# FEATURE SELECTION IN METABOLOMICS WITH PLS-DERIVED METHODS

Martin, Manon and Bernadette Govaerts

DISCUSSION PAPER | 2019 / 20

UCLouvain

ISBA | LIDAM
Louvain Institute of Data Analysis and Modeling in economics and statistics

# Feature Selection in metabolomics with PLS-derived methods.

Manon Martin & Bernadette Govaerts

September 12, 2019

## Contents

# Abstract

Current feature selection in metabolomics rely essentially on t-tests, (O)PLS and their sparse counterparts, but their validation as feature selection technique in this research field is lacking as the true biomarkers location is usually unknown. In this work, we use a set of semi-artificial $^{1}$H NMR spectra from rat urine where artificial alterations are known id advance to evaluate the features accuracy and stability, among other measurements, of the supervised methods. Several conditions are stressed with different levels of noise in the data, true feature intensities and sample sizes. Results showed that PLS-vip with a threshold of 1 was the overall best method for our dataset. Nevertheless, all approaches have advantages and drawbacks that should be considered prior to their use for feature selection.

# 1 Introduction

## 1.1 Feature selection in metabolomics

Compared to the budgetary limited number of experimental samples, modern high-throughput technologies produce an enormous amount of variables. In such wide and short data tables, computational and statistical challenges are substantial to extract all the meaningful information from them.

There is a myriad of variable selection techniques (see Guyon & Elisseeff (2003) for a comprehensive review of the different kinds of feature selection) that have been developed accordingly, some are domain-specific while others are more general. They all lead to a subset selection of relevant variables called *features*. One main purpose of Feature Selection (FS) is to identify potential biomarkers to answer the biological question of interest. Furthermore, after their selection, these potential biomarkers should latter be double-checked by an independent and targeted confirmatory study. Note that in this paper, *features* is defined as the set of variables selected by the discriminant techniques whereas *biomarkers* involve the whole metabolites' descriptors, *e.g.* 16 true features corresponds to 4 different true biomarkers.

Variable selection techniques can be distinguished into three main categories: filters, wrappers or embedded (Guyon & Elisseeff, 2003). They are either ranking the features or selecting a subset of them. *Filters* are independent of the classifier. They will assign a ranking to the variables that is based on a predefined univariate or multivariate importance measure. For their part, *wrappers* are classifier-specific. They evaluate several subsets of interest based on prediction performances. Since the number of potential subset partitions is too important to examine each of them, an iterative search procedure is implemented to only evaluate the most interesting ones (*e.g.* with backward/forward selection). The latter category, *embedded*, lies in between filters and wrappers: the selection procedure is integrated to the training proceure of the classifier as *e.g.* in LASSO regression.

In the area of feature selection for biomarker discovery, and depending on the domain of study, the researcher must first ask himself: "what do I want as features?" and "What is a good feature?". The algorithms are usually designed to select a minimal set of uncorrelated and relevant features that will increase the predictive performances of the model to detect *e.g.* a patient state. This approach is the *minimal-optimal problem* (Shi *et al.* , 2018). However another possibility exists. It selects all the relevant features *e.g.* to explain complex biological mechanisms with various varying parameters. This second approach is called the *all-relevant problem* (Shi *et al.* , 2018).

Beside their high dimensionality and inherent noise, metabolomics data are particularly challenging to be used for feature selection: they contain a high number of correlated data since metabolites are spanned across multiple descriptors and they can be correlated to each other. Both cases of feature selection (minimal-optimal and all-relevant) could therefore be interesting for biomarker discovery with these data. In this research domain, Partial Least Squares (PLS, Abdi, 2010) and Orthogonal PLS (OPLS, Trygg & Wold, 2002) regression are the current preferred method to build a prediction model and perform feature selection. A review of existing filter, wrapper and embedded selection approaches that have been combined with PLS is given in Mehmood *et al.* (2012). Among the most popular selection techniques are filters based on the regression coefficients or the Variable Importance in Projection (VIP). Other common approaches are embedded techniques where PLS is constrained by a form of regularisation.

## 1.2 Objectives

Recent articles suggested various feature selection approaches in metabolomics (see for example Rinaudo *et al.* (2016), Grissa *et al.* (2016) or Shi *et al.* (2018)). Nevertheless, they are all based on experimental datasets for which the exact identification of features cannot be exactly verified.

Similarly to Rousseau (2011) and Christin *et al.* (2013), the intent of this paper is to compare methods performances based on a list of known discriminant features and under different realistic conditions: different sample sizes and different degrees of noise in the signal.

To that purpose, we used spectra derived from a large set of rat "control" urine samples, and we artificially altered some of them. Using such baseline spectra renders possible a genuine extension to experimental data. In this work, resampling techniques are performed to evaluate the true performances from a statistical point of view and undercover the methods properties. It further enables to study inter and

intra-method the features and classifiers parameters stabilities as well as prediction accuracies that can be extended to other feature selection techniques.

With regards to the methods to be compared, we decided to focus, on the one hand, on filters associated with PLS and OPLS and on the other hand on embedded techniques considered as the most natural extensions to the original PLS regression: Sparse-PLS (SPLS, Chun & Keleş, 2010) and Light Sparse OPLS (LSOPLS, Féraud *et al.* , 2017).

Knowing that classifier performances are heavily dependent on the dataset at hand, this paper does not intent to give general recommendations on which supervised approach should be routinely used. It rather gives insights on how are behaving the compared methods, on keys to select an appropriate method, and on how to evaluate them in the scope of feature selection.

### 1.3 Content

This paper is organised as follows: Section 2 describes first the Semi-artificial database, then Section 3 reviews the methods that are used to find potential features of interest in the spectral profiles. The next part, Section 4 describes how the Meta Parameters (MP) are tuned to train the classifiers. Section 5 presents classic data analysis results by applying the previously described methods onto one subsample. Afterward Section 6 presents a generic evaluation of those methods based on resampled sub-datasets and multiple criteria. Finally, Section 7 ends this article by concluding remarks.

## 2 The Semi-artificial urine database

The Semi-Artificial Urine database is fully described in Rousseau (2011). In this dataset, random artificial alterations were added to control rat urine spectra in order to assess the quality of potential biomarkers' identification techniques when the biomarkers' location is known in advance. This raw dataset is issued from the COMET database (Lindon *et al.* , 2003). The final dataset is composed of a spectral data matrix of size ($960 \times 500$) where half of the spectra were altered. The other half of the data are original spectra. It also includes a binary vector characterising the presence of these alterations and it corresponds to the class to be predicted. Details about pre-processing and alterations of the spectra are given in Rousseau (2011).

