3, 20 and 100). Three loadings related to the water, ethanol and water-ethanol interaction spectrum were common to the 3 simulations. The other corresponded to artificial spectra. Simulations 1a, 1b, 1c were carried out following the Table 2.

| | **Simulation 1a** | **Simulation 1b** | **Simulation 1c** |
|---|---|---|---|
| $P_u$ | Pure spectrum of ethanol | | |
| $T_u$ | Folded-normal distribution | | |
| $P_d$ | Pure spectrum of water<br>Pure spectrum of glucose<br>Spectrum of water-ethanol Interaction<br>**0 Artificial loadings** | Pure spectrum of water<br>Pure spectrum of glucose<br>Spectrum of water-ethanol Interaction<br>**17 Artificial loadings** | Pure spectrum of water<br>Pure spectrum of glucose<br>Spectrum of water-ethanol Interaction<br>**97 Artificial loadings** |
| $T_d$ | Folded-normal distribution<br>Folded-normal distribution<br>Product of $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution | | |

| | |
|---|---|
| E | Gaussian distribution |
| f | Linear |
| F | Gaussian distribution |

**Table 2: The different choices in the simulation 1**

## 3.3. Simulation 2

For Simulation 2, the numbers of $\mathbf{P_d}$ loadings (3,100,200) were chosen. In this simulation all the simulated spectra were not constructed with all the dimensions of $\mathbf{P_d}$. Each spectrum was constructed with a part of the $\mathbf{P_d}$ dimensions selected randomly. To achieve this, the $\mathbf{T_d}$ scores associated with the artificial loadings were constructed with a combination of a random selection (0,1) and a folded-normal distribution. The different choices of parameters are shown in Table 3.

| | **Simulation 2a** | **Simulation 2b** | **Simulation 2c** |
|---|---|---|---|
| $P_u$ | Pure spectrum of ethanol | | |
| $T_u$ | Folded-normal distribution | | |
| $P_d$ | Pure spectrum of water<br>Pure spectrum of glucose<br>Spectrum water-ethanol Interaction<br>**0 Artificial spectra** | Pure spectrum of water<br>Pure spectrum of glucose<br>Spectrum water-ethanol Interaction<br>**97 Artificial spectra** | Pure spectrum of water<br>Pure spectrum of glucose<br>Spectrum water-ethanol Interaction<br>**197 Artificial spectra** |
| $T_d$ | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>**Random selection) + folded-normal distribution** | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>**Random selection) + folded-normal distribution** | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>**Random selection) + folded-normal distribution** |
| E | Gaussian distribution | | |
| f | Linear | | |
| F | Gaussian distribution | | |

**Table 3: The differents choice in the simulation 2**

## 3.4. Simulation 3

Three different structures were simulated. The distributions of the scores $\mathbf{T_u}$ and $\mathbf{T_d}$ were changed. For the first two cases, 2 and 3 classes were simulated. In this aims, $\mathbf{T_u}$ were modified using two and three folded-normal distributions. In the third case, a relation between the scores $\mathbf{T_u}$ (ethanol) and $\mathbf{T_d}$ (water) was created. This relation was given by Eq.(7):

$$\mathbf{T_{water} = f(T_{eth}) = \sqrt{(1 - T_{eth})^2}} \qquad \mathbf{(7)}$$

For the other two groups, $\mathbf{T_u}$ (ethanol) and $\mathbf{T_d}$ (water) scores were simulated with Gaussian distributions; for $\mathbf{T_d}$ (water) the standard deviation of the two classes was the same but the average was different.

| | **Simulation 3a** | **Simulation 3b** | **Simulation 3c** |
|---|---|---|---|
| $P_u$ | Pure spectrum of ethanol | | |
| $T_u$ | Folded-normal distribution | | |
| $P_d$ | Pure spectrum of water<br>Pure spectrum of glucose<br>Spectrum of water-ethanol interaction<br>17 Artificial spectra | Pure spectrum of water<br>Pure spectrum of glucose<br>Spectrum of water-ethanol interaction<br>17 Artificial spectra | Pure spectrum of water<br>Pure spectrum of glucose<br>17 Artificial spectra |
| $T_d$ | **Two folded-normal distribution**<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution | **Three folded-normal distribution**<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution | **Two folded-normal distribution and (7)**<br>Folded-normal distribution<br>Folded-normal distribution |
| E | Gaussian distribution | | |
| f | Linear | | |
| F | Gaussian distribution | | |

**Table 4: The different choices in the simulation 3**

# 4. Results and discussion

For all the simulations, the first step was the unmixing of the pure spectra from the 22 measured spectra. Figure 1a and 1b show the unmixed and real pure spectra of ethanol and water.

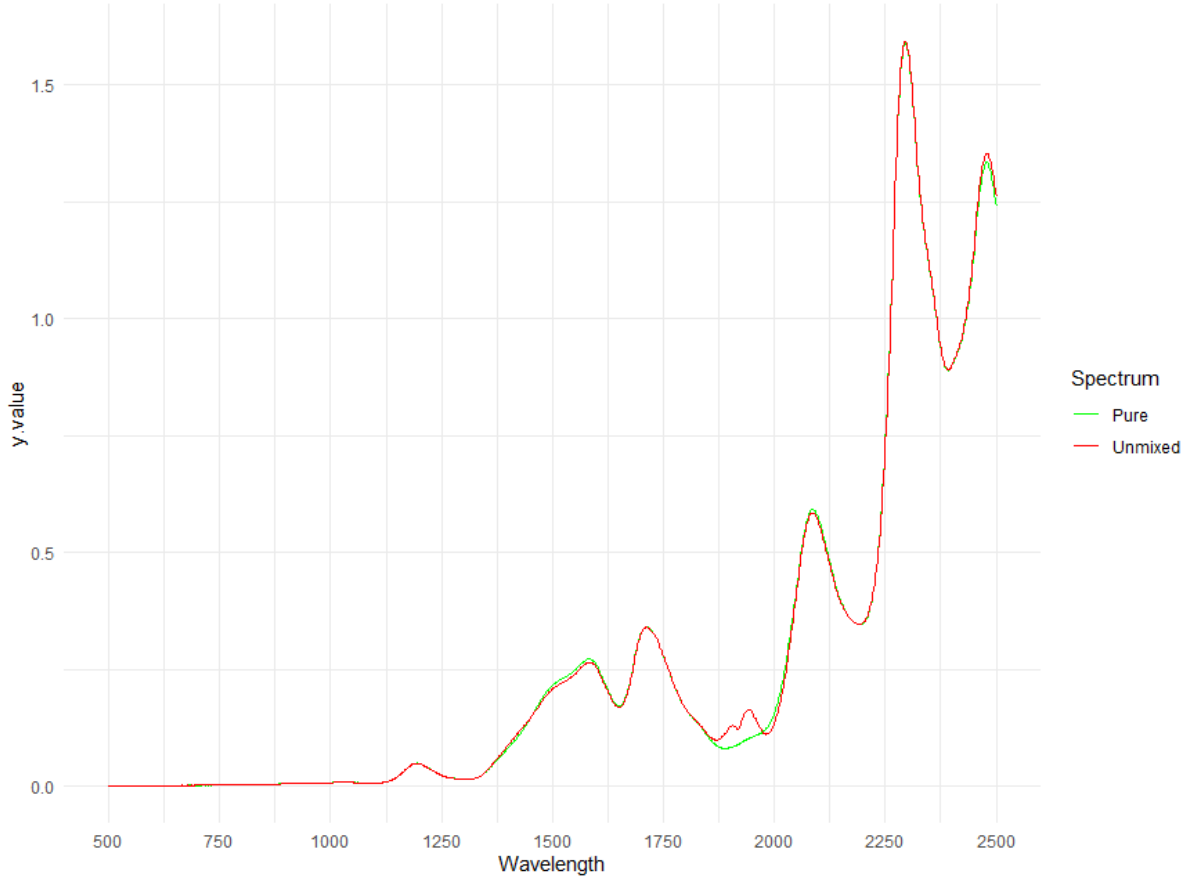



**Figure 1a: Spectra plot of the pure (green) and unmixed spectra (red) of ethanol**

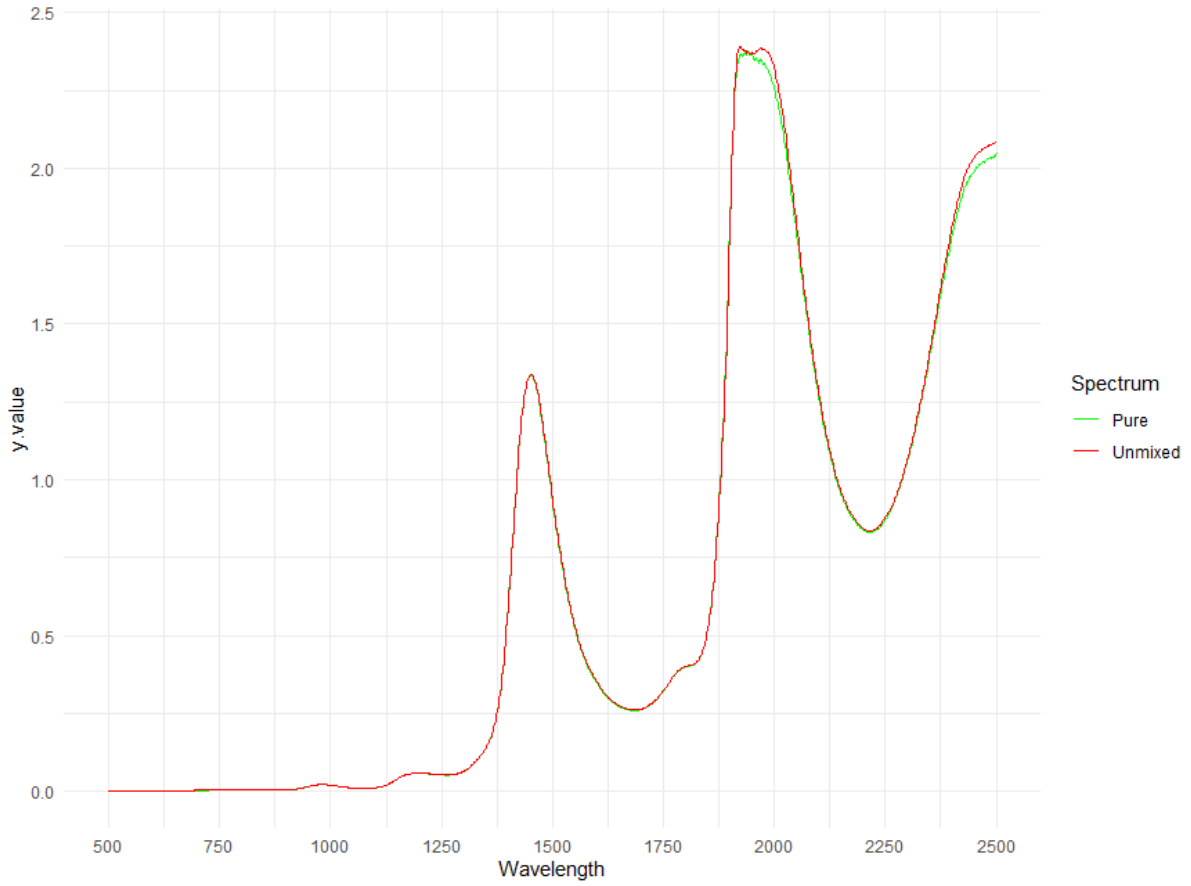



**Figure 1b: Spectra plot of the pure (green) and unmixed spectra (red) of water**

## 4.1. Simulation 1

In Simulation 1, the detrimental space part was varied. Three databases were simulated by adding dimensions in $\mathbf{P}_d$. Ten PLSR models were calculated with different calibration and test sets.

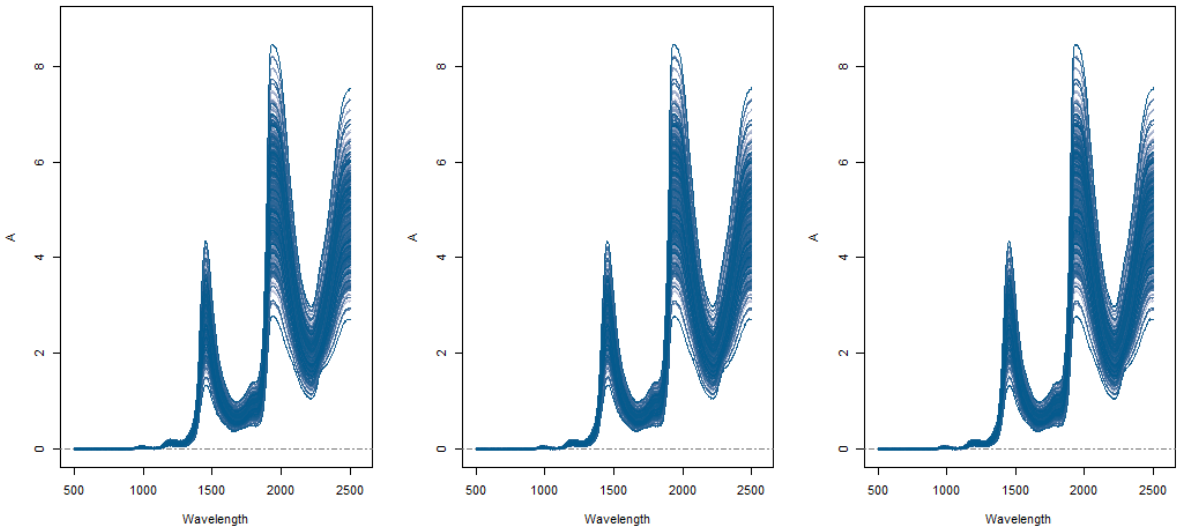



**Figure 2: Spectra plot of the 3 simulation sets (1a, 1b, 1c)**

Figure 2 shows the 3 sets of simulated spectra. In this figure, the spectra of the 3 sets have real spectral appearances. The 3 simulated sets look alike. On this graph it is not possible to deduce that the 3 sets of spectra are different. The spectra have a real appearance because the scores that have the highest values are associated with loadings representing ethanol, glucose and water. It is not possible to observe the differences between the 3 sets because the added artificial loadings are associated with low value scores. The generic frame allows using the scores to set the final form of the simulated spectra. The score generation allows integrating unobservable sources of variability (loadings) with a spectra plot.
Having a source of main variability and other auxiliary in a very small proportion in the spectrum makes it possible to simulate real situations, for example traces of compounds.

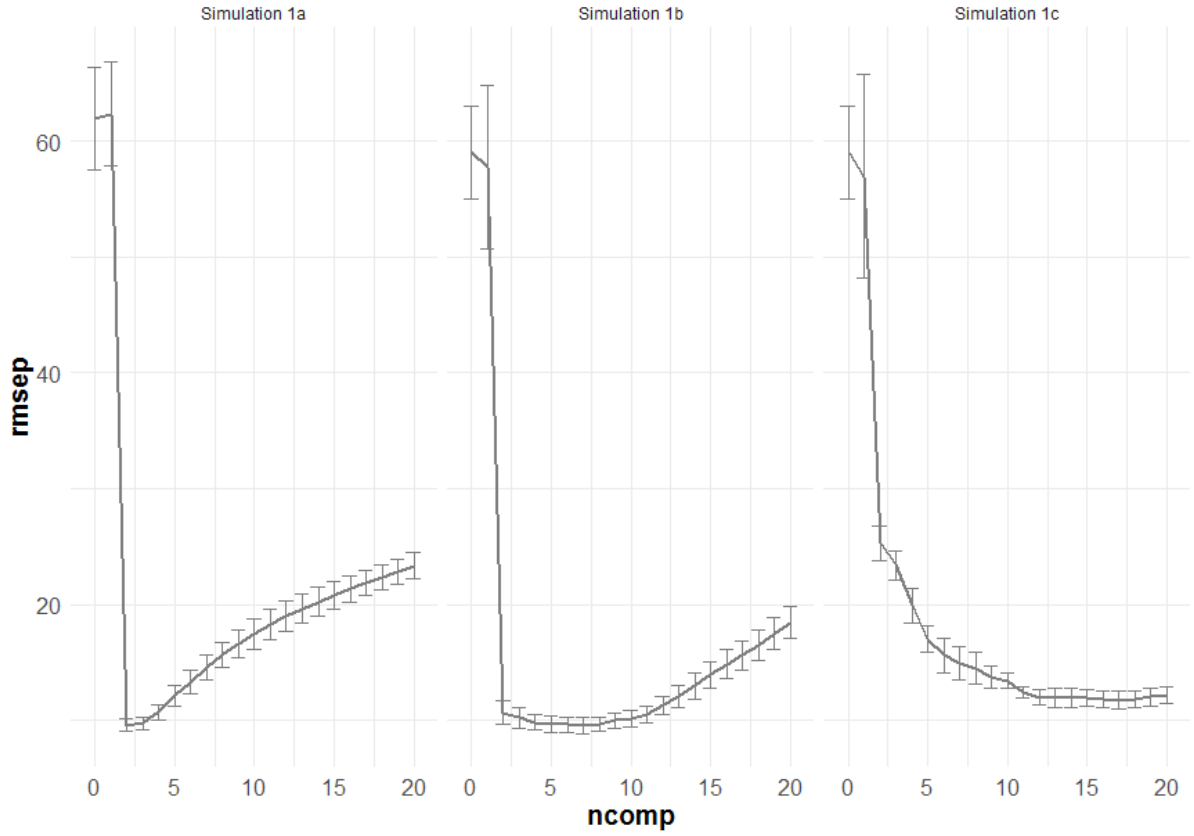



**Figure 3: Evolution of the RMSEP for different numbers of loadings $P_d$, as a function of the number of latent variables of the PLS. Left: simulation 1a, Middle:simulation 1b, Right: simulation 1c**

Figure 3 represents the Root Mean Square Error in prediction (RMSEP) as a function of the number of latent variables. This figure is divided into 3 graphs corresponding to the different dimensions of $\mathbf{P_d}$ (3, 20, 100). Each plot shows the mean error curve and error bars corresponding to the standard deviation of the 10 repetitions. The first graph in Figure 3 (simulation 1a) corresponds to 3-dimensional $\mathbf{P_d}$ loadings in the simulation. An "elbow" and a minimum of 2 latent variables can be observed. Then, the error increases with the number of latent variables. The second graph (simulation 1b) corresponds to 20-dimensional $\mathbf{P_d}$ loadings in the simulation. An "elbow" for 2 latent variables and a minimum around 5 latent variables can be observed. Then, a "plateau" followed by an increase in the error with the number of latent variables is observed. The third graph (simulation 1c) corresponds to 100-dimensional $\mathbf{P_d}$ loadings in the simulation. An "elbow" between 2 and 5 latent variables and a minimum around 15 latent variables are observed. Then, a "plateau" up to 20 latent variables is observed. The "elbow" present on the curves with 3 loadings changes with the number of loadings in $\mathbf{P_d}$. On the 3 graphs the minimal error and the position of the minimal error of the curves increase with the dimension of $\mathbf{P_d}$.

These observations allow us to deduce several effects of $\mathbf{P_d}$ on the simulated data. The scores are used to define the shape of the simulated spectra. The dimension of $\mathbf{P_d}$ makes it difficult to extract the information about **Y** contained in **X**. Indeed, in Figure 3, when the

dimension of $\mathbf{P_d}$ increases, the number of optimal latent variables deviates from the real dimension of $\mathbf{T_u}$. In conclusion, with the proposed generic framework, it is possible to make it more difficult to extract information about **Y** in **X** (with $\mathbf{P_d}$) and also to keep a real appearance. The generic framework also allows to mix different types of loadings such as pure / interaction / artificial spectra. Finally, cases that mimic real behaviour could be simulated. Indeed, it is common to obtain PLS models with a large number of latent variables when the compounds analyzed are complex.

## 4.2. Simulation 2

In Simulation 2, the detrimental part was varied like in the simulation 1. Three databases were simulated by changing the number of $\mathbf{P_d}$ loadings (3,100,200). In this case, the loadings are not all common to each simulated spectrum. Ten PLSR models were calculated with different calibration and test sets.

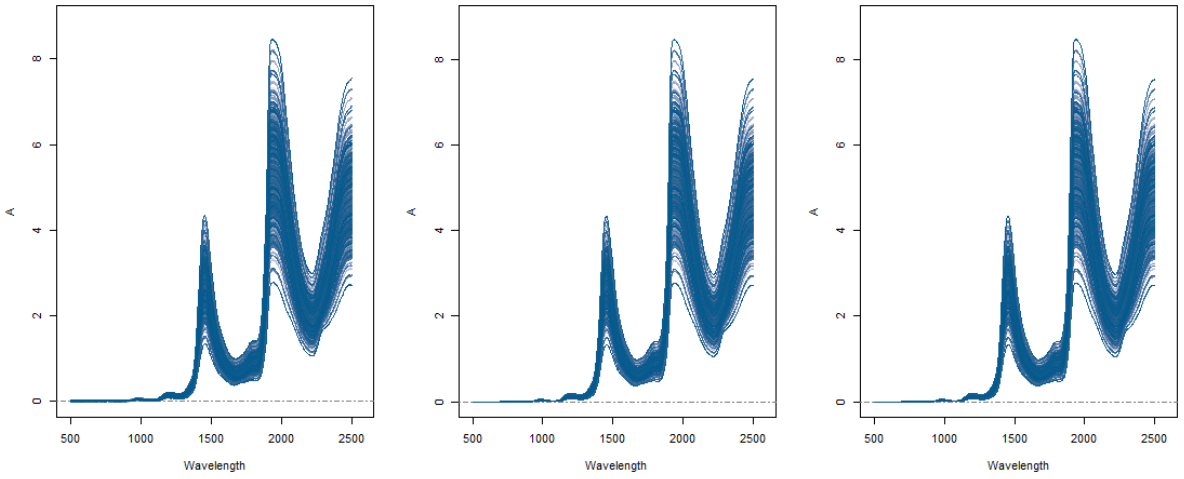



**Figure 4: Spectra plot of the 3 simulation sets (2a, 2b, 2c)**

Figure 4 shows the 3 sets of simulated spectra. In this figure, the spectra of the 3 sets have real spectral appearances. The simulated spectra for 3, 100, 200 loadings are similar. On this graph it is possible to make the same observations and conclusions as on the Figure 2. Spectra have a real appearance because the loadings that make up the 3 sets are the pure spectra (ethanol, glucose, water). It is not possible to observe the difference between the 3 sets because the loadings added in the simulation have a low value of scores.

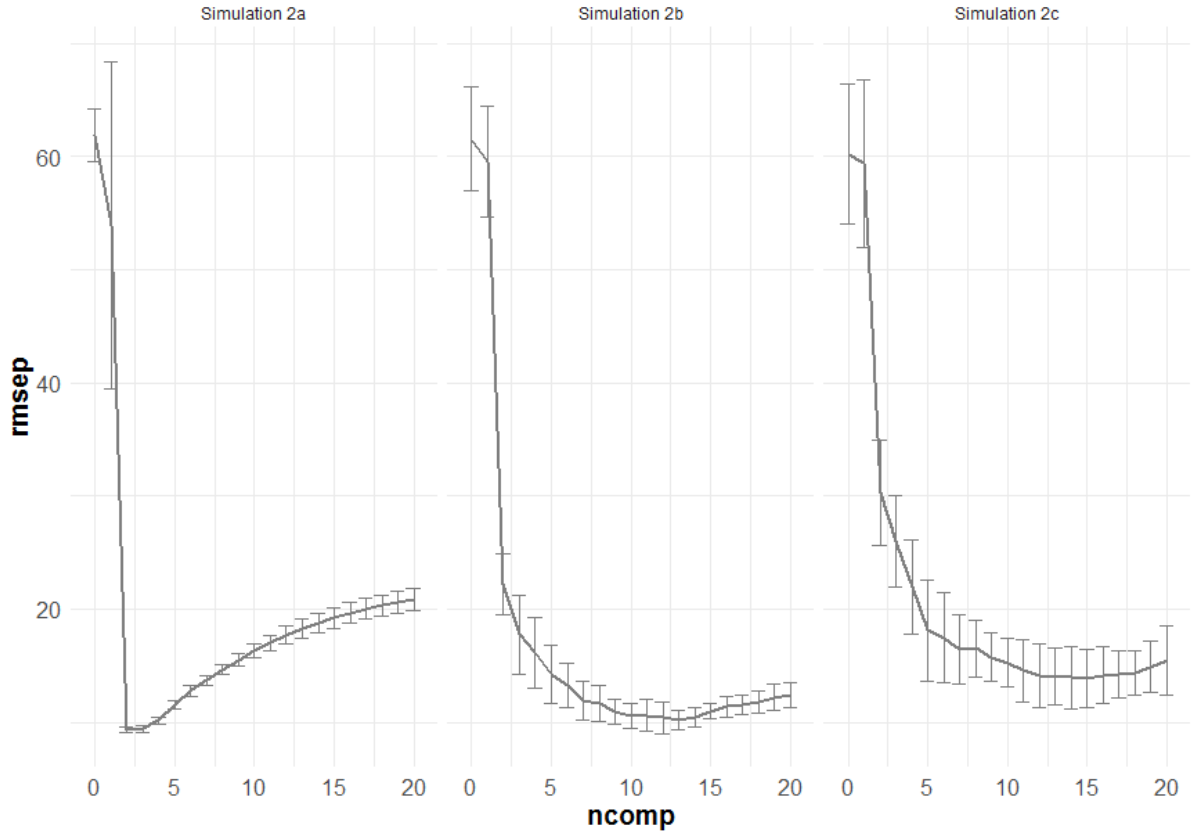



**Figure 5: Evolution of the mean error of predictions for the different numbers of loadings $P_d$, as a function of the number of latent variables**

The first graph in Figure 5 (simulation 2a) represents the same simulation choices as the first graph in Figure 3 (simulation 1a). In the first plot in Figure 5, the error bars are very small and shows a little variation with the latent variable numbers. This means that repetitions lead to very small variations. In the second graph of Figure 5 (simulation 2b), the curve is formed by “elbow” followed by a “plateau”. Also, the error bar variations are much greater than the variations in the graphs of Figure 3. The third graph in Figure 5 (simulation 2c), shows a higher overall level of error curves than the other graphs. An increase of the minimal error with the dimension of $\mathbf{P_d}$ can be observed. There is the same impact of the dimension $\mathbf{P_d}$ on the error curves as in simulation 1. However, big variations of error bars are observed. Indeed, in simulation 2 the test and calibration sets have a strong influence on the calculation of the model and therefore on the associated error curves.

The test and calibration sets have a strong influence on the error curves when a random selection (0.1) is combined with a folded normal distribution (simulation 2b, and 2c) to build the scores. This is due to the fact that only a few $\mathbf{P_d}$ dimensions are used to construct a spectrum. This means that loadings are not common to each spectrum. In conclusion, the construction of scores allows to create a robustness problem. In reality, this problem related to robustness is common. For example, it is possible to observe a robustness problem when some samples have been contaminated.

## 4.3. Simulation 3

In the simulation 3, the scores $\mathbf{T_u}$ were modified in order to create different groups of data. Here, the relation between $\mathbf{T_u}$ and **Y** remains the same for each group. For simulation 3, to describe the topological modification of the database, the spectra and scores of the first two PLS latent variables are observed.

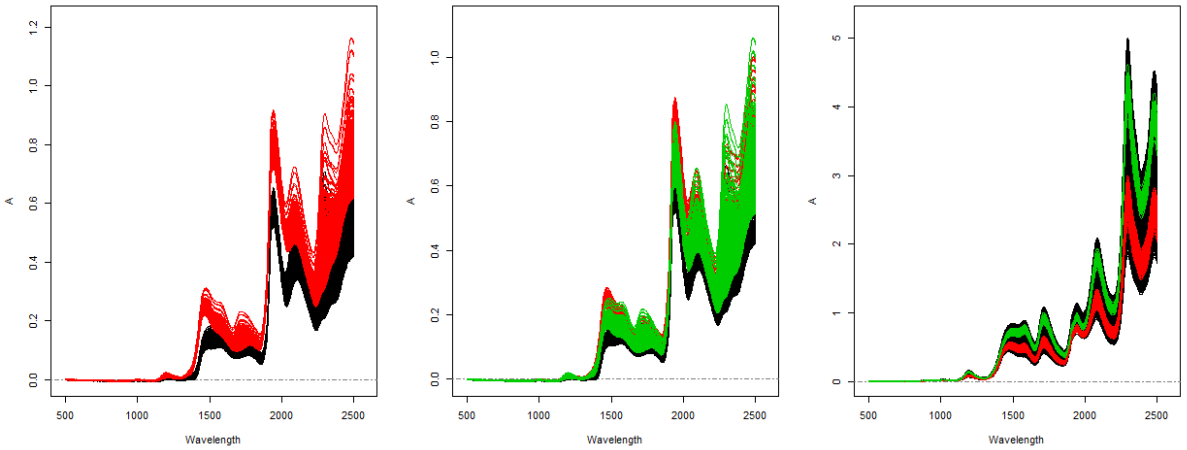



**Figure 6: Spectra plot of the 3 simulation sets (3a, 3b, 3c), the colors represent the different simulated groups**

Figure 6 shows the 3 sets of simulated spectra. In this figure the spectra of the 3 sets have real spectral appearances. The simulation 3a shows 2 groups of spectra, these two groups are sufficiently distant to visualize this difference on the spectra plot. Simulation 3b, shows 3 groups of spectra (red, green, black). The 3 groups of spectra are not sufficiently separated to distinguish them on the spectra plot. For the simulation 3c , 3 groups of spectra are not sufficiently separated to distinguish them on the plot spectra. The groups of spectra are visible on the simulation 3a because the scores corresponding to the majority loadings in the spectra consist of two normal distributions of very different means and low standard deviations. On the other two graphs ( simulation 3b/3c) the groups do not stand out because the averages of each group of scores constructed are close or have high standard deviations. This interpretation highlights the possibility to build groups of spectra thanks to the **T** scores. By modifying the parameters of simulation of the scores (ex: average and standard deviation for a normal distribution) it is possible to make visible differences through the spectra plot or not.

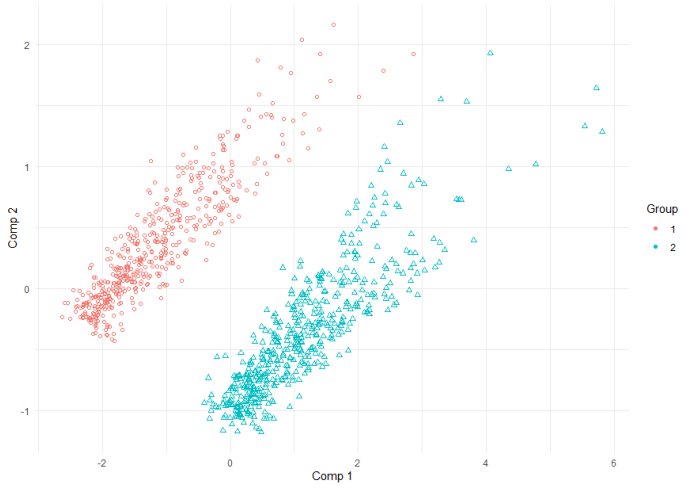



**Figure 7a: Plot of the first two Pls scores for the first simulation**

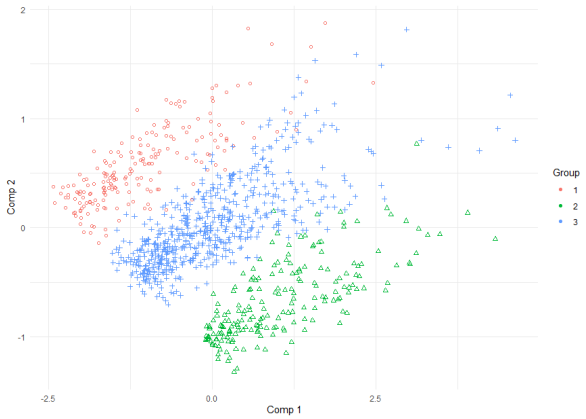



**Figure 7b: Plot of the first two PLS scores for the second simulation**

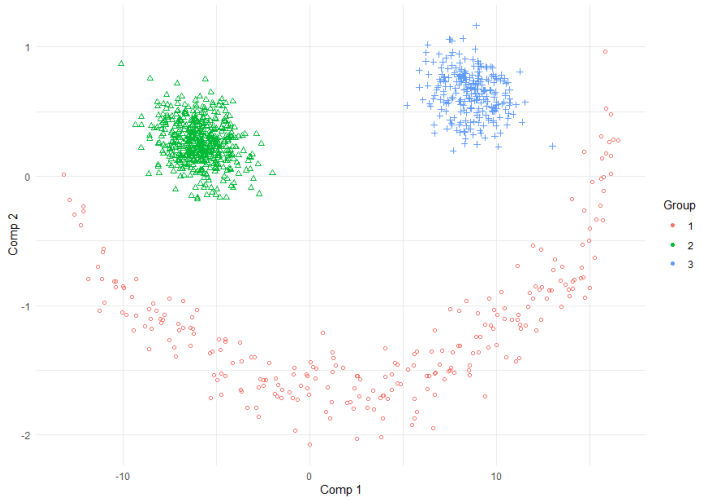



**Figure 7c: Plot of the first two PLS scores for the third simulation**

Figures 7a/7b/7c represent the scores of the two latent variables of a PLS model. Figure 7a shows 2 groups of data. The groups have a rectangle shape, and are very distinct. Figure 7b, shows 3 groups of data. Groups have a rectangle shape. It is difficult to identify groups by observing the plot score. Figure 7c, shows 3 groups of data. Groups have special shapes. Two groups form circles while the other forms an arc. The formation of rectangle in figures 8a and 8b is due to the fact that there is an interaction between water and ethanol. In Figure 7a, the 2 groups are distinguished because the average scores of each group are very different and the standard deviation values are not high. For the Figure 7b the groups are not separated on the score plot because the averages of the scores of each group are close and the standard deviation high enough for classes to overlap. For the Figure 7c the 3 groups of simulated scores have a particular shape corresponding to an interaction. Indeed, the arcuate shape of group 3 is due to a relation between ethanol and water. The circular form of the other two classes is due to the fact that there is no relation between the scores.
We can conclude that the proposed frame makes it possible to act on the database topology. This topology can be observed directly from a spectra plot if it is due to scores representing a loadings mainly present in the simulated spectra. This topology can also be observable by a PCA or PLS if the loadings are not mainly present in the spectra. It is possible to integrate interactions between the scores. One solution is to create loadings that correspond to an interaction spectrum. The scores associated with these loadings are linked by a relation. Another solution is to directly create a relation between two dimensions of scores, e.g. using a function as in equation (7).

To conclude, the scores make it possible to form groups. But the scores also make it possible to simulate relations between scores. The generic frame thus allows to generate an infinity of different topology.

## 4.4. Conclusion

In this paper, a generic spectral data simulation frame, based on the notion of useful and detrimental spaces is proposed. It allowed us simulating 3 different databases. The first one represents the difficulty of a method of extracting relevant information. The second one aims at imitating problems of robustness in spectroscopy. Finally, the third one focus on the topology of the simulated database. These examples have highlighted the fact that it is very easy to use this generic frame to simulate a lot of problems. But these 3 examples are only a small example of all the possibilities offered by this generic frame. Indeed, it is possible to vary a large number of parameters, for instance, noise on data, relations between **X** and **Y**, relations between scores, etc. In addition, it is often very difficult to effectively explain the simulation process in an article. With this generic frame it is very easy to explain and summarize the simulation process. Finally, this generic frame offers a better perspective for the study of chemometric methods. Indeed, the generic frame uses principles proposed in chemometrics to simulate data close to reality. This simulation approach therefore makes it possible to promote the development and study of future methods used in chemometrics.

# 5. Acknowledgements

This work was supported by the French National Research Agency under the Investments for the Future Program, referred as ANR-16-CONV-0004.