The semi-artificial dataset involves 46 artificially modified descriptors defined as true biomarkers separated into 10 different areas of the spectrum (*cf.* Figure 1), some of these are correlated with each other, leaving a total of 6 independent groups of biomarkers. Furthermore, each spectrum is only altered by 2 out of the 6 randomly chosen independent biomarkers, to mimic various responses from one organism to another. Alterations on the left side of the spectra (ranging from 7.26 to 6.48 ppm) are located in a "noise-free" area and have a low intensity. In contrast, alterations present on the right side of the spectral window (6.48 to 3.37 ppm) are located in a noisy area but have a higher intensity.

Due to the alteration, built-in correlation is then present between and within them: correlation between descriptors of the same peak but also correlated pairs of alterations since the addition of alterations was designed to always add 2 different biomarkers in each altered spectrum. This is intended to mimic correlation between different peaks of a same molecule or between peaks of correlated metabolites.

## 3 Methods

Consider a data matrix $\boldsymbol{X}$ with $n$ observations and $m$ variables, and a univariate response vector $\boldsymbol{y}$ of size $(n \times 1)$ representing a binary class indicator $\{0, 1\}$ in the context of a discrimination problem. Note that to simplify the mathematical notations, we will now consider $\boldsymbol{X}$ and $\boldsymbol{y}$ as being their mean-centred-by-column equivalents.

Every method is described below with its statistical properties and algorithms, as well as how they perform feature selection in this context. Note that they are all presented in the case of a single $\boldsymbol{y}$ response vector but could easily be generalised to the multivariate case of a matrix response $\boldsymbol{Y}$.

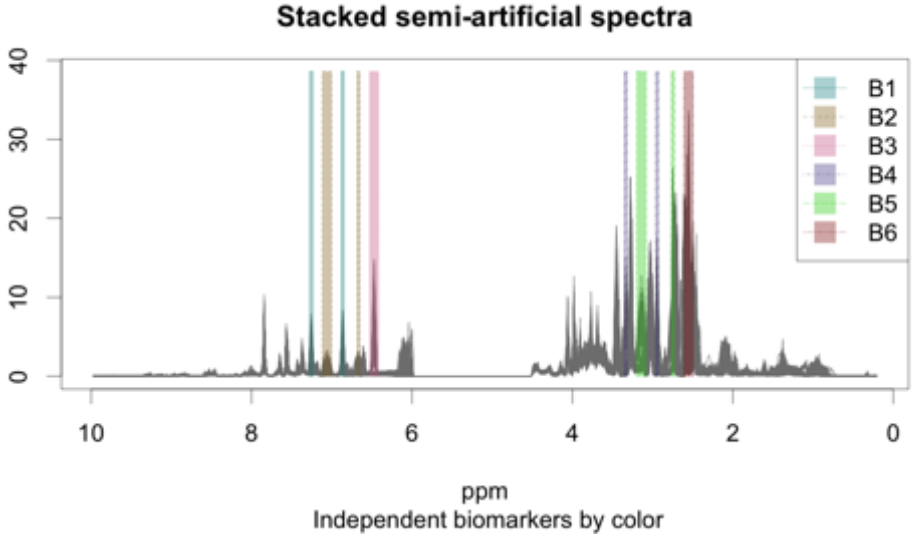


Figure 1: Stacked representation of the spectral profiles in the Semi-Artificial Urine database. The 6 alterations' locations are outlined with the colour boxes.

From a single classifier, there can be multiple feature selection procedures. For each of them, we will define a *Feature Score* vector $\boldsymbol{B}$ of size $(1 \times m)$ as the value on which the decision to select or not the variables will be based. Then a decision rule will be defined to determine the set of retained variables as the active variables – the *features* – that will be contained in $\mathcal{A}$ of size $(1 \times m^*)$ with $m^*$ the number of retained variables.

The distinction between filters and embedded methods will disappear altogether, in favour of a clearer homogenised definition of what is stored in $\boldsymbol{B}$ and the decision rules to determine the variables in $\mathcal{A}$. They will be explained in each methods section and resumed in Table 6.

In Section 6, they are later compared to each other through a double loop scheme resampling method.

## 3.1 Multiple t-tests

### 3.1.1 Description

Univariate t-tests are applied as benchmark method (as in Rousseau *et al.* (2008) or Christin *et al.* (2013)) to be compared against the multivariate classifiers. Though this is not a classification technique *per se*, the test statistic and its p-value are good indicators of the differences between two classes of observations for a given variable. This test is still widely applied in bioinformatics applications, typically to measure differential expression in Microarray Data (Jeanmougin *et al.* , 2010; Kuyuk & Ercan, 2017), single-cell RNA-Seq (Poirion *et al.* , 2016), etc.

The test statistic is applied in parallel for each spectral descriptor. It measures the difference in means of two different samples and, based on that value, one can formally test if the difference is significant.

Because several tests are performed simultaneously, the overall Type I error is inflated. Hence, the derived p-values need a correction factor for multiple tests. The Benjamini-Yekutieli (B-Y) approach (Benjamini & Yekutieli, 2001) controls the False Discovery Rate (FDR) under arbitrary dependency assumptions, which is appropriate in our case since spectral descriptors are correlated. It provides a less conservative control of Type I errors, on the contrary to procedures used to control the family-wise error rate such as the Bonferroni correction.

### 3.1.2 Statistical methodology

For each descriptor $i = 1, \ldots, m$, the two-sample Welch's t-statistic for unequal variances is:

$$\delta_i = \frac{\bar{x}_{0i} - \bar{x}_{1i}}{\sqrt{s_{0i}^2/n_0 + s_{1i}^2/n_1}} \; ; \tag{1}$$

where $n_0$ and $n_1$ are the sample sizes of each class, $\bar{x}_{0j}$ and $\bar{x}_{1j}$ are the sample means, and $s_{0j}^2$ and $s_{1j}^2$ are the sample variances.

Under $H_0$, the Welch's t-statistic follows a Student's t-distribution $\tau_\nu$ with $\nu$ degrees of freedom approximated by the Welch-Satterthwaite equation[1]. The corresponding p-values are calculated for each variable as: $\rho_i = 2P(\tau_\nu > |\delta_i|)$ with $i = 1, \ldots, m$.

The B-Y FDR controlling procedure is the following: $\boldsymbol{\rho}$ is the vector of $p$-values. The B-Y rule to select significant test statistics consists in finding the index $m^*$ such that:

$$m^* = \max \left\{ i : \rho_{(i)} \leq \frac{iq}{m \sum_{i=1}^{m} 1/i} \;\text{ for }\; i = 1, \ldots, m \right\} ; \tag{2}$$

where $q$ is the FDR to be controlled (usually set to 10 or 5 %) and $(i)$ corresponds to the index $i$ of ordered $\boldsymbol{\rho}$ in descending order. It rejects the null hypotheses for ordered $i = 1, \ldots, m^*$. Alternatively, the corrected p-value $\boldsymbol{\rho}_{fdr}$ can be directly compared to the FDR level $q$.

### 3.1.3 Feature selection

If based on the test statistic (t-stat), the Feature score is $\boldsymbol{B}^{t-stat} = |\boldsymbol{\delta}|$ and an arbitrary threshold $m^*$ determines which descriptors are included in the selection set: $\mathcal{A}^{t-stat} = \{i : |\delta|_{(i)} \geq |\delta|_{(m^*)}\}$. On the other hand, if the decision rule relies on the p-value (t-fdr), the Feature score remains the same $(\boldsymbol{B}^{t-fdr}) = |\boldsymbol{\delta}|$ but only variables with $\boldsymbol{\rho}_{fdr} \leq 0.05$ are kept: $\mathcal{A}^{t-fdr} = \{i : \rho_{(i),fdr} \leq 0.05\}$.

## 3.2 Partial Least Squares

### 3.2.1 Description

The original Partial Least Squares (PLS) regression as well as the Nonlinear estimation by Iterative PArtial Least Squares (NIPALS) algorithm were first introduced by Herman Wold in the 1960s-1970s (Geladi, 1988). They were later combined in Sjöström *et al.* (1983) to solve ill-conditioned linear regression problems with a high number of multi-collinear variables. PLS can be defined as a combinaison of dimension reduction and a prediction tool that is, as for Principal Component Analysis (PCA), aimed at extracting a small number of orthogonal Latent Variables (LV) – also called Predictive Components (PrC) here – that are linear combinations of the original variables to best resume the available information. Unlike PCA that only maximises the $\boldsymbol{X}$ variance explained, here both explained variances of $\boldsymbol{X}$ and $\boldsymbol{y}$ are maximised, as well as the correlation between these two matrices.

Besides the original NIPALS algorithm, several others have been suggested in the literature (see Varmuza & Filzmoser (2016) for more details about the most common ones and Andersson (2009) for a comparison between various algorithms) such as Kernel PLS (Lindgren *et al.* , 1993), Canonical Powered PLS (Indahl *et al.* , 2009) or Statistically Inspired Modification of PLS (SIMPLS) (de Jong, 1993) algorithms, all with varying scores and loadings properties and different ways to find the solution to the maximisation problem.

Here, a distinction arises between having a single or multivariate response (Boulesteix & Strimmer, 2006). In the case of a single response vector, the method is called PLS1. In contrast, to predict one or multiple responses, the two main variants are: PLS2 (obtained usually with NIPALS or Kernel-PLS

[1] This equation estimates the effective pooled degrees of freedom of a linear combination of independent sample variances: $\frac{(\frac{s_{0i}^2}{n_0} + \frac{s_{1i}^2}{n_1})^2}{(\frac{s_{0i}^4}{n_0^2(n_0-1)} + \frac{s_{1i}^4}{n_1^2(n_1-1)})}$.

algorithms) and SIMPLS. Both multivariate versions are equivalent, though PLS2 has different and less intuitive contraints on the $X$-weights (ter Braak & de Jong, 1998). SIMPLS can also be seen as the direct multivariate extension of PLS1, as its shares the same criterion and the same constraints (Boulesteix & Strimmer, 2006).

PLS is therefore a widely spread method for modelling relations between dependent ($\boldsymbol{y}$) and independent ($\boldsymbol{X}$) variables (see Mehmood & Ahmed (2016) for examples of applications). The technique is also popular and extensively used in chemometrics and metabolomics (Wold *et al.* , 2001; Antti *et al.* , 2004) where it has proven its usefulness and efficiency to underline metabolic changes in biofluids *e.g.* due to toxicity or disease process. This technique was developed so as to better handle the overdetermined regression problems usually addressed with Principal Component Regression (PCR), where one needs to predict a set of dependent variables from a very large set of correlated predictors (Abdi, 2010), resulting in a much higher degree of multicollinearity, which is almost systematically the case in the field of metabolomics. Indeed, PLS is an improvement of the classical Multiple Linear Regression and PCR as the method is more robust in its parameters estimation (Geladi & Kowalski, 1986) and is explicitly built with the incentive to predict $\boldsymbol{y}$.

Originally, PLS was not intended to solve classification problems. Nevertheless, Barker & Rayens (2003) showed that the method is well suited to fit that particular application. It gives good separation results emphasising their benefits compared to unsupervised methods such as PCA[2]. In this case, the response variables are coded with binary values, typically $\{0, 1\}$. When the class response has $F > 2$ levels, $\boldsymbol{y}$ is converted into a series of $F$ binary vectors $\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_F$ gathered in a matrix of size $(n \times F)$, such that for $i = 1, \ldots, n$ and $f = 1, \ldots, F$:

$$\tilde{y}_{if} = \begin{cases} 1 & \text{if } y_i = f \\ 0 & \text{otherwise.} \end{cases}$$

### 3.2.2 Statistical methodology

#### Principles

PLS is fundamentally a dimension reduction technique based on a basic latent structure of the data (Singular Value Decomposition), that is built for each block of data $\boldsymbol{X}$ and $\boldsymbol{y}$ and between them.

Its purpose is to find the final linear relationship between $\boldsymbol{X}$ and $\boldsymbol{y}$ such that $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^{pls} + \boldsymbol{e}$ with the regression coefficients vector $\boldsymbol{\beta}^{pls}$ of size $(m \times 1)$ and an error vector $\boldsymbol{e}$ of size $(n \times 1)$. To that purpose, PLS will build $a$ new orthogonal variables (the LV estimates), with corresponding $X$-scores, $\boldsymbol{t}_j$ with $j = 1, \ldots, a$, which are linear combinations of the original $\boldsymbol{X}$ variables, as well as good predictors of $\boldsymbol{y}$. The model components are extracted sequentially: every iteration deflates the variability from the data that is associated with the extracted component. The remaining information is used to estimate the next component, etc.

$X$ can be decomposed into scores, loadings and weights with the following relations:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}' + \boldsymbol{E}_X \; ; \tag{3}$$

where $\boldsymbol{T}$ is the $(n \times a)$ $X$-scores matrix, $\boldsymbol{P}$ is the $(m \times a)$ $X$-weights matrix. Their product is considered a good "summary" of $\boldsymbol{X}$ if one obtains small $X$-residuals $\boldsymbol{E}_{X(n \times m)}$.