# 6. References

**[1] Azzouz, T., Tauler, R., 2008. Application of multivariate curve resolution alternating least squares (MCR-ALS) to the quantitative analysis of pharmaceutical and agricultural samples. Talanta 74, 1201–1 210.**

**[2] B.D. Ripley, Stochastic Simulation, vol. 316, John Wiley & Sons, 2009**

**[3] Biancolillo, A., Liland, K.H., Måge, I., Næs, T., Bro, R., 2016. Variable selection in multi-block regression. Chemometrics and Intelligent Laboratory Systems 156, 89–101. https://doi.org/10.1016/j.chemolab.2016.05.016**

**[4] Denham, M.C., 2000. Choosing the number of factors in partial least squares regression: estimating and minimizing the mean squared error of prediction. Journal of Chemometrics 14, 351–361.**

**[5] Faber, M., Bro, R., 2002. Standard error of prediction for multiway PLS 1. Background and a simulation study. Chemometrics and Intelligent Laboratory Systems 17.**

**[6] Frank, Ildiko E., Friedman, J.H., 1993. A Statistical View of Some Chemometrics Regression Tools. Technometrics 35, 109–135.**

**[7] Helland, I.S., Almøy, T., 1994. Comparison of Prediction Methods when Only a Few Components are Relevant. Journal of the American Statistical Association 89, 583–591.**

**[8] Jørgensen, K., Segtnan, V., Thyholt, K., Naes, T., 2004. A comparison of methods for analysing regression models with both spectral and designed variables: Methods for analysing regression models. J. Chemometrics 18, 451–464.**

**[9] Roger, J.-M., Boulet, J.-C., 2018. A review of orthogonal projections for calibration. Journal of Chemometrics 32, e3045.**

**[10] Rutledge, D.N., Jouan-Rimbaud Bouveresse, D., 2013. Independent Components Analysis with the JADE algorithm. TrAC Trends in Analytical Chemistry 50, 22–32. https://doi.org/10.1016/j.trac.2013.03.013**

**[11] Sæbø, S., Almøy, T., Helland, I.S., 2015. simrel — A versatile tool for linear model data simulation based on the concept of a relevant subspace and relevant predictors. Chemometrics and Intelligent Laboratory Systems 146, 128–135.**

**[12] Vajna, B., Patyi, G., Nagy, Z., Bódis, A., Farkas, A., Marosi, G., 2011. Comparison of chemometric methods in the analysis of pharmaceuticals with hyperspectral Raman imaging. J. Raman Spectrosc. 42, 1977–1986.**

**[13] Wold, S., Sjöström, M., Eriksson, L., 2001. PLS-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems 58, 109–130. https://doi.org/10.1016/S0169-7439(01)00155-1**

**[14] Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemometrics and Intelligent Laboratory Systems 2, 37–52. https://doi.org/10.1016/0169-7439(87)80084-9**

# Chapitre 6

## Article 3 : A “big-data” algorithm for KNN-PLS

Référence :

Maxime Metz, Matthieu Lesnoff, Florent Abdelghafour, Reza Akbarinia, Florent Masseglia, and Jean-Michel Roger. A “big-data” algorithm for KNN-PLS. Chemometrics and Intelligent Laboratory Systems, 203 :104076, August 2020.ISSN 0169-7439. doi:10.1016/j.chemolab.2020.104076. URL http://www.sciencedirect.com/science/article/pii/S0169743920301908

# A “big-data” algorithm for local-PLS

Maxime Metz [1,2], Matthieu Lesnoff [2,3,4], Florent Abdelghafour[1,2], Reza Akbarinia[5], Florent Masseglia[5], Jean-Michel Roger [1,2]

[1]ITAP, Univ Montpellier, INRAE, Institut Agro, Montpellier, France

[2]ChemHouse Research Group, Montpellier, France

[3]CIRAD, UMR SELMET, Montpellier, France

[4]SELMET, Univ Montpellier, CIRAD,INRA,Institut Agro, Montpellier,France

[5]Inria & LIRMM, Univ Montpellier, France

## Corresponding author

Maxime Metz

**Email:** maxime.metz@inrae.fr

**Postal address:** 361 Rue Jean François Breton, 34196 Montpellier

**Keywords :** KNN-PLSDA, PLSDA, parSketch, local model , Big-data

# Abstract

PLS is a common tool used in chemometrics, however, it does not apprehend well nonlinearities in data. An extension of the PLS, developed in order to resolve this issue, is the "local-PLS", also known as "KNN-PLS". With the recent developments in spectroscopic instrumentation (especially hyperspectral imaging), it became convenient to acquire considerable amounts of data. However, the KNN-PLS method is based on a neighbourhood

selection algorithm whose execution time is highly dependent on the size of the database leading to prohibitive response times. An alternative to solve this issue is to use "big-data" methods. This article proposes a new method for processing large volumes of data designated as "parSketch-PLS". This method combines a "big-data" domain neighbour selection method, called "parSketch", and the PLS method. parSketch has already been studied in terms of calculation costs in relationship with the data size / dimension and has been proven very efficient. However, this method is based on approximation of sample neighbourhoods. It is then necessary to investigate the relevance of these neighbourhoods for PLS models and predictions. This article compares PLS and KNN-PLS methods with the parSketch-PLS method. In this context, PLS allows to process large volumes of data quickly but performs poorly in prediction while the KNN-PLS method returns accurate predictions, yet with much higher computational time. In addition, a comprehensive study of the input parameters of parSketch-PLS is conducted. The objective being understanding the influence of these parameters on the prediction performances. Finally, this article presents parSketch-PLS as an alternative within the KNN-PLS method. This pairing offers a good operational trade-off between prediction performances and computational cost.

# 1. Introduction

Chemometrics offers a wide range of tools for the analysis and interpretation of spectroscopic data. One of the objectives of these tools is to associate spectral information with physicochemical properties in order to predict these properties. Among them, a reference method, PLSR (Partial Least Squares Regression) [1], enables to realise very efficient predictive models when there is a linear relationship between the spectra and the physicochemical property(ies) of interest. PLSR is composed of a dimension reduction step (PLS) followed by a regression on the scores produced. Similarly, it is possible to carry out a

discrimination calculating a discrimination model on the PLS scores [20]. This article focuses indistinctly on the two methods, under the term PLS. Due to the diversity of applications in the field, it is common to be confronted with data resulting from the aggregation of measurements carried out on samples of different natures. This aggregation often introduces non-linearities in the data (curvatures, clustering). These nonlinearities can significantly alter the quality of the predictions [2]–[4]. A solution to this problem is the use of "local" methods [2], [5]–[10]. These methods essentially consist in considering subsets of the data within which it becomes relevant to apply linear methods such as PLS. One of the most common local methods is "KNN-PLS" (K Nearest Neighbours - PLS) [2], [5], [9], [11]–[19]. The KNN-PLS method consists in determining a neighbourhood of the sample to be predicted, using a similarity criterion, and then calculating a PLS model on the neighbourhood of this sample. This method solves regression problems by computing a PLSR model on each neighbourhood. This method can also be applied to classification problems by computing a PLSDA (PLS Discriminant Analysis) model [20] on the neighbourhood. Similarity criteria is one of the most studied issues regarding implementations of KNN-PLS models [21], [22]. However, paradoxically, only a few studies have been conducted on neighbours selection and on the associated algorithms. Current KNN-PLS methods are all implementations of the *“brute-force”* algorithm, which consists in calculating all dissimilarities between the sample to be predicted and all the samples in the database, then ordering these samples according to the dissimilarities. The *"brute-force"* algorithm has the advantage of being a straightforward and accurate calculus. However, it is fastidious or even unfeasible to process large databases using this method.

Recent developments in spectroscopic instrumentation (especially hyperspectral imaging) make it possible to acquire large volumes of data. It becomes then unreasonable to apply methods as computationally intensive as the *"brute-force"* algorithm on these data. An alternative is the use of *"big-data"* methods. Numerous methods have been developed in this

field in order to accelerate the search for neighbours. All these methods share the central notion of indexation [23]–[27]. An index is a data structure that enables the search for samples in a time that is sub-linear (often logarithmic) to the size of the database. Therefore, indexing a database consists in adding its data to the index structure to be able to find them quickly. However, the conventional indexing structures are not suitable for spectral data, as they contain large numbers of highly inter-correlated variables. Recently, indexing methods have been developed to deal with time series [28]–[35] which present similar issues with spectral data. However, only a few time series indexing methods are suitable for large volumes of data. Two methods have recently been developed to be processed with extensively parallel architectures in order to process quickly large databases: DPiSAX (Distributed Partitioned indexed Symbolic Aggregate approXimation) [36] and parSketch [37], [38]. In this work, it is proposed to apply and study parSketch in the context of chemometrics.

The parSketch method combines dimension reduction, achieved by projection on random vectors [37] , with the creation of lists of samples based on grids. parSketch's efficiency has been illustrated in terms of calculation cost compared to the *"brute-force"* method [37], [38]. In this article, parSketch has been evaluated to replace the *"brute-force"* algorithm in the KNN-PLS method, *i.e.* to use the parSketch method and then compute a PLS model on the resulting neighbourhood ("parSketch-PLS"). However, parSketch approximates a neighbourhood. It is therefore necessary to test the influence of this approximation on the quality of PLS results.

The rest of this article is organised as follows. Chapter 2 describes the parSketch theory. Chapter 3 describes the data used. The amount of data considered in this work is chosen to be at the limits of what KNN-PLS algorithms can usually process, in order to compare the results achieved by parSketch-PLS, KNN-PLS and parSketch-PLS. Chapter 4 is dedicated to the analysis of the influence of the parSketch parameters on the results.

# 2. Theory

## 2.1. Notations

Capital bold characters will be used to designate matrices, e.g. **X** ; small bold characters for column vectors, *e.g.* $\mathbf{x}_j$ will denote the $j^{th}$ column of **X**; row vectors will be denoted by the transpose notation, *e.g.* $\mathbf{x}^T_i$ will denote the $i^{th}$ row of **X**; non bold italic characters will be used for scalars, *e.g.* matrix elements $x_{ij}$ or indices *i*.

## 2.2. Method description

The creation of an index with parSketch is done in two steps: dimension reduction and grid creation [37], [38]. Let $\mathbf{X}_{(np)}$ be the matrix of *n* spectra per *p* wavelengths. Let $\mathbf{P}_{(pv)}$ be a matrix of *v* vectors of dimension *p*, containing the values -1 or 1 according to a random selection. The dimension reduction is achieved by calculating the matrix $\mathbf{T}_{(nv)}$, obtained by equation (1), shown in Figure 1. Each line $\mathbf{t}^T_i$ of **T** corresponds to the sketch of $\mathbf{x}_i$.

$$\mathbf{T} = \mathbf{XP} \qquad (1)$$

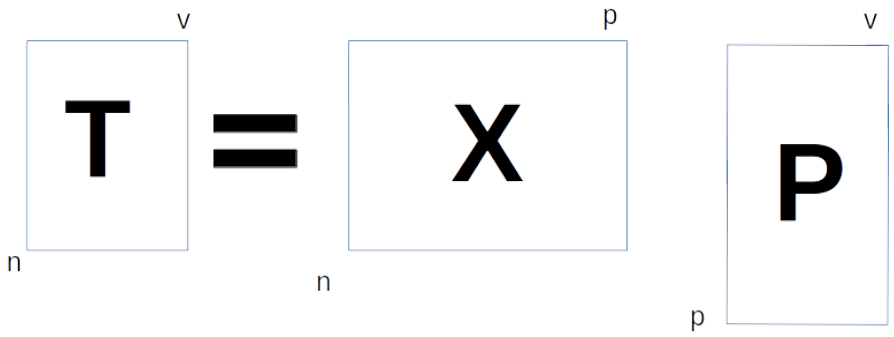



**Figure 1 : Sketch creation**

Adjacent pairs of **T** columns are grouped together to form two-dimensional spaces (in default setting). Such formed, the spaces are segmented to form *g* grids (*g=v/2* where *v* is the number of random vectors generated). The position of all samples in the grid cells is recorded, as illustrated by Figure 2. In this figure, T, the matrix of sketches, contains 6 samples described with four variables. T is then mapped into two dimensional grids divided into 3 segments for both of the variables. Each of the 6 samples are then assigned to a cell within the grids according to their values. For example sample “t(1)” is assigned to cell [10-20][10-20] in grid 1 and to cell [0-10][10-20] in grid 2.

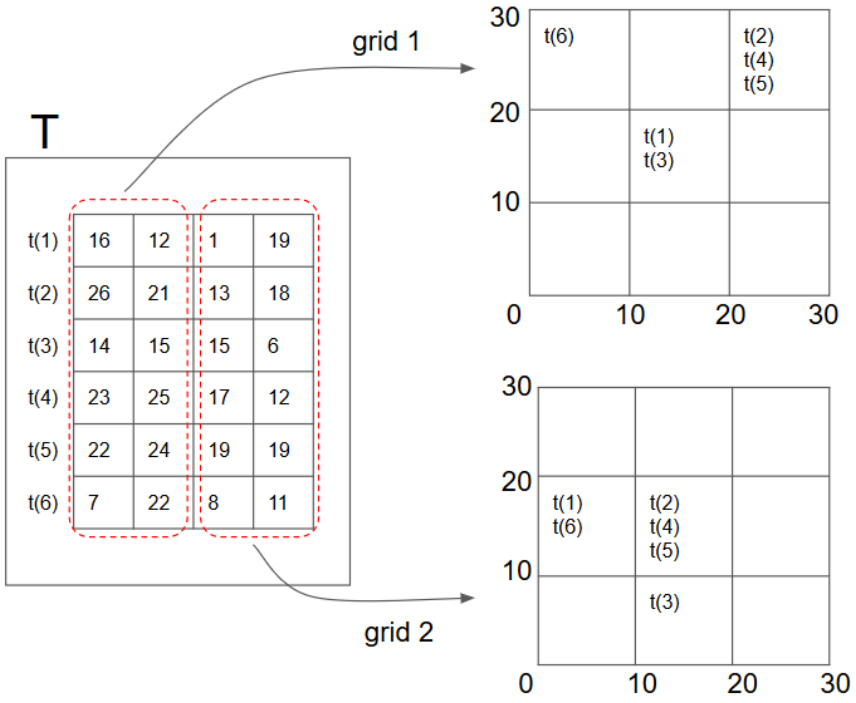



**Figure 2 : Grid creation**

The search for the neighbours of any unknown sample $\mathbf{x}_{new}$ uses the indexes created in the following way:

$\mathbf{x}_{new}$ is converted into a sketch using the loadings **P**: $\mathbf{t}_{new}=\mathbf{x}^{T}_{new}\,\mathbf{P}$. For each grid $u$, let $c_u$ be the cell where $\mathbf{t}_{new}$ is located. The samples present in the $c_u$ cell for at least $m$ % of the grids are selected as neighbours of $\mathbf{x}_{new}$.

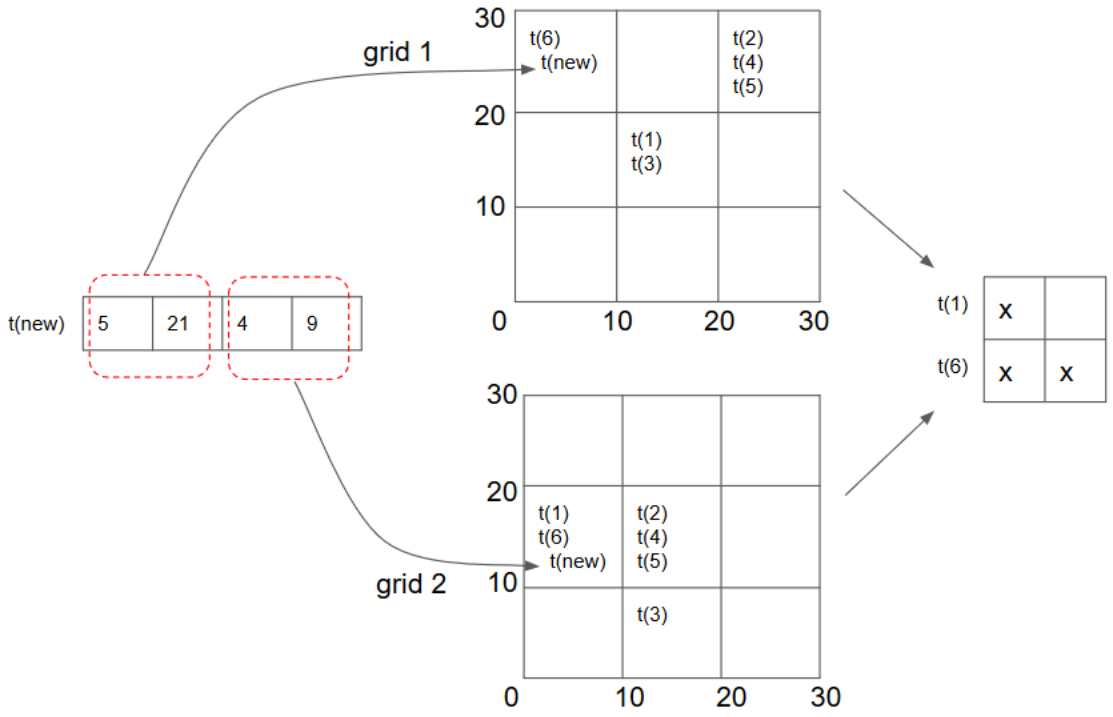



**Figure 3 : neighbour search**

This process is illustrated by Figure 3, where a new sample is searched for best candidate neighbours in **T**. The sample is first converted as a sketch, t(new), and placed in the grids using the same process. Consequently, the neighbours of t(new) are chosen within the samples that occur in the same cells as t(new) with a minimum threshold of *m*. In Figure 3, t(1) and t(new) co-occur one, while t(1) and t(new) co-occur twice. With a minimum threshold corresponding to 2 co-occurrences, t(6) would be considered candidate to be a nearest neighbour to t(new).

## 2.3. Method properties

The parSketch method has interesting properties for processing massive data.

Firstly, the parSketch method has three adjustable parameters to define a trade-off between the computation costs and the accuracy of resulting neighbourhood. The first one,

is the number of random vectors *v* used to generate **P**. This parameter improves the approximation quality of the neighbourhood. The greater the number of random vectors generated (*v*), the better the approximation. The second one, is the number of segments *s* in the grids. This parameter allows to obtain neighbours closer to the sample to be predicted and reduces the number of returned neighbours. The larger the number of segments, the smaller the neighbourhood returned by parSketch. The third parameter, is a threshold regarding the minimum number of grids *m* in percentage. The higher this threshold, the closer the neighbours returned by parSketch are to the sample to be predicted but fewer in number.

Secondly, the dimension reduction is very efficient. In contrast to the usual dimension reduction methods used in chemometrics (*e.g.* principal component analysis or partial least squares) the dimension reduction performed within parSketch is essentially a single matrix product. This reduction is performed through the matrix **P** which is very easy to generate (from random selection). And more importantly, the Johnson-Lindenstrauss lemma [39], guarantees to preserve an approximation of the Euclidean distances between the samples.

Thirdly, the application of grids to **T** is facilitated. Indeed, there are no predominant variables in the sketch matrix **T** because this matrix is obtained using **P**, constructed from random vectors . It is therefore possible to create grids without having to take precautions regarding the space of the **T** variables. If factorial methods such as the PCA were used for this dimension reduction, the grids should take into account the variance expressed by each component.

Fourthly, parSketch uses a large number of low dimensional grids (2 dimensions in this case). This makes possible to discard some constraints inherent to the curse of dimensionality [40]. Indeed, if large grids were exploited, it would generate hollow subspaces *i.e*. the returned neighbourhoods would be too small or even non-existent.

Fifthly, the parSketch method considerably reduces calculation times, thanks to dimension reduction and indexing. In addition, all parSketch steps can be parallelised. This allows parSketch to process large amounts of data. For example, in [37], parSketch made it possible to process databases of $3x10^8$ samples in a short time.

# 3. Material and methods

## 3.1. Data and software

In this article a dataset for classification has been created using hyperspectral images. The initial database contained 360,000 reflectance spectra of wheat leaves. The samples measured belong to four classes, corresponding to four different genotypes. The spectra were acquired using a hyperspectral camera at p=256 wavelengths ranging from 410 to 1000 nm. One hyperspectral image was acquired for each class containing 90,000 spectra. For each image and thus each class, 100 test samples were selected using the Kennard-Stone method [41] applied to the coordinates of the pixels in the image. Because samples are selected from images, spectra resulting from adjacent pixels are very likely to be highly correlated. Consequently, a reasonable manner to construct unbiased validation sets is to select only one sample in every 7*7 neighbourhoods. The resulting database included a test set of 400 and a calibration set of 354,426 samples. Calculations were performed with the R software (version 3.6.1 [42]), and the Rnirs toolbox for PLS. The R package rnirs is available at https://github.com/mlesnoff/rnirs.

## 3.2. Prediction models

The objective of this article was to compare the properties and classification performance of three types of methods: PLSDA, KNN-PLSDA, parSketch-PLSDA. It was first intended to illustrate the contribution of the KNN-PLS method in relation to PLSDA. Then, it was used to compare KNN and the parSketch algorithm in terms of returned neighbourhoods and in terms of cost to performance potentials.

The first model was derived from a PLSDA, the model consisted in transforming a univariate variable y (containing q classes) into an n*q matrix $\mathbf{Y}_{\mathbf{dummy}}$ of q 0/1 dummy variables then to apply a PLS2 model on ($\mathbf{X}$ ,$\mathbf{Y}_{\mathbf{dummy}}$) [20] and then to carry out a linear discriminant analysis (LDA) between the PLS2 scores and **Y**.

In the second model, derived from a KNN-PLSDA [43], the search for neighbours (thanks to brute-force algorithm) is conducted on the first 10 scores of a PLS model [44]. In this experiment, the influence of two parameters is tested: the size of the neighbourhoods and the number of latent variables. Neighbourhoods of 400, 1000, 3000, 5000 and 10000 samples are considered.

For the first two models (PLSDA and KNN-PLSDA), the criterion for assessing the quality of the calculated models was the percentage error of prediction on the test set. A third model was estimated by replacing the brute-force algorithm by parSketch in the KNN-PLSDA method.

The three parameters of parSketch (*v,s,m*), are set with values ranging as: v ∈{10, 20, 30, 50, 80, 100}, s ∈{5, 7, 9, 11, 13, 15}, m ∈{30, 50, 70, 90}.

In the following this model will be referred to as “parSketch-PLSDA”. For the third model, 3 evaluation criteria of the parSketch parameters have been selected. First, the number of predictable samples, *i.e.* the number of test samples having more than 30 neighbours. Then,

the distribution of the number of neighbours is observed. Finally, the prediction error of parSketch-PLSDA is observed through 5 notable combinations of [*v,s,m*] parameters.

# 4. Results and discussion

## 4.1. Data visualisation

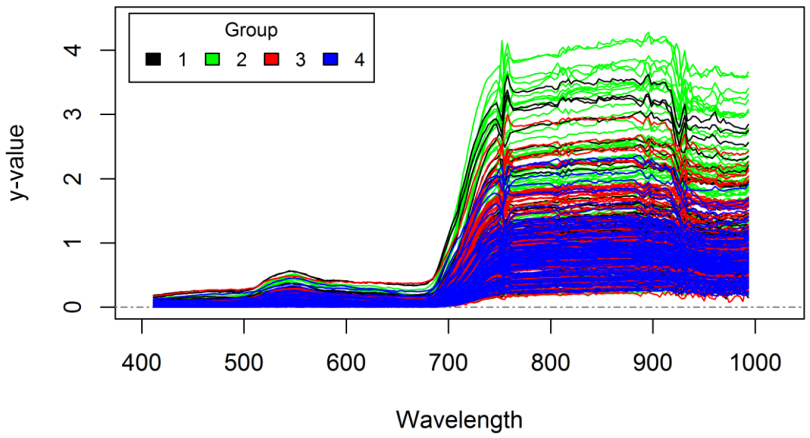



**Figure 4 : spectra plot of the 4 genotypes**

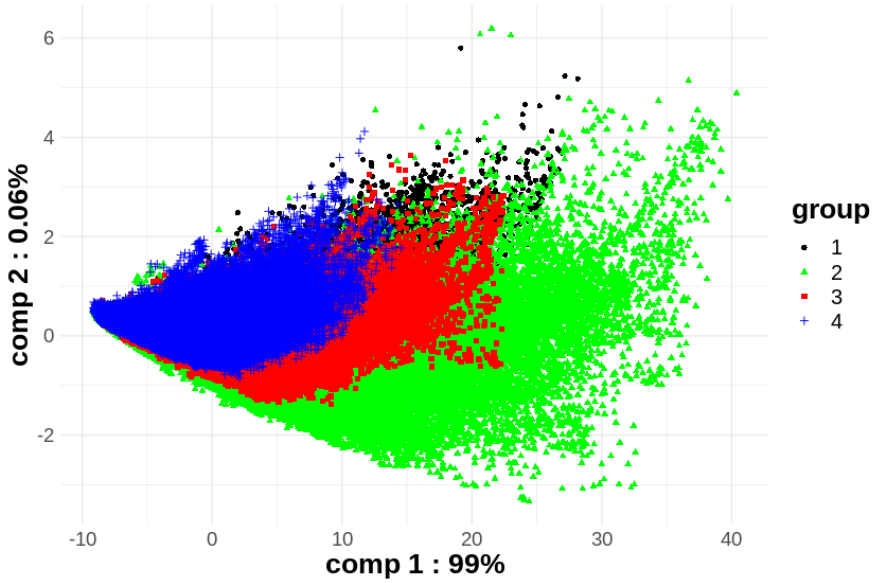



**figure 5 : Plot of the first two PCA scores**

Figure 4 shows sampled spectra of the calibration set covering the 4 genotypes. Spectra from each group present similar general shapes. There is no significant peak in any wavelength that can clearly discriminate spectra from the different genotypes. Moreover, there is no specific spectral domain in which the four genotypes seem to diverge.

Figure 5 presents a projection of the whole spectra on the two first components of a PCA. All genotypes follow a single trajectory with significant overlaps. Similar examinations conducted with up to 20 components lead to the same results. This shows that genotypic differences cannot be explained by the PCA model.

## 4.2. PLSDA

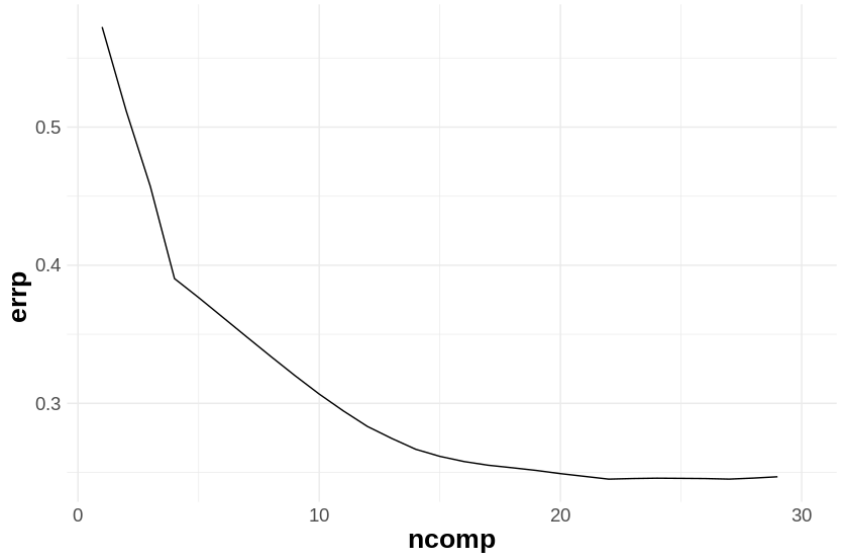



**Figure 6 : Error in classification of the test set according to the number of latent variables PLS**

Figure 6 shows the classification error of the test set performed by the PLSDA as a function of the number of PLS latent variables. It is difficult to observe a minimum error. The optimal number of latent variables is between 20 and 30 components. This is a very high number of components, which can be related to non-linearities in the data. The minimum classification error is close to 0.24(24%). It therefore seems that the “PLS” model does not perform well in this case prediction. The classes are divided by genotypes, which are very close from a physicochemical point of view. Therefore, it is not surprising that a global linear model fails to discriminate them.

## 4.3. KNN-PLSDA

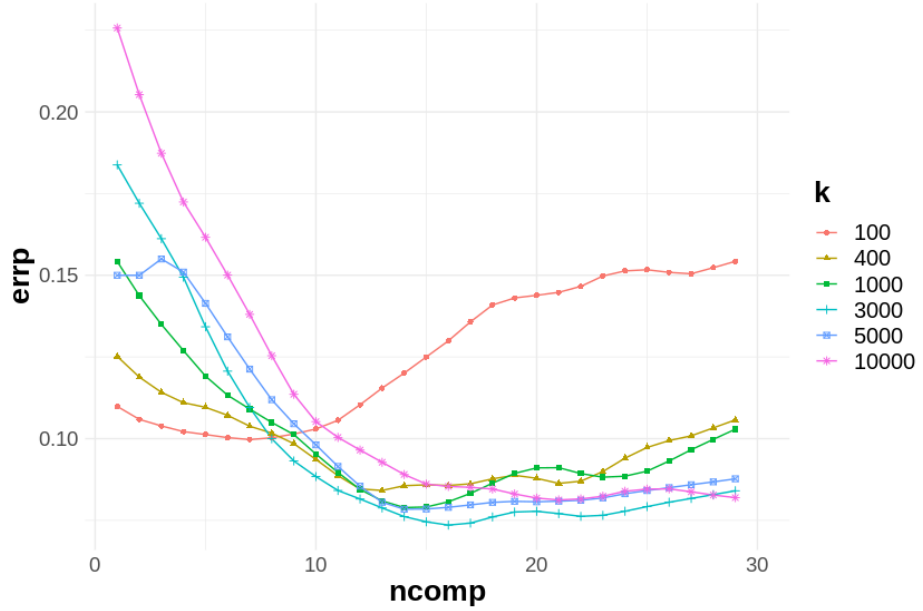



**Figure 7 : Classification error of the test set of a KNN-PLSDA, depending on the number of PLS latent variables and the number of neighbours**

Figure 7 shows the prediction error of a KNN-PLSDA model as a function of the number of PLS latent variables for a given number of neighbours k∈{100,400,1000,3000,5000,10 000}. These results are much better compared to the PLSDA model (see figure 6) with a minimal prediction error divided by 2.5. This confirms the non-linearity within the data.

Table 1 : Optimal classification error according to the number of neighbours

| Number of neighbours | Optimal number of LVs | Misclassification error (%) |
|---|---|---|
| 100 | 7 | 10 |
| 400 | 13 | 8 |
| 1000 | 14 | 8 |
| 3000 | 16 | 7 |
| 5000 | 14 | 8 |
| 10000 | 22 | 8 |

Table 1 summarises the minimum misclassification errors and the associated latent variables depending on the size of the considered neighbourhood (from figure 7). It illustrates that as the number of neighbours increases, errors decrease but with a growing optimal number of latent variables. Table 1 shows that the minimum prediction error is 7% for an optimal number of neighbours of 3000 and a number of latent variables of 16. It can then be concluded that the KNN-PLSDA method for genotype discrimination is more efficient than the PLSDA. Furthermore, it is observed that to achieve minimal prediction errors, it is necessary to create local models with a large number of PLS latent variables, which means that genotype discrimination is difficult to achieve.

## 4.4. parSketch-PLSDA

In this section, two points will be discussed: the neighbourhood of samples and the prediction error of the parSketch-PLSDA method.

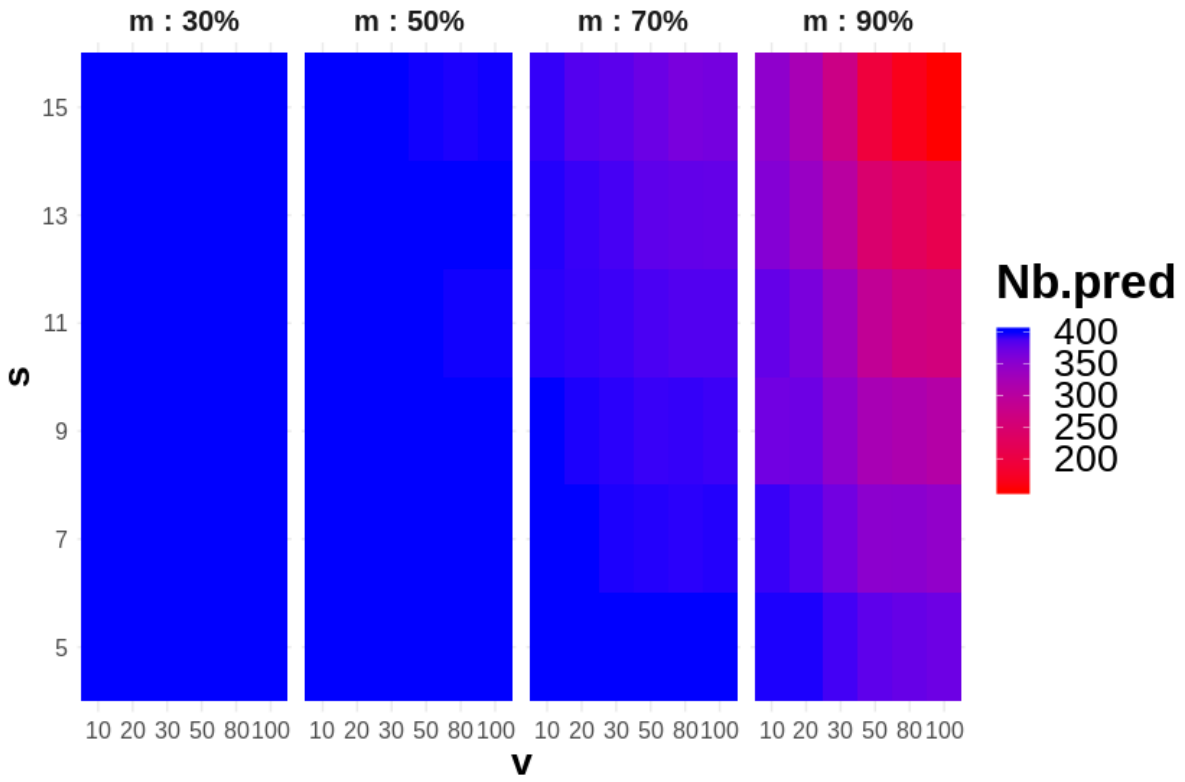



**Figure 8 : heatmap of the number of predictable samples found by parSketch, as a function of the 3 parameters v, s, m (number of random vectors, number of segments, minimum % of grids). Figure 8 is divided into 4 graphs, each graph corresponds to a value of parameter m.**

For each sample in the test set, parSketch returns a neighbourhood whose size depends on the parameters of the algorithm. An sample in the test set is said to be predictable, if the neighbourhood returned by parSketch contains at least 30 samples.

Figure 8 shows that the number of predictable samples decreases as the value of *m* increases. The parameter *m* is used to select the samples most often present in the same

cell as the sample to be predicted (see section 2). The parSketch method constructs grids on sketches that are derived from matrix **P** random vectors. It is therefore very unlikely to obtain a large number of neighbours if the threshold *m* is too high (Cf section 2.2).

Figure 8 shows that the number of predictable samples decreases as the values of s (number of segments) and *v* (number of random vectors) increase. *s* defines the number of cells in each grid (see section 2). When the number of cells in each grid increases, the cells are smaller and therefore contain fewer samples.

The number of random vectors (*v*) is used to preserve the Euclidean distances in an approximate way. This parameter is not directly related to the size of the neighbourhood but *v* is correlated to the threshold *m* and thus will have an indirect impact on the number of predictable samples.

To conclude, *s* and *m* have a strong impact on the number of predictable samples while *v* has a weak (indirect) impact on the number of predictable samples.