In the scope of estimating orthogonal $X$-scores, $\boldsymbol{T}$ are defined as linear combinations of the original variables $\boldsymbol{X}$ with $\boldsymbol{W}_{(m \times a)}$ the $X$-weights matrix (also denoted as the set of *direction vectors*):

$$\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W}. \tag{4}$$

Calculating those weights represents the key computation of the PLS methodology.

[2]Since the dimension reduction of PLS is guided by the variability existing among groups rather than by the total variability of the data for PCA.

During the estimation process, $\boldsymbol{P}$ is estimated by OLS regression of $\boldsymbol{T}$ on $\boldsymbol{X}$: $\boldsymbol{P'} = (\boldsymbol{T'T})^{-1}\boldsymbol{T'X}$.

Similarly, $\boldsymbol{y}$ is defined as the addition of summarised information (through scores and loadings) and an error terms:

$$\boldsymbol{y} = \boldsymbol{Uq} + \boldsymbol{e}_y \; ; \tag{5}$$

where $\boldsymbol{U}$ is the $(n \times a)$ matrix of $y$-scores, $\boldsymbol{q}$ is the $(a \times 1)$ vector of $y$-loadings and $\boldsymbol{e}_y$ is the $(n \times 1)$ vector of $y$-residuals.

For the same technical reasons as in Eq. (4), the $y$-weights vector $\boldsymbol{c}$ of size $(1 \times a)$ needs to be introduced: $\boldsymbol{U} = \boldsymbol{yc}$.
These weights are estimated by OLS regression of $\boldsymbol{T}$ on $\boldsymbol{y}$: $\boldsymbol{c} = (\boldsymbol{T'T})^{-1}\boldsymbol{T'y}$.

The two data tables are linked by the following equation:

$$\boldsymbol{y} = \boldsymbol{Tc} + \boldsymbol{e} = \boldsymbol{XWc} + \boldsymbol{e} = \boldsymbol{X}\boldsymbol{\beta}^{pls} + \boldsymbol{e}. \tag{6}$$

And therefore, regression coefficients are estimated as: $\boldsymbol{b}^{pls} = \boldsymbol{Wc} = \boldsymbol{W}(\boldsymbol{T'T})^{-1}\boldsymbol{T'y}$.

PLS can be seen as a constrained maximisation problem to derive direction vectors $\boldsymbol{w}_j$. In the case of one response vector, the objective function is:

$$\begin{aligned} \boldsymbol{w}_j &= \text{argmax}_{\boldsymbol{w}} \boldsymbol{w'X'yy'Xw} \;\text{ for }\; j = 1, \ldots, a \; ; \\ &= \text{argmax}_{\boldsymbol{w}}\; cov^2(\boldsymbol{Xw}, \boldsymbol{y}) \;\text{ for }\; j = 1, \ldots, a \; ; \\ &= \text{argmax}_{\boldsymbol{w}}\; cor^2(\boldsymbol{Xw}, \boldsymbol{y}) var(\boldsymbol{Xw}) var(\boldsymbol{y}) \;\text{ for }\; j = 1, \ldots, a \; ; \end{aligned} \tag{7}$$

Hence, it is clear that the direction vectors contained in $\boldsymbol{W}$ seek to capture most of the $X$-variance as well as a high correlation between $\boldsymbol{X}$ and $\boldsymbol{y}$.

Note that the space spanned by the direction vectors $\boldsymbol{w}_j$ is an orthogonal basis for the Krylov[3] subspace: $K_a(\boldsymbol{X'X}, \boldsymbol{X'y})$. It implies that PLS estimator can be seen as the solution of the following optimisation problem (Rosipal & Krämer, 2006): $\boldsymbol{b} = \text{argmin}_{\boldsymbol{b}} \parallel \boldsymbol{y} - \boldsymbol{Xb} \parallel^2$, subject to $\boldsymbol{b} \in K_a(\boldsymbol{X'X}, \boldsymbol{X'y})$. PLS can therefore be viewed as a regularised least squares fit or shrinkage estimator in a similar way than Ridge (Krämer, 2007). The directions of the estimator that are responsible for a high variance are indeed shrunk. This introduces a biais, which is hopefully not too large and compensated by a lowered variance, hence a lowered Mean Square Error of the estimators. Therefore, compared to classic Ordinary Least Squares (OLS) estimators for $\boldsymbol{\beta}$, PLS leads to lowered estimated coefficients.

**Details on the algorithms**

Recall that in PLS, the properties of the scores, loading, etc. are subject to the choice of the applied algorithm. NIPALS and SIMPLS are reviewed here in details as they will be used to build our classifiers.

Here are a few remarks concerning these two algorithms:

- The scores vectors are usually orthogonal to each other but not necessarily normalised. Hence, $\boldsymbol{T'T}$ is diagonal. In SIMPLS, they are orthonormal *i.e.* $\boldsymbol{t}_j^{\prime simpls}\boldsymbol{t}_j^{simpls} = 1 \; \forall j = 1, \ldots, a$ and $\boldsymbol{T}^{\prime simpls}\boldsymbol{T}^{simpls} = \boldsymbol{I}_a$.
- The loadings vectors $\boldsymbol{p}$ are usually not orthogonal to each other (if the scores are) and the norm of each $\boldsymbol{p}$ will depends on the algorithm.
- The SIMPLS loading matrix $\boldsymbol{P}^{simpls}$ can be obtained from the NIPALS one as follows: $\boldsymbol{P}^{simpls} = \boldsymbol{P}^{nipals}(\boldsymbol{T}^{\prime nipals}\boldsymbol{T}^{nipals})$.

[3]Krylov methods are iterative methods especially suited for wide and short data tables, used to solve a large system of linear equations by avoiding matrix-matrix computations (Ipsen & Meyer, 1998).

- If the SIMPLS algorithm is used, the weight matrix $\boldsymbol{W}$ that can be applied to the original mean-centred $\boldsymbol{X}$ matrix can be directly obtained, and we omit the $^{simpls}$ notation in this case. If NIPALS is used, $\boldsymbol{W}^{nipals}$ is originally related to the deflated $\boldsymbol{X}$ matrix. To recover $\boldsymbol{W}$ from the original NIPALS weights matrix $\boldsymbol{W}^{nipals}$, the following formula is applied: $\boldsymbol{W} = \boldsymbol{W}^{nipals}(\boldsymbol{P}'^{nipals}\boldsymbol{W}^{nipals})^{-1}$.
- For univariate $\boldsymbol{y}$, NIPALS and SIMPLS are equivalent to each other (de Jong, 1993; ter Braak & de Jong, 1998).

**NIPALS algorithm**

NIPALS algorithm has been the first algorithm to solve the PLS problem. Scores and loadings are obtained iteratively component-wise with the deflation of the $\boldsymbol{X}$ matrix at the end of each iteration. The latter is a crucial step in order to obtain orthogonal scores vectors (Varmuza & Filzmoser, 2016).

Its corresponding contraints to be applied to the optimisation function in Equ. (7) are:

$$\boxed{\text{Subject to } \boldsymbol{w}_j'(\boldsymbol{I}_m - \boldsymbol{W}\boldsymbol{W}^+)\boldsymbol{w}_j = 1 \text{ and } \boldsymbol{t}_j'\boldsymbol{t}_k = \boldsymbol{w}_j'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{w}_k = 0 \text{ for } j = 1, \ldots, a \text{ and } k = 1, \ldots, j-1; \text{ where } \boldsymbol{W}^+ \text{ is the unique Moore-Penrose inverse of } \boldsymbol{W}.} \tag{8}$$

NIPALS algorithm for univariate $\boldsymbol{y}$ is detailed below:

Table 1: NIPALS algorithm for a single response

| | | |
|---|---|---|
| **A.** | **Extract PLS components** | |
| (1) | Initialisation: $\boldsymbol{X}_1 = \boldsymbol{X}$ | Start from the initial mean-centred data matrix $\boldsymbol{X}$ to be deflated. Recall that $\boldsymbol{y}$ is also mean-centred. |
| (2) | Iterate for $j = 1, \ldots, a$ : | |
| (2.1) | $\boldsymbol{w}_j = \boldsymbol{X}_j'\boldsymbol{y}(\boldsymbol{y}'\boldsymbol{y})^{-1}$ | Calculate the weights vector $\boldsymbol{w}_j$ by regressing $\boldsymbol{X}_j$ on $\boldsymbol{y}$ |
| (2.2) | $\boldsymbol{w}_j = \boldsymbol{w}_j(\boldsymbol{w}_j'\boldsymbol{w}_j)^{-1/2}$ | Normalize $\boldsymbol{w}_j$ |
| (2.3) | $\boldsymbol{t}_j = \boldsymbol{X}_j\boldsymbol{w}_j$ | Calculate scores vector $\boldsymbol{t}_j$ as a linear combination of original variables |
| (2.4) | $\boldsymbol{p}_j = \boldsymbol{X}_j'\boldsymbol{t}_j(\boldsymbol{t}_j'\boldsymbol{t}_j)^{-1}$ | Calculate corresponding loadings vector by regressing $\boldsymbol{X}$ on $\boldsymbol{t}_j$ |
| (2.5) | $\boldsymbol{X}_{j+1} = \boldsymbol{X}_j - \boldsymbol{t}_j\boldsymbol{p}_j'$ | Deflate matrix $\boldsymbol{X}_j$ |
| **B.** | **Build global matrices** | |
| | $\boldsymbol{T} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_a)$ | The $(n \times a)$ matrix of scores |
| | $\boldsymbol{P} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_a)$ | The $(m \times a)$ matrix of loadings |
| | $\boldsymbol{W}^{nipals} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_a)$ | The $(m \times a)$ "deflated" form of matrix of weights |
| | $\boldsymbol{W} = \boldsymbol{W}^{nipals}(\boldsymbol{P}'\boldsymbol{W}^{nipals})^{-1}$ | The $(m \times a)$ "undeflated" form of the weights matrix |
| **C.** | **Calculate the regression coefficients** | |
| | $\boldsymbol{b}^{pls} = \boldsymbol{W}(\boldsymbol{T}'\boldsymbol{T})^{-1}\boldsymbol{T}'\boldsymbol{y}$ | |

Note that the NIPALS scores matrix $\boldsymbol{T}$ can be directly calculated from non deflated matrix $\boldsymbol{X}$ by using the "undeflated" form of the weight matrix: $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W}$.

**SIMPLS algorithm**

The strength of SIMPLS is that it is not just a series of iteration steps, as it was first developed to solve an optimisation problem. In that regard, SIMPLS has a direct statistical meaning. It has further advantages of computational speed and direct computation of $\boldsymbol{T}$ in terms of linear combinaisons of the $\boldsymbol{X}$ matrix, and we will now focus on this algorithm.

Its corresponding contraints on the optimisation function are:

$$\boxed{\text{Subject to } \boldsymbol{w}_j'\boldsymbol{w}_j = 1 \text{ and } \boldsymbol{t}_j'\boldsymbol{t}_k = \boldsymbol{w}_j'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{w}_k = 0 \text{ for } j = 1,\ldots,a \text{ and } k = 1,\ldots,j-1.} \quad (9)$$

After the estimation of $\boldsymbol{w}_j$ and once the latent variables $\boldsymbol{t}_j$ are estimated, the loadings $\boldsymbol{p}_j$ can be calculated via OLS regression: $\boldsymbol{t}_j = \boldsymbol{X}\boldsymbol{p}_j + \boldsymbol{e}$.

The entire SIMPLS algorithm for univariate $\boldsymbol{y}$ is detailed here:

Table 2: PLS1/SIMPLS algorithm for a single response

| | | |
|---|---|---|
| **A.** | **Extract PLS components** | |
| (1) | Initialisation: $\boldsymbol{s}_1 = \boldsymbol{X}'\boldsymbol{y}$ | Start from the cross-product of mean-centred $\boldsymbol{X}$ and $\boldsymbol{y}$ matrices to be deflated *i.e.* their initial covariance vector |
| (2) | Iterate for $j = 1,\ldots,a$ : | |
| | (2.1) $\boldsymbol{w}_j = \boldsymbol{s}_j(\boldsymbol{s}_j'\boldsymbol{s}_j)^{-1/2}$ | Calculate the weights vector $\boldsymbol{w}_j$ from the normalised $\boldsymbol{s}_j$ vector |
| | (2.2) $\boldsymbol{t}_j = \boldsymbol{X}\boldsymbol{w}_j$ | Calculate scores vector $\boldsymbol{t}_j$ as a linear combination of the original variables |
| | (2.3) $\boldsymbol{t}_j = \boldsymbol{t}_j(\boldsymbol{t}_j{}'\boldsymbol{t}_j)^{-1/2}$ | Normalize $\boldsymbol{t}_j$ |
| | (2.4) $\boldsymbol{w}_j = \boldsymbol{w}_j(\boldsymbol{t}_j{}'\boldsymbol{t}_j)^{-1/2}$ | Adapt weights $\boldsymbol{w}_j$ to normalised scores |
| | (2.5) $\boldsymbol{p}_j = \boldsymbol{X}'\boldsymbol{t}_j(\boldsymbol{t}_j'\boldsymbol{t}_j)^{-1} = \boldsymbol{X}'\boldsymbol{t}_j$ | Calculate corresponding loadings vector by regressing $\boldsymbol{X}$ on $\boldsymbol{t}_j$ |
| | (2.6) $\boldsymbol{s}_{j+1} = \boldsymbol{s}_j - \boldsymbol{P}_{1:j}(\boldsymbol{P}_{1:j}'\boldsymbol{P}_{1:j})^{-1}\boldsymbol{P}_{1:j}'\boldsymbol{s}_j$ | Deflate covariance vector $\boldsymbol{s}_{j+1}$ as the residuals of the regression of $\boldsymbol{s}_j$ on previous loadings $\boldsymbol{P}_{1:j}$ where $\boldsymbol{P}_{1:j} = (\boldsymbol{p}_1,\ldots,\boldsymbol{p}_j)$ |
| **B.** | **Build global matrices** | |
| | $\boldsymbol{T} = (\boldsymbol{t}_1,\ldots,\boldsymbol{t}_a)$ | The $(n \times a)$ matrix of scores |
| | $\boldsymbol{P} = (\boldsymbol{p}_1,\ldots,\boldsymbol{p}_a)$ | The( $m \times a$) matrix of loadings |
| | $\boldsymbol{W} = (\boldsymbol{w}_1,\ldots,\boldsymbol{w}_a)$ | The $(m \times a)$ matrix of weights |
| **C.** | **Calculate the regression coefficients** | |
| | $\boldsymbol{b}^{pls} = \boldsymbol{W}\boldsymbol{T}'\boldsymbol{y}$ | |