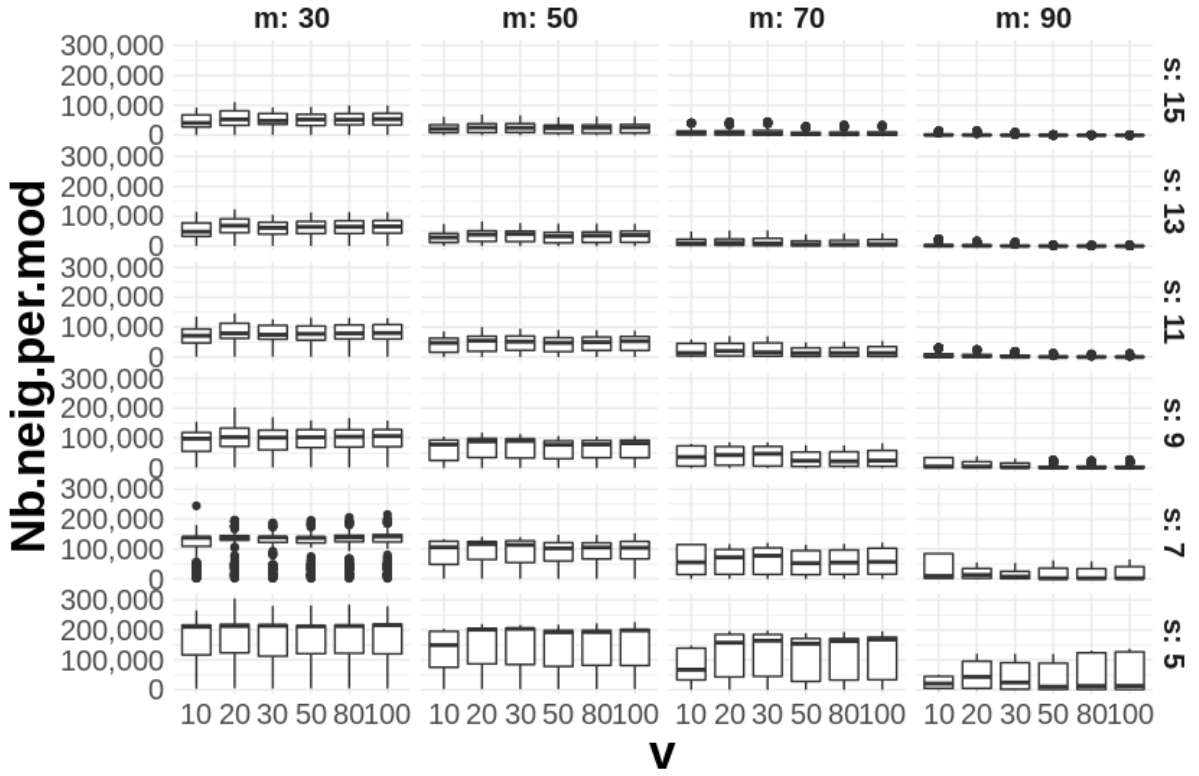


**Figure 9 : Distribution of the number of neighbours per test sample as a function of the parSketch parameters (v, *m*, *s*).**

Figure 9 shows that when the value of parameters *m* and *s* is too high, the number of returned neighbours is low or even zero. Moreover, a stochastic behavior of the neighbourhood distributions can be observed when the value of the threshold *m* is high (e.g. *m* = 90%). It is therefore not possible to make conclusions about the influence of the parSketch parameters when the value of the parameters *m* and *s* are too high. For future observations, the distributions of the neighbourhoods obtained with parameters *m* = 90 or *s* = 15 are not studied.

In Figure 9, it is possible to observe the impact of the three parSketch parameters on the neighbourhood distributions of the samples to be predicted.

Firstly, when v varies, the median number of neighbours and the interquartile range of neighbourhoods are constant. Indeed, the number of random vectors has no impact on the quantity of returned neighbours (see section 2).

Secondly, as *s* increases, the median number of neighbours per sample to be predicted decreases. Indeed, *s* defines the number of cells in each grid, the more *s* will have a high value the more cells there will be and thus the fewer samples in each cell. When *s* increases, the interquartile range of neighbourhoods decreases. Indeed, the stronger the segmentation, the less influence there will be on the structure of the position of the samples to be predicted in the database.

Thirdly, when *m* increases, the median number of neighbours and the interquartile neighbourhood gap per sample decreases. *m* is a threshold, the higher the value of *m* the more similar the samples returned by parSketch will be. Therefore, if the value of *m* is high, fewer neighbours will be returned by parSketch.

To conclude, with the help of figures 8 and 9 it is possible to eliminate certain combinations of parameters. Indeed, figure 8 makes it easy to select combinations that allow us to predict a certain number of samples. Then, figure 9 allows the selection of combinations of parameters according to the characteristics of the neighbourhoods (e.g. a high number of neighbours and a low variability of the neighbourhoods). Five combinations were chosen to calculate a PLSDA model (see Table 2).

| Combinaison | *m* | *v* | *s* |
|---|---|---|---|
| 1 | 50 | 10 | 9 |
| 2 | 50 | 10 | 11 |
| 3 | 50 | 20 | 11 |
| 4 | 50 | 20 | 11 |
| 5 | 50 | 100 | 13 |

Table 2 : Combinations of the selected parSketch parameters

These combinations of parameters were chosen because the resulting neighbourhoods were small and all samples were predictable. However, in these combinations, the median neighbourhood returned by parSketch is much larger (100,000) than the neighbourhood used in the PLSDA room, which was 3,000 neighbours.

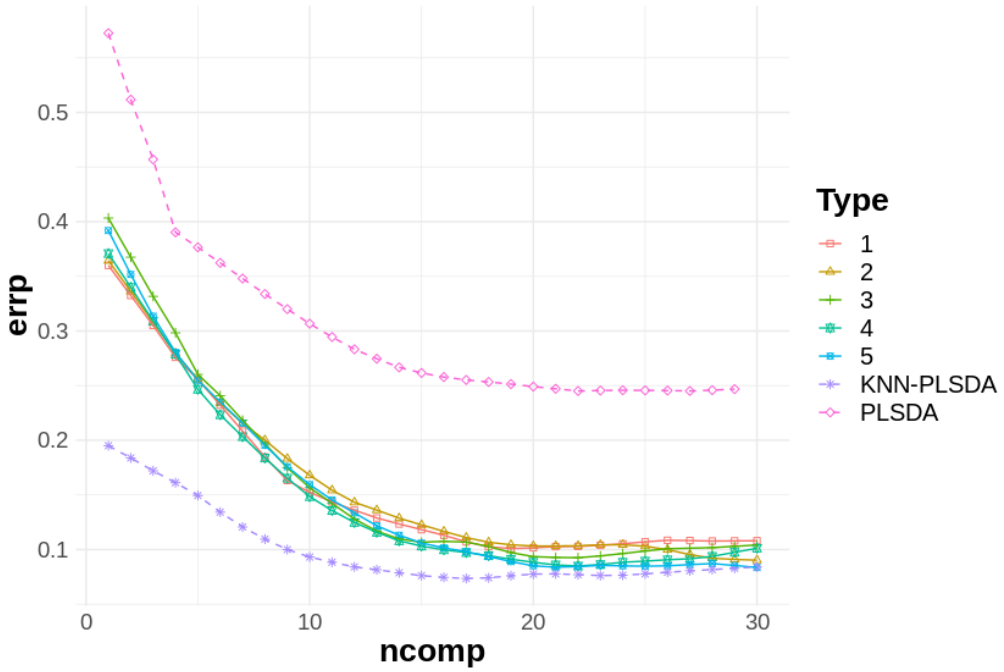



**Figure 10 : Method classification error: parSketch-PLSDA (all 5 combinations); PLSDA and KNN-PLSDA (3000 neighbours), depending on the number of PLS latent variables**

In Figure 10, the error curves of the parSketch-PLSDA are all very close and provide an optimal error of approximately 10% which is much more interesting than the result of the PLSDA. The best combination of parameters approaches the best result of the KNN-PLSDA. To conclude, on the example discussed in the article, the neighbourhood returned by parSketch may provide an alternative. parSketch provides a neighbourhood that allows us to improve the prediction qualities but this neighbourhood can be very large and therefore not all the neighbours returned by pasketch are useful.

# 5. Conclusion / perspective

This article shows that the combination of a big-data indexing algorithm (parSketch) with PLSDA (parsketch-PLSDA) approaches the best results of KNN-PLSDA on the treated example (wheat leaf genotype discrimination by hyperspectral imaging). The parSketch method is not able to obtain as good classification results as the KNN-PLSDA method, but allows better results than a PLSDA. It can be concluded that parSketch will be efficient to handle non-linear relations between X and Y. parSketch is less efficient than the brute-force method to obtain relevant neighbours for PLS model calculation, but allows to realise a fast estimation of the neighbourhood on massive databases. Indeed, the prediction of the 400 test samples took several hours with the brute force method while the prediction using parSketch took only a few minutes. This article also shows that it is possible to use parSketch to study the database. Indeed, the parameters *s* and *m* (number of segments and minimum % of grids) are dependent on the database structure, for example if the database is very compact, the neighbourhoods returned by parSketch will be very large.

A major problem with parSketch is that it may return too large neighbourhoods. If the neighbourhoods returned by parSketch are too large, it is possible to fail to handle non-linearity. Indeed, if too many neighbours are returned it is possible that these samples do not all belong to the same linear model. Moreover, the computation time of a PLS model is related to the number of samples. A solution to obtain smaller neighbourhoods and better predictions is to combine parSketch with a method for selecting samples. For example, parSketch could be combined with the brute-force method to reduce the number of distances to be computed for each sample to predict. This means that within a neighbourhood returned by parSketch it would be possible to apply the brute force method. It would therefore be interesting to study parSketch as a filter and then to combine parSketch with another approach of selection or weighting of samples.

# 6. Acknowledgements

This work was supported by the French National Research Agency under the Investments for the Future Program, referred as ANR-16-CONV-0004.

The authors want to thank Oleksandra Levchenko for the productive discussions and relevant insights regarding the parSketch approach and its application to various domains, including chemometrics.

# 7. References

[1] S. Wold, M. Sjöström, et L. Eriksson, « PLS-regression: a basic tool of chemometrics », *Chemom. Intell. Lab. Syst.*, vol. 58, n° 2, p. 109-130, oct. 2001, doi: 10.1016/S0169-7439(01)00155-1.

[2] P. Dardenne, G. Sinnaeve, et V. Baeten, « Multivariate Calibration and Chemometrics for near Infrared Spectroscopy: Which Method? », *J. Infrared Spectrosc.*, vol. 8, n° 4, p. 229-237, oct. 2000, doi: 10.1255/jnirs.283.

[3] F. Davrieux *et al.*, « LOCAL Regression Algorithm Improves near Infrared Spectroscopy Predictions When the Target Constituent Evolves in Breeding Populations »:, *J. Infrared Spectrosc.*, janv. 2016, doi: 10.1255/jnirs.1213.

[4] M. Clairotte *et al.*, « National calibration of soil organic carbon concentration using diffuse infrared reflectance spectroscopy », *Geoderma*, vol. 276, p. 41-52, août 2016,

doi: 10.1016/j.geoderma.2016.04.021.

[5] D. Pérez-Marín, A. Garrido-Varo, et J. E. Guerrero, « Non-linear regression methods in NIRS quantitative analysis », *Talanta*, vol. 72, nº 1, p. 28‑42, avr. 2007, doi: 10.1016/j.talanta.2006.10.036.

[6] M. Bevilacqua et F. Marini, « Local classification: Locally weighted–partial least squares-discriminant analysis (LW–PLS-DA) », *Anal. Chim. Acta*, vol. 838, p. 20‑30, août 2014, doi: 10.1016/j.aca.2014.05.057.

[7] A. M. C. Davies, H. V. Britcher, J. G. Franklin, S. M. Ring, A. Grant, et W. F. McClure, « The application of fourier-transformed near-infrared spectra to quantitative analysis by comparison of similarity indices (CARNAC) », *Mikrochim. Acta*, vol. 94, nº 1‑6, p. 61‑64, janv. 1988, doi: 10.1007/BF01205839.

[8] K. Hazama et M. Kano, « Covariance-based locally weighted partial least squares for high-performance adaptive modeling », *Chemom. Intell. Lab. Syst.*, vol. 146, p. 55‑62, août 2015, doi: 10.1016/j.chemolab.2015.05.007.

[9] B. Igne, J. B. Reeves, G. McCarty, W. D. Hively, E. Lund, et C. R. Hurburgh, « Evaluation of Spectral Pretreatments, Partial Least Squares, Least Squares Support Vector Machines and Locally Weighted Regression for Quantitative Spectroscopic Analysis of Soils », *J. Infrared Spectrosc.*, vol. 18, nº 3, p. 167‑176, juin 2010, doi: 10.1255/jnirs.883.

[10] Tormod. Naes, Tomas. Isaksson, et Bruce. Kowalski, « Locally weighted regression and scatter correction for near-infrared reflectance data », *Anal. Chem.*, vol. 62, nº 7, p. 664‑673, avr. 1990, doi: 10.1021/ac00206a003.

[11] D. Andueza, F. Picard, D. Dozias, et J. Aufrère, « Fecal Near-Infrared Reflectance Spectroscopy Prediction of the Feed Value of Temperate Forages for Ruminants and Some Parameters of the Chemical Composition of Feces: Efficiency of Four Calibration Strategies »:, *Appl. Spectrosc.*, juin 2017, doi: 10.1177/0003702817712740.

[12] C. Ariza-Nieto, O. Mayorga, B. Mojica, D. Parra, et G. Afanador-Tellez, « Use of LOCAL algorithm with near infrared spectroscopy in forage resources for grazing systems in Colombia », *J. Infrared Spectrosc.*, vol. 26, nº 1, p. 44‑52, févr. 2018, doi: 10.1177/0967033517746900.

[13] P. Berzaghi, J. S. Shenk, et M. O. Westerhaus, « LOCAL Prediction with near Infrared Multi-Product Databases », *J. Infrared Spectrosc.*, vol. 8, nº 1, p. 1‑9, janv. 2000, doi: 10.1255/jnirs.258.

[14] I. I. F. E. Barton, J. S. Shenk, M. O. Westerhaus, et D. B. Funk, « The Development of near Infrared Wheat Quality Models by Locally Weighted Regressions »:, *J. Infrared Spectrosc.*, févr. 2017, doi: 10.1255/jnirs.280.

[15] J. A. Fernández Pierna et P. Dardenne, « Soil parameter quantification by NIRS as a Chemometric challenge at ‘Chimiométrie 2006’ », *Chemom. Intell. Lab. Syst.*, vol. 91, nº 1, p. 94‑98, mars 2008, doi: 10.1016/j.chemolab.2007.06.007.

[16] E. Fernández-Ahumada *et al.*, « Reducing NIR prediction errors with nonlinear methods and large populations of intact compound feedstuffs », *Meas. Sci. Technol.*, vol. 19, nº 8, p. 085601.

[17] E. Fernández-Ahumada, T. Fearn, A. Gómez-Cabrera, J. E. Guerrero-Ginel, D. C. Pérez-Marín, et A. Garrido-Varo, « Evaluation of Local Approaches to Obtain Accurate Near-Infrared (NIR) Equations for Prediction of Ingredient Composition of Compound Feeds », *Appl. Spectrosc.*, vol. 67, nº 8, p. 924‑929, août 2013, doi: 10.1366/12-06937.

[18] J. S. Shenk, M. O. Westerhaus, et P. Berzaghi, « Investigation of a LOCAL Calibration Procedure for near Infrared Instruments », *J. Infrared Spectrosc.*, vol. 5, nº 4, p. 223‑232, oct. 1997, doi: 10.1255/jnirs.115.

[19] G. Sinnaeve, P. Dardenne, et R. Agneessens, « Global or Local? A Choice for NIR Calibrations in Analyses of Forage Quality », *J. Infrared Spectrosc.*, vol. 2, nº 3, p. 163‑175, juin 1994, doi: 10.1255/jnirs.43.

[20] M. Barker et W. Rayens, « Partial least squares for discrimination », *J. Chemom.*, vol. 17, nº 3, p. 166-173, 2003, doi: 10.1002/cem.785.

[21] T. Fearn et A. M. C. Davies, « Locally-Biased Regression », *J. Infrared Spectrosc.*, vol. 11, nº 6, p. 467-478, déc. 2003, doi: 10.1255/jnirs.397.

[22] F. Gogé, R. Joffre, C. Jolivet, I. Ross, et L. Ranjard, « Optimization criteria in sample selection step of local regression for quantitative analysis of large soil NIRS database », *Chemom. Intell. Lab. Syst.*, vol. 110, nº 1, p. 168-176, janv. 2012, doi: 10.1016/j.chemolab.2011.11.003.

[23] R. Bayer, « Binary B-trees for Virtual Memory », in *Proceedings of the 1971 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control*, New York, NY, USA, 1971, p. 219–235, doi: 10.1145/1734714.1734731.

[24] J. L. Bentley, « Multidimensional binary search trees used for associative searching », *Commun. ACM*, vol. 18, nº 9, p. 509-517, sept. 1975, doi: 10.1145/361002.361007.

[25] R. A. Finkel et J. L. Bentley, « Quad trees a data structure for retrieval on composite keys », *Acta Inform.*, vol. 4, nº 1, p. 1-9, mars 1974, doi: 10.1007/BF00288933.

[26] A. Guttman, « R-trees: A Dynamic Index Structure for Spatial Searching », in *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 1984, p. 47–57, doi: 10.1145/602259.602266.

[27] N. Roussopoulos, S. Kelley, et F. Vincent, « Nearest Neighbor Queries », in *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 1995, p. 71–79, doi: 10.1145/223784.223794.

[28] I. Assent, R. Krieger, F. Afschari, et T. Seidl, « The TS-Tree: Efficient Time Series Search and Retrieval », p. 12.

[29] Y. Cai et R. Ng, « Indexing spatio-temporal trajectories with Chebyshev polynomials », in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data - SIGMOD '04*, Paris, France, 2004, p. 599, doi: 10.1145/1007568.1007636.

[30] A. Camerra, J. Shieh, T. Palpanas, T. Rakthanmanon, et E. Keogh, « Beyond one billion time series: indexing and mining very large time series collections with iSAX2+ », *Knowl. Inf. Syst.*, vol. 39, nº 1, p. 123-151, avr. 2014, doi: 10.1007/s10115-012-0606-6.

[31] A. Camerra, T. Palpanas, J. Shieh, et E. Keogh, « iSAX 2.0: Indexing and Mining One Billion Time Series », in *2010 IEEE International Conference on Data Mining*, Sydney, Australia, 2010, p. 58-67, doi: 10.1109/ICDM.2010.124.

[32] C. Faloutsos, M. Ranganathan, et Y. Manolopoulos, « Fast Subsequence Matching in Time-Series Databases », p. 11.

[33] T. Rakthanmanon *et al.*, « Data Mining a Trillion Time Series Subsequences Under Dynamic Time Warping », p. 5.

[34] J. Shieh et E. Keogh, « iSAX: disk-aware mining and indexing of massive time series datasets », *Data Min. Knowl. Discov.*, vol. 19, nº 1, p. 24-57, août 2009, doi: 10.1007/s10618-009-0125-6.

[35] Y. Wang, P. Wang, J. Pei, W. Wang, et S. Huang, « A data-adaptive and dynamic segmentation index for whole matching on time series », *Proc. VLDB Endow.*, vol. 6, nº 10, p. 793-804, août 2013, doi: 10.14778/2536206.2536208.

[36] D. E. Yagoubi, R. Akbarinia, F. Masseglia, et T. Palpanas, « DPiSAX: Massively Distributed Partitioned iSAX », in *2017 IEEE International Conference on Data Mining (ICDM)*, New Orleans, LA, 2017, p. 1135-1140, doi: 10.1109/ICDM.2017.151.

[37] O. Levchenko, D.-E. Yagoubi, R. Akbarinia, F. Masseglia, B. Kolev, et D. Shasha, « Spark-parSketch: A Massively Distributed Indexing of Time Series Datasets », in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management - CIKM '18*, Torino, Italy, 2018, p. 1951-1954, doi: 10.1145/3269206.3269226.

[38] D. E. Yagoubi, R. Akbarinia, F. Masseglia, et D. Shasha, « RadiusSketch: Massively Distributed Indexing of Time Series », in *2017 IEEE International Conference on Data*

*Science and Advanced Analytics (DSAA)*, Tokyo, Japan, 2017, p. 262‑271, doi: 10.1109/DSAA.2017.49.

[39] W. B. Johnson, J. Lindenstrauss, et G. Schechtman, « Extensions of lipschitz maps into Banach spaces », *Isr. J. Math.*, vol. 54, nº 2, p. 129‑138, juin 1986, doi: 10.1007/BF02764938.

[40] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.

[41] R. W. Kennard et L. A. Stone, « Computer Aided Design of Experiments », *Technometrics*, vol. 11, nº 1, p. 137‑148, févr. 1969, doi: 10.1080/00401706.1969.10490666.

[42] « R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/. »

[43] M. Lesnoff, M. Metz, et J.-M. Roger, « Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data », *J. Chemom.*, vol. n/a, nº n/a, p. e3209, doi: 10.1002/cem.3209.

[44] G. Shen *et al.*, « Local partial least squares based on global PLS scores », *J. Chemom.*, vol. 33, nº 5, p. e3117, 2019, doi: 10.1002/cem.3117.

# Chapitre 6

## Article 4 : A novel robust pls regression method inspired from boosting principles : Roboost-PLSR

Référence :

Maxime Metz, Florent Abdelghafour, Jean-Michel Roger, and Matthieu Lesnoff. A novel robust pls regression method inspired from boosting principles : Roboost-PLSR. Analytica Chimica Acta, 1179 :338823, 2021. ISSN 0003-2670. doi:https://doi.org/10.1016/j.aca.2021.338823. URL https://www.sciencedirect.com/science/article/pii/S0003267021006498

# Highlights

**A novel robust PLS regression method inspired from boosting principles : RoBoost-PLSR**

Maxime Metz, Florent Abdelghafour, Jean-Michel Roger, Matthieu Lesnoff

- A novel robust PLSR method inspired from boosting principle is proposed
- The novel method is based on sample downweighting
- The novel method is evaluated thanks to simulated and real dataset
- This new method has achieved performance equivalent to the calibrated PLS without outliers

# A novel robust PLS regression method inspired from boosting principles : RoBoost-PLSR

Maxime Metz[a,b], Florent Abdelghafour[a,b], Jean-Michel Roger[a,b], Matthieu Lesnoff[b,c,d]

[a] *ITAP, Univ Montpellier, INRAE, Institut Agro, Montpellier, France*
[b] *ChemHouse Research Group, Montpellier, France*
[c] *CIRAD, UMR SELMET, Montpellier, France*
[d] *SELMET, Univ Montpellier, CIRAD,INRA, Institut Agro, Montpellier,France*

**Abstract**

The calibration of Partial Least Square regression (PLSR) models can be disturbed by outlying samples in the data. In these cases the models can be unstable and their predictive potential can be depreciated. To address this problem, some robust versions of the PLSR algorithm were proposed. These algorithms rely on the downweighting of these outliers during calibration. To this end, it is necessary to estimate an inconsistency measurement between the samples and the model. However, this estimation is not trivial in high dimensions. This paper proposes a novel robust PLSR algorithm inspired from the principles of boosting : RoBoost-PLSR. This method consists of realising a series of one latent variable weighted PLSR. RoBoost-PLSR is compared with the PLSR algorithm calibrated with and without outliers and also with Partial Robust M-regression (PRM), a reference robust method. This evaluation is conducted on the basis of three simulated datasets and a real dataset. Finally Roboost-PLSR proves to be resilient to the tested outliers, and can achieve the performances of the reference PLSR calibrated without any outlier.

*Keywords:* Partial least squares, Outliers, Robustness, Boosting ;

*Email address:* maxime.metz@inrae.fr (Maxime Metz)

## 1. Introduction

Partial Least Square Regression (PLSR) [1] is a usual data analysis method and a well-established tool in analytical chemistry. PLSR is particularly relevant for the processing of high dimensional data, especially when the number of explanatory variables exceeds the number of samples. The successful processing of these data is partly conditioned by the fact that the samples can be assimilated to a well-defined distribution. However, if some samples do not share the properties of this distribution, the PLSR model can be disturbed and its predictive quality depreciated [2]. These samples are designated as outliers in comparison with the other ones called inliers. In order to deal with the presence of outliers, numerous strategies have been developed in chemometrics [3–15]. This type of methods are called robust methods. Robust methods place confidence in the main mass the of data. These methods must be parsimonious so as not to exclude major samples who contribute strongly to the good predictive quality of the model. According to [16], "*For high-dimensional data this would result in a severe loss of information as long as the outliers still contain some valuable information, and thus intelligent robust methods adapt the weights according to the outlyingness or inconsistency of the observations.*". In fact, a major difficulty is therefore to determine relevant outlying measurements in order to give low importance to outliers (*e.g.* through weighting), while retaining some of their relevant properties.

In this article, the attention is focused on methods intended for the calibration of PLS1 models in presence of potential outliers. This means that the methods weight the samples through the PLSR in order to reduce the impact of outliers on model calibration. In that sense, only a few robust methods were proposed along with an available algorithm.

One of the first methods was proposed in [10]. This method carries out a robust least square regression for each explanatory variable. This means that the method considers independent variables with this procedure. This aspect was particularly argued in [17] because this process does not capture the multidimensional aspect of outliers.

To address to this problem, [18], developed the Partial Robust M-regression (PRM) method. PRM is frequently studied and used in chemometrics. PRM is based on the NIPALS algorithm trained on the iteratively reweighted matrices (representing the explanatory variables and responses). PRM consists of weighting the samples on the basis of a PLSR model with a

predefined number of latent variables (LVs). This means that the weights are defined for a specific model (*i.e.* PLSR with $K$ latent variables). To determine the $k < K$ models, weights must be specifically recomputed for each given $k$, as opposed to PLSR where each 1 to $K$ LVs model cand be deduced at once from a $K$ model. In PRM, an outlier is defined by a combination of the leverage estimation (*i.e.* the Euclidean distance between scores and the median of scores) and Y-residuals. A limitation of this method, is that outliers are detected using a PLSR model with a number of latent variables that is defined beforehand. In [10], this limitation is lifted by weighting the samples independently of the number of latent variables.
Considering these perspectives, authors propose a new robust PLSR algorithm that combines principles of gradient boosting within a modified framework derived from [10] : RoBoost-PLSR. Boosting is a statistical and machine learning principle consisting in assembling a series of weak models (*i.e.* partially explanatory models) that are adjusted between them. Finally, the prediction by the strong model is the sum of the predictions of each weak model.

The link between PLS and gradient boosting has already been studied and resulted in implementations for the processing of chemical data [19–23] Essentially, these approaches use numerous weak learners, computed sequentially from different sub-samples. Each new weak learner is computed from the previous ones using a loss function. Finally, the weak learners are all combined in a weight function according to their predictive potential. As for the RoBoost-PLSR framework, it proposes to apply the basic idea of gradient boosting : *i.e.* combining an ensemble of weak learners. The weak learners are defined here as weighted one-latent variable PLSR models. The weights are defined iteratively in order to reduce the contribution of outliers on the calculated model. The weak learners are then combined using an unweighted sum of the predictions of each weak learner.

This strategy enables the weighting of samples in the calibration set independently of the number of latent variables (LVs) while considering the multivariate nature of the samples.
The objective of this paper is to provide a study of the proposed new RoBoost-PLSR method using simulated and real data. These data represent different types of outliers that could be present in spectral databases. The first section presents the theoretical principles of RoBoost-PLSR and the associated algorithm. The following section presents the data and the methods used to evaluate and compare RoBoost-PLSR with standard PLSR

and PRM. Finally, the last section presents applications for the calibration and prediction performances of RoBoost-PLSR on the basis of simulated and real data.

## 2. Theoretical background of the RoBoost-PLSR method

### *2.1. Notations*

Capital bold characters will be used for matrices, *e.g.* $\mathbf{X}$; small bold characters for column vectors, *e.g.* $\mathbf{x}_j$ will denote the $j^{th}$ column of $\mathbf{X}$; row vectors will be denoted by the transpose notation, *e.g.* $\mathbf{x}_i^{\mathrm{T}}$ will denote the $i^{th}$ row of $\mathbf{X}$; italicised characters will be used for scalars, *e.g.* matrix elements $x_{ij}$ or indices $i$. Constant scalars will be denoted with italicised characters, *e.g.* number of samples $n$. $\mathbb{1}$ will represent a column vector of ones, of proper dimension.

### *2.2. Principle of the method*

RoBoost-PLSR consists in achieving a series of $K$ unidimensional (1 LV) iteratively reweighted PLSR [24] models. The weigthed PLSR algorithm used is weighted-NIPALS [25] (steps 6-8,12). Each $K+1$ model is calibrated with the residuals ($\mathbf{X}$ and $\mathbf{Y}$) of the previous $K$ models. Sample weights are defined thanks to a Bisquare function [26]. This weight function requires the optimisation of a hyperpameter. This optimisation can be done through a cross-validation procedure or an optimisation on an external validation set. The more the samples deviate from the model, the closer the weights must be to 0. Iteratively, models are updated according to the weights previously attributed until convergence to a stable solution.
Within each PLSR model. Weights are computed according to a combination of three measurements :

- $X$-residuals
- $Y$-residuals
- Leverage

### *2.3. Algorithm*

Let $\mathbf{X}$ be an $[n \times m]$ matrix containing $n$ samples described by $m$ variables. Let $\mathbf{y}$ be a response vector containing $n$ samples. In this article $\mathbf{y}$ is always considered as a vector, *i.e.* the response is univariate.

For a definite number of $K$ latent variables, the algorithm proceeds as described below :

**Algorithm** RoBoost-PLSR for K LV

**Calibration**$(\mathbf{X}, \mathbf{y}, K)$

1: Set $k = 1$
2: Set $\mathbf{X}_0 = \mathbf{X}$
3: Initialise the $[n \times n]$ weight matrix $\mathbf{D}$ :
$\mathbf{D} = diag(d_1, d_2, ..., d_n)$ such as $\forall i \in [1, n], d_i = \frac{1}{n}$
4: Derive the weighted means :
$\bar{\mathbf{x}}_k^{\mathrm{T}} = \mathbb{1}^{\mathrm{T}}\mathbf{D}\mathbf{X}_{k-1}$
$\bar{y}_k = \mathbb{1}^{\mathrm{T}}\mathbf{D}\mathbf{y}_{k-1}$
5: Center the data :
$\mathbf{X}_k = \mathbf{X}_{k-1} - \mathbb{1}\bar{\mathbf{x}}_k^{\mathrm{T}}$
$\mathbf{y}_k = \mathbf{y}_{k-1} - \mathbb{1}\bar{y}_k$
6: Derive the $k^{th}$ weighted loading's weights :

$$\mathbf{w}_k = \frac{\mathbf{X}_k^{\mathrm{T}}\mathbf{D}\mathbf{y}_k}{||\mathbf{X}_k^{\mathrm{T}}\mathbf{D}\mathbf{y}_k||}$$

7: Derive the $k^{th}$ scores :
$\mathbf{t}_k = \mathbf{X}_k\mathbf{w}_k$
8: Derive the $k^{th}$ weighted *loading vectors* of $\mathbf{X}_k$ and the $k^{th}$ regression coefficient vector :

$$\mathbf{p}_k = \frac{\mathbf{X}_k^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}{\mathbf{t}_k^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}$$

$$q_k = \frac{\mathbf{y}_k^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}{\mathbf{t}_\mathbf{k}^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}$$

9: Derive the Y-residuals ($\mathbf{f}$), X-residuals ($\mathbf{E}$), leverage estimation ($\mathbf{l}$) corresponding to the current $k^{th}$ latent variable :
$\mathbf{E} = \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^{\mathrm{T}}$
$\mathbf{f} = \mathbf{y}_k - \mathbf{t}_k q_k$
$\mathbf{l} = \mathbf{t}_k$

10: Estimate and update the weights for each $i \in [1, n]$ sample
$\alpha_i = B(\frac{||\mathbf{e}_i||}{c_\alpha \times s_\alpha})$
$\beta_i = B(\frac{f_i}{c_\beta \times s_\beta})$
$\gamma_i = B(\frac{l_i}{c_\gamma \times s_\gamma})$
$d_i = \frac{1}{n} \times \alpha_i \times \beta_i \times \gamma_i$
With $s_\alpha$ , $s_\beta$ , $s_\gamma$ being respectively the median of $\{||\mathbf{e}_i||\}_n$, $\{|\mathbf{f}_i|\}_n$ and $\{|l_i|\}_n$ $\forall i \in [1, n]$. $c$ respectively denotes fixed constants in each weight function. In this case the weight function $B$ is the Bisquare function defined as :
$B(x) = (1 - x^2)^2$, for $|x| < 1$, B(x) = 0, for |x| > 1

11: Go back to step (4) until convergence of successive $q$'s.

12: while $k < K$
$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^{\mathrm{T}}$
$\mathbf{y}_{k+1} = \mathbf{y}_k - \mathbf{t}_k q_k$
set $k = k + 1 \rightarrow$ then go to step (3)

**End Calibration**

---

**Prediction**($\mathbf{x}^*$, fitted model)

Fitted model { $[q_1, q_2, ..., q_K], [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K], [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_K], [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, ..., \bar{\mathbf{x}}_K]$}

The estimation of $\hat{y}^*$ for a given new sample $\mathbf{x}^*$ is :
$\hat{y}^* = \sum_{k=1}^{K} \hat{y}_k^*$

The computation of $\hat{y}_k^*$ is given by :
$\hat{y}_k^* = \mathbf{t}_k q_k$ with,
$\mathbf{t}_k = \mathbf{x}_k^* \mathbf{w}_k$ and,
$\mathbf{x}_k^* = (\mathbf{x}_{k-1}^* - \bar{\mathbf{x}}_k^{\mathrm{T}}) - (\mathbf{x}_{k-1}^* - \bar{\mathbf{x}}_k^{\mathrm{T}}) \mathbf{w}_k \mathbf{p}_k^{\mathrm{T}}$

---

*2.4. Method properties*

The RoBoost-PLSR framework is designed foremost to facilitate the estimation of the samples weights, *i.e.* estimating the deviation from a model in large dimensions (a large number of latent variables).

Firstly, estimating the weights of samples independently for each latent variable provides a simpler estimation of leverage points. Indeed, in usual robust PLSR algorithms, leverage is computed either thanks to Euclidean or Mahalanobis distances between the scores and the centre of the model. In high dimensional spaces (numerous LVs), this estimation is not so trivial. As a matter of fact, in the case of a Euclidean distance, the latest LVs have only a minor contribution to the leverage value. This is naturally due to the decreasing magnitude of scores. Nevertheless, the predictive potential of these latest LVs is not necessarily lesser. In the case of a Mahalanobis distance, the contributions of all LVs become equal in the computation of the leverage value. This can be equally detrimental, since the predictive potentials of the LVs are most oftenly uneven.

Secondly, the proposed method considers X-residuals, which is not the case in usual robust PLSR methods. The inclusion of these residuals provides additional information that cannot be expressed solely by leverage and Y-residuals.

Thirdly, the method does not provide regression coefficients. Contrary to other robust methods such as PRM, in this case, it is not trivial to compute them. Indeed, the proposed algorithm for RoBoost-PLSR does not allow an estimation of the rotation matrix $\mathbf{R}$. Models can nevertheless be interpreted by analysing the loadings, although it is less convenient. Indeed, it is possible to observe the loadings and derive the most influential variables within each 1LV model. However, unlike in conventional PLSR, it is not possible yet to determine the relative influences of variables at the scale of the whole K-LV model

Fourthly, like PLSR, RoBoost-PLSR makes it possible to deduce any of the 1 to $K$ LVs models from the calibration of a single $K$ LVs model. This preserves the operability during the validation and parameterisation process of the RoBoost-PLSR method.

## 3. Material and methods

### *3.1. Data and software*

RoBoost-PLSR was evaluated on three simulated datasets and one real dataset. Simulations were used to introduce controlled disturbances while the real dataset was used to confirm and support the simulations results. The algorithms were developed using the R software packages. RoBoost-PLSR was developed on the basis of "rnirs" functions. The functions and data associated with RoBoost-PLSR are available on Github "RoBoost-PLSR". PRM was implement with the "prms" function available in the "sprm" package.

### *3.2. Simulated Data*

The three simulations were generated according to the generic framework proposed by [27]. Contrary to the simulation strategies usually used to evaluate robust methods, the data were not simulated from a real model. The data are simulated from a combination of spectral signatures, some of which are related to one or more variables to be predicted ($\mathbf{Y}$ matrix).

The simulations were based on a combination of pure artificial spectra and controlled noises. The aim of each simulation was to reproduce the common external disturbances that can occur when calibrating a predictive model. It consisted of adding to the dataset an additional set of predefined outliers that have a negative effect on the performance of the models. The first simulation introduced pure $Y$ outliers. The second simulation introduced contaminant induced outlier *i.e.* $X$-outliers occurring when an external substance pollutes the calibration samples. These individuals are strong outliers because they can be easily distinguished from inliers (*e.g.* by a spectra plot). The third simulation introduced slight $X$-outliers. For all simulations, 900 inliers and 100 outliers were simulated. Descriptions of the simulation are available in the appendix in table form. The differences between simulated inliers and outliers are highlighted in bold in the tables.

### *3.3. Real dataset*

The real dataset consisted of NIR spectral samples acquired from two types of feed materials : soybean and meat and bone meal. Each sample-spectrum was associated with its Y-response *i.e.* the chemical reference measurement of its protein content. The spectra were measured with a *Foss* spectrometer in the spectral range $[1100 - 2498$ nm$]$ with a 2 nm

spectral resolution. These data were extracted from the "PROT" database provided by the CRA-W (Agronomic Research Centre of Wallonia, Belgium). This database was already used for the development and comparison of local methods [28].

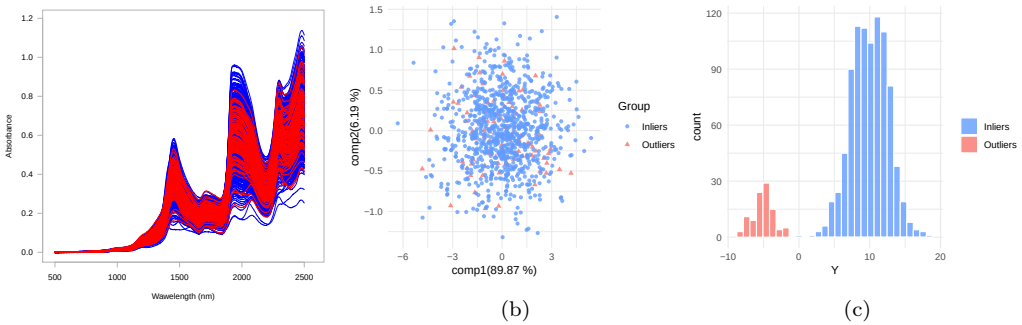
### *3.4. Evaluation strategies*

The purpose is to evaluate the behaviour of the newly introduced RoBoost-PLSR methods in presence of outliers during calibration. The calibrated model is then evaluated on a validation set. The reference against which all models were compared was a PLSR calibrated on a dataset without outliers (and will be designated as such). Roboost-PLSR was evaluated and compared with two standard regression algorithms : PLSR and PRM.
In the case of the simulations, the weight parameters of PRM and RoBoost-PLSR were optimised according to the validation set. Only the results of the optimal (*i.e.* the parameters that provide the minimum value of RMSEP) parameters of RoBoost-PLSR and PRM were presented in the following section. The calibration sets were generated from 500 samples (400 inliers and 100 outliers). The resulting models were studied with validation sets containing 500 inliers. The prediction performance of the RoBoost-PLSR method was studied also as a function of the proportion of outliers . It varied from 10% to 40%. These performances were compared to the reference model (PLSR without outliers). This study was carried out with the three simulated datasets.

In the case of the real dataset, the weights parameters of PRM (using the Hampel function) and RoBoost-PLSR were optimised according to the validation set. Only the results of the optimal parameters of RoBoost-PLSR and PRM were presented in the following section. The calibration set was composed of 457 soybean protein (TTS) samples and 100 animal-protein (ANF) samples that represent the outliers. The validation was conducted on 50 additional samples of soybean and results were evaluated through Root Mean Square Error of Prediction (RMSEP).
The evaluation strategy also aimed at assessing the weights attributed to each sample. Weights are evaluted for the number of latent variable resulting in the minimum RMSEP respectively for PRM and Roboost.

## 4. Results and discussion

### *4.1. Simulation 1 : pure Y-outliers*




(a) (b) (c)

FIGURE 1: Simulated dataset 1. (a) Spectra, (b) PCA projection of spectra and (c) distributions of Y-responses. Inliers are represented in blue, while outliers are in red.

Figure 1 presents the properties of the simulated dataset with Y-outliers. Figure 1a shows that inliers spectra (in red) blend perfectly with the rest of the population. Likewise, there is no separation of the two populations of spectra when projected on the two first principal components of a PCA (see Figure 1b). The same behaviour is observed up to the $10^{th}$ component. In this simulation, the outliers are simulated to display significant differences in terms of $Y$. Figure 1c shows that the distribution of the outliers is not similar to the distribution of inliers. The samples are distinguished only by their response values ($\mathbf{y}$) and not by their explanatory variables ($\mathbf{X}$).

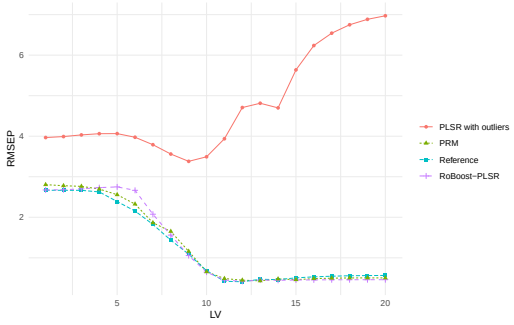



FIGURE 2: Evolution of the RMSEP as a function of latent variables for the reference, PLSR without outliers, PRM and RoBoost-PLSR for the dataset 1

Figure 2 presents four curves showing the RMSEP evolution as a function of the number of LVs, for PLSR calibrated with outliers, the reference, PRM and RoBoost-PLSR, both calibrated with outliers. This figure shows that pure Y-outliers have an impact on the calibration of a PLSR model. In this case, the standard PLSR model calibrated on data including outliers (red curve) achieves very poor prediction performances compared to the reference model (blue curve).
Figure 2 also shows that PRM achieves similar performances with the reference. The behaviour of the RMSEP curve of PRM is similar to the one of the reference along the LVs.
When RoBoost-PLSR (purple curve) is calibrated with Y-outliers, it achieves also similar performances with the reference along the LVs. This means that RoBoost-PLSR attributes low weights to outliers and reaches the best performance of the reference. The behaviour of the RoBoost-PLSR RMSEP curve is close to the reference. This means that the attribution of a weights close to 0 to the outliers for RoBoost-PLSR is independent of the selected number of LVs.

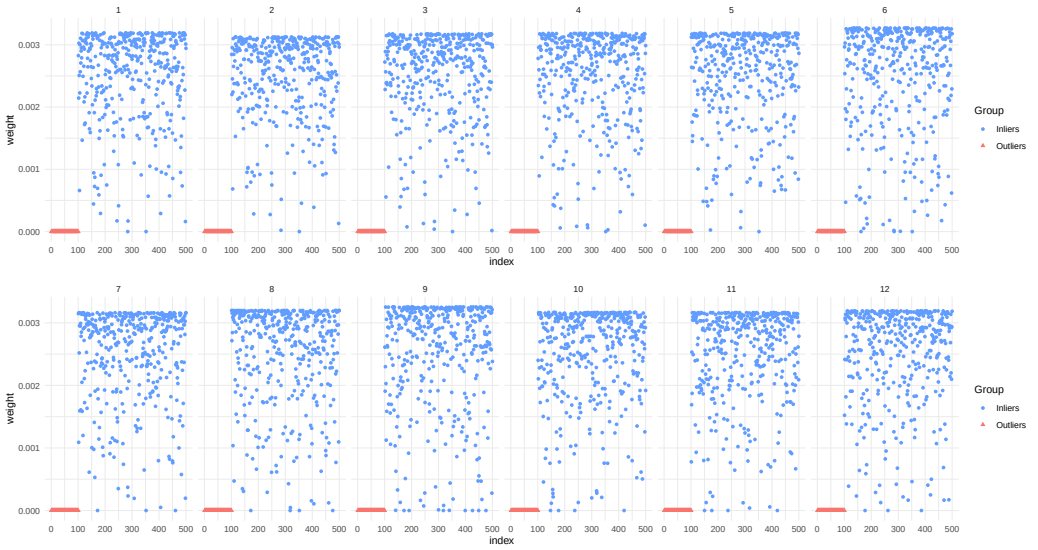



FIGURE 3: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 12 latent variables

Figure 3 shows the weights attributed by RoBoost-PLSR to outliers and

inliers for the best performing model (12 LVs). Since the first LV, the outliers weights are close to 0. Few inliers are also erroneously assigned low weights during calibration. However, in this simulation, this distortion has no impact on prediction performance of RoBoost-PLSR, as shown by Figure 2. It can be observed that there are differences in ceiling values for the weights between LVs. This is due to the normalization of the weights. Actually, the weight of a given sample varies for each LV. At some point, it is possible that an increasing amount of samples are attributed high weights. Therefore, due to normalisation, the maximum value of the weights decreases as more samples are considered relevant from RoBoost-PLSR.

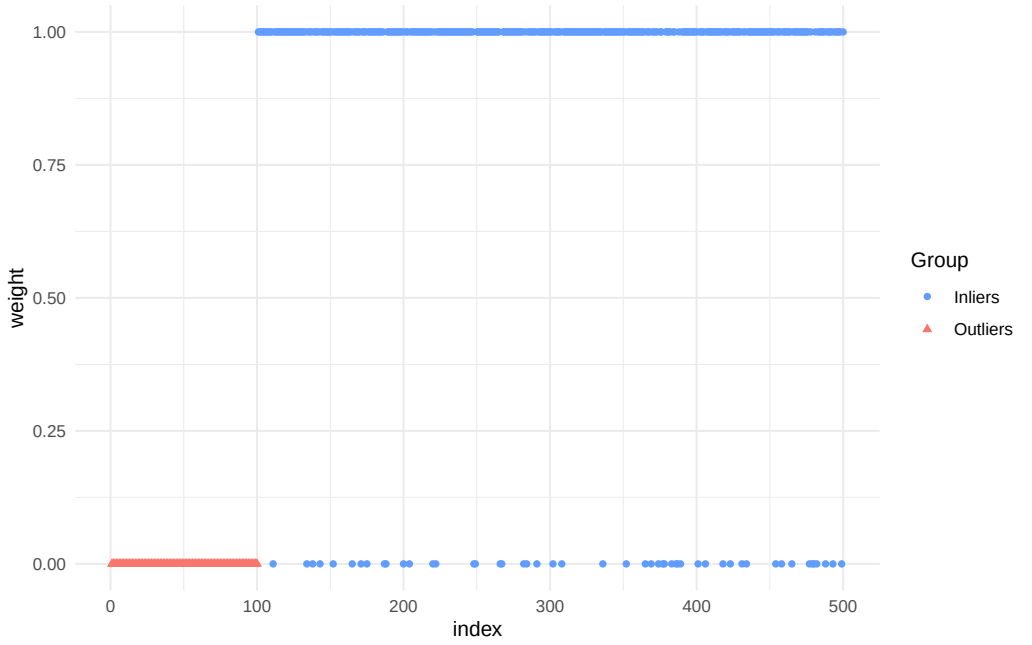
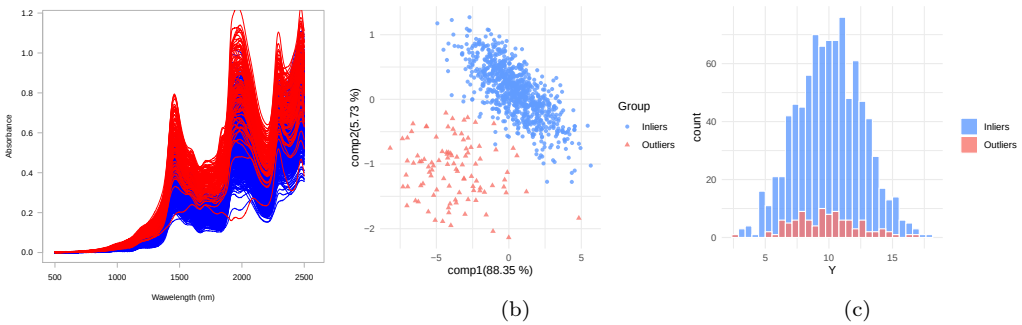



FIGURE 4: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for respectively 13 LVs

Figures 4 show the weights attributed to outliers and inliers in the calibration set for 13 LVs (the best performing model)
Figure 4 shows a clear separation between outliers and inliers weigths with a 13 LVs PRM model. Some inlier samples have a weight of 0 but the vast majority of inlier samples have a weight of 1. As the performance curves of PRM and the reference are almost similar, this does not disturb model calibration.

*4.2. Simulation 2 : contaminant induced outliers*




(a) (b) (c)

FIGURE 5: Simulated dataset 2. (a) Spectra, (b) PCA projection of spectra and (c) distributions of Y-responses. Inliers are represented in blue, while outliers are in red.

Dataset 2 introduces X-outliers. The purpose is to simulate the impact of a contamination of samples during spectral measurements without any anomaly for the reference measures $\mathbf{y}$. Figure 5a shows that such outliers (in red) overlap with standard observations. The difference between the two groups is very faintly apparent on the spectra plot. Figure 5b shows a separation of outliers from inliers on a projection onto the two first principal components of PCA. The two groups are contiguous though, which implies that some outliers could be confounded with inliers. Figure 5c shows the distribution of $Y$ reponses for both outliers and inliers. Data are simulated so that the outliers responses match the same distribution as inliers. In practice, this situation corresponds to the possibility of conducting rigorous reference measurements in controlled laboratory conditions for chemical measures, while the spectral measurements are high-throughput and possibly conducted in outdoor or uncontrolled conditions. In these cases the extraction of information related to the spectra is more complex and probably requires additional LVs. Figure 5 therefore shows that the samples are distinguished only by their explanatory variables ($\mathbf{X}$) and not by their responses ($\mathbf{y}$).

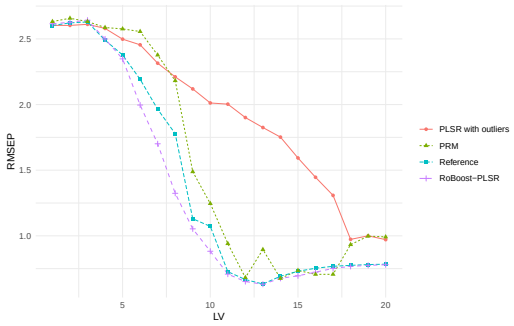



FIGURE 6: Evolution of the RMSEP as a function of latent variables for the reference, PLSR with outliers, PRM and RoBoost-PLSR for the dataset 2

Figure 6 shows that the PLSR with outliers (red curve) is less performant than the reference (blue curve). Indeed, the minimal RMSEP with outliers is $\simeq 1$ for 19 LVs whereas minimal RMSEP without outliers is $\simeq 0.4$ for 13 LVs. This means that the PLSR is sensitive to these outliers. In addition, Figure 6 shows that the number of LVs necessary to achieve the best performances is considerably higher (19 LVs vs. 13 LVs for the reference). This means that outliers add a detrimental information that requires the calculation of a PLSR model with a larger number of LVs [27].
Figure 6 shows that the PRM performance curve is close to the reference curve. This means that PRM can handle the presence of these outliers in the calibration set
Figure 6 shows that the RoBoost-PLSR curve reaches a minimum error close to the reference. RoBoost-PLSR has a behaviour similar to the reference with the minimum RMSEP at 12 LVs. This means that RoBoost-PLSR attributes very low weights to the outliers but also to some inliers.
Both PRM and RoBoost-PLSR prove to be robust to "contaminant induced" which are simple X-outliers. RoBoost-PLSR seems to perform well and have the same behaviour as the reference.

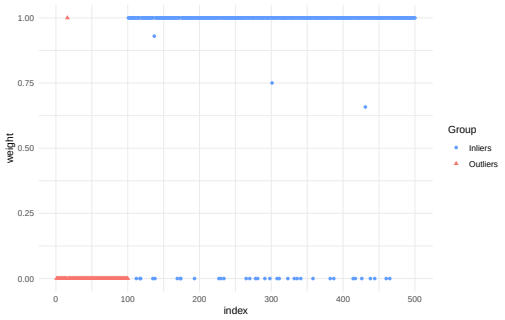



FIGURE 7: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for respectively 12 LVs

Figure 7 shows that the majority of inliers weights are 1 and the outliers weigths 0 for 12 LVs.

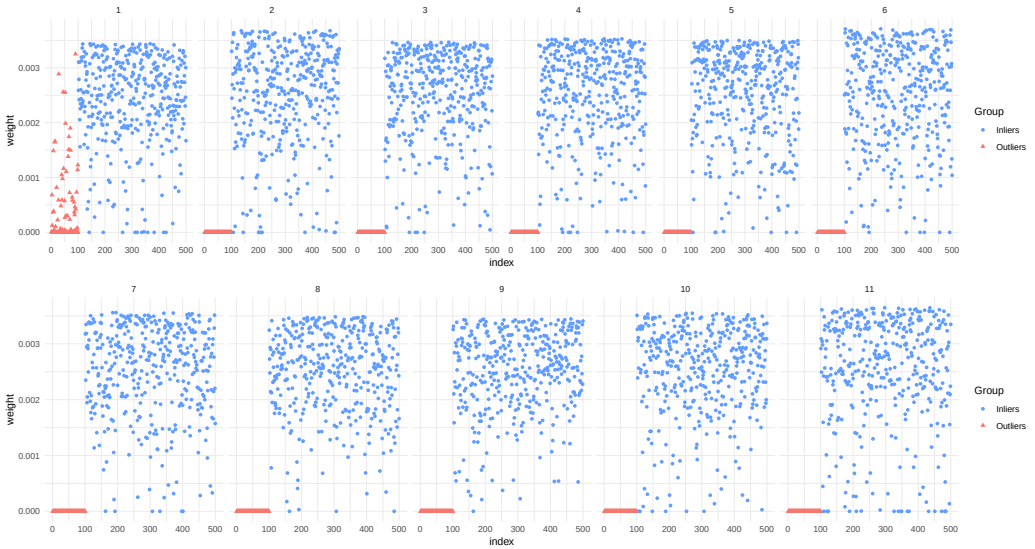
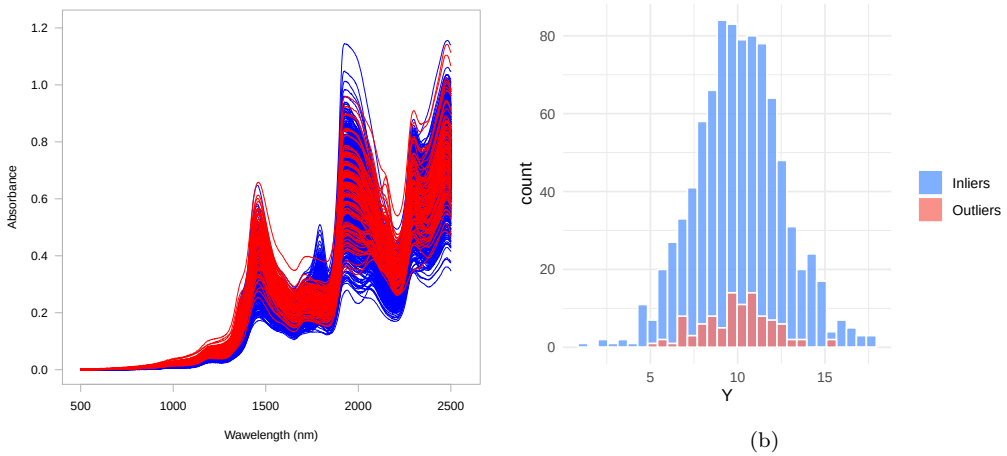



FIGURE 8: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 12 latent variables

Figure 8 compares the weights assigned to outliers and inliers during the calibration process for RoBoost-PLS. It shows that for each LV, RoBoost-PLSR assigns to outliers a weight close to 0. As soon as the $2^{nd}$ latent variable, all outliers have a 0 weight. This result is due to the fact that the simulated spectra (outliers and inliers) have a first common source

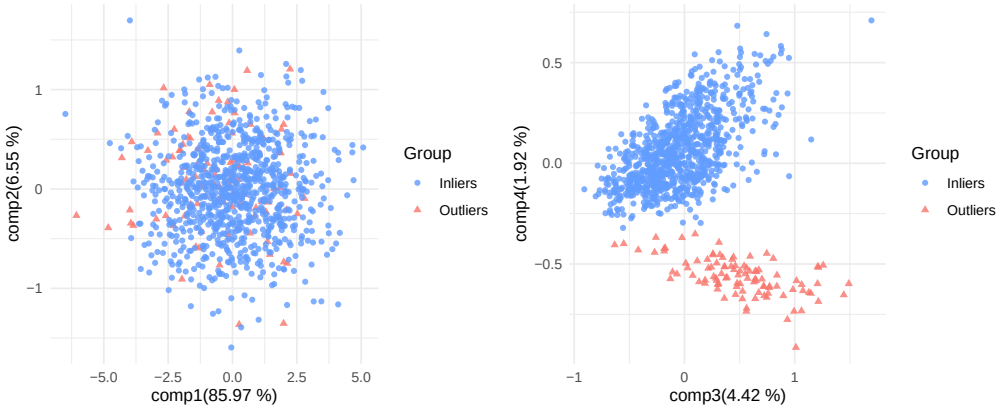
of variability and that, for the first LV, outliers are not detrimental to the model.

### *4.3. Simulation 3 : X-outliers induced by microvariations of the measuring environment*




(a) (b)

FIGURE 9: Simulated dataset 3. (a) Spectra, (b) distributions of Y-responses. Inliers are represented in blue, while outliers are in red.




(a) (b)

FIGURE 10: Simulated dataset 3. (a) PCA projection of spectra onto components 1 and 2, (b) PCA projection of spectra onto components 3 and 4. Inliers are represented in blue, while outliers are in red.

Dataset 3 introduces further X-outliers. The purpose is to simulate the effect of microvariations of the measurement environment, such as temperature or hygrometry shifts *e.g.* when there is a timelapse between spectral measurements. The occurrence of such minor disturbances can alter the resulting spectra in imperceptible ways, yet, sufficiently to deteriorate PLSR models. Figure 9 shows the similarities between the outliers and the inliers. Spectra overlap so that the two populations are indistinguishable. Figure 10 presents their projection PCA axis. The first two components cannot help to differentiate outliers. It is only from the fourth component that the two groups are discriminated. However, this axis represents less than 2% of the total variability which describes the difficulty to determine the presence of such outliers beforehand. In terms of Y-responses, the distribution of outliers is simulated to match the inliers, hence the visible overlay on figure 9. Finally, these samples could have been detected through the appropriate analysis. For instance, some outliers can be distinguished on PCA axes in this case. Nevertheless it is difficult to justify the removal of such samples from the presented graphs. Inherently, a sample should be discarded if it is detrimental to the prediction quality. To determine that, more elaborate methods should be considered, e.g. using a PLS model to detect the samples that diverge from the model. These types of approaches are very useful to understand the phenomena generating these outliers, but require considerable time to study the data. To reduce the time needed

to detect outliers, it would be therefore relevant to use automated robust methods.

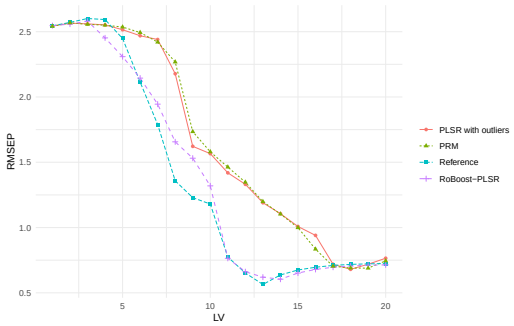



FIGURE 11: Evolution of the RMSEP as a function of latent variables for the reference, PLSR with outliers, PRM and RoBoost-PLSR for the dataset 3

Figure 11 shows that the PLSR with outliers (red curve) is less performant than the reference (blue curve). Indeed, the minimal RMSEP for the PLSR with outliers is $\simeq 0.7$ for 18 LVs whereas the minimal RMSEP of the reference is $\simeq 0.55$ for 13 LVs. This means that PLSR is sensitive to these outliers.
Figure 11 shows that the PRM performance curve is close to the PLSR with outliers curve. This means that PRM does not completely capture the nature of these outliers. It is fair to conjecture that PRM will perform much better for these data if based on a reweighting scheme that accounts for the residuals in the X-space as well
Figure 11 shows that the RoBoost-PLSR curve reaches a minimum error with 14 LVs, which is close to the reference. RoBoost-PLSR has a behaviour very similar to that of the reference. The minimum RMSEP of RoBoost-PLSR curve (14 LVs) is higher than the minimum RMSEP of the reference (13 LVs). This means that RoBoost-PLSR attributes a 0 weights to the outliers but also to some inliers. This leads to an increase in the number of LVs for a higher minimum RMSEP than the minimum RMSEP of the reference.

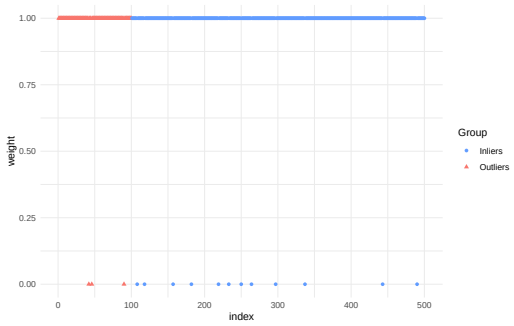



FIGURE 12: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for 18 latent variables

Figure 12 does not show a clear separation between the majority of outlier weights and inlier weights with a 18 LVs PRM model. This is due to the fact that outliers are not detected by PRM. This limitation of PRM could be explained by the absence of X-residuals in the computation of weights. This also could be explained by the fact that outliers are weighted using a model with a predefined number of LVs.

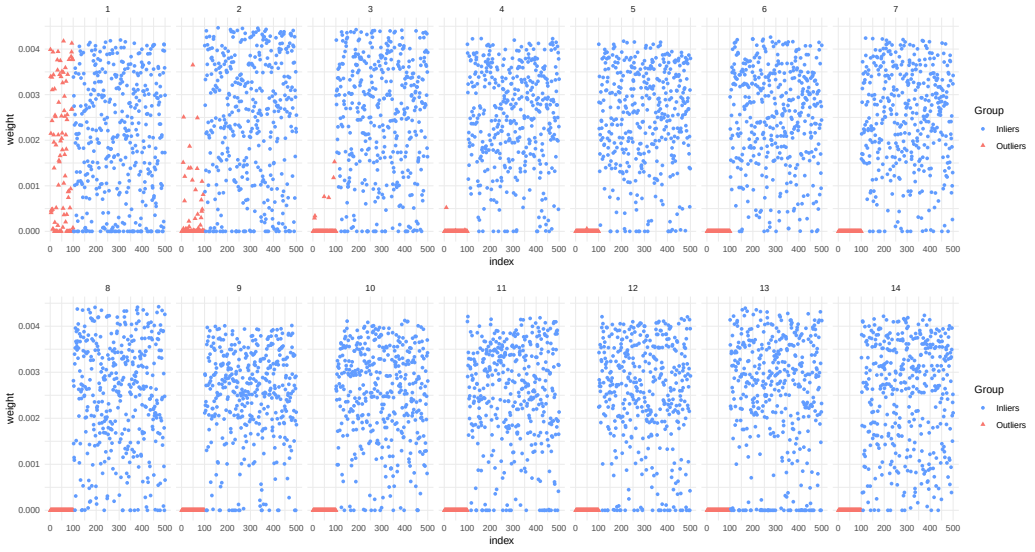
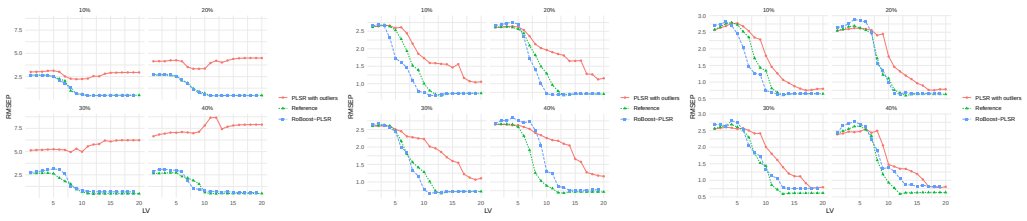



FIGURE 13: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 14 latent variables

Figure 13 shows the weights assigned to outliers and inliers by

RoBoost-PLSR. It shows that RoBoost-PLSR begins to assign 0 weights to the outliers from the $3^{rd}$ LV. RoBoost-PLSR also attributes very low weights to a significant number of inliers while some outliers are attributed higher weights along the three first LVs. This means that some *a priori* informative samples are not necessarily favourable or even relevant for some LVs. It also means that outliers are not necessarily detrimental for the determination of all LVs. For example, the first LV can often be assimilated to baselines. In these cases, outliers sharing a similar baseline are not detrimental while inliers with minor baseline shifts can be detrimental.
RoBoost-PLSR seems to be able to taking into account the variability of the beneficial samples and even sometimes the non-abnormal properties of outliers.

### *4.4. Influence of the proportion of outliers within calibration*




(a) simulation1 (b) simulation2 (c) simulation3

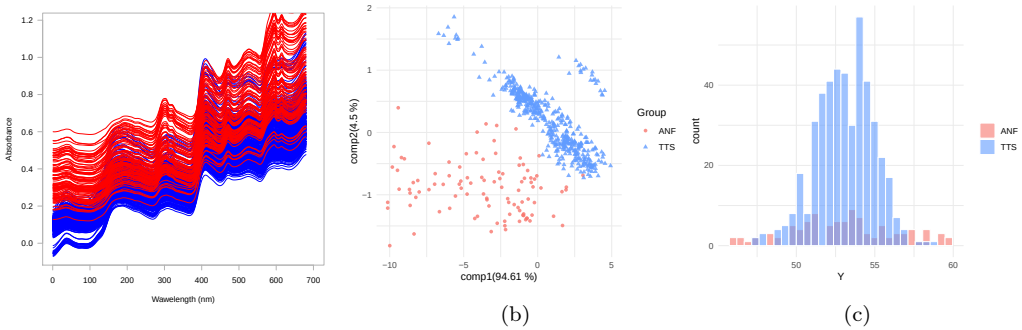
FIGURE 14: RMSEP depending on the proportion of outliers in the calibration set

Figure 14 shows the prediction performances obtained with proportions of outliers varying between 10% and 40% for PLSR and RoBoost-PLSR. Figure 14 shows that the proportion of outliers affects the PLSR performance for each simulation. The higher the proportion of outliers, the lower the quality of outlier prediction.
Figure 14a shows that the proportion of outliers does not affect the performance of RoBoost-PLSR until 30% of outliers. This is due to the fact that the Y-distributions of the two groups of samples are differentiable (see Figure 1) and therefore the separation between outliers and inliers is easy. However, with 40% outliers, RoBoost-PLSR methods can not produce the same result as the PLSR method without outliers. Indeed, when the proportion of outliers is close to the proportion of inliers, it becomes really difficult to focus the model on the main mass. Despite this, RoBoost-PLSR method has a stable curve and is generally close to reference even with 40%

outliers.
Figure 14b shows that the proportion of outliers has little effect on the performance of the RoBoost-PLSR method. This means that the outliers are correctly detected by RoBoost-PLSR even when the proportion of outliers is close to the inliers proportion.
Figure 14c shows that the proportion of outliers has little effect on the performance of the RoBoost-PLSR method until 30%. For 40%, the curve of RoBoost-PLSR is between the PLSR witout outliers and the PLSR with outliers. This means that the method detects some but not all outliers. In conclusion, the RoBoost-PLSR method supports these three types of outliers up to 30% with prediction performances approaching the reference.

*4.5. Real dataset and application : prediction of protein content.*

The present section intends to deal with real agronomic data, with the example of a common animal nutrition application : the prediction of the protein content of feed materials and the presence of incorrectly categorised samples. In this database the samples resulting from animal bonemeal (noted ANF) represent the outliers polluting the regular soyabean cakes (noted TTS).




(a) (b) (c)

FIGURE 15: Properties of TTS and ANF proteins. (a) Spectra, (b) PCA projection of spectra and (c) distributions of Y-responses. TTS Inliers are represented in blue, while ANF outliers are in red.

In the proposed application, the proteins contained in ANF outliers, present spectral similarities with soya proteins (TTS). Therefore, even in minor proportions, these outliers can alter PLSR models. Figure 15 shows the similarities between the outliers (ANF) and the inliers (TTS). Spectra

overlap so that the two populations are indistinguishable. There is supposedly an overall difference in baselines, yet insufficient to separate the data into consistent clusters. Figure 15 presents the data projected on the first two PCA axes. On the basis of this projection, it is also difficult to attest the presence of two distinct groups. With the beforehand knowledge regarding the affiliation of samples, inliers (in blue) seem to follow a precise trajectory while the outliers (in red) form a sparse cloud. However without this knowledge, it would not represent a reliable clustering of data, all the more since inliers present a second marginal distribution, parallel to the main one. Therefore in practice it is not trivial to discard unknown outliers, accidentally introduced in a calibration set. In terms of response, (see fig 15), the outliers also present a similar distribution (in red) overlapping the distribution of inliers responses (in blue). It is often the case in food applications where different raw materials can present comparable nutrient contents.

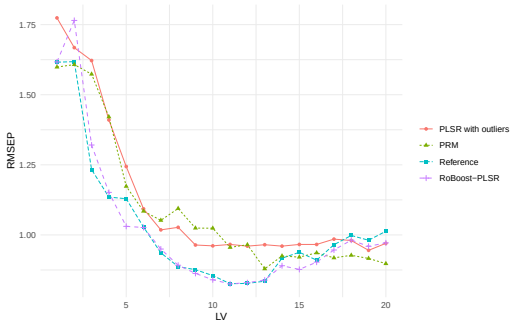



FIGURE 16: Evolution of the RMSEP as a function of latent variables for the reference, PLSR with outliers, PRM and RoBoost-PLSR for the real set

Figure 16 shows that the PLSR with ANF samples (red curve) is less performant than the reference (PLSR calibrated without ANF samples, blue curve). Indeed, the minimal RMSEP for the PLSR with ANF samples is $\simeq$ 0.95 for 19 LVs whereas the minimal RMSEP of the reference is $\simeq$ 0.83 for 11 LVs. This means that PLSR is sensitive to these ANF samples.
Figure 16 shows that the PRM performance curve is between the PLSR with and without ANF samples curves. At best, it achieves an RMSEP equal to 0.87 for 13 LVs.
Figure 16 shows that the RoBoost-PLSR curve reaches a minimum error with 11 LVs, that is the same as the reference (RMSEP = 0.83). RoBoost-PLSR has a behaviour very similar to that of the reference. This means that

RoBoost-PLSR attributes a 0 weights to the ANF samples but also to some TTS samples.

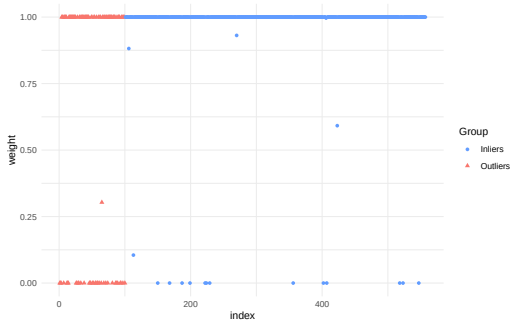



FIGURE 17: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for 13 latent variables with the best weights constant setting

Figure 17 presents the repartition of the weights attributed within the calibration of PRM. ANF samples weights are not distinguished from the TTS samples weights. This result explains the poor prediction performances of PRM on this real dataset.