Note that an advantage of SIMPLS is that the weights $\boldsymbol{W}$ are directly related to $\boldsymbol{X}$ (and not calculated from the deflated matrices as in NIPALS) and the deflation Step (2.6) sets by construction the orthogonality constraint on the scores vector $\boldsymbol{t}_j$ to all previous ones extracted.

### 3.2.3 PLS results interpretation

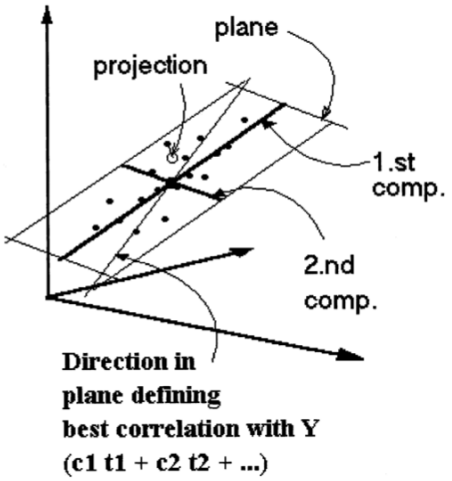


Figure 2: Geometrical representation of a PLS model (Figure from Wold *et al.* (2001)).

The geometrical representation of PLS is depicted in Figure 2: the $\boldsymbol{X}$ matrix is projected onto an $a$-dimensional hyper-plane so as the coordinates of the projection, which turns out to be the $X$-scores, $\boldsymbol{t}_j$, appear to be predictors of $\boldsymbol{y}$.

The scores ($\boldsymbol{t}_j$) indicate how are projected the observations onto the latent variables and their (dis)similarities, while the loadings ($\boldsymbol{p}_j$) indicate the contribution of a particular $\boldsymbol{X}$ variable to the above-mentioned latent variable.

One can also inspect the percentage of information from the response $\boldsymbol{y}$ that is explained by each component of the model $SSY_j$. In the case of PLS1, $SSY_j = \boldsymbol{c}_j^2 \boldsymbol{t}_j' \boldsymbol{t}_j$ for $j = 1, \ldots, a$.

Weight matrices $\boldsymbol{W}$ and $\boldsymbol{c}$ provide information about how the variables are combined in order to form the relation between $\boldsymbol{X}$ and $\boldsymbol{y}$. Furthermore, the $X$-weights $\boldsymbol{w}_j$ serve to express both "positive" and "compensation" correlations respectively between $\boldsymbol{X}$ and $\boldsymbol{y}$ and needed to predict $\boldsymbol{y}$ from $\boldsymbol{X}$.

Regression coefficients $\boldsymbol{b}$, which are independent of the algorithmic details of the method, give a global view summarising the influence of all PLS components over the response variable.

Graphical representations of the weights $\boldsymbol{w}_j$ and $\boldsymbol{c}_j$, loadings $\boldsymbol{p}_j$, scores $\boldsymbol{t}_j$ and the regression coefficients $\boldsymbol{b}$ and combinations of them are therefore expected to allow a comprehensive interpretation of the model.

Finally, the residuals that are not explained by the model can be used for model diagnostics and outliers detection.

### 3.2.4 Feature selection

The first and more common Feature selection with PLS is based on the regression coefficients $\boldsymbol{b}^{pls}$ that can be employed to rank spectral descriptors: $\boldsymbol{B}^{PLS-coef} = |\boldsymbol{b}^{pls}|$. In this case, an arbitrary threshold is set to select the $m^*$ higher absolute coefficients and $\mathcal{A}^{pls-coef} = \{i : |b^{pls}|_{(i)} \geq |b^{pls}|_{(m^*)}\}$. The disadvantage of this measure is the dependence to an arbitrary scale and variance of the $\boldsymbol{X}$ columns.

Another solution would be to rely on the Variable Importance in Projection (VIP) (Farrés *et al.* , 2015) value $\boldsymbol{v}$. This is a well-known criterion applied in PLS-derived techniques for variable selection. It summarises the influence of an individual variable to build the model. They are measured on the basis of a weighted sum of the Sum of Squares for explained variation in the response (SSY). The VIP value for the $i^{th}$ variable is given as:

$$v_i = \sqrt{\frac{m \cdot \sum_{j=1}^{a} [SSY_j \cdot (w_{ij} / \parallel \boldsymbol{w}_j \parallel)^2]}{SSY_{tot}}} \; ; \tag{10}$$

where $w_{ij}$ represents the $X$-weight of variable $i$ and LV $j$, $SSY_j$ is the Sum of Squares of the $y$-variance explained by the $j^{th}$ LV and $SSY_{tot}$ is the total sum of squares of the dependent variable $\boldsymbol{y}$: $SSY_{tot} = \sum_{j=1}^{a} SSY_j$.

Since $(w_{ij} / \parallel \boldsymbol{w}_j \parallel)^2$ represents the importance of the $i^{th}$ variable in the construction of the $j^{th}$ LV, the VIP value is a global measure of the contribution of this variable in the whole PLS model.

Given that the average of the squared VIP scores is equal to 1[4], the rule of thumbs is to select variables having a VIP value greater than 1.