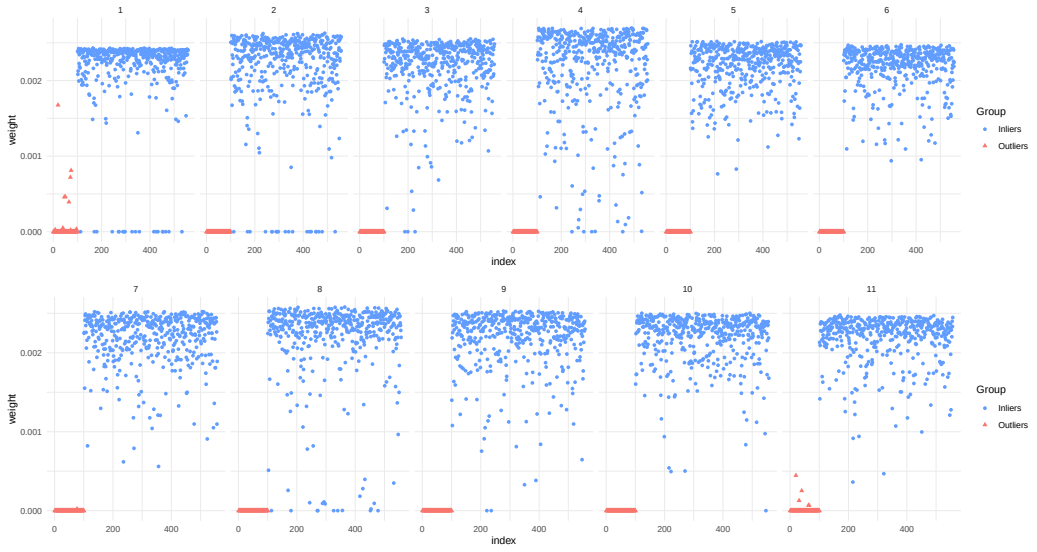



FIGURE 18: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 11 latent variables with the best weights constants settings

Figure 18 presents the weights attributed to samples for the eight considered LVs within the calibration of RoBoost-PLSR. From the first LV, most ANF samples are assigned null weights along with a few TTS samples. From the second latent variable, all ANF samples weights are close to 0.

## 5. Conclusion & Perspectives

This article showed the potential of the RoBoost-PLSR method. This method offers a relevant solution for the calibration of PLSR models in the presence of various types of outliers. At this stage the proposed algorithm is mainly based on weighting strategies within a of series unidimensional PLS. The method is designed to detect outliers within the calibration, through iterations where dissimilarity measurements take into account the hypothesis of robust linear models. The four evaluated applications demonstrate that the introduced outliers are predominantly detected and discarded from the model. As a result, RoBoost-PLSR is able to attain performance on par with the reference.

One dataset was found to be particularly difficult to process by the PRM method. This dataset was the one with X-outliers. It would be interesting to integrate within PRM a weighting criterion related to the X-residuals (as in RoBoost-PLSR).This would enable to observe the benefit of considering X-residuals compared to the benefit of estimating weights based on the score space alone. Eventually, RoBoost-PLSR proved to be a promising framework to deal with various practical issues. Further studies should be carried out in practical context for more diverse applications ; Including smaller datasets, where it is yet undetermined if the estimation of weight criteria is still relevant / functional. Indeed, the observations carried out in this paper are based on large learning databases. This implies that it is potentially possible to apply stricter weights without degrading the prediction quality of the method.

To this end, the method requires further studies on the following issues : Firstly, a comprehensive study regarding the weight functions and their optimisation should be carried out, in order to better adjust the models. Indeed, the parametrisation of RoBoost-PLSR can be a difficult task (three parameters have to be optimised for each LV). In this paper, the constants were fixed for all the latent variables. However, it would be relevant to define specific constants optimised for each latent variable. It is not conceivable to manually optimise constants for each latent variable. Secondly, in real applications, outliers can be present both in the calibration and validation

sets. In this paper, the validation sets are not contaminated. To obtain a fully operational method, it should be completed with the development of a metric intended to determine the consistency of unknown samples with the model. This would enable to predict datasets containing potential outliers, and then only process data for which the model is calibrated for.

Thirdly, the interpretation of the RoBoost-PLSR model is complex. Indeed, the proposed algorithm does not provide an estimation of regression coefficients unlike approaches such as PRM. In order to allow better interpretability, it would be essential in future work to propose an algorithm that enables an estimation of the regression coefficients. Fourthly, the initial estimators (centering) and the estimation of the sample weights can be corrupted. In RoBoost-PLS, data are centred about the arithmetic mean. It is well known that the arithmetic mean is non-robust and can thereby provide distorted starting values for the algorithm. A potential solution is to replace these estimators with robust alternatives (*e.g.* robust multivariate location). As for the bisquare weight function, it uses the (coordinatewise) median, which can lie outside the convex hull of the data than breakdown. It would be relevant to consider other weight functions that take into account these aspects. Fifthly, the cross-validation of robust methods, for instance for the optimisation of hyperparameters, is not a straightforward procedure. In this paper, this limitation has been overcome by optimising and studying the behaviour of the methods on an unpolluted validation set. In future work it will be interesting to develop tools to cross-validate the RoBoost-PLSR method in order to allow the development of this method on real cases.

Finally, the robust multivariate methods have proven their reliable predictive quality in classification issues [29]. RoBoost-PLSR could also be adapted to classification problems. This implies that RoBoost-PLSR should be adapted to multidimensional $Y$.

**Acknowledgement**

This work was supported by the French National Research Agency under the Investments for the Future Program, referred as ANR-16-CONV-0004. Authors care to thank Vincent Baeten and Pierre Dardenne from the CRA-W (Agronomic Research Centre of Wallonia, Belgium) for providing the real dataset used in this article. Authors wish to adress a special thanks to Gilbert Saporta from the CNAM for his thorough proofreading of the paper and his precious advice.

## Referencse


[1] S. Wold, H. Martens, H. Wold, The multivariate calibration problem in chemistry solved by the pls method, in : B. Kågström, A. Ruhe (Eds.), Matrix Pencils, Springer Berlin Heidelberg, Berlin, Heidelberg, 1983, pp. 286–293.

[2] S. Serneels, C. Croux, P. J. Van Espen, Influence properties of partial least squares regression, Chemometrics and Intelligent Laboratory Systems 71 (1) (2004) 13–20. doi:https://doi.org/10.1016/j.chemolab.2003.10.009.

[3] P. Filzmoser, S. Höppner, I. Ortner, S. Serneels, T. Verdonck, Cellwise robust M regression, Computational Statistics & Data Analysis 147 (2020) 106944. doi:10.1016/j.csda.2020.106944.

[4] M. Griep, I. Wakeling, P. Vankeerberghen, D. Massart, Comparison of semirobust and robust partial least squares procedures, Chemometrics and Intelligent Laboratory Systems 29 (1) (1995) 37–50. doi:10.1016/0169-7439(95)80078-N.

[5] I. Stanimirova, S. Serneels, P. J. Van Espen, B. Walczak, How to construct a multiple regression model for data with missing elements and outlying objects, Analytica Chimica Acta 581 (2) (2007) 324–332. doi:10.1016/j.aca.2006.08.014.

[6] R. J. Pell, Multiple outlier detection for multivariate calibration using robust statistical techniques, Chemometrics and Intelligent Laboratory Systems 52 (1) (2000) 87–104. doi:10.1016/S0169-7439(00)00082-4.

[7] J. A. Gil, R. Romera, On robust partial least squares (PLS) methods, Journal of Chemometrics 12 (6) (1998) 365–378. doi:10.1002/(SICI)1099-128X(199811/12)12:6<365::AID-CEM519>3.0.CO;2-G.

[8] S. Acitas, P. Filzmoser, B. Senoglu, A new partial robust adaptive modified maximum likelihood estimator, Chemometrics and Intelligent Laboratory Systems 204 (2020) 104068. doi:10.1016/j.chemolab.2020.104068.

[9] J. González, D. Peña, R. Romera, A robust partial least squares regression method with applications, Journal of Chemometrics 23 (2) (2009) 78–90. doi:10.1002/cem.1195.

[10] I. N. Wakeling, H. J. H. Macfie, A robust PLS procedure, Journal of Chemometrics 6 (4) (1992) 189–198. doi:10.1002/cem.1180060404.

[11] J. Peng, S. Peng, Y. Hu, Partial least squares and random sample consensus in outlier detection, Analytica Chimica Acta 719 (2012) 24–29. doi:10.1016/j.aca.2011.12.058.

[12] P. Filzmoser, R. Maronna, M. Werner, Outlier identification in high dimensions, Computational Statistics & Data Analysis 52 (3) (2008) 1694–1711. doi:10.1016/j.csda.2007.05.018.

[13] M. Hubert, K. V. Branden, Robust methods for partial least squares regression, Journal of Chemometrics 17 (10) (2003) 537–549. doi:10.1002/cem.822.

[14] U. Kruger, Y. Zhou, X. Wang, D. Rooney, J. Thompson, Robust partial least squares regression : Part II, new algorithm and benchmark studies, Journal of Chemometrics 22 (1) (2008) 14–22, _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.1095. doi:10.1002/cem.1095.

[15] I. Hoffmann, S. Serneels, P. Filzmoser, C. Croux, Sparse partial robust M regression, Chemometrics and Intelligent Laboratory Systems 149 (2015) 50–59. doi:10.1016/j.chemolab.2015.09.019.

[16] P. Filzmoser, V. Todorov, Review of robust multivariate statistical methods in high dimension, Analytica Chimica Acta 705 (1-2) (2011) 2–14. doi:10.1016/j.aca.2011.03.055.

[17] S. F. Møller, J. v. Frese, R. Bro, Robust methods for multivariate data analysis, Journal of Chemometrics 19 (10) (2005) 549–563, _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.962. doi:10.1002/cem.962.

[18] S. Serneels, C. Croux, P. Filzmoser, P. J. Van Espen, Partial robust M-regression, Chemometrics and Intelligent Laboratory Systems 79 (1) (2005) 55–64. doi:10.1016/j.chemolab.2005.04.007.

[19] J. Betzin, Pls-regression in the boosting framework, in : M. Vilares, M. Tenenhaus, P. Coelho, A. Morineau, V. Esposito Vinzi (Eds.), PLS and Related Methods, DECISIA, Levallois Perret, 2003, pp. 261–269.

[20] A.-L. Boulesteix, PLS Dimension Reduction for Classification with Microarray Data, Statistical Applications in Genetics and Molecular Biology 3 (1), publisher : De Gruyter Section : Statistical Applications in Genetics and Molecular Biology (Nov. 2004). doi:10.2202/1544-6115.1075.

[21] X. Shao, X. Bian, W. Cai, An improved boosting partial least squares method for near-infrared spectroscopic quantitative analysis, Analytica Chimica Acta 666 (1-2) (2010) 32–37. doi:10.1016/j.aca.2010.03.036.

[22] R. Rosipal, N. Krämer, Overview and Recent Advances in Partial Least Squares, in : C. Saunders, M. Grobelnik, S. Gunn, J. Shawe-Taylor (Eds.), Subspace, Latent Structure and Feature Selection, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2006, pp. 34–51. doi:10.1007/11752790_2.

[23] M. H. Zhang, Q. S. Xu, D. L. Massart, Boosting partial least squares, Analytical Chemistry 77 (5) (2005) 1423–1431, pMID : 15732927. doi:10.1021/ac048561m.

[24] D. J. Cummins, C. W. Andrews, Iteratively reweighted partial least squares : A performance analysis by monte carlo simulation, Journal of Chemometrics 9 (6) (1995) 489–507. doi:10.1002/cem.1180090607.

[25] S. Schaal, C. G. Atkeson, S. Vijayakumar, Scalable Techniques from Nonparametric Statistics for Real Time Robot Learning, Applied Intelligence 17 (1) (2002) 49–60. doi:10.1023/A:1015727715131.

[26] W. S. Cleveland, Robust Locally Weighted Regression and Smoothing Scatterplots, Journal of the American Statistical Association (1979) 9.

[27] M. Metz, A. Biancolillo, M. Lesnoff, J.-M. Roger, A note on spectral data simulation, Chemometrics and Intelligent Laboratory Systems 200 (2020) 103979. doi:10.1016/j.chemolab.2020.103979.

[28] M. Lesnoff, M. Metz, J.-M. Roger, Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data, Journal of Chemometrics 34 (5) (2020) e3209. doi:10.1002/cem.3209.

[29] I. Hoffmann, P. Filzmoser, S. Serneels, K. Varmuza, Sparse and robust PLS for binary classification, Journal of Chemometrics 30 (4) (2016) 153–162. doi:10.1002/cem.2775.

## Appendix

TABLE .1: The different choices in the simulation 1

| | Inliers | Outliers |
|---|---|---|
| $\mathbf{P}_u$ | Pure spectrum of glucose | |
| $\mathbf{T}_u$ | Folded-normal distribution | |
| $\mathbf{P}_d$ | Pure spectrum of water<br>Pure spectrum of ethanol<br>Spectrum of water-ethanol Interaction<br>10 Artificial spectra | |
| $\mathbf{T}_d$ | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution | |
| $\mathbf{E}$ | Gaussian distribution | |
| $f$ | $Y = 10 * T_{glucose}$ | $\mathbf{Y} = -\mathbf{5} * \mathbf{T_{glucose}}$ |
| $\mathbf{F}$ | Gaussian distribution | |

u define useful space, d define detrimental space, E define the spectral noise and F the response noise.

TABLE .2: The different choices in the simulation 2

| | Inliers | Outliers |
|---|---|---|
| $\mathbf{P}_u$ | Pure spectrum of glucose | |
| $\mathbf{T}_u$ | Folded-normal distribution | |
| $\mathbf{P}_d$ | Pure spectrum of water<br>Pure spectrum of ethanol<br>Spectrum of water-ethanol Interaction<br>10 Artificial spectra | Pure spectrum of water<br>Pure spectrum of ethanol<br>Spectrum of water-ethanol Interaction<br>10 Artificial spectra<br>**100 Artificial spectra** |
| $\mathbf{T}_d$ | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution<br>**Folded-normal distribution** |
| $\mathbf{E}$ | Gaussian distribution | |
| $f$ | $Y = 10 * T_{glucose}$ | |
| $\mathbf{F}$ | Gaussian distribution | |

u define useful space, d define detrimental space, E define the spectral noise and F the response noise.

TABLE .3: The different choices in the simulation 3

| | Inliers | Outliers |
|---|---|---|
| $\mathbf{P}_u$ | Pure spectrum of glucose | |
| $\mathbf{T}_u$ | Folded-normal distribution | |
| $\mathbf{P}_d$ | Pure spectrum of water<br>Pure spectrum of ethanol<br>Spectrum of water-ethanol Interaction<br>10 Artificial spectra | Pure spectrum of water<br>Pure spectrum of ethanol<br>Spectrum of water-ethanol Interaction<br>10 Artificial spectra<br>**10 Artificial spectra** |
| $\mathbf{T}_d$ | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution<br>**Folded-normal distribution** |
| $\mathbf{E}$ | Gaussian distribution | |
| $f$ | $Y = 10 * T_{glucose}$ | |
| $\mathbf{F}$ | Gaussian distribution | |

u define useful space, d define detrimental space, E define the spectral noise and F the response noise.

# Chapitre 6

## Article 5 : Massivespectral data analysis for plant breeding using parSketch-PLSDA method :Discrimination of sunflower genotypes

Référence :

Maxime Ryckewaert, Maxime Metz, Daphné Héran, Pierre George, Bruno Grèzes-Besset, Reza Akbarinia, Jean-Michel Roger, and Ryad Bendoula. Massive spectral data analysis for plant breeding using parSketch-PLSDA method :Discrimination of sunflower genotypes. Biosystems Engineering, 210 :69–77,2021. ISSN 1537-5110. doi : https://doi.org/10.1016/j.biosystemseng.2021.08.005. URL https://www.sciencedirect.com/science/article/pii/S1537511021001896

MDPI

*Article*

# Massive spectral data analysis for plant breeding using parSketch-PLSDA method: discrimination of sunflower genotypes

**Maxime Ryckewaert [1,2], Maxime Metz[1,2], Daphné Héran [1], Pierre George[3], Bruno Grèzes-Besset[3], Reza Akbarinia[4], Jean-Michel Roger[1,2], Ryad Bendoula [1]**

1 ITAP, Univ Montpellier, INRAE, Institut Agro, Montpellier, France
2 ChemHouse Research Group, Montpellier, France
3 Innolea, 6 chemin des Panedautes, 31700 Mondonville, France
4 Inria & LIRMM, Univ Montpellier, France
† Current address: maxime.ryckewaert@inrae.fr

Version October 11, 2021 submitted to Sensors

**Abstract:** In precision agriculture and plant breeding, the amount of data tends to increase. This massive data is becoming more and more complex, leading to difficulties in managing and analyzing it. Optical instruments such as NIR Spectroscopy or hyperspectral imaging are gradually expanding directly in the field, increasing the amount of spectral database. Processing this massive amount of spectral data is challenging. In a context of genotype discrimination, we propose to apply a method called parSketch-PLSDA to analyze such a massive amount of spectral data. For this purpose, a spectral database was formed by collecting 1,300,000 spectra from hyperspectral images of leaves of four different sunflower genotypes. ParSketch-PLSDA is compared to a PLSDA which is a reference method. Both methods use the same set of calibration and test. The prediction model obtained by PLSDA has a classification error close to 23% on average across all genotypes. ParSketch-PLSDA method outperforms PLSDA by greatly improving prediction qualities by 10%. These results are encouraging and allow us to anticipate the future bottleneck related to the generation of a large amount of data from phenotyping.

**Keywords:** Spectroscopy; Massive data; Digital Agriculture; Precision Agriculture; Chemometrics

## 1. Introduction

In recent years, precision agriculture and plant breeding have tended to increase the quantity and complexity of phenotyping related data [1–3]. Managing and analyzing huge amounts of data are identified as a future bottleneck in phenotyping [2]. Indeed, over the last few years, high throughput phenotyping (HTP) platforms in the laboratory or directly in the field have been flourishing [4,5]. These platforms provide a monitoring of one or more phenotypic traits of the vegetation. This information can be obtained at different spatial, spectral or temporal scales depending on the studied level, which could be vegetation organ, individual or even population [1,6,7]. The higher the spatial, spectral or temporal resolution, the larger the amount of data.

Spectroscopy in the visible and near-infrared range (VIS-NIR) has proven to be relevant for providing useful information for vegetation monitoring. Several plant phenotyping issues can be tackled with high spectral resolution measurements such as biochemical variable access [8,9], disease [10,11] or stress detection [12,13].

From a technological point of view, spectral acquisitions directly in the field have been made possible thanks to spectrometer miniaturization [14,15] or hyperspectral imager evolution [16,17].

Associated with mobile vectors (such as UAV, tractor, pedestrian), these tools become high-throughput phenotyping instruments and generate a large amount of spectral data. However, simple computations on this amount of data such as outlier detection or the use of pre-processing become difficult to perform and very time consuming [18]. Processing this massive amount of spectral data is challenging.

In chemometrics, most popular methods as Partial Least Square (PLS) [19] are based on an assumption of linear relationship between spectral data and specific variables[20]. These methods are popular because of their good predictive performances and low computation time. Conversely, using these methods may not provide good prediction models when relationships between spectra and variable of interest are non-linear.

When dealing with a large amount of spectral data, complex structures and non-linear relationships can arise which may compromise linear regression approaches. [21]. In practice, using a linear classification or regression to predict a complex database would lead to degraded results [22,23]. As a consequence, linear methods are challenged on large amounts of data. Furthermore, some methods, called local methods, exploit data based on a restricted neighborhood of individuals which greatly improves prediction quality [21,24–26]. These methods can be used to overcome non-linearity problems under the assumption that with a restricted neighborhood, the relationship between spectra and variables becomes linear. The parSketch method has recently been proposed to implement a local approach to a large volume of data [27] Therefore, parSketch can be used to address the complex analysis of large amount of spectral data from phenotyping.

In this paper, we propose to study the use of the parSketch method to exploit a large amount of spectral data. Additionally, we compare this method with a reference method in an application of discrimination of different sunflower varieties.

## 2. Materials and methods

### *2.1. Biological material*

Four sunflower genotypes (called A, B, C and D) were grown in a greenhouse at INRAE France in comfortable water and light conditions. Water and lighting conditions were similar for each pot with a day-night cycle of 12 h/8 h. The greenhouse was equipped with multispectral lighting (450 nm, 560 nm, 660 nm, 730 nm and 6000°K) controlled by Herbro automaton (GreenHouseKeeper). Irrigation occurred every two days and corresponded to water comfort condition. For the four selected genotypes, two potted plants (called P1 and P2) of each were grown. Four leaves were collected at the upper and middle parts of each plant, except for the genotype D where only two leaves of each plant were collected. Leaf petioles were immediately wrapped with water soaked paper before measurements. In total, 28 leaves were then collected and measured.

### *2.2. Spectral acquisitions*

Spectral data of the prepared leaf samples were acquired in the reflectance mode by using a laboratory-based line scanning Hyperspectral Imaging System (HIS). The HIS system was composed of a halogen light source, a translation rail, and a detection system. The sample was placed on a translation rail, synchronized with the acquisition software (NEO Hyspex, Norsk Elektro Optikk AS) which can record images when sample was scanned under the hyperspectral camera (NEO Hyspex VNIR-1600 with 30cm-objective, Norsk Elektro Optikk AS, Skedsmokorest, Norway). Spectral data were acquired in the $400 - 1000$ nm wavelength range with 3.7 nm intervals.

For each sample, the reflected light intensity ($I_s(\lambda)$) was measured at each wavelength . Dark current image ($I_b(\lambda)$) was also recorded for each measure. A white reference (SRS99, Spectralon ®) was used as a reference ($I_o(\lambda)$) to standardize images from non-uniformities of all components of the instrumentation (light source, lens, detector). From these measurements, reflectance ($R_s(\lambda)$) was calculated for each sample:

$$R_s(\lambda) = \frac{I_s(\lambda) - I_b(\lambda)}{I_0(\lambda) - I_b(\lambda)} \quad (1)$$

For all hyperspectral images, vegetation pixels were selected to form a spectral data set. This selection was made by threshold values (Fig. 1). Leaf spectra were collected from the 28 hyperspectral images, representing more than 1,300,000 spectra.

**(a)** **(b)**

**Figure 1.** Pixel selection from a hyperspectral image of a sunflower leaf (a) image, (b) mask based on threshold values

### *2.3. Data analysis*

Two methods were used to compare their ability to discriminate sunflower genotypes. Both methods were applied to a similar data set, called test set, built out of the spectra database. Calculations were performed with the R software (version 3.6.1 [28]) and rnirs package (https://github.com/mlesnoff/rnirs) was used for classical discrimination methods (PLSDA).

#### 2.3.1. PLSDA method

The Partial Least Squares for Discrimination Analysis (PLSDA) [29] was used as reference method for classification. This method consisted of building models between multivariate data and a vector coding different classes (here, the four genotypes).

Multivariate data was represented by a matrix $\mathbf{X}$ of size ($n$, $p$) where $n$ was the observation number and $p$ the variable number. The $n$ observations were identified by their corresponding class in the vector $y$ of size ($n$,1) where values ranged from 1 to $q$, where $q$ was the class number. The first step was to transform $y$ into a dummy matrix $\mathbf{Y}$ of size ($n$, $q$) also called disjunctive table.

$$y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{bmatrix} \rightarrow \mathbf{Y} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

An example of a dummy matrix is given in equation 2 with nine observations belonging to three classes. The matrix $\mathbf{Y}$ contains binary values (0,1) where each column corresponded to a class. For a given observation, the class-corresponding column has a value of 1 while other columns were equal to 0.

Then, a Partial Least Square (PLS) model [19] was applied between $\mathbf{X}$ and $\mathbf{Y}$. $\mathbf{Y}$ being multidimensional, the algorithm PLS2 adapted to the prediction of several responses was used. Finally, a linear discriminant analysis (LDA) [30] was applied between the PLS2 scores and $\mathbf{Y}$.

### 2.3.2. ParSketch-PLSDA method

The other strategy was to apply the parSketch-PLSDA method [27], an extension of the KNN-PLSDA method for massive data processing [31]. ParSketch-PLSDA was used to combine an indexation strategy (parSketch) and the PLSDA. An approximation of the neighborhood was defined for each spectrum to be classified. This neighborhood was then used to compute a PLSDA model and to predict which class belong new spectra.

ParSketch was performed in three steps: dimension reduction, grid creation, neighborhood approximation. Three method parameters ($v$, $s$, $m$) were defined, corresponding to these three steps, and are described below.

First, a dimension reduction was achieved by calculating the matrix **T** corresponding to the sketch of the matrix **X** as follows:

$$\mathbf{T} = \mathbf{XP} \tag{3}$$

Where **P** was a matrix of size ($p$,$v$) containing values of -1 or 1 according to a random selection. The first parameter of ParSketch ($v$), corresponding to the column number of **P** was then defined. The higher the value of $v$ the better the approximation of the neighborhood. However, the larger the value of $v$ the longer the parSketch method computation time.

The second step corresponding to the grid creation process (see Fig. 2) was to segment the space (2d) formed by adjacent pairs of **T** columns. The number of segments ($s$) is the second parSketch parameter. The higher the value of $s$ the better the approximation of the nearest neighbors. However, the greater the value of $s$, the smaller the number of neighbors.

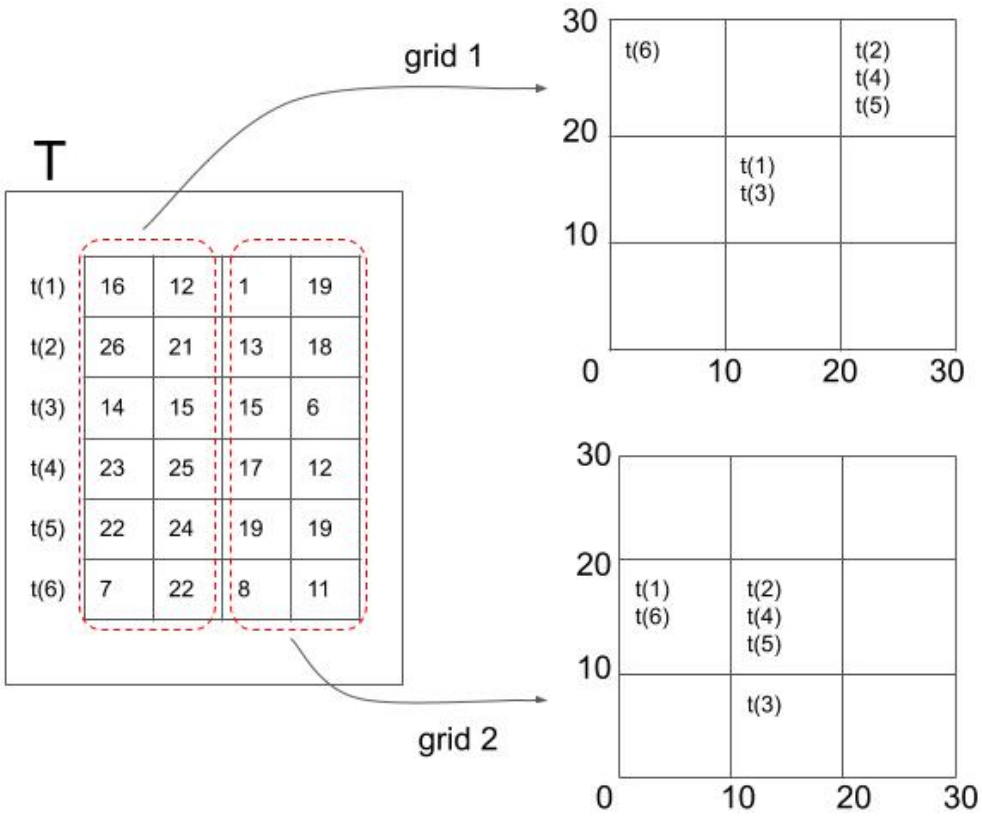



**Figure 2.** An illustrated example of grid creation with a segment number $s = 3$

The last step to configure parSketch was to define the minimal number $m$ of grid returned in the neighbor's search. This step corresponded to the neighborhood approximation for the grid search process (see Fig. 3). The higher the value of $m$ the better the approximation of the nearest neighbors. Observations present in the same cell for at least $m$ grid number are selected as neighbors of the individual to be predicted. However, the greater the value of $m$, the smaller the number of neighbors returned by parSketch method.

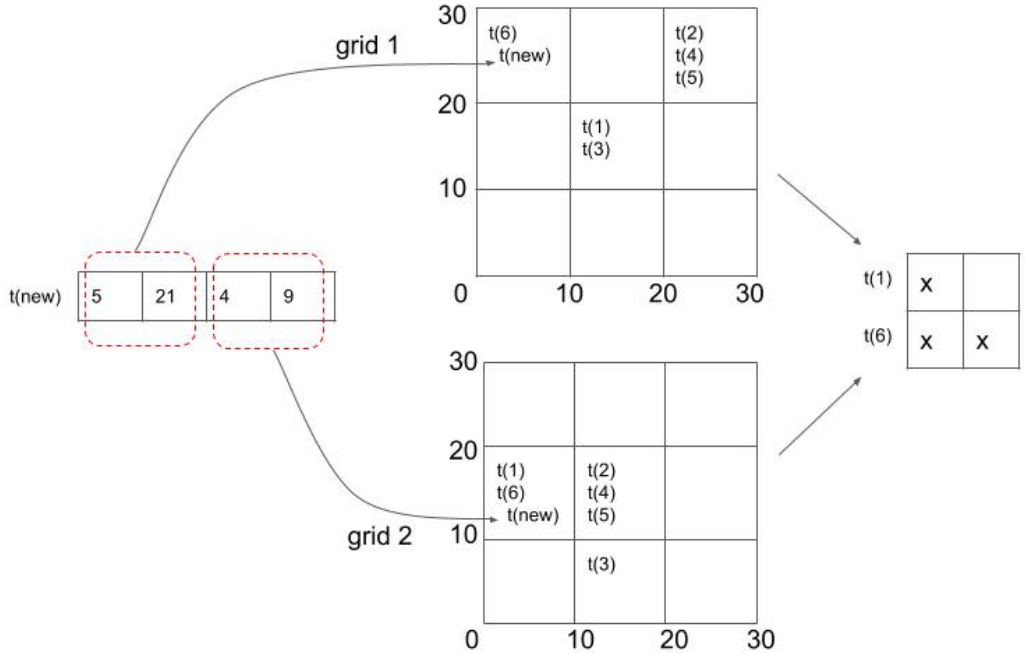



**Figure 3.** An illustrated example of grid search. For a new measure $t(new)$ and a $m$ value equals to 2, $t(6)$ will be returned as a neighbor because it is present in two grids next to $t(new)$, whereas $t(1)$ will not be considered as a neighbor

### *2.4. Evaluation strategies and method parameterization*

The data set was divided into two independent data sets: a calibration set and an independent test set . The calibration set was formed with the 14 images acquired on P1 plants and corresponding to about 650,000 spectra. The PLSDA model and parSketch-PLSDA method were both calibrated using all spectra of this calibration set.

The test set was formed with the other 14 images acquired on P2 plants (independent from P1). 1000 spectra were randomly selected in each image totalling 14,000 spectra for the test set. Spectra that could not be predicted by parSketch due to lack of neighbors were removed from the test set. In the end, the same test set were used for parSketch-PLSDA and for PLSDA.

For both methods, validation steps were performed on the calibration set in order to minimize overfitting.

To build the PLSDA model, the cross-validation step consisted of splitting the calibration data set into different blocks in order to calculate calibration and validation errors. This approach, also called k-fold validation [32,33] was carried out with five blocks repeated three times. Validation errors were then computed and led to the number of latent variables to be retained.

For parSketch-PLSDA, a parametrisation step was performed to configure the three parSketch parameters. This step was performed by analyzing distributions of returned neighbors according to two parSketch parameters: number of segments $s$ and the common minimum grids $m$. Here, the number of random vectors $v$ was set to a value of 20. Afterwards, a PLSDA model was established. The number of latent variables was optimized for a subset of the calibration set, called the validation

set. This validation set was formed with four images of the calibration set by randomly selecting 1000 spectra in each image.

In order to compare both methods, confusion matrices were obtained and percentages of precision and recall were calculated according to the following equations:

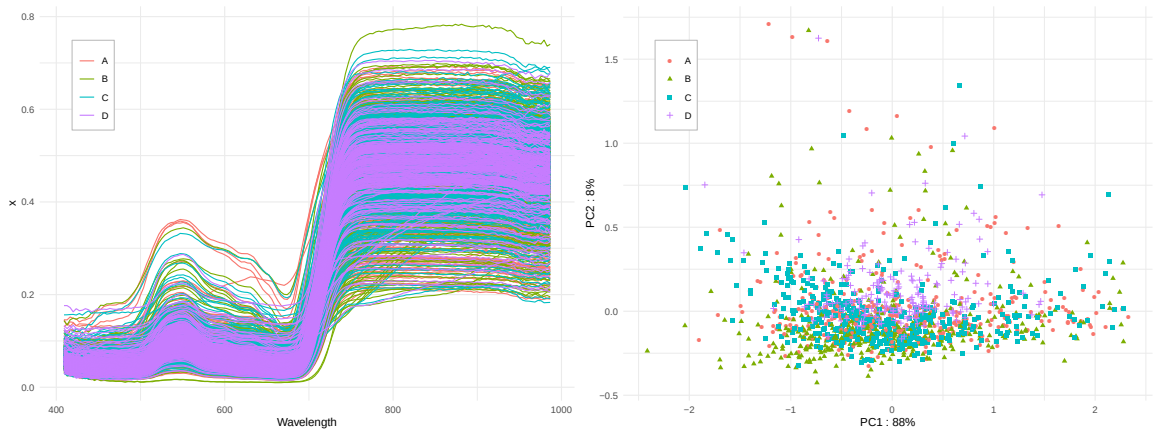
$$\text{Precision} = \frac{tp}{tp + fp} \tag{4}$$

$$\text{Recall} = \frac{tp}{tp + fn} \tag{5}$$

Where $tp$, $fp$ and $fn$ corresponded to true positives, false positives and false negatives respectively. On the one hand, for a given class, precision value assessed the predictive quality of the model based on the proportion of well-classified observations among all observations that were classified in the same corresponding class. On the other hand, recall, also called sensitivity, evaluated the number of well-classified observations compared to the total number of observations of the given class. These two criteria are complementary to evaluate the model performances. These two figures of merit were expressed as percentages. The higher the values, the better the model performance.

## 3. Results and discussion

### *3.1. Data visualization*




**(a)** **(b)**

**Figure 4.** All data set: (a) spectra, (b) score plot of the first two principal components

Reflectance spectra shown in Fig. 4a correspond to 1000 spectra per class randomly selected among all the data set. These spectra correspond to vegetation spectra [34] : specific hollows at 450 nm and 650 nm related to chlorophyll content, anthocyanin content at 550 nm; the red-edge towards 780 nm and a plateau in the near-infrared between 780 nm and 1000 nm. Besides, the main observed variability in the spectrum plot corresponds to an additive effect due to the scattering effect of the structure of the leaves. However, the number of spectra is too large to be able to describe difference between classes.

A principal component analysis was applied to these spectra. Figure 4b shows the score plot of the two first components. The first component represents 88% of the spectra variability and 8% for the second component. On these two components, scores are uniformly distributed without any evident distinction between genotypes. The exploratory study of the spectra shows that there are no outliers and that there is no distinct group on the first two components.

## 3.2. Model calibration

### 3.2.1. PLSDA

Figure 5 shows the cross-validated error rate curve for PLSDA applied to all spectra of the calibration data set. The behaviour of the curve decreases continuously according to the latent variable (LV) number.

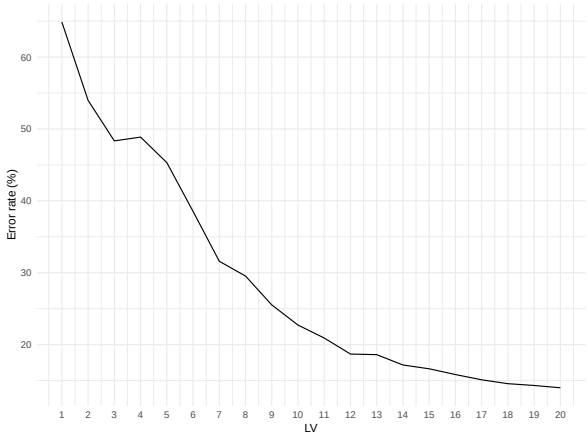



**Figure 5.** Evolution of the cross-validated error rate as a function of latent variables (LV) for the PLSDA applied to all spectra of the calibration data set

A high value of LV number generally shows the complex structure of a data set. This is expected with spectral measurements on vegetation [27]. With 16 LVs, an error rate with a value close to 12% is obtained. However, after 16 LVs the predictive performance gain is very small. Consequently, the PLSDA model is set to 16 LVs.

### 3.2.2. ParSketch-PLSDA

ParSketch parameters $s$ and $m$ are studied according to the statistical distribution of the number of returned neighbors (Fig. 6).

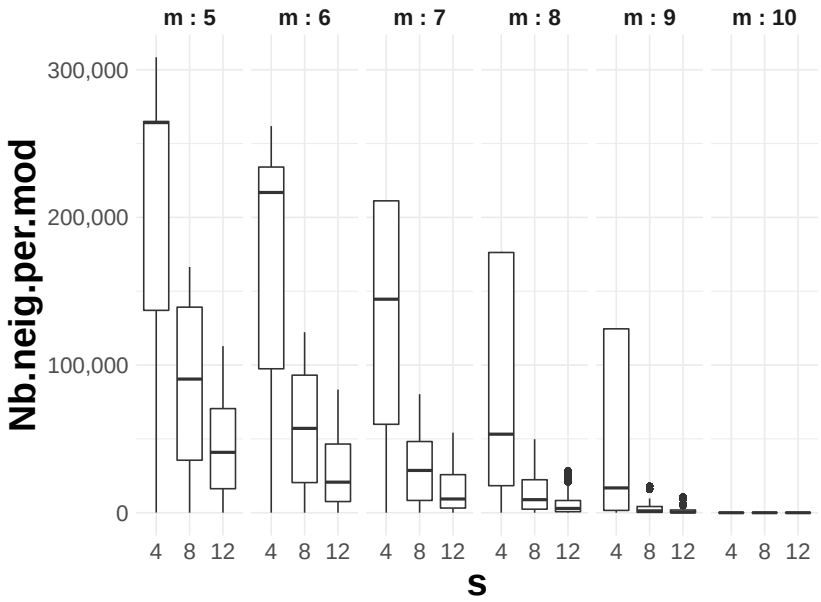



**Figure 6.** Distribution of the returned neighbors according to parSketch parameters $s$ and $m$ (number of segments and common minimum grids). Here, $v$ (number of random vectors) parameter is fixed to 20

The number of neighbors decreases to a value close to zero when parameter values ($s$,$m$) increase. Indeed, on the one hand, an increase of number of segments $s$, the number of returned neighbors will be lower. And on the other hand, by increasing the minimum number of common grids $m$, the risk of not having neighbors is high. This global trend is expected [27].

By contrast, when parameter values are low, the number of returned neighbors is high (close to 300 000 neighbors by individual to be predicted). This situation is not desirable, as it may cause problems related to computation time constraints. As a result, parameters $m$ and $s$ must be chosen to have a sufficient number of neighbors, neither too much nor too little. As several values are possible, four combinations of the parSketch parameters are selected (Table 1) to compare their model performances.

**Table 1.** Combinations of the selected parSketch parameters and the corresponding median number of returned neighbors

| **Combination** | $m$ | $v$ | $s$ | Median neighbor number |
|---|---|---|---|---|
| **(a)** | 9 | 20 | 8 | 1246 |
| **(b)** | 8 | 20 | 12 | 2903 |
| **(c)** | 7 | 20 | 12 | 9303 |
| **(d)** | 6 | 20 | 12 | 27030 |

Table 1 shows the retained values of the three parSketch parameters for these four combinations. The combination **(a)** was selected because the median number of neighbors is 1246. This low number of neighbors enables to quickly calibrate PLSDA models but it could be insufficient to have a good predictive quality. Indeed, a low median number of neighbors means that a large amount of observations do not have neighbor at all. For the combinations **(b)** and **(c)**, higher numbers of neighbors are returned, with median values of 2903 and 9303 respectively. Finally, the highest median number of neighbors returned by parSketch is chosen with the combination **(d)** with a value equal to 27030. In this case, constraints in computation time might appear. Moreover, the linear relationship between spectra and class variable of a small neighborhood might be lost.

Validation error curves for the four retained combinations for parSketch-PLSDA are shown in the figure 7.

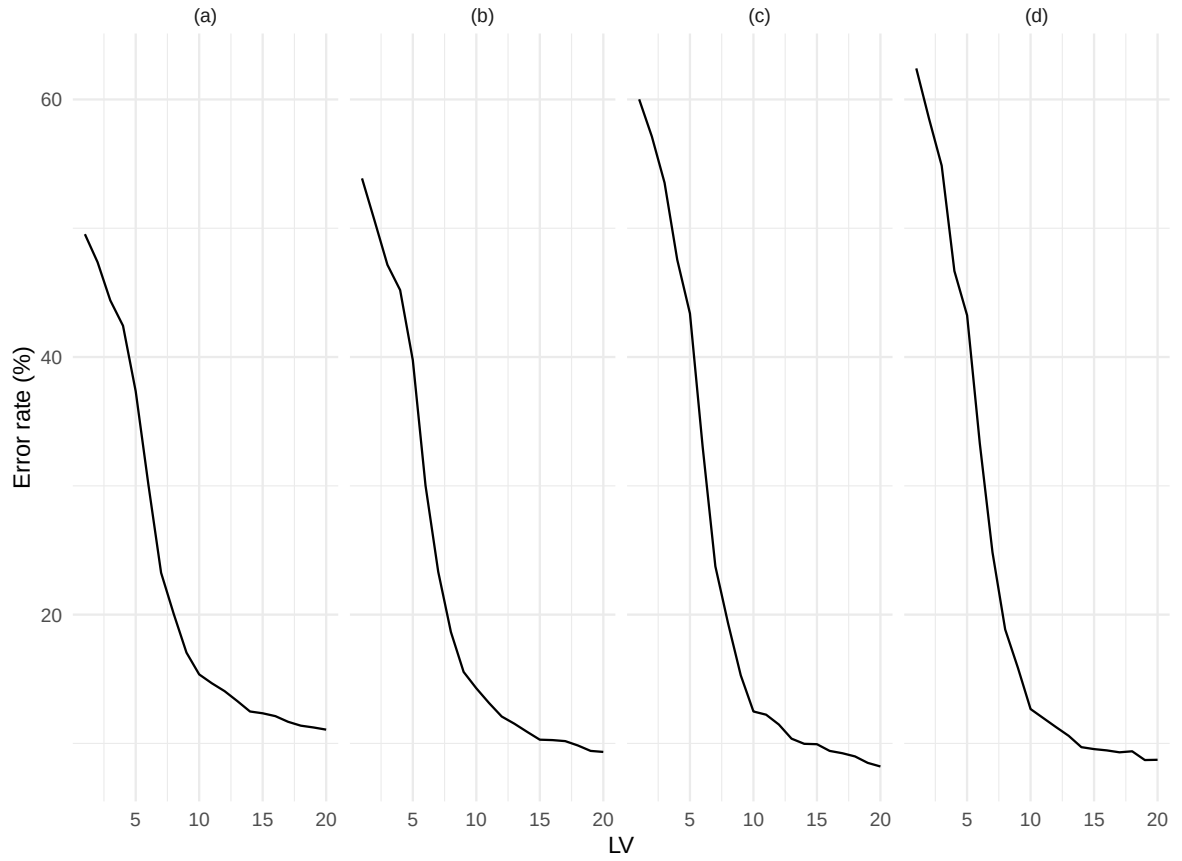



**Figure 7.** Evolution of the validation error rate as a function of latent variables for the four parameter combinations of parSketch-PLSDA

With higher error values, the combination **(a)** is less predictive than other parameter combinations. As expected, this combination having the smallest number of neighbors, the resultant model has poorer predictive performances than the three other ones.

The error curve obtained with the combination **(b)** reaches lower rates than the combination **(a)** curve. This means that the predictive capabilities of the model can be improved by slightly increasing the number of neighbors. The combination **(c)** has best predictive performance for the validation set with lowest values of classification error. The combination **(d)** has lower prediction quality than the combination **(c)** for a larger median number of neighbors per sample to be predicted (cf. Table 1).

The model with the lowest predictive quality has a high number of neighbors. This degradation reflects a non-linear aspect of the data set. By further increasing the number of returned neighbors, prediction qualities of parSketch will be close to the PLSDA method on the whole data set.

Finally, for this validation set the optimal parameter combination is the combination **(c)**. In this case, the number of latent variables is not easy to define. The number of latent variables is defined by a trade-off between the size of the model and the benefit of adding an extra dimension to the model. The number of latent variables chosen is therefore 16.

### *3.3. Model testing*

The PLSDA model has been calibrated with all the spectra of the calibration set. Then this model is applied to the test set defined previously and its prediction performances are assessed in Table 2.

**Table 2.** Confusion matrix for PLSDA (16 LVs)

| | **A** | **B** | **C** | **D** | **Recall (%)** |
|---|---|---|---|---|---|
| **A** | 3120 | 121 | 339 | 250 | **81** |
| **B** | 238 | 2812 | 619 | 154 | **74** |
| **C** | 127 | 241 | 3463 | 75 | **89** |
| **D** | 283 | 173 | 270 | 1213 | **63** |
| **Precision(%)** | **83** | **84** | **74** | **72** | |

Precision and recall values are high for all classes A, B, C and D with values ranging from 72% to 84% for precision and from 63% to 89% for recall. Genotypes A and B have the highest precision values with values of 83% and 84%, respectively. This means that 83% of spectra classified in genotype A, actually belong to genotype A. Few other genotypes are found in this class. The same argumentation holds true for 84% of genotype B spectra. For recall, genotypes A and C have the best values with 81% and 89%, respectively. For these genotypes, spectra are mainly well-classified that is infrequently assigned to other classes.

The percentage missings from recall values correspond to the prediction error for each class. The prediction error of the whole data set, corresponding to the average error, is close to 23%. It is expected to have value for the test error slightly higher than the 12% observed during calibration (see Fig. 5). This means that the calibration set samples are representative of the test set despite their independence (as mentioned above, the test set corresponds to other plants of the same genotype).

**Table 3.** Confusion matrix for parSketch-PLSDA with combination**(c)** ($m = 7$, $v = 20$, $s = 12$)

| | **A** | **B** | **C** | **D** | **Recall (%)** |
|---|---|---|---|---|---|
| **A** | 3547 | 114 | 115 | 54 | **93** |
| **B** | 40 | 3256 | 473 | 54 | **85** |
| **C** | 58 | 211 | 3590 | 47 | **92** |
| **D** | 51 | 112 | 260 | 1516 | **78** |
| **Precision(%)** | **96** | **88** | **81** | **91** | |

Table 3 shows the parSketch-PLSDA prediction performance by giving percentages of precision and recall for each genotype. Genotypes A and D have the highest precision values with values of 96% and 91% respectively. Besides, genotypes A and C have the highest recall values with values of 93% and 92% respectively. Genotype D has low recall values with both methods (63% for PLSDA and 78% for parSketch-PLSDA). This is probably due to the under-representation in the data set which may degrade the model calibration. Indeed, only two images were acquired per plant of genotype D compared to four images for the other genotypes.

Finally, overall recall and precision values have increased by almost 10% with parSketch-PLSDA compared to PLSDA applied on all spectra. Consequently, the model prediction error decreases to a value of 13%. This implies that parSketch-PLSDA model performs better than the reference discriminant strategy. This improvement in the classification results demonstrates the advantage of using a limited number of neighbors to create a model.

As the methods used are locally linear, this improvement confirms the hypothesis that with a limited number of neighbors, the problem becomes linear. The prediction improvement obtained with parSketch method highlights the presence of non-linear relationships between spectra and a class variable in the whole data set. which can be encountered when building a large spectral database.

## 4. Conclusion

In this study, we propose to use parSketch-PLSDA method to analyze a massive amount of spectral data in order to discriminate different sunflower varieties. This method was compared with PLSDA, a reference method of discrimination.

For this purpose, we formed a spectral database from an experimental design containing four sunflower varieties. Part of the data set, containing 650,000 spectra, was used for the model calibration. Another part was used to form an independent test set of 14,000 spectra. We compared the two classification strategies on the same calibration and test data sets.

For both methods, classification results are encouraging and confirm the interest of VIS-NIR spectroscopy for variety discrimination. Results showed that parSketch-PLSDA method outperforms PLSDA by improving prediction qualities by 10%. The use of the parSketch procedure in the exploitation of massive spectral data is confirmed and shows the interest of using a close neighborhood of the spectra to be predicted.

It would be interesting to test such methods on a larger number of genotypes. This increase in the spectral database can potentially lead to an increase in complexity hence reducing the data set quality. Therefore, it would be interesting, in perspective, to evaluate other methods dealing with non-linearity.

In the framework of plant breeding, hyperspectral imaging or field microspectrometers as tools for high-throughput plant phenotyping could be considered in real time with this method. In an applicative aspect, parSketch procedure is parallelizable, which shows the possibility of fast real-time prediction of a large amount of data. We used parSketch-PLSDA on spectral data for close-range plant phenotyping. Other applications to plant breeding (disease, biotic/abiotic stress) or other applications related to precision agriculture could be considered. More generally, this method can be applied to any other application in analytical chemistry or metabolomics.

## Acknowledgment

This work has benefited from a financial support from the "Programme Investissement d'Avenir" fund, managed by the National Research Agency under the reference ANR-16-CONV-0004 and

**Funding:** This work was conducted within the OPTIPAG Project supported by the grant ANR-16-CE04-0010 from the French Agence Nationale de la Recherche.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Mahlein, A.K. Plant disease detection by imaging sensors–parallels and specific demands for precision agriculture and plant phenotyping. *Plant disease* **2016**, *100*, 241–251. Publisher: Am Phytopath Society.
2. Tripodi, P.; Massa, D.; Venezia, A.; Cardi, T. Sensing technologies for precision phenotyping in vegetable crops: current status and future challenges. *Agronomy* **2018**, *8*, 57. Publisher: Multidisciplinary Digital Publishing Institute.
3. Awada, L.; Phillips, P.W.B.; Smyth, S.J. The adoption of automated phenotyping by plant breeders. *Euphytica* **2018**, *214*, 148. doi:10.1007/s10681-018-2226-z.
4. Chawade, A.; van Ham, J.; Blomquist, H.; Bagge, O.; Alexandersson, E.; Ortiz, R. High-throughput field-phenotyping tools for plant breeding and precision agriculture. *Agronomy* **2019**, *9*, 258. Publisher: Multidisciplinary Digital Publishing Institute.
5. Shakoor, N.; Lee, S.; Mockler, T.C. High throughput phenotyping to accelerate crop breeding and monitoring of diseases in the field. *Current opinion in plant biology* **2017**, *38*, 184–192. Publisher: Elsevier.
6. Dhondt, S.; Wuyts, N.; Inzé, D. Cell to whole-plant phenotyping: the best is yet to come. *Trends in Plant Science* **2013**, *18*, 428–439. doi:10.1016/j.tplants.2013.04.008.
7. Mutka, A.M.; Fentress, S.J.; Sher, J.W.; Berry, J.C.; Pretz, C.; Nusinow, D.A.; Bart, R. Quantitative, image-based phenotyping methods provide insight into spatial and temporal dimensions of plant disease. *Plant Physiology* **2016**, p. pp.00984.2016. doi:10.1104/pp.16.00984.
8. Vigneau, N.; Ecarnot, M.; Rabatel, G.; Roumet, P. Potential of field hyperspectral imaging as a non destructive method to assess leaf nitrogen content in wheat. *Field Crops Research* **2011**, *122*, 25–31. doi:10.1016/j.fcr.2011.02.003.

9. Jay, S.; Gorretta, N.; Morel, J.; Maupas, F.; Bendoula, R.; Rabatel, G.; Dutartre, D.; Comar, A.; Baret, F. Estimating leaf chlorophyll content in sugar beet canopies using millimeter-to centimeter-scale reflectance imagery. *Remote Sensing of Environment* **2017**, *198*, 173–186. Publisher: Elsevier.
10. Lu, J.; Ehsani, R.; Shi, Y.; de Castro, A.I.; Wang, S. Detection of multi-tomato leaf diseases ( late blight , target and bacterial spots ) in different stages by using a spectral-based sensor. *Scientific Reports* **2018**, *8*, 2793. Number: 1 Publisher: Nature Publishing Group, doi:10.1038/s41598-018-21191-6.
11. Yu, K.; Anderegg, J.; Mikaberidze, A.; Karisto, P.; Mascher, F.; McDonald, B.A.; Walter, A.; Hund, A. Hyperspectral Canopy Sensing of Wheat Septoria Tritici Blotch Disease. *Frontiers in Plant Science* **2018**, *9*. Publisher: Frontiers, doi:10.3389/fpls.2018.01195.
12. Behmann, J.; Steinrücken, J.; Plümer, L. Detection of early plant stress responses in hyperspectral images. *ISPRS Journal of Photogrammetry and Remote Sensing* **2014**, *93*, 98–111.
13. Christensen, L.K.; Upadhyaya, S.K.; Jahn, B.; Slaughter, D.C.; Tan, E.; Hills, D. Determining the Influence of Water Deficiency on NPK Stress Discrimination in Maize using Spectral and Spatial Information. *Precision Agriculture* **2005**, *6*, 539–550. doi:10.1007/s11119-005-5643-7.
14. Yan, H.; Siesler, H.W. Hand-held near-infrared spectrometers: State-of-the-art instrumentation and practical applications. *NIR news* **2018**, *29*, 8–12. Publisher: SAGE Publications Sage UK: London, England.
15. Beć, K.B.; Grabska, J.; Siesler, H.W.; Huck, C.W. Handheld near-infrared spectrometers: Where are we heading? *NIR news* **2020**, *31*, 28–35. Publisher: SAGE Publications Sage UK: London, England.
16. Mishra, P.; Lohumi, S.; Ahmad Khan, H.; Nordon, A. Close-range hyperspectral imaging of whole plants for digital phenotyping: Recent applications and illumination correction approaches. *Computers and Electronics in Agriculture* **2020**, *178*, 105780. doi:10.1016/j.compag.2020.105780.
17. Fiorani, F.; Schurr, U. Future Scenarios for Plant Phenotyping. *Annual Review of Plant Biology* **2013**, *64*, 267–291. doi:10.1146/annurev-arplant-050312-120137.
18. Szymańska, E. Modern data science for analytical chemical data – A comprehensive review. *Analytica Chimica Acta* **2018**, *1028*, 1–10. doi:10.1016/j.aca.2018.05.038.
19. Wold, S.; Sjöström, M.; Eriksson, L. PLS-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems* **2001**, *58*, 109–130.
20. Mark, H.; Workman, J. *Chemometrics in spectroscopy*; Elsevier/Academic Press: Amsterdam, 2007. OCLC: 255877127.
21. Dardenne, P.; Sinnaeve, G.; Baeten, V. Multivariate Calibration and Chemometrics for near Infrared Spectroscopy: Which Method? *Journal of Near Infrared Spectroscopy* **2000**, *8*, 229–237. doi:10.1255/jnirs.283.
22. Bertran, E.; Blanco, M.; Maspoch, S.; Ortiz, M.C.; Sánchez, M.S.; Sarabia, L.A. Handling intrinsic non-linearity in near-infrared reflectance spectroscopy. *Chemometrics and Intelligent Laboratory Systems* **1999**, *49*, 215–224. doi:10.1016/S0169-7439(99)00043-X.
23. Ni, W.; Nørgaard, L.; Mørup, M. Non-linear calibration models for near infrared spectroscopy. *Analytica Chimica Acta* **2014**, *813*, 1–14. doi:10.1016/j.aca.2013.12.002.
24. Pérez-Marín, D.; Garrido-Varo, A.; Guerrero, J.E. Non-linear regression methods in NIRS quantitative analysis. *Talanta* **2007**, *72*, 28–42. doi:10.1016/j.talanta.2006.10.036.
25. Davrieux, F.; Dufour, D.; Dardenne, P.; Belalcazar, J.; Pizarro, M.; Luna, J.; Londoño, L.; Jaramillo, A.; Sanchez, T.; Morante, N.; Calle, F.; Lopez-Lavalle, L.B.; Ceballos, H. LOCAL Regression Algorithm Improves near Infrared Spectroscopy Predictions When the Target Constituent Evolves in Breeding Populations. *Journal of Near Infrared Spectroscopy* **2016**, *24*, 109–117. doi:10.1255/jnirs.1213.
26. Naes, T.; Isaksson, T.; Kowalski, B. Locally weighted regression and scatter correction for near-infrared reflectance data. *Analytical Chemistry* **1990**, *62*, 664–673. doi:10.1021/ac00206a003.
27. Metz, M.; Lesnoff, M.; Abdelghafour, F.; Akbarinia, R.; Masseglia, F.; Roger, J.M. A "big-data" algorithm for KNN-PLS. *Chemometrics and Intelligent Laboratory Systems* **2020**, *203*, 104076. doi:10.1016/j.chemolab.2020.104076.
28. Core Team, R. R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. *Available* **2013**.
29. Barker, M.; Rayens, W. Partial least squares for discrimination. *Journal of Chemometrics* **2003**, *17*, 166–173. doi:10.1002/cem.785.

30. Fisher, R.A. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics* **1936**, *7*, 179–188. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-1809.1936.tb02137.x, doi:https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.
31. Lesnoff, M.; Metz, M.; Roger, J. Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data **2020**. p. 13.
32. Wold, S. Cross-Validatory Estimation of the Number of Components in Factor and Principal Components Models. *Technometrics* **1978**, *20*, 397–405. Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality], doi:10.2307/1267639.
33. Camacho, J.; Ferrer, A. Cross-validation in PCA models with the element-wise k-fold (ekf) algorithm: theoretical aspects. *Journal of Chemometrics* **2012**, *26*, 361–373. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.2440, doi:https://doi.org/10.1002/cem.2440.
34. Xu, J.L.; Gobrecht, A.; Héran, D.; Gorretta, N.; Coque, M.; Gowen, A.A.; Bendoula, R.; Sun, D.W. A polarized hyperspectral imaging system for in vivo detection: Multiple applications in sunflower leaf analysis. *Computers and Electronics in Agriculture* **2019**, *158*, 258–270. doi:10.1016/j.compag.2019.02.008.

# Chapitre 6

## Article 6 : Roboost-PLS2-R : An extension of Roboost-PLSR method for multi-responses

Référence (article soumis) :

Maxime Metz, Maxime Ryckewaert, Silvia Mas Garcia, Ryad Bendoula, Matthieu Lesnoff, and Jean-Michel Roger. Roboost-PLS2-R : An extension of Roboost-PLSR method for multi-responses. Chemometrics and Intelligent Laboratory Systems,XXXX(under revision)

[2] ChemHouse Research Group, Montpellier, France

# Graphical Abstract

**Roboost-PLS2-R : An extension of RoBoost-PLSR method for multi-responses**

Maxime Metz, Maxime Ryckewaert, Silvia Mas Garcia, Ryad Bendoula, Matthieu Lesnoff, Jean-Michel Roger

# Highlights

**Roboost-PLS2-R : An extension of RoBoost-PLSR method for multi-responses**

Maxime Metz, Maxime Ryckewaert, Silvia Mas Garcia, Ryad Bendoula, Matthieu Lesnoff, Jean-Michel Roger

- An extension of the RoBoost-PLSR method was proposed.
- This extension enables to deal with outliers in a multi-responses context.
- The new approach was tested on real and simulated data sets.
- This extension presents similar performances to PLSR model calibrated without outliers.

# Roboost-PLS2-R : An extension of RoBoost-PLSR method for multi-responses

Maxime Metz[a,b], Maxime Ryckewaert[a,b], Silvia Mas Garcia[a,b], Ryad Bendoula[a,b], Matthieu Lesnoff[c,b], Jean-Michel Roger[a,b]

[a] *ITAP Univ Montpellier INRAE Institut Agro Montpellier France*
[b] *ChemHouse Research Group Montpellier France*
[c] *SELMET Univ Montpellier CIRAD INRAE Montpellier SupAgro Montpellier France*

## Abstract

Recently, a novel robust PLSR method was developed to address the problem of outliers in the data. In this paper, an extension of this method, called Roboost-PLS2-R is proposed to predict multi-response variables. Robustness and efficiency of this new approach have been validated on two simulated data sets and one real data set containing different outlier scenarios. Its performance was also compared with reference methods (PLS2-R and RSIMPLS) for predicting multi-response variables. Results confirm that Roboost-PLS2-R greatly reduces prediction errors when data contain outliers. Prediction performances of Roboost-PLS2-R are close to the optimal model (PLS2-R) calibrated without outliers and also to RSIMPLS method. This method seems to be a reliable and a competitive robust regression tool for predicting multi-response variables.

*Keywords:* Robust regression methods, outliers, multi-responses, multivariate data analysis
*PACS:* 0000, 1111
*2000 MSC:* 0000, 1111

## 1. Introduction

Partial Least Square Regression (PLSR) [1] is a common data analysis method and a well-established tool in chemometrics. PLSR calculates a linear relationship between explanatory variables ($\mathbf{X}$) and response variables ($\mathbf{Y}$). PLSR can be used to predict one response (PLS1) or several responses (PLS2). PLSR is particularly useful for processing

high-dimensional data, especially when the number of explanatory variables exceeds the number of samples. This method is widely used in analytical chemistry for predicting constituent concentrations of a sample based on its spectrum obtained by spectroscopic techniques, such as near-infrared (NIR) spectroscopy, Fluorescence spectroscopy and ultraviolet (UV) spectroscopy. The PLSR model is known to be affected by the presence of atypical observations (outliers) in the data set. Outliers can negatively affect the calibration of PLSR models. To deal with outliers, several robust PLSR methods were proposed in the literature [2–12]. These methods were particularly developed to deal with outliers when the response matrix is uni-dimensional [13] (PLS1-R). However, robust methods that address the case of multi-responses (PLS2) are few. Among them, RSIMPLS is one of the most used method [14]. RSIMPLS proposes to robustly estimate the cross-covariance matrix $\mathbf{C_{xy}}$ and the empirical covariance matrix $\mathbf{C_x}$ used in SIMPLS algorithm. For this, a robust principal component analysis (ROBPCA) is performed on the concatenated data matrix of $\mathbf{X}$ and $\mathbf{Y}$. RSIMPLS uses additional information from the previous ROBPCA step to perform a reweighted multiple linear regression.

Recently, a new robust method called RoBoost-PLSR has been developed [15]. RoBoost-PLSR aims to determine the measure of relevance of the samples for PLSR model calibration. Indeed, in practical cases, the samples of a database are not defined as outliers, i.e. not relevant for the calibration of a PLSR model. RoBoost-PLSR proposes to calculate a weight on each latent variable to define the relevance of the samples. The relevance measurement is defined according to three criteria calculated for each latent variable ($X$-residuals, $Y$-residuals, leverage). This method has proven to be effective for outliers in both $\mathbf{Y}$ and $\mathbf{X}$. However, this algorithm was only developed for a one-dimensional PLSR response variable (PLS1). This paper contributes to the RoBoost-PLSR method which will be able to manage outliers in a multiple response context.

The first section introduces the extension of RoBoost-PLSR named RoBoost-PLS2-R and the associated algorithm. The following section presents the data and the methods used to evaluate and compare the predictive ability of RoBoost-PLS2-R. Finally, the prediction performances of RoBoost-PLS2-R and its comparison with standard methods are shown in the last section.

## 2. Notations

Capital bold characters will be used for matrices, *e.g.* $\mathbf{X}$; small bold characters for column vectors, *e.g.* $\mathbf{x}_j$ will denote the $j^{th}$ column of $\mathbf{X}$; row vectors will be denoted by the transpose notation, *e.g.* $\mathbf{x}_i^{\mathrm{T}}$ will denote the $i^{th}$ row of $\mathbf{X}$; italicised characters will be used for scalars, *e.g.* matrix elements $x_{ij}$ or indices $i$. Constant scalars will be denoted with italicised characters, *e.g.* number of samples $n$. $\mathbb{1}$ will represent a column vector of ones, of proper dimension. *med* defines the median. $\mathbf{X}$ and $\mathbf{Y}$ are the spectral and the responses matrices. $g$ is the weight function. $\mathbf{D}$ is the matrix of sample weights where the diagonal of the matrix is the sample weight and the other terms are zero.

## 3. RoBoost-PLSR extension for multi-responses

### *3.1. Algorithm*

The new algorithm allowing an extension in a multi-response context is the following :

**Algorithm** RoBoost-PLSR for $K$ LV

For a definite number of $K$ latent variables, the algorithm proceeds as described below :

1: Initialisation step

$$k = 1$$

$$\mathbf{D} = diag(d_1, d_2, ..., d_n) \text{ with } d_i = \frac{1}{n}$$

2: Center the data :

$$\mathbf{X}_k = \mathbf{X} - \mathbb{1}\mathbb{1}^{\mathrm{T}}\mathbf{D}\mathbf{X}$$

$$\mathbf{Y}_k = \mathbf{Y} - \mathbb{1}\mathbb{1}^{\mathrm{T}}\mathbf{D}\mathbf{Y}$$

3: Define $\mathbf{u}_k$ as an arbitrary column of $\mathbf{Y}$

4: Calculate one weighted latent variable NIPALS :

$$\mathbf{w}_k = \frac{\mathbf{X}_k^{\mathrm{T}}\mathbf{D}\mathbf{u}_k}{||\mathbf{X}_k^{\mathrm{T}}\mathbf{D}\mathbf{u}_k||}$$

$$\mathbf{t}_k = \mathbf{X}_k\mathbf{w}_k$$

$$\mathbf{p}_k = \frac{\mathbf{X}_k^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}{\mathbf{t}_k^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}$$

$$\mathbf{q}_k = \frac{\mathbf{Y}_k^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}{\mathbf{t}_\mathbf{k}^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}$$

$$c_k = \frac{\mathbf{u}_k^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}{\mathbf{t}_\mathbf{k}^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}$$

5: Derive ($\mathbf{F}$), ($\mathbf{E}$), ($\mathbf{l}$) :

$$\mathbf{E} = \mathbf{X}_k - \mathbf{t}_k\mathbf{p}_k^{\mathrm{T}}$$

$$\mathbf{F} = \mathbf{Y}_k - \mathbf{t}_k\mathbf{q}_k$$

$$\mathbf{l} = \mathbf{t}_k$$

6: Update the weights for each $i \in [1, n]$ sample :

$$d_i = \frac{1}{n} \times g(||\mathbf{e}_i||, \alpha) \times \prod_{j=1}^{m} g(f_{ij}, \beta), \times g(l_i, \gamma)$$

7: Return to (step (2) for k = 1, otherwise return to step (4) ) until convergence of successive $c$'s.

8: Deflation step

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^{\mathrm{T}}$$

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k - \mathbf{t}_k \mathbf{q}_k$$

$$\mathbf{u}_{k+1} = \mathbf{Y}_k \mathbf{q}_k$$

set $k = k + 1 \rightarrow$ then go to step (4)

The regression coefficients resulting for $K$ latent variables are estimated as follows :

$$\mathbf{B} = \mathbf{R}\mathbf{c}^{\mathrm{T}}$$

With $\mathbf{R}$ :

$$\mathbf{R} = \mathbf{W}(\mathbf{P}^{\top}\mathbf{W})^{-1}$$

### *3.2. Theoretical discussions*

The algorithm RoBoost-PLS2-R have similar properties to the algorithm proposed in [15], but also new properties :

— The RoBoost-PLS2-R framework is designed foremost to facilitate the leverage measurement. Leverage is defined as the distance to the centre of the model. In usual strategies, to define distances between the model centre and individuals, different metrics can be used. Euclidean or Mahalanobis distances between scores and the model

centre are strategies commonly used in chemometrics. However, in the case of a Euclidean distance, the latest LVs could have a minor contribution to the leverage value. This is due to the decreasing magnitude of scores. Nevertheless, the predictive potential of these latest LVs is not necessarily less. In the case of a Mahalanobis distance, contributions of all LVs become equal in the computation of the leverage value. This can be also detrimental, since the predictive potentials of the LVs are most usually uneven. Considering these limitations, RoBoost-PLSR proposes to estimate the sample leverage for each latent variable. This avoids the need to define specific metrics for the leverage calculation.

— The proposed method takes into account X-residuals. Usually only Y-residuals are considered in robust PLS approaches. The inclusion of these residuals provides additional information that cannot be expressed by leverage and Y-residuals alone.

— The algorithm proposed in this article provides regression coefficients. This makes the constructed RoBoost-PLSR models more easily interpretable. Contrary to the first algorithm proposed in [15], the rotation matrix $\mathbf{R}$ used to estimate the regression coefficients can be estimated. This is due to data centring which is only done for the first model with a single latent variable. In the previous algorithm, repeated centring of $\mathbf{X}$ and $\mathbf{Y}$ matrices led to a bias which made it impossible to estimate the rotation matrix.

— Like PLSR, RoBoost-PLSR makes it possible to deduce any of the 1 to $K$ LVs models from the calibration of a single $K$ LVs model. This preserves the operability during validation and parameterisation process of the RoBoost-PLSR method.

— The algorithm proposed in [15], determines the convergence with $q$. However, $\mathbf{q}$ is multidimensional when $\mathbf{Y}$ is multidimensional. In the new algorithm convergence estimation is facilitated by using $c$ which is a scalar when responses matrix $\mathbf{Y}$ is multidimensional.

— The weights of the sample according to the $\mathbf{Y}$-residuals are the product of the estimated weights for each $\mathbf{Y}$-variable. A specific sample weight for each residual of each $\mathbf{Y}$ variable is calculated and

then multiply them to give an overall weight. This strategy enables sample weights to be estimated in a way that is appropriate to the multivariate nature of $\mathbf{Y}$. This strategy takes in consideration the fact that $\mathbf{Y}$ variables may have different variances. If this aspect is not taking into account, some outliers could be considered as inliers by the method. For instance, atypical samples on a specific variable of $\mathbf{Y}$ can mask the outliers of other columns of $\mathbf{Y}$ which present a lower variability. This strategy also allows a fast operation by applying the bisquare function on each column of $Y$-residuals matrix for each LV according to the $\beta$ hyperparameter. Finally, the global weights associated with $Y$-residuals are defined as a product of each weight calculated on the $Y$-residual. This strategy of combining weights is a commonly used strategy. It is basically used to combine the weights calculated according to the three criteria ($X$-residuals, $Y$-residuals, leverage) in RoBoost-PLSR. However, different strategies are possible. Like calculating the Mahalanobis distances on $\mathbf{Y}$ or making a combination of weights different from the product. In particular, it is possible to perform a sum of weights, so that the weighting strategy can eliminate individuals who only have weights at 0 for each criterion.

— In this article, the weight function $g$ is the bisquare function :

$$B(z_i) = (1 - z_i^2)^2 \text{ for } |z_i| < 1 \text{ and } B(z_i) = 0 \text{ for } |z_i| > 1$$

with $z_i$ :

$$\frac{x_i}{c \times med(\mathbf{x})}$$

However, any weight function can be considered and tested in order to improve the algorithm to obtain better predictive capacity.

## 4. Materials and methods

### *4.1. Simulated Data*

To evaluate the performance of Roboost-PLS2-R in comparison with standard PLS2-R and RSIMPLS, two simulations were performed. The first simulation represents the $Y$-outlier case and the second simulation the $X$-outlier case. For each simulation, 1000 samples were generated according to the framework proposed by [16]. Among these samples, 200 outliers

were generated. The spectral signatures used for the simulations were the spectral signatures of water, ethanol and glucose estimated in [16]. Using this approach, the matrix of explanatory variables ($\mathbf{X}$) was generated by :

$$\mathbf{X} = \mathbf{T_u}\mathbf{P_u} + \mathbf{T_d}\mathbf{P_d} + \mathbf{E} \tag{1}$$

And the relationship $f$ between $\mathbf{X}$ and $\mathbf{Y}$ by :

$$\mathbf{Y} = f(\mathbf{T_u}) + \mathbf{F} \tag{2}$$

Where $\mathbf{P_u}$ and $\mathbf{P_d}$ are spectral signatures in the useful space and the detrimental space. $\mathbf{T_u}$ and $\mathbf{T_u}$ are their associated contributions. The $\mathbf{E}$ and $\mathbf{F}$ matrices are defined as gaussian noises of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

The parameters of the simulations are represented in tables (Table 1 and Table 2) where differences between simulated inliers and outliers were highlighted in bold in the tables.