These VIP values will be used as well as features ranking. The decision rule to include a variable $i$ in $\mathcal{A}$ is either based on a predefined number of features $m^*$ to retain (pls-vip), or on the rule of thumbs threshold of 1 (pls-vip1) with $m^* = \max\{i : |v^{pls}| \geq 1\}$ for $i = 1, \ldots, m$. More formally, $\mathcal{A}^{pls-vip} = \{i : |v^{pls}|_{(i)} \geq |v^{pls}|_{(m^*)}\}$ and $\mathcal{A}^{pls-vip1} = \{i : |v^{pls}| \geq 1\}$.

[4]Proof: $\frac{\sum_{i=1}^{m}(v_i)^2}{m} = \frac{m}{m} \cdot \sum_{j=1}^{a} \sum_{i=1}^{m} \frac{[SSY_j \cdot (w_{ij}/\|\boldsymbol{w}_j\|)^2]}{SSY_{tot}} = 1 \cdot \frac{\sum_{j=1}^{a} SSY_j}{SSY_{tot}} \cdot \sum_{j=1}^{a} \sum_{i=1}^{m} (w_{ij} / \parallel \boldsymbol{w}_j \parallel)^2 = 1 \cdot 1 \cdot \sum_{j=1}^{a} 1 = 1$; since $\sum_{i=1}^{m} (w_{ij} / \parallel \boldsymbol{w}_j \parallel)^2 = 1$.

## 3.3 Orthogonal Partial Least Squares

### 3.3.1 Description

Multivariate tools and projection methods such as PLS are widely used in metabolomics, though they remain strongly affected by the occurrence of systematic orthogonal variations - also called structured noise - in $\boldsymbol{X}$ that is not associated with the response variable $\boldsymbol{y}$. Indeed, the structured noise in the data matrix coming from spectrometric measures can affect the precision of a new sample prediction and deteriorate the model robustness at some point (Trygg & Wold, 2002).

To overcome this issue, Wold *et al.* (1998) first developed the idea to build a filter that only removes from the spectral matrix $\boldsymbol{X}$ the portion that is orthogonal to the dependent variable and thus obtain more concise and interpretable models. In this first investigation, they developed a filter called Orthogonal Signal Correction (OSC). Several different algorithms have since been proposed and compared, in particular in Svensson *et al.* (2002). The latter concluded that OSC filters do not improve predictive abilities but do lead to more parsimonious models, with an easier interpretation of the filtered data. An additional and interesting advantage is the possibility to analyse the structured noise that is extracted from the data.

One well known OSC variant, called Orthogonal Projections to Latent Structures or Orthogonal PLS (OPLS) has emerged in chemometrics. It has been first introduced by Trygg & Wold (2002) and can be seen as a preprocessing technique or as part of an integrated filter in a classic PLS model. OPLS extends the original NIPALS-PLS regression algorithm, which is already known for it usefulness in multiple application areas thanks to the fact that it can cope with variables collinearity, noise in the data and can deal with a moderate amount of missing values. Indeed, a filtering step is added to remove the $y$-orthogonal variation from the $X$-scores. Thus, the variance of interest is extracted into a single predictive component from the variance that is unrelated (or orthogonal, defined by multiple Orthogonal Components (OC)) to the response variable. This approach results in a rotation of the standard PLS model so that the variance of interest is focused into the predictive component and the variance unrelated to the tested hypothesis is filtered into orthogonal components.

A comparison of geometrical representations of PLS and O-PLS models in a 2D-plane with two components can be seen on Figure 3 where the OPLS model is a rotation of the PLS components.

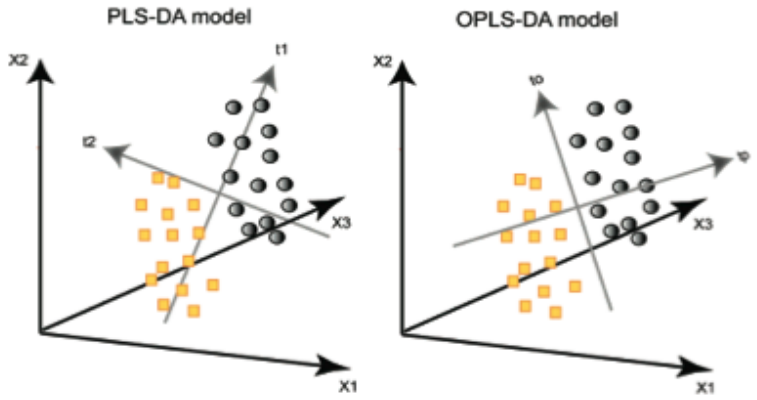


Figure 3: Comparison between geometrical representations of PLS and OPLS models (Figure from Wiklund *et al.* (2008)). $\boldsymbol{t}_p$ represents the predictive component while $\boldsymbol{t}_o$ is a $y$-orthogonal component.

Similarly to PLS, OPLS can also be applied to discriminant problems (Bylesjö *et al.* , 2006). In this case, the between-class variation resides in the predictive component whereas the within-class variation is contained in the $y$-orthogonal components.

### 3.3.2 Statistical methodology

As for PLS, OPLS creates $a$ new orthogonal variables that are the estimates of the latent variables and correspond to the $X$-scores, $\boldsymbol{t}_j$, which are linear combinations of the original $\boldsymbol{X}$ variables for $j = 1, \cdots, a$. Among the $X$-scores, only the last one is predictive, $\boldsymbol{t}_p$, and hence a good predictor of $\boldsymbol{y}$. The other $X$-scores $\boldsymbol{t}_{jo}$ do not contain information from $\boldsymbol{y}$ and are designed as $y$-orthogonal $X$-scores for $j = 2, \cdots, a-1$.

The final predictive model is:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^{opls} + \boldsymbol{e} = \boldsymbol{X}_p\boldsymbol{\beta}_p^{opls} + \boldsymbol{e}\ ; \tag{11}$$

where $\boldsymbol{X}_p$ is the filtered $\boldsymbol{X}$ data matrix and $\boldsymbol{\beta}_p^{opls}$ are the predictive regression coefficients linked to $\boldsymbol{X}_p$.

As for PLS, the OPLS model can also be seen as a set of relations that define either separately $\boldsymbol{X}$ and $\boldsymbol{y}$ or their relationship to each other.