#### *4.1.1. Simulation 1, Y-outliers*

The $Y$-outliers were defined by their relationship $f$ between $\mathbf{X}$ and $\mathbf{Y}$. All other simulation parameters were common between inliers and outliers. The construction of the simulated data set 1 is represented in table 1.

TABLE 1: The different choices in the simulation 1

| | Inliers | Outliers |
|---|---|---|
| $\mathbf{P}_u$ | Pure spectrum of glucose | |
| $\mathbf{T}_u$ | Folded-normal distribution | |
| $\mathbf{P}_d$ | Pure spectrum of water<br>Pure spectrum of ethanol<br>Spectrum of water-ethanol Interaction<br>10 Artificial spectra | |
| $\mathbf{T}_d$ | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution | |
| $\mathbf{E}$ | Gaussian distribution | |
| $f$ | $Y_1 = 10 * T_{ethanol}$<br>$Y_2 = 10 * T_{glucose}$<br>$Y_3 = 10 * T_{water}$ | $Y_1 = 10 * T_{ethanol}$<br>$\mathbf{Y_2 = -10 * T_{glucose}}$<br>$Y_3 = 10 * T_{water}$ |
| $\mathbf{F}$ | Gaussian distribution | |

### *4.1.2. Simulation 2, X-outliers*

The $X$-outliers were defined by others artificial spectral signatures. These signatures correspond to minority compounds. All other simulation parameters were common between inliers and outliers. The simulation is represented in table 2.

TABLE 2: The different choices in the simulation 2

| | Inliers | Outliers |
|---|---|---|
| $\mathbf{P}_u$ | Pure spectrum of glucose | |
| $\mathbf{T}_u$ | Folded-normal distribution | |
| $\mathbf{P}_d$ | Pure spectrum of water<br>Pure spectrum of ethanol<br>Spectrum of water-ethanol Interaction<br>10 Artificial spectra | Pure spectrum of water<br>Pure spectrum of ethanol<br>Spectrum of water-ethanol Interaction<br>10 Artificial spectra<br>**10 Artificial spectra** |
| $\mathbf{T}_d$ | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution | Folded-normal distribution<br>Folded-normal distribution<br>Product between $T_{water}$ and $T_{ethanol}$<br>Folded-normal distribution<br>**Folded-normal distribution** |
| $\mathbf{E}$ | Gaussian distribution | |
| $f$ | $Y_1 = 10 * T_{ethanol}$<br>$Y_2 = 10 * T_{glucose}$<br>$Y_3 = 10 * T_{water}$ | $Y_1 = 10 * T_{ethanol}$<br>$Y_2 = 10 * T_{glucose}$<br>$Y_3 = 10 * T_{water}$ |
| $\mathbf{F}$ | Gaussian distribution | |

### *4.2. Real data set*

The real data set was formed by 261 spectra of raw cow milk collected from farms in Wallonia in 2014 and 2015. Spectra were recorded over a spectral range 397-4000 cm-1 with a resolution of 4 cm-1 by using a FTIR spectrometer (Delta LactoScope, PerkinElmer). For each sample, chemical measurements were performed to obtain two-responses variable : fat content and protein content. Fact and Protein content were determined in accordance with reference methods "ISO 1211 :2010 [IDF 1 :2010]" and "ISO 8968-1 :2014 [IDF 20-1 :2014]", respectively. This database is particularly interesting because it contains missing data whose values have been replaced by 0.

#### *4.2.1. Evaluation strategies*

Roboost-PLS2-R was evaluated and compared with two standard regression algorithms : PLS2-R and RSIMPLS.

In the case of the simulations, the 1000 samples were divided into two groups : 800 for calibration and 200 for validation. The reference method in terms of prediction performance was PLS2-R calibrated without outliers. For the real data set, calibration set was composed of 209 samples. The validation was conducted on 52 samples. These samples were selected from a study of the data in order to represent the samples as well as possible without containing potential outliers. The reference method in terms of prediction performance was RSIMPLS.

The method performance was evaluated according to the validation sets and Root Mean Square Error of Prediction (RMSEP) as a figure of merit. Only the results achieved using the optimal parameters (*i.e.* the parameters that provide the minimum value of the RMSEP) of RoBoost-PLS2-R and RSIMPLS were presented.

The evaluation strategy also aimed to assess the weights attributed to each sample. Indeed, The RoBoost-PLS2-R method allows the visualisation of the weight given to each sample for each LV. In this work, these weights that best approaches the minimum RMSEP value of the responses were evaluated.

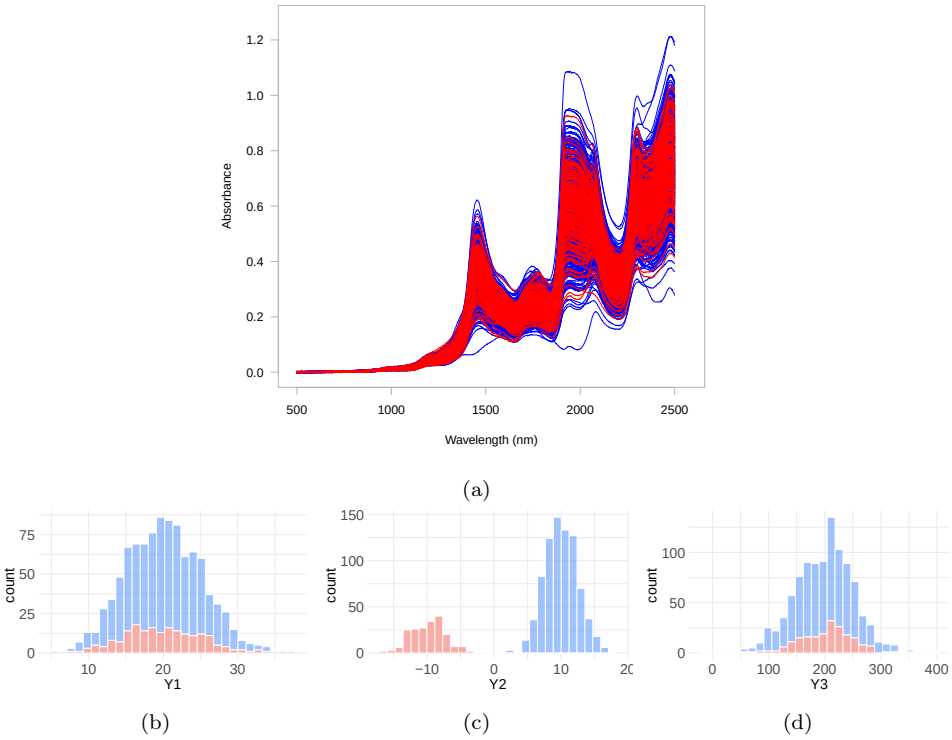
### *4.3. Software*

PLS2-R was performed with "rnirs" and RoBoost-PLS2-R is available RoBoost-PLSR functions available in R. RSIMPLS was performed using the function of the LIBRA package available in MALTLAB.

## 5. Results and discussions

### *5.1. Simulation set 1*

#### *5.1.1. Data visualisation*




(a) (b) (c) (d)

FIGURE 1: Graphical representation of simulation 1 : (a) spectral data (b) value distribution of Y1 response variable (c) value distribution of Y2 response variable (d) value distribution of Y3 response variable. Outliers are shown in red and inliers in blue.

Figure 1 shows the graphical representation of simulation 1. From the spectra plot (Figure 1a), it can be seen that is difficult to identify outliers (in red) from a simple visual inspection. In this case, the outliers were defined by a distinct relation $f$ on one of the response variables (see Table 1). Therefore, no spectral difference between the two groups is expected. From the plot of value distributions of the response variables (see Figure 1b,c,d) it can be observed that Y1 and Y3 variables present the same distribution for both outliers and inliers. However, different distribution for

these two groups is presented in Y2 variable. Moreover, the variances of Y1 are smaller than the variance of Y3.

### *5.1.2. Method evaluation*

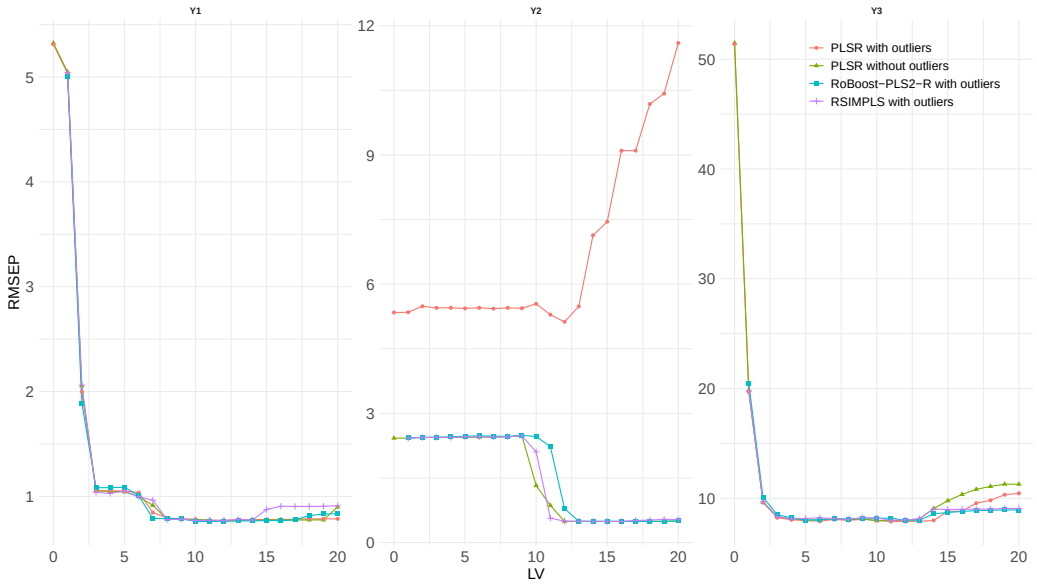
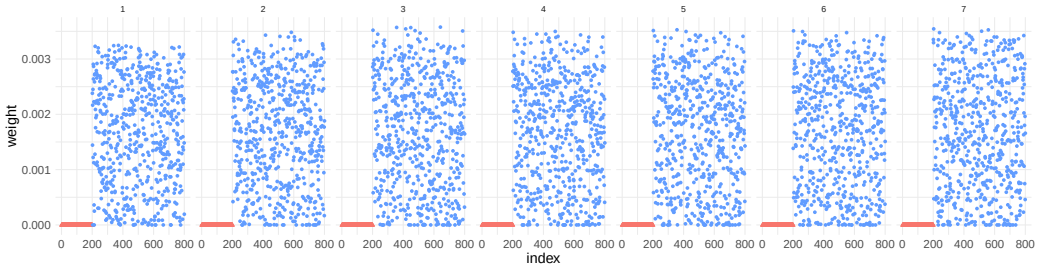
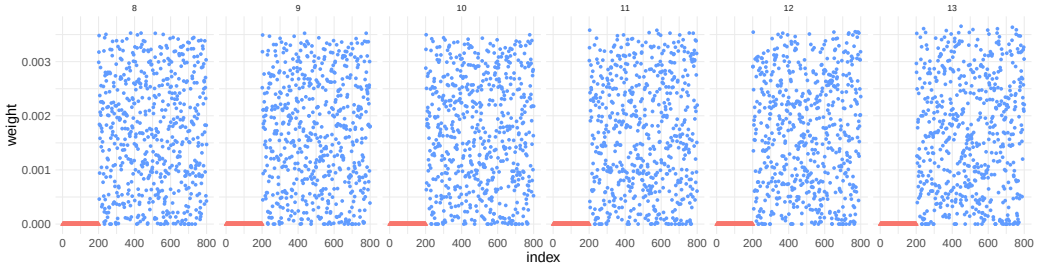



FIGURE 2: Evolution of the RMSEP as a function of the number of latent variables for the PLS2-R with and without outliers, RSIMPLS and RoBoost-PLS2-R for the simulation 1 set

Figure 2 shows the prediction performances for each method and response variable Y on the basis of simulation 1. For the variables Y1 and Y3, the error curves obtained by PLS2-R with and without outliers, RSIMPLS and RoBoost-PLS2-R are similar. This is due to the fact that outliers are only atypical on Y2 and hence, no impact on the Y1 and Y3 predictions is expected. For the variables Y2 the error curves obtained by PLS2-R with and without outliers are different. The PLS2-R model calibrated with outliers perform poorly in inliers prediction. The prediction performance of RSIMPLS is close to the PLS2-R without outliers. This means that the RSIMPLS method can deal with these outliers and provides satisfactory results. These results show that RoBoost-PLS2-R performs as well as RSIMPLS on this dataset. Therefore, RoBoost-PLS2-R can handle the presence of outliers in the response variables regardless of the variance of the responses.




(a)




(b)

FIGURE 3: Weights assigned to samples by the RoBoost-PLS2-R method for the simulation set 1 according to the number of LV from 1 to 13. Outliers and inliers are in red and blue, respectively.

Figure 3 shows the weights assigned to the samples of simulation 1 by the RoBoost-PLS2-R method as a function of the number of LV with the best performing hyperparameters. It can be noted that outliers have a very low weight while some inliers have a weight close to zero. This may be due to three reasons. Firstly, the hyperparameters of bisquare function must be strict enough to assign a weight close to 0 to the outliers for each LV. Taking into account that some inliers could be very similar to some outliers, assignation of low weights to these inliers could be expected. Secondly, the weights associated to $Y$-residuals are a combination of weights defined for each Y variables. The hyperparameter *beta*(see Section 3) is assumed to be constant for each variable in Y. This means that the higher the number of variables, the more dispersed the weights assigned to the inliers could be. To achieve a more homogeneous weighting on the outliers, the multivariate aspect of Y should be taken into account. For example, a potential solution can be to calculate the robust Mahalanobis distance at the centre of the data on the residuals of Y for each Latent Variable. Thirdly, some outliers are not detrimental to the model but are also irrelevant and can therefore have a

low weight without impacting on the prediction performance of the model. In conclusion, RoBoost-PLS2-R has assigned a low weight to a large number of samples without impacting on the prediction performance of the model. However, it is potentially possible to improve this approach by modifying the weighting criteria associated with the Y residuals.

## *5.2. Simulation 2*

### *5.2.1. Data visualisation*

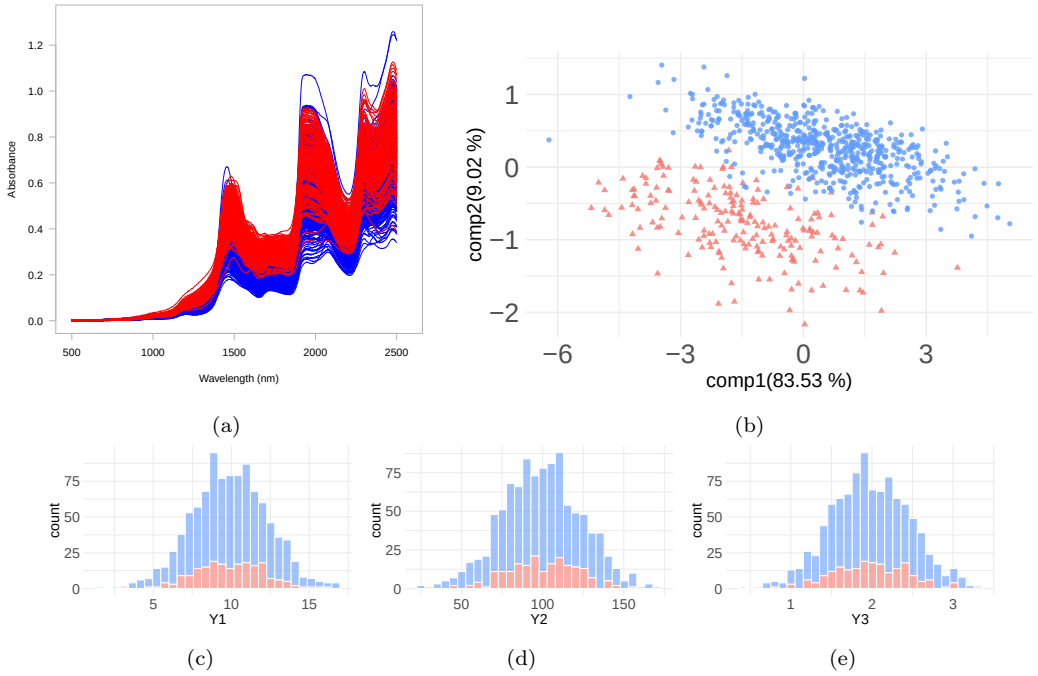



FIGURE 4: Graphical representation of simulation 2 : (a) spectral data (b) PCA score plot of the two first components (c) value distribution of Y1 response variable (d) value distribution of Y2 response variable (e) value distribution of Y3 response variable. Outliers are shown in red and inliers in blue.

Figure 4 shows the graphical representation of simulation 2. From spectra plot of the sample (Figure 4 a), it can be seen that outliers are not identifiable. Indeed, in this simulation, outliers are different only for spectral signatures and hence, they contribute slightly to the construction of the spectra. Figures 4b represents the score plot on the two first principal components. It can be

seen that there is no clear separation between the inliers and outliers. This is due to the outliers having their major compounds in common (see Table 2). From the value distributions plot of the responses (see : Figures 4c,d,e), it can be seen that outliers and inliers present similar distribution in all Y response variables. Outliers are different only on the basis of the spectral signatures that compose them.

### *5.2.2. Method evaluation*

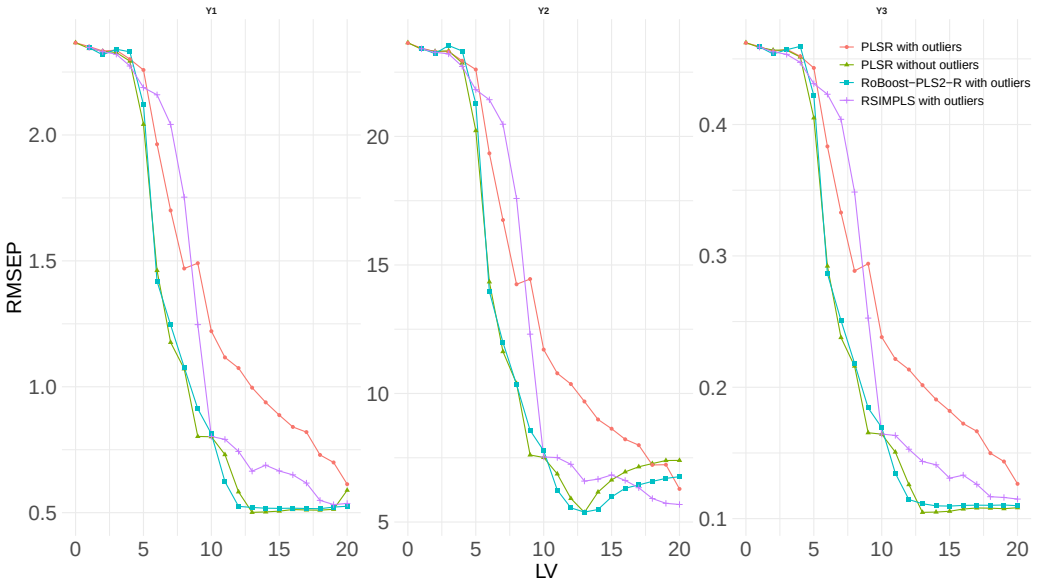
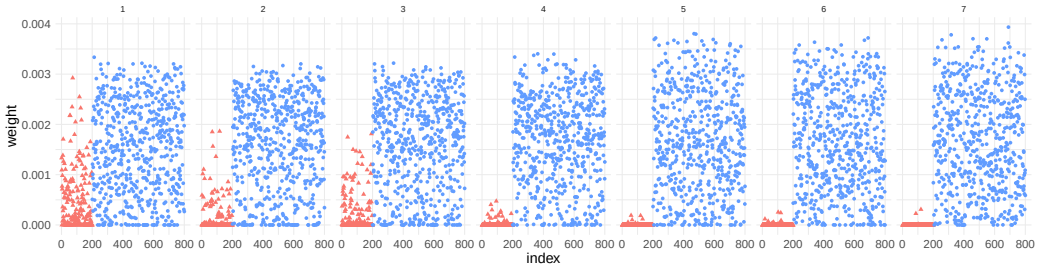


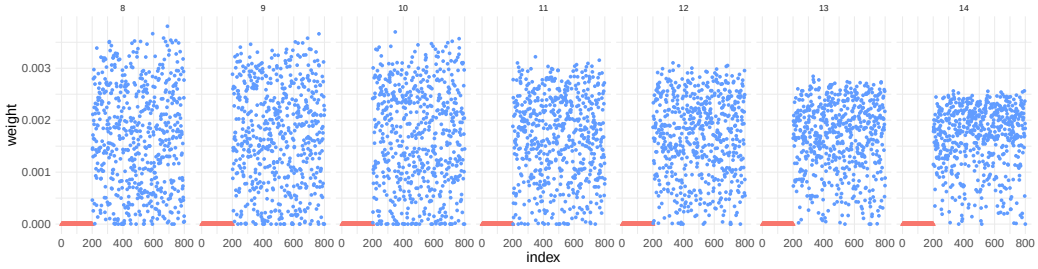

FIGURE 5: Evolution of the RMSEP as a function of the number of latent variables for the PLS2-R with and without outliers, RSIMPLS and RoBoost-PLS2-R for the simulation 2 set

The figure 5 represents the prediction performances of the applied methods on validation set for each response variable on the basis of the simulation. As expected, the outliers impact negatively the predictive capacity of the PLS2-R for all responses. For the RSIMPLS method, all performance curves are between those of the PLS2-R method with and without outliers. However, with a large number of latent variables, the prediction performances of RSIMPLS approach the best performance of PLS2-R without outliers. For the RoBoost-PLS2-R method, it can also be seen that for the three responses, performance curves are close to those of

PLS2-R without outliers. However the optimal number of components is higher for RoBoost-PLS2-R than the PLS2-R without outliers. To conclude, these results highlight the fact that RoBoost-PLS2-R can reach the best performance of PLS2-R without outliers. Thus, RoBoost-PLS2-R can handle these $X$-outliers for the prediction of multiple responses.




(a)




(b)

FIGURE 6: Weights assigned to samples in simulation set 2 according to the chosen number of latent variables from 1 to 14. Outliers and inliers are in red and blue, respectively

Figure 6 shows the weight assigned to samples by RoBoost-PLS2-R according to the number of LV. It can be observed that the weights of outliers decrease progressively when the number of LV increases. This gradual decrease is partly explained by the fact that both outliers and inliers were simulated using common majority spectral signatures. Indeed, only some minor spectral signatures differentiate the inliers from the outliers (see Section 4). After 8 latent variables, all outliers have a weight equal to 0, whereas almost all inliers present a high weight. Nevertheless, it is possible to note that the majority of the inliers have a strong weight and therefore a large number of them are used to calculate the model.

### 5.3. Real data set

#### 5.3.1. Data visualisation

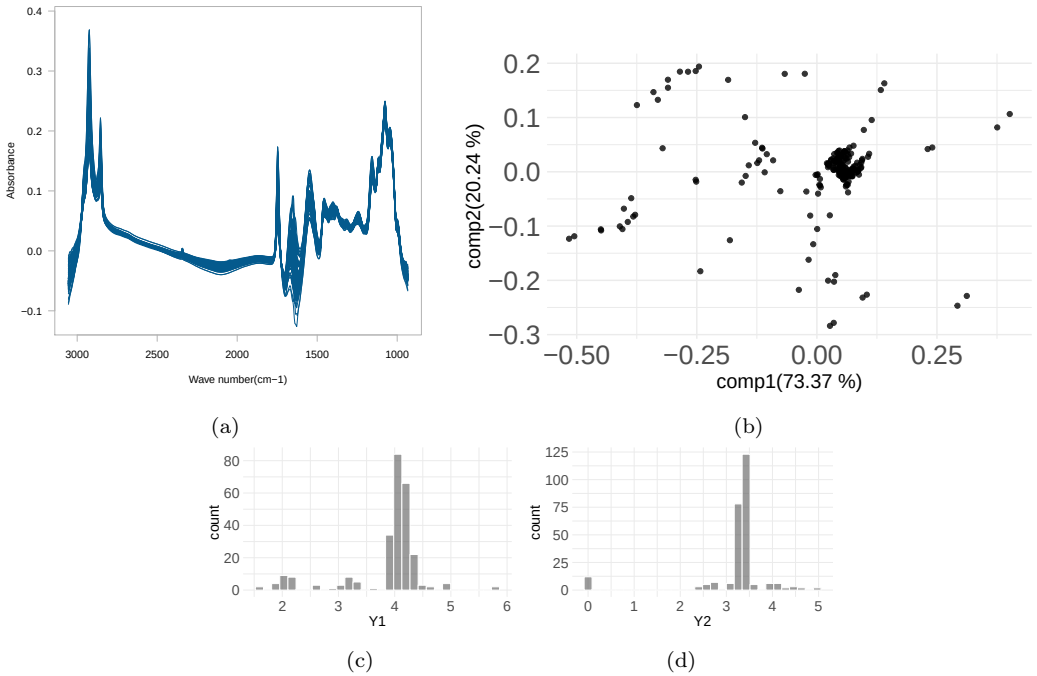



FIGURE 7: Graphical representation of real data set : (a) spectral data (b) PCA score plot of the two first components (c) value distribution of Y1 (c) value distribution of Y2

Figure 7 shows the graphical representation of real data set. From the spectra plot (Figure. 7a), it can be seen that there is no visible atypical spectrum. This means that is not possible to identify or detect outliers in this data set based on spectra visualisation. Figure 7b shows the PCA score plot of the two first components. It can be observed that some samples scores are really different from those of other samples. It is possible that some atypical samples are outliers but some sample can be also relevant to calculate a model. From the value distributions plot of the responses (see Figures 7c,d), it can be seen that some samples show extreme response values in Y1 and Y2. In conclusion, this real data set potentially contains samples that are detrimental to the model.

### 5.3.2. Method evaluation

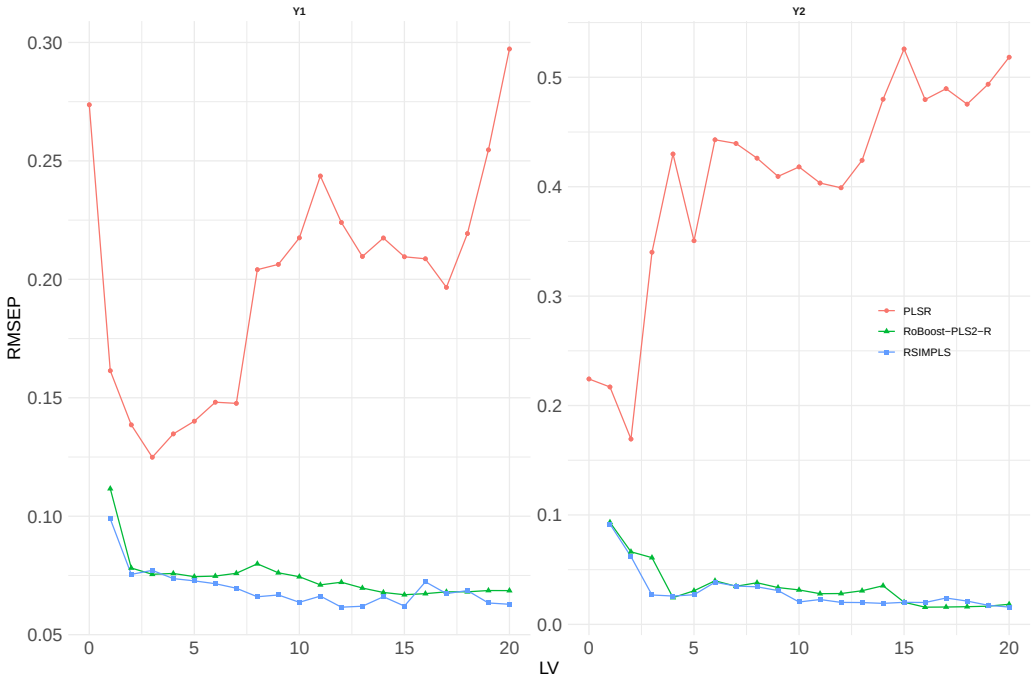
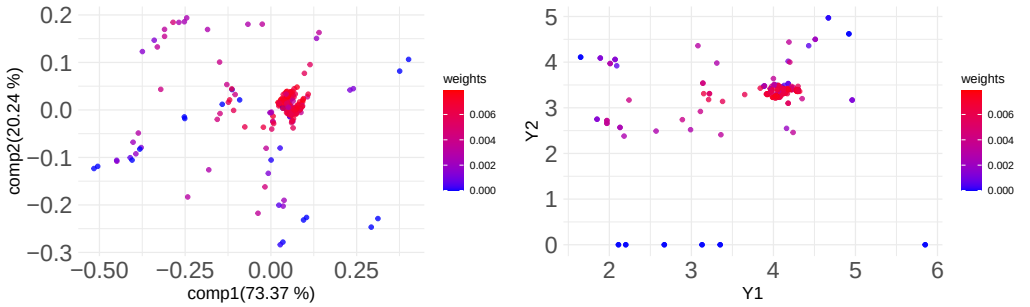



FIGURE 8: Evolution of the RMSEP as a function of the number of latent variables for the PLS2-R, RSIMPLS and RoBoost-PLS2-R for the real data set

Figure 8 represents the prediction performances of the methods on validation set for each reference Y. As there are not all known outliers in the calibration set, it was not possible to define a PLS2-R with and without outliers. Therefore, only the PLS2-R has been calculated on the data with potential outliers. In the figure 8 it can be seen that for both responses the PLSR performance curve is higher than those of the two robust methods.This means that RSIMPLS and RoBoost-PLS2-R method have higher prediction performances than the PLS2-R method applied on this data set. Therefore, some samples are detrimental in the calibration set to the calculation of a PLS2-R model that predicts the samples in the validation set. The two methods RoBoost-PLS2-R and RSIMPLS have close results in terms of RMSEP for a number of latent variables close to 15. This means that both methods were able to deal with potential outliers samples and therefore enable more accurate predictions.




(a) (b)

FIGURE 9: Graphical Representation of the mean weights (for 15 LV) assigned by RoBoost-PLS2-R through PCA score plot of the first two components(a) and Y2 as a function of Y1(b). A colour gradient from blue to red represents the weights assigned to the samples (smallest to largest).

Figure 9 shows the weights assigned to the samples by RoBoost-PLS2-R through PCA score plot of the first two components and the Y2 as a function of Y1 plot. It can be seen in figure 9a that not all samples far from the centre were considered as potential outliers (*i.e.* with low weights). Some extreme samples seem to be relevant for the model and were therefore given high weights. The figure 9b shows that some sample for Y has extreme value (0), this sample has a 0 weight value in RoBoost-PLS2-R. This is due to missing value. In this data set, missing data has a value of 0 assigned. It can be concluded through these observations that the RoBoost-PLS2-R method can eliminate outliers on Y but also on X while limiting the assignment of low weights to extreme samples.

## 6. Conclusion

In this paper, Roboost-PLS2-R method is proposed to predict multi-response. This method was evaluated and compared to reference methods on two simulated data sets and one real data set containing different outlier scenarios. For all data sets, prediction performances of Roboost-PLS2-R are close to those of PLS2-R models calibrated without outliers and to RSIMPLS method. Simulations have shown that RoBoost-PLS2-R extension was very effective when outliers are defined by their spectral properties. In the case of real data, results obtained for both robust methods are better than the PLS2-R method. To conclude, Roboost-PLS2-R seems to be a reliable and robust regression tool for

predicting multi-response variables when data potentially contain outliers. However, some method developments are possible. First of all, the estimation of the criterion evaluated on the $Y$-residuals can be estimated in another way to take into account the multivariate aspect of Y. In addition, the optimisation of the hyperparameters allowing the weighting of the individuals is complex, it would be relevant to look at automatic parameterisation approaches. Moreover, it could be interesting to use the formalism of the Roboost-PLS2-R method for cases of categorical variables and thus propose a robust discriminant method. Finally, new RoBoost-PLS2-R algorithm now enables the estimation of regression coefficients contrary to the previous algorithm proposed for RoBoost-PLS1-R. It would be interesting to study these regression coefficients to assess the method's behaviour outside the prediction capacities. In future work, it would be relevant to use the Roboost formalism for concrete applications involving multi-response variables. This formalism handles the presence of outliers in the spectral data and could be used for other multivariate data opening up other disciplines such as analytical chemistry, metabolomics.

## Références


[1] S. Wold, M. Sjostrom, L. Eriksson, PLS regression : a basic tool of chemometrics, Chemometrics and Intelligent Laboratory Systems 58 (2) (2001) 109–130. doi:10.1016/S0169-7439(01)00155-1.

[2] M. Griep, I. Wakeling, P. Vankeerberghen, D. Massart, Comparison of semirobust and robust partial least squares procedures, Chemometrics and Intelligent Laboratory Systems 29 (1) (1995) 37–50. doi:10.1016/0169-7439(95)80078-N.

[3] I. Stanimirova, S. Serneels, P. J. Van Espen, B. Walczak, How to construct a multiple regression model for data with missing elements and outlying objects, Analytica Chimica Acta 581 (2) (2007) 324–332. doi:10.1016/j.aca.2006.08.014.

[4] R. J. Pell, Multiple outlier detection for multivariate calibration using robust statistical techniques, Chemometrics and Intelligent Laboratory Systems 52 (1) (2000) 87–104. doi:10.1016/S0169-7439(00)00082-4.

[5] J. A. Gil, R. Romera, On robust partial least squares (PLS) methods, Journal of Chemometrics 12 (6) (1998) 365–378. doi:10.1002/(SICI)1099-128X(199811/12)12:6<365::AID-CEM519>3.0.CO;2-G.

[6] J. González, D. Peña, R. Romera, A robust partial least squares regression method with applications, Journal of Chemometrics 23 (2) (2009) 78–90. doi:10.1002/cem.1195.

[7] I. N. Wakelinc, H. J. H. Macfie, A robust PLS procedure, Journal of Chemometrics 6 (4) (1992) 189–198. doi:10.1002/cem.1180060404.

[8] J. Peng, S. Peng, Y. Hu, Partial least squares and random sample consensus in outlier detection, Analytica Chimica Acta 719 (2012) 24–29. doi:10.1016/j.aca.2011.12.058.

[9] P. Filzmoser, R. Maronna, M. Werner, Outlier identification in high dimensions, Computational Statistics & Data Analysis 52 (3) (2008) 1694–1711. doi:10.1016/j.csda.2007.05.018.

[10] M. Hubert, K. V. Branden, Robust methods for partial least squares regression, Journal of Chemometrics 17 (10) (2003) 537–549. doi:10.1002/cem.822.

[11] U. Kruger, Y. Zhou, X. Wang, D. Rooney, J. Thompson, Robust partial least squares regression : Part II, new algorithm and benchmark studies, Journal of Chemometrics 22 (1) (2008) 14–22, _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.1095. doi:10.1002/cem.1095.

[12] I. Hoffmann, S. Serneels, P. Filzmoser, C. Croux, Sparse partial robust M regression, Chemometrics and Intelligent Laboratory Systems 149 (2015) 50–59. doi:10.1016/j.chemolab.2015.09.019.

[13] P. Filzmoser, S. Serneels, R. Maronna, C. Croux, Robust multivariate methods in Chemometrics, arXiv :2006.01617 [stat] (2020) 393–430ArXiv : 2006.01617. doi:10.1016/B978-0-12-409547-2.14642-6.

[14] M. Hubert, K. V. Branden, Robust methods for partial least squares regression, Journal of Chemometrics 17 (10) (2003) 537–549. doi:10.1002/cem.822.

URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.822

[15] M. Metz, F. Abdelghafour, J.-M. Roger, M. Lesnoff, A novel robust PLS regression method inspired from boosting principles: RoBoost-PLSR, Analytica Chimica Acta (2021) 338823doi:10.1016/j.aca.2021.338823.
URL https://linkinghub.elsevier.com/retrieve/pii/S0003267021006498

[16] M. Metz, A. Biancolillo, M. Lesnoff, J.-M. Roger, A note on spectral data simulation, Chemometrics and Intelligent Laboratory Systems 200 (2020) 103979. doi:10.1016/j.chemolab.2020.103979.

# Chapitre 6

## Article 7 : Evaluation of a robust regression method (Roboost-PLSR) to predict biochemical variables for agronomicapplications : case study of grape berry maturity monitoring.

Référence (article soumis) :

Aldrig Courand, Maxime Metz, Daphné Héran, Carole Feilhes, Fanny Prezman,Eric Serrano, Ryad Bendoula, and Maxime Ryckewaert. Evaluation of a robust regression method (roboost-plsr) to predict biochemical variables for agronomic applications : case study of grape berry maturity monitoring. Chemometrics and Intelligent Laboratory Systems, XXXX(under revision)

MDPI

*Article*

# Evaluation of a robust regression method (RoBoost-PLSR) to predict biochemical variables for agronomic applications: case study of grape berry maturity monitoring

**Aldrig Courand[3], Maxime Metz[1,2], Daphné Héran[1], Carole Feilhes[4], Fanny Prezman[4], Eric Serrano[4], Ryad Bendoula[1], Maxime Ryckewaert[1,2]**

1 ITAP, Univ Montpellier, INRAE, Institut Agro, Montpellier, France
2 ChemHouse Research Group, Montpellier, France
3 UBO,Brest,France
4 IFV, 1920 Route de Lisle-sur-Tarn, 81310 Peyrole, France
† Current address: ryad.bendoula@inrae.fr

Version October 11, 2021 submitted to Sensors

**Abstract:** Visible and near infrared spectroscopy (VIS-NIR) is increasingly being transferred from laboratory to industry for in-lign and portable applications in various domains. By using intensively VIS-NIR spectroscopy, some abnormal observations may certainly arise. It is then important to handle properly outliers to elaborate effective prediction models. The objective of this study is to investigate the potential of using a robust method called Roboost-PLSR to improve prediction model performances for agricultural applications. This work focuses on a case study to predict sugar content in grape berries of three different grape varieties of Vitis Vinifera in a maturity monitoring context. Hyperspectral images were acquired on grape berries of Syrah, Fer-Servadou and Mauzac varieties. Reference measurements of sugar levels were made in the laboratory by densimetric baths. Performances of RoBoost-PLSR models were compared to performances of reference models using Partial Least Square Regression (PLSR). Reference prediction criteria using PLSR were obtained for all varieties with these following values: Syrah ($R^2_p$ = 0.971; $RMSE_p$= 5.36 g/l), Fer-servadou ($R^2_p$ = 0.788; $RMSE_p$= 11.69 g/l) and Mauzac ($R^2_p$ = 0.690; $RMSE_p$ = 15.61 g/l). Prediction qualities are improved with RoBoost-PLSR: Syrah ($R^2_p$ = 0.990; $RMSE_p$=3.14 g/l), Fer-Servadou ($R^2_p$ = 0.848; $RMSE_p$= 10.20 g/l) and Mauzac ($R^2_p$ = 0.927; $RMSE_p$= 7.58 g/l). Results confirm that Roboost-PLSR method allows a better consideration of outliers within the calibration set.

**Keywords:** Robust regression; Chemometrics; Spectroscopy; Grapes

## 1. Introduction

It is increasingly common that visible and near-infrared (VIS-NIR) spectroscopy transfers from laboratory to industry for in-lign and portable applications in various domains. By using intensively VIS-NIR spectroscopy, some abnormal observations may certainly arise. Among these, observations are called leverage points when they have a strong impact on the construction of a prediction model. When they are detrimental to the prediction model, they are called outliers. It is then important to handle properly these outliers to elaborate effective prediction models. In chemometrics, Partial Least Square Regression (PLSR) [1] is a widely-used tool. Particularly, PLSR is effective when dealing with high-dimensional data such as spectral data, where sample number is lower than variable number. Besides, PLSR method performs admirably when the relationship between explanatory variables

and response variable to be predicted is linear. However, estimating this linear relationship may be disturbed in presence of outliers [2].

These outlier data are generally due to variations of measurement conditions (view angle, reference, sensor temperature), physico-chemical variations in measured samples or experimental errors (annotation, operator). All these variations require efforts to identify and remove outliers from the calibration set. In addition, inspecting each observation manually is complicated and time-consuming in the case of large databases.

These problems are also found in agronomy, where the use of VIS-NIR spectroscopy is tending to be more frequently used [3]. Indeed, rich spectral information is an added value to predict biochemical variables to assess agronomic parameters for various agronomic applications. This technological trend operates at different scales depending on the objectives: prediction models can be used at fruit scale for quality control, at the leaf/canopy scale for plant health monitoring or at the plot scale for production monitoring. Multiple use cases of spectral data encourage a particular development on the management of outliers.

Robust methods have been developed to address this issue [4–9]. Indeed, this type of method aims at reducing the outlier impact automatically on PLSR model calibration. Recently, a method called Roboost-PLSR has been developed [9] and has shown its effectiveness to manage PLSR model calibration in the presence of outlier data.

This article highlights the interest of RoBoost-PLSR method to improve prediction models for agronomic applications and more particularly in the case of monitoring grape berry maturity of Vitis Vinifera. For this purpose, Roboost-PLSR method was compared to the reference method PLSR to predict sugar content in grape berries of three different grape varieties.

## 2. Materials and methods

### *2.1. Biological material and reference measurements*

Grape berries were collected during a campaign carried out in Gaillac (France), in summer 2020. The sampling started one or two weeks after veraison and preharvest, on three plots corresponding to three different grape varieties of the experimental vineyard Domaine Expérimental Viticole Tarnais: Syrah, Fer Servadou and Mauzac. Thirty bunches were randomly sampled in each plot about once a week.

**Figure 1.** Picture of densimetric baths used for maturity degree sorting of grape berries.

In the laboratory, grape berries were cut from bunches at the pedicel level to preserve entire fruits. Grape berries were then sorted in batches with same maturity degree using sodium chloride (NaCl) baths to achieve a densimetric sorting (see fig. 1). Indeed, the increase in berry density during ripening is mainly due to sugar accumulation in berries [10,11]. To this end, twelve NaCl baths with increasing concentrations from 80 to 190 g/l were used to classify berry density corresponding to sugar

concentrations from 110 to 279 g/l [12]. First, berries were immersed in the highest NaCl concentration solution. Then, floating fruits were removed and immersed in a solution of lower concentration, whereas sinking fruits were removed and sorted into the density level corresponding to the NaCl solution. The procedure was repeated for all baths in order to obtain twelve classes of homogeneous maturity.

### 2.2. Spectral acquisition

Reflectance spectra were acquired with a hyperspectral camera (Specim IQ, Specim, Finland) having a spectral range from 400 nm to 1000 nm and a spectral resolution equals to 7 nm (see Fig 2).

**Figure 2.** Hyperspectral acquisition of grape berries.

For each sample, reflected light intensity ($I_s(\lambda)$) was measured at each wavelength $\lambda$. Dark current image ($I_b(\lambda)$) was also recorded for each measure. A certified reflectance standard (Labsphere, SRS-40-010) was used as a reference reflected intensity ($I_o(\lambda)$) to standardise images from non-uniformities of instrumentation (light source, lens, detector). From these measurements, a reflectance image ($R_s(\lambda)$) was obtained for each sample where each pixel of this image is a reflectance spectrum:

$$R_s(\lambda) = \frac{I_s(\lambda) - I_b(\lambda)}{I_0(\lambda) - I_b(\lambda)} \tag{1}$$

### 2.3. Image preprocessing

A segmentation process was implemented to retrieve berry reflectance spectra from images. First, three reference spectra were defined, corresponding to each grape variety, by calculating an averaged spectrum from a manual selection of an area of a berry. Then, the segmentation was performed by comparing each image pixel with these previously defined spectra (see fig. 3).

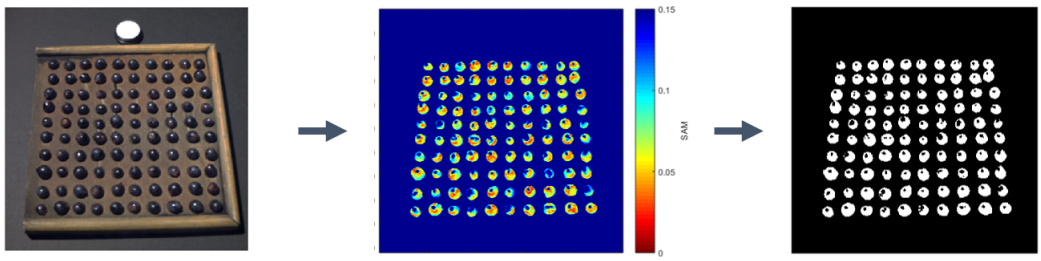



**Figure 3.** Segmentation by using spectral similarity threshold.

To this end, Spectral Angle Mapper (SAM) [13,14] was selected to evaluate spectral similarity between the reference spectrum defined for a given grape variety and spectra contained in hyperspectral images. Indeed, this criterion corresponds to an angle between two spectra (assimilated to vectors) and is favourably independent to intensity levels. The angle $\alpha$ defined between the

corresponding variety reference spectrum **y** and the spectrum of a given pixel **x**, was calculated as follows:

$$\alpha = cos^{-1} \frac{\sum_\lambda \mathbf{xy}}{\sqrt{\sum(\mathbf{x})^2 \sum(\mathbf{y})^2}} \quad (2)$$

By defining a spectral similarity threshold, berry spectra were retrieved from the images (see fig. 3). Finally, for each image a berry average spectrum was computed, to consider a unique sugar content.

### *2.4. Data analysis*

#### 2.4.1. Regression methods used

RoBoost-PLSR [9] was used as a robust regression method to predict sugar content **Y** from spectral data **X**. This methods reduces outlier effect on model calibration by weighting them. A particularity of this method is that outliers are defined latent variable by latent variable. For each model with one latent variable, observation weights are calculated according to three criteria: **X** residuals, **Y** residuals and leverage points with the hyperparameters $\alpha$, $\beta$ and $\gamma$ respectively. In this study, sixty-four combinations of values for $\alpha$, $\beta$ and $\gamma$ were tested to optimise the model with these following possible values: 2, 4, 6, and infinite. RoBoost-PLSR was compared to the reference regression method PLSR [1].

Calculations were performed with the R software (version 3.6.1 [15]), rnirs package for PLSR (https://github.com/mlesnoff/rnirs) and roboost package for RoBoost-PLSR (https://github.com/maxmetz/RoBoost-PLSR).

#### 2.4.2. Calibration and test set definition

To compare PLSR method with RoBoost-PLSR method, models were established from three data sets corresponding to the three different grape varieties. For each grape variety, data were split into two sets, one calibration set and one test set. The calibration set was formed with 75% of the whole data set whereas the test set was formed with the remaining 25%. As showed in table 1, the total number of observations was different depending on the grape variety.

**Table 1.** Number of observations constituting the whole data set, the calibration set and the test set, for the three grape varieties, Syrah, Fer and Mauzac.

| Number of observations | Syrah | Fer | Mauzac |
|---|---|---|---|
| Whole dataset | 126 | 63 | 85 |
| Calibration set | 95 | 48 | 67 |
| Test set | 31 | 15 | 18 |

Besides, test sets were created avoiding abnormal observations according to [9].

#### 2.4.3. Assessment criteria

PLSR models were calibrated by performing a cross-validation procedure [16]. For each grape variety, a k-fold cross-validation with five blocks was defined on the corresponding calibration data set.

Model evaluation was performed using several criteria: root-mean-square error (RMSE), median absolute deviation (MAD) and determination coefficient $R^2$. Besides, the number of latent variables was optimised thanks to the RMSE parameter and was chosen to be lower than twenty. These criteria were calculated thanks to the following equations:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}} \quad (3)$$

$$\text{MAD} = median(|y_i - \tilde{y}|) \tag{4}$$

$$\text{R}^2 = 1 - \frac{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}{\frac{\sum_{i=1}^{N}(y_i - y_m)^2}{N}} \tag{5}$$

with $\hat{y}_i$ the predicted value, $y_i$ the observed value, $y_m$ the average of all response values and $N$ the total number of observations. $\text{RMSE}_{cv}$, $\text{MAD}_{cv}$ and $\text{R}^2_{cv}$ denoted criteria obtained in the cross-validation step whereas $\text{RMSE}_p$, $\text{MAD}_p$ and $\text{R}^2_p$ denoted those obtained with the independent test set.

Likewise PLSR, RoBoost-PLSR models were calibrated by performing a k-fold cross-validation procedure with five blocks. However, so-called robust evaluation criteria were calculated by using a procedure of trimming [17] Trimming consisted in sorting out observations according to their weights before removing a percentage of observations having the weaker weights. Moreover, this percentage was adapted to each of the three grape varieties: 5% for Syrah, 15% for Fer and 20% for Mauzac. Among these new criteria, r-$\text{RMSE}_{cv}$ and r-$\text{R}^2_{cv}$ were defined, corresponding respectively to the trimmed RMSE and the trimmed coefficient of determination. The MAD calculated previously (eq. 4) was retained as it is considered a criterion for evaluating robustness.

So-called robust evaluation criteria were chosen according to Filzmoser and al work [17]. MAD, considered as a robustness evaluation criterion, was computed. r-$\text{RMSE}_{cv}$ and r-$\text{R}^2_{cv}$ were computed as follows:

$$\text{r-RMSE}_{cv} = \sqrt{\frac{\sum_{i=1}^{N_t}(\hat{y}_i - y_i)^2}{N_t}} \tag{6}$$

$$\text{r-R}^2_{cv} = 1 - \frac{\frac{\sum_{i=1}^{N_t}(\hat{y}_i - y_i)^2}{N_t}}{\frac{\sum_{i=1}^{N_t}(y_i - y_m)^2}{N_t}} \tag{7}$$

With $\hat{y}_i$ the predicted y, $y_i$ the observed y, $y_m$ the average y and $N_t$ the number of trimmed observations. The r-RMSE was chosen as the criterion to minimise during cross-validation.

## 3. Results and discussion

### *3.1. Data visualization*

Sugar content distributions measured on grape berries of the three varieties (Fer Servadou, Mauzac and Syrah) can be seen in Figure 4.

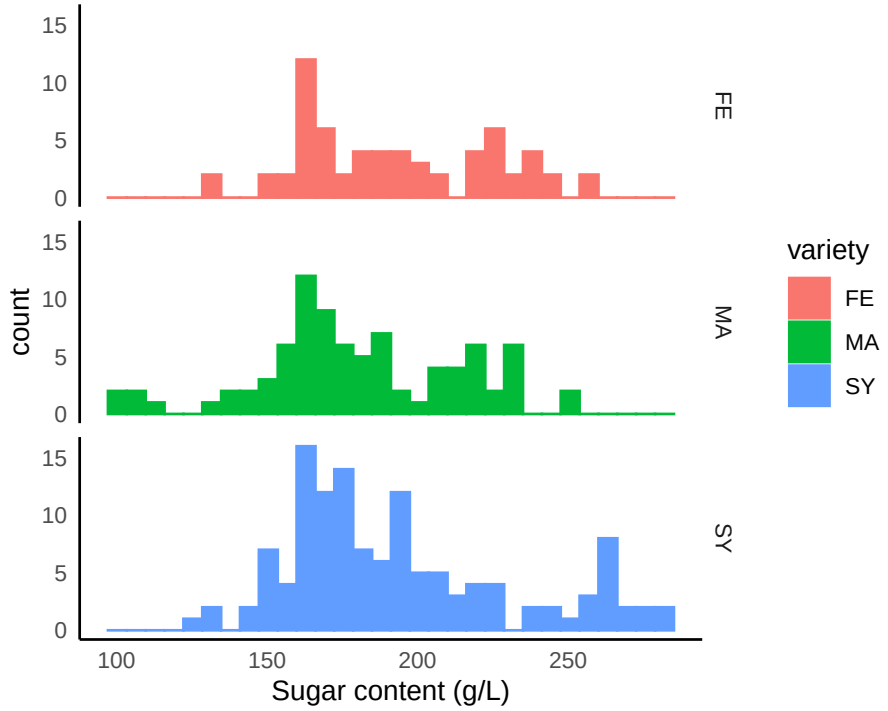



**Figure 4.** Sugar content (g/l) histograms for the three grape varieties: Fer Servadou (FE), Mauzac (MA) and Syrah (SY)

For the three varieties, sugar content values are similar and comprised between 100 and 300 g/l. Most values lie between 150 and 200 g/l which correspond to expected sugar contents for grape berries at different maturity stages. As sugar content values cover the same range for the three varieties, comparing results obtained for each grape variety is relevant.

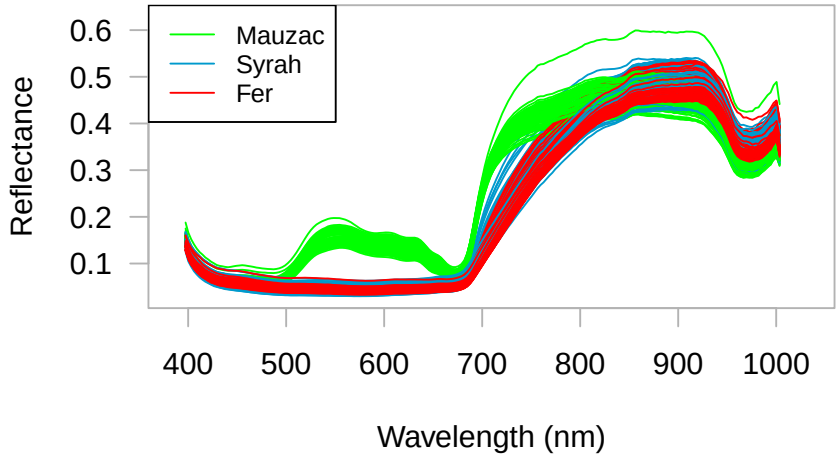



**Figure 5.** Reflectance spectra of the whole data set

Reflectance spectra comprised between 400 nm and 1000 nm of the whole data set are shown in figure 5. The two varieties Syrah and Fer Servadou are similar over the whole spectral range. However, Mauzac spectra differ from the two other varieties. Mainly, reflectance values are higher in

the spectral range comprised between 500 nm and 680 nm. Moreover, the spectrum slope is steeper around 700 nm.

Syrah and Fer Servadou are red grape varieties and are known to possess high anthocyanin contents. Besides, visible light is largely absorbed by anthocyanins which causes low reflectance values between 500 nm and 700 nm, as can be seen on spectra for these two varieties. Spectra visualisation confirms the establishment of prediction models by variety.

### *3.2. Prediction models*

#### 3.2.1. PLSR models

Table 2 presents the values of the four criteria, latent variable number (nLV), prediction error ($RMSE_{cv}$), median ($MAD_{cv}$) and determination coefficient ($R^2_{cv}$), based on the cross-validation of the three grape variety PLSR models.

**Table 2.** Selected criteria obtained for cross-validation of PLSR prediction models on calibration data set: latent variable number (nLV), prediction error ($RMSE_{cv}$), median ($MAD_{cv}$) and determination coefficient ($R^2_{cv}$)

| Model | Variety | nLV | $RMSE_{cv}$ (g/l) | $MAD_{cv}$ (g/l) | $R^2_{cv}$ |
|---|---|---|---|---|---|
| PLSR | Syrah | 6 | 9.31 | 8.09 | 0.937 |
| | Fer Servadou | 7 | 19.45 | 15.84 | 0.623 |
| | Mauzac | 5 | 28.78 | 18.40 | 0.298 |

Results show large disparities between grape varieties. Indeed, Syrah has the best results with a higher $R^2_{cv}$ of 0.937 and lower $RMSE_{cv}$ and $MAD_{cv}$ ,of respectively 9.31 g/l and 8.09 g/l. For Fer Servadou variety, $RMSE_{cv}$ and $MAD_{cv}$ have values equal to 19.45 g/l and 15.84 g/l, which are nearly twice as large as the Syrah values. For Mauzac variety, $RMSE_{cv}$ value is equal to 28.78 g/l and $MAD_{cv}$ value is 18.40 g/l. These values are two to three times higher than the ones obtained for Syrah.

Likewise, determination coefficient values differ between the three grape varieties. $R^2_{cv}$ obtained for Fer Servadou and Mauzac varieties are equal to 0.623 and 0.298 respectively, much lower than Syrah result, especially for Mauzac. High discrepancies can be seen among the three grape varieties.

#### 3.2.2. RoBoost-PLSR models

**Table 3.** Selected criteria obtained for cross-validation of RoBoost-PLSR prediction models on calibration data set: trimming, hyperparameters, latent variable number (nLV), prediction error (r-$RMSE_{cv}$), median ($MAD_{cv}$) and determination coefficient (r-$R^2_{cv}$)

| Model | Variety | Trimming | Hyperparameters ($\alpha$; $\beta$; $\gamma$) | nLV | r-$RMSE_{cv}$ (g/l) | $MAD_{cv}$ (g/l) | r-$R^2_{cv}$ |
|---|---|---|---|---|---|---|---|
| RoBoost-PLSR | Syrah | 5% | Inf; 4; 6 | 6 | 8.57 | 6.86 | 0.951 |
| | Fer | 15% | Inf; 4; Inf | 7 | 12.5 | 14.3 | 0.844 |
| | Mauzac | 20% | Inf; 4; 6 | 6 | 12.1 | 15.50 | 0.794 |

The table 3 shows parameters from cross validation of RoBoost-PLSR method. These parameters are trimming percentage, hyperparameters ($\alpha$, $\beta$, $\gamma$), latent variable number, r-$RMSE_{cv}$, $MAD_{cv}$ and r-$R^2_{cv}$. Hyperparameter values $\alpha$, $\beta$ and $\gamma$ are respectively equal to infinite, 4, 6 for Syrah; infinite, 4, infinite for Fer Servadou; and infinite, 4, 6 for Mauzac. Hyperparameters $\alpha$, $\beta$ and $\gamma$ are selective criteria for outlier detection respectively on $\mathbf{X}$, $\mathbf{Y}$ and leverage points. The lower the hyperparameter, the higher the outlier number identified by the model. Conversely, an infinite value means no outlier identified. This implies that there is no outlier detected by cross-validation on $\mathbf{X}$ for the three grape varieties ($\alpha$ = Inf). However, this is not the case for $\mathbf{Y}$ (i.e. measures of sugar content), where $\beta = 4$ for the three grape varieties and means that several outliers are detected. Indeed, outliers could be

introduced during sugar content measurements by densimetric bath. Finally, based on hyperparameter $\gamma$ values, no leverage point is identified for Fer Servadou variety whereas some are detected for Mauzac and Syrah.

Among the three grape varieties, Syrah obtains the best results with a r-RMSE$_{cv}$ equals to 8.57 g/l which corresponds to the lowest value. Furthermore, this value is slightly lower than the one obtained with the PLSR model (see table 2). Regarding Fer Servadou and Mauzac varieties, r-RMSE$_{cv}$ values are close from each other with values equal to 12.5 g/l and 12.1 g/l respectively. These results are improved compared to the values previously obtained with PLSR cross-validation (see table 2) and closer to Syrah value. Indeed, outlier points are taken into account in a more effective way during RoBoost-PLSR model design.

Besides, the same analysis can be done for r-$R^2_{cv}$ values. Syrah obtains the best value with 0.951 whereas Fer Servadou and Mauzac obtain 0.844 and 0.764. Again, these values are lower and closer to each other than the ones previously obtained with PLSR cross-validation (see table 2).

Finally, the comparison of both cross-validation results, PLSR (table 2) and RoBoost-PLSR (table 3), indicates that RoBoost-PLSR decreases the prediction quality discrepancies between grape varieties. This result confirms the presence of outlier points among Fer Servadou and Mauzac data sets.

#### 3.2.3. Observed vs. predicted values of calibration models

The visualisation of observed values by predicted values shown in figure 6 helps to better understand criteria values obtained in cross-validation (tab 2 and 3). It provides a means to assess model quality, observation by observation.

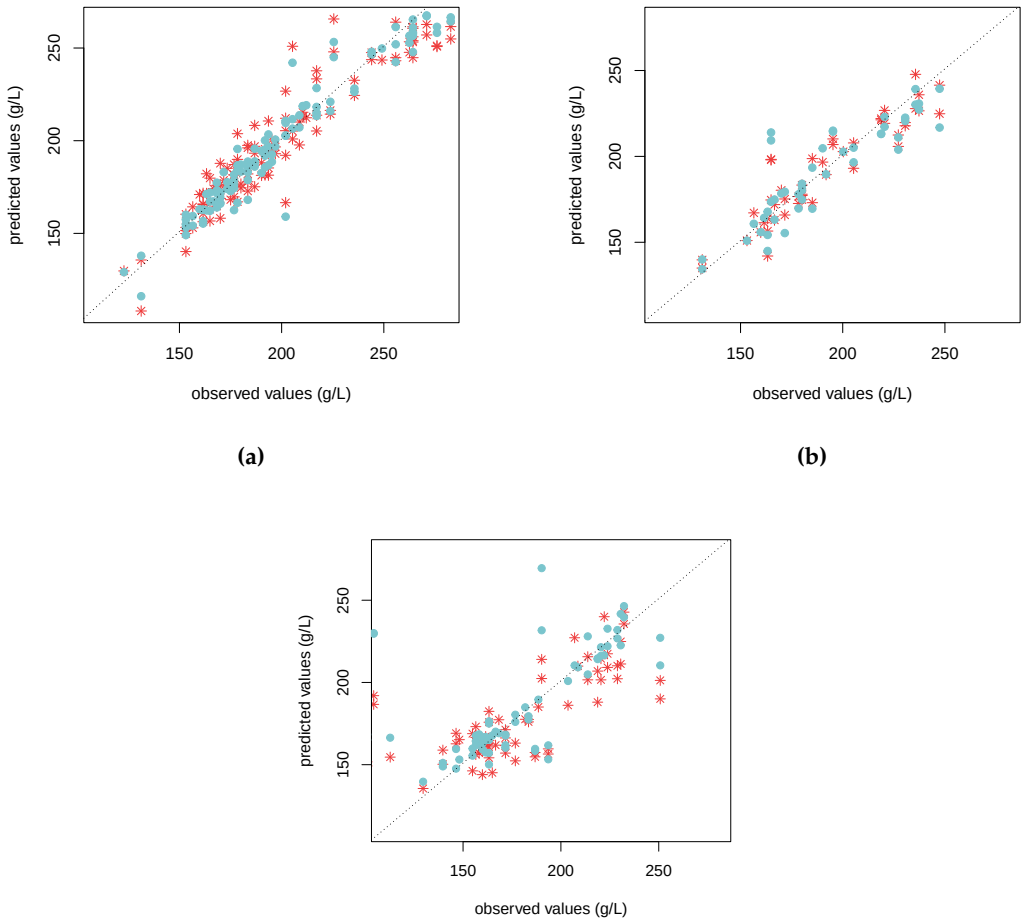



**Figure 6.** Sugar content observed values versus predicted values based on (*, red) PLSR and (•, blue) RoBoost-PLSR for the three grape varieties: (a) Syrah, (b) Fer, (c) Mauzac

Figures 6a, 6b and 6c compare predicted values of the calibration data set obtained with RoBoost-PLSR and PLSR for Syrah, Fer and Mauzac respectively.

Regarding Syrah variety (fig. 6a), relationship between predicted **Y** and observed **Y** is linear and point dispersion is the lowest obtained for the three varieties and this with RoBoost and PLSR. The same holds true for Fer Servadou variety (fig. 6b), where a linear tendency between predicted **Y** and observed **Y** can be noticed. However, several points obtained with PLSR deviate from this tendency. These same points are further deviated from the linear trend with RoBoost-PLSR. The identified points deviating from the linear tendency are possibly outliers (also called vertical outliers) or leverage points.

As far as Mauzac is concerned (fig. 6c), the relationship between predicted **Y** and observed **Y** deviates from a linear tendency with several points strongly dispersed. Some points deviate more strongly from this trend than previously. These same points are even further apart with RoBoost-PLSR, while an improvement appears on the majority of the other points. These points are clearly identified by the RoBoost-PLSR method as vertical outliers or leverage points. These points are discarded when building the prediction model with RoBoost-PLSR. RoBoost-PLSR thus improves the linearity between predicted and observed values.

By comparing these three figures (6a, 6b and 6c), calibration data set which have the best predictions are Syrah first, then Fer Servadou and finally Mauzac. This confirms the results obtained in cross-validation (table 3).

### 3.3. Model prediction on independent test sets

For each grape variety, PLSR and RoBoost-MLSR models previously parameterized during cross-validation steps and calibrated with calibration data sets are now tested on the test data sets.

**Table 4.** Performance evaluation of PLSR and RoBoost-PLSR prediction models on test data sets: latent variable number (nLV), prediction error ($RMSE_p$), median ($MAD_p$) and determination coefficient ($R^2_p$)

| Model | Variety | nLV | $RMSE_p$ (g/l) | $MAD_p$ (g/l) | $R^2_p$ |
|---|---|---|---|---|---|
| PLSR | Syrah | 6 | 5.36 | 4.99 | 0.971 |
| | Fer Servadou | 7 | 11.69 | 12.04 | 0.788 |
| | Mauzac | 5 | 15.61 | 10.97 | 0.690 |
| RoBoost PLSR | Syrah | 6 | 3.14 | 3.38 | 0.990 |
| | Fer Servadou | 7 | 10.20 | 10.50 | 0.848 |
| | Mauzac | 6 | 7.58 | 9.36 | 0.927 |

Table 4 outlines the prediction quality of both PLSR and RoBoost-PLSR models, applied to the test data sets of each grape variety. To this end, the following criteria are presented: latent variable number (nLV), $RMSE_p$, $MAD_p$ and $R^2_p$.

First of all, a higher heterogeneity among results can be noticed for PLSR models than for RoBoost-PLSR ones. Regarding PLSR models, Syrah has the best performances, with the lowest $RMSE_p$ and $MAD_p$ values, equal to 5.36 g/l and 4.99 g/l respectively, and the highest $R^2_p$ value, equals to 0.971. Fer Servadou and Mauzac have $RMSE_p$ and $MAD_p$ values, two to three times higher than Syrah ones. $RMSE_p$ are equal to 11.69 g/l and 15.61 g/l for Fer and Mauzac respectively, whereas $MAD_p$ values are 12.04 g/l and 10.97 g/l respectively. Moreover, $R^2_p$ are lower than for Syrah, with respective values of 0.788 and 0.690. As said before during cross-validation step (section 3.2.1), discrepancies among varieties arise with PLSR models.

As far as RoBoost-PLSR models are concerned, all three varieties predictions are improved compared to PLSR models. This is all the more true in the case of Mauzac and Syrah. Indeed, Syrah obtains $R^2_p$, $RMSE_p$ and $MAD_p$ values equal to 0.990, 3.14 g/l and 3.38 g/l respectively. Besides, Fer Servadou obtains $R^2_p$, $RMSE_p$ and $MAD_p$ values equal to 0.848, 10.20 g/l and 10.50 g/l. Lastly, Mauzac obtains $R^2_p$, $RMSE_p$ and $MAD_p$ values equal to 0.927, 7.58 g/l and 9.36 g/l. These last results outperform PLSR models and lead to performances close to Syrah ones.

It is worth noticing that PLSR model allows to predict sugar content for Syrah in an effective way. This implies that there is a limited number of outlier points in the data set. The same does not hold true for Fer Servadou and Mauzac, as noticed in figure 6. In all cases, RoBoost-PLSR method allows to build predictive models with higher performances than PLSR when dealing with outliers points among calibration data sets.

## 4. Conclusion

The potential of RoBoost-PLSR method to calibrate prediction models in the presence of outliers in an agronomic context was studied. The method was evaluated on a case of Vitis Vinifera grapes berry maturity context and especially to predict berry sugar content. RoBoost-PLSR method was compared to the reference method (PLSR) on spectral data from berries of three grape varieties (Syrah, Mauzac, Fer Servadou). For these three varieties, results obtained from RoBoost-PLSR method outperformed those from the PLSR method. The improvements in the prediction of sugar content for Fer Servadou and Mauzac are the most significant due to a potentially higher outliers number in the calibration set.

This study validates the use of the RoBoost-PLSR method for monitoring grapes berries maturity in the laboratory. The advantage of this method is to provide good prediction models despite outliers presence. Despite optimal measurement conditions, outliers were identified as detrimental to the model calibration. This method could be challenged on data collecting directly in the field where measurement conditions most often lead to outliers. This would open up multiple possibilities for the

use of VIS-NIR spectroscopy for agronomic applications. This method also contributes to perspectives in other disciplines where multivariate data is involved such as analytical chemistry.

## Acknowledgement

**Funding:** This work has benefited from a financial support from the InterregSudoe under the reference SOE3/P2/E0911.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wold, S.; Sjöström, M.; Eriksson, L. PLS-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems* **2001**, *58*, 109–130.
2. Serneels, S.; Croux, C.; Filzmoser, P.; Van Espen, P.J. Partial robust M-regression. *Chemometrics and Intelligent Laboratory Systems* **2005**, *79*, 55–64. doi:https://doi.org/10.1016/j.chemolab.2005.04.007.
3. Ryckewaert, M.; Metz, M.; Héran, D.; George, P.; Grèzes-Besset, B.; Akbarinia, R.; Roger, J.M.; Bendoula, R. Massive spectral data analysis for plant breeding using parSketch-PLSDA method: Discrimination of sunflower genotypes. *Biosystems Engineering* **2021**, *210*, 69–77. doi:10.1016/j.biosystemseng.2021.08.005.
4. Serneels, S.; Croux, C.; Filzmoser, P.; Van Espen, P.J. Partial robust M-regression. *Chemometrics and Intelligent Laboratory Systems* **2005**, *79*, 55–64. doi:10.1016/j.chemolab.2005.04.007.
5. Hubert, M.; Branden, K.V. Robust methods for partial least squares regression. *Journal of Chemometrics* **2003**, *17*, 537–549. _eprint: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cem.822, doi:https://doi.org/10.1002/cem.822.
6. Filzmoser, P.; Maronna, R.; Werner, M. Outlier identification in high dimensions. *Computational Statistics & Data Analysis* **2008**, *52*, 1694–1711. doi:10.1016/j.csda.2007.05.018.
7. Filzmoser, P.; Serneels, S.; Maronna, R.; Croux, C. Robust Multivariate Methods in Chemometrics. In *Comprehensive Chemometrics*; Elsevier, 2020; pp. 393–430. doi:10.1016/B978-0-12-409547-2.14642-6.
8. Griep, M.I.; Wakeling, I.N.; Vankeerberghen, P.; Massart, D.L. Comparison of semirobust and robust partial least squares procedures. *Chemometrics and Intelligent Laboratory Systems* **1995**, *29*, 37–50. doi:10.1016/0169-7439(95)80078-N.
9. Metz, M.; Abdelghafour, F.; Roger, J.M.; Lesnoff, M. A novel robust PLS regression method inspired from boosting principles: RoBoost-PLSR. *Analytica Chimica Acta* **2021**, p. 338823. doi:10.1016/j.aca.2021.338823.
10. M.r, L.; J.r, M. Density separation of muscadine grapes. *Arkansas Farm Research* **1978**.
11. M.r, L.; J.r, M. Maturation rates of muscadine grapes. *Arkansas Farm Research* **1978**.
12. Bigard, A. Varietal differences in solute accumulation and grape development. phdthesis, Montpellier SupAgro, 2018.
13. Kruse, F.A.; Lefkoff, A.B.; Boardman, J.W.; Heidebrecht, K.B.; Shapiro, A.T.; Barloon, P.J.; Goetz, A.F.H. The spectral image processing system (SIPS)—interactive visualization and analysis of imaging spectrometer data. *Remote sensing of environment* **1993**, *44*, 145–163.
14. Yuhas, R.H.; Goetz, A.F.; Boardman, J.W. Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm **1992**.
15. Core Team, R. R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. *Available* **2013**.
16. Browne, M.W. Cross-Validation Methods. *Journal of Mathematical Psychology* **2000**, *44*, 108–132. doi:10.1006/jmps.1999.1279.
17. Filzmoser, P.; Nordhausen, K. Robust linear regression for high-dimensional data: An overview. *Wiley Interdisciplinary Reviews: Computational Statistics* **2021**, *13*, e1524. Publisher: Wiley Online Library.