The information contained in the dependent variable $\boldsymbol{y}$ is used to break down the matrix $\boldsymbol{X}$ into three different blocks: one that contains the common variation, another that only involves variation only for $\boldsymbol{X}$, *i.e.* the structured noise, and the last part corresponding to the residual variance (Bylesjö *et al.* , 2006).

$$\boldsymbol{X} = \boldsymbol{t}_p\boldsymbol{p}_p{}' + \boldsymbol{T}_{ortho}\boldsymbol{P}'_{ortho} + \boldsymbol{E}_X; \tag{12}$$

with the predictive and orthogonal $X$-scores *resp.* $\boldsymbol{t}_p$ of size $(n \times 1)$ and $\boldsymbol{T}_{ortho}$ of size $(n \times (a-1))$. Note that $\boldsymbol{T}_{ortho} \cdot \boldsymbol{y} = 0$, *i.e.* the orthogonal $X$-scores are orthogonal to $\boldsymbol{y}$. The predictive scores $\boldsymbol{t}_p$ are estimated on the basis of the predictive $X$-weights: $\boldsymbol{t}_p = \boldsymbol{X}\boldsymbol{w}_p$.

Similarly, $\boldsymbol{y}$ is defined as the addition of summarised information and an error terms:

$$\boldsymbol{y} = \boldsymbol{u}q + \boldsymbol{e}_y; \tag{13}$$

Where $\boldsymbol{u}$ is the $(n \times 1)$ vector of $y$-scores, $q$ is here a constant term (due to the single $\boldsymbol{y}$ and single predictive component) and corresponds to the $y$-loading and $\boldsymbol{e}_y$ of size $(n \times 1)$ is the vector of $y$-residual.

Since $\boldsymbol{t}_p$ is a good predictor of $\boldsymbol{y}$, we have the following inter-block relationship (as in Eq. (6) for PLS):

$$\boldsymbol{y} = \boldsymbol{t}_p\boldsymbol{c} + \boldsymbol{e} = \boldsymbol{X}_p\boldsymbol{\beta}_p^{opls} + \boldsymbol{e} = \boldsymbol{X}\boldsymbol{\beta}^{opls} + \boldsymbol{e} \tag{14}$$

$\boldsymbol{\beta}_p^{opls}$ can now be estimated as: $\boldsymbol{b}_p^{opls} = \boldsymbol{w}_p\boldsymbol{c} = \boldsymbol{w}_p\boldsymbol{y}'\boldsymbol{t}_p(\boldsymbol{t}'_p\boldsymbol{t}_p)^{-1}$. To predict a new unfiltered centred observation, the regression coefficients need to be back-transformed as: $\boldsymbol{b}^{opls} = (\boldsymbol{I}_m - \boldsymbol{W}_o\boldsymbol{P}'_o)\boldsymbol{b}_p^{opls}$, where $\boldsymbol{P}'_o$ and $\boldsymbol{W}_o$ will be described bellow.

#### OPLS algorithm

The OPLS algorithm is made of 2 different operating blocks : (1) the orthogonalisation of $\boldsymbol{X}$ by extracting step by step one or several orthogonal components and (2) the derivation of predictive components based on the corrected predictive matrix $\boldsymbol{X}_p$ as in a regular PLS model. In the case of a univariate $\boldsymbol{y}$, only one predictive PLS component is extracted in the final model.

Its algorithm is based on slight modifications of the original NIPALS-PLS algorithm and is fully described in Table 3.

Note that orthogonal scores $\boldsymbol{T}_o$ can be directly calculated from the non deflated matrix $\boldsymbol{X}$ by using the "undeflated" form of the orthogonal weight matrix: $\boldsymbol{T}_o = \boldsymbol{X}\boldsymbol{W}_o$ and $\boldsymbol{P}_o = \boldsymbol{X}_o{}'\boldsymbol{T}_o(\boldsymbol{T}_o{}'\boldsymbol{T}_o)^{-1}$.

Table 3: OPLS algorithm for a single response

| | | |
|---|---|---|
| **A.** | **Orthogonalisation step** | Compute the part of $\boldsymbol{X}$ orthogonal to $\boldsymbol{y}$ where both matrices are mean-c |
| (1) | Initialisation: $\boldsymbol{X}_1 = \boldsymbol{X}$ | Start from the initial mean-centred data matrix $\boldsymbol{X}$ to be deflated |
| (2) | Iterate for $j = 1, \ldots, (a-1)$ | First 4 steps are identical than the NIPALS-PLS algorithm |
| (2.1) | $\boldsymbol{w}_j = \boldsymbol{X}_j'\boldsymbol{y}(\boldsymbol{y}'\boldsymbol{y})^{-1}$ | Calculate weight vector $\boldsymbol{w}_j$ by regressing $\boldsymbol{X}_j$ on $\boldsymbol{y}$ |
| (2.2) | $\boldsymbol{w}_j = \boldsymbol{w}_j(\boldsymbol{w}_j{}'\boldsymbol{w}_j)^{-1/2}$ | Normalize $\boldsymbol{w}_j$ |
| (2.3) | $\boldsymbol{t}_j = \boldsymbol{X}_j\boldsymbol{w}_j$ | Calculate score vector $\boldsymbol{t}_j$ as a linear combination of original variables |
| (2.4) | $\boldsymbol{p}_j = \boldsymbol{X}_j'\boldsymbol{t}_j(\boldsymbol{t}_j'\boldsymbol{t}_j)^{-1}$ | Calculate corresponding loading vector by regressing $\boldsymbol{X}_j$ on $\boldsymbol{t}_j$ |
| (2.5) | $\boldsymbol{w}_{oj} = \boldsymbol{p}_j - (\boldsymbol{w}_j{}'\boldsymbol{w}_j)^{-1}(\boldsymbol{w}_j{}'\boldsymbol{p}_j)\boldsymbol{w}_j$ | Calculate orthogonal weight vector as the residual of the regression of $\boldsymbol{p}_j$ |
| (2.6) | $\boldsymbol{w}_{oj} = \boldsymbol{w}_{oj}(\boldsymbol{w}_{oj}{}'\boldsymbol{w}_{oj})^{-1/2}$ | Normalize $\boldsymbol{w}_{oj}$ |
| (2.7) | $\boldsymbol{t}_{oj} = \boldsymbol{X}_j\boldsymbol{w}_{oj}$ | Calculate orthogonal score vector $\boldsymbol{t}_{oj}$ |
| (2.8) | $\boldsymbol{p}_{oj} = \boldsymbol{X}_j'\boldsymbol{t}_{oj}(\boldsymbol{t}_{oj}'\boldsymbol{t}_{oj})^{-1}$ | Calculate corresponding orthogonal loading vector by regressing $\boldsymbol{X}_j$ on |
| (2.9) | $\boldsymbol{X}_{j+1} = \boldsymbol{X}_j - \boldsymbol{t}_{oj}\boldsymbol{p}_{oj}'$ | Deflate matrix $\boldsymbol{X}_j$ by removing the orthogonal component $j$<br>And go to step(1) to derive the next orthogonal component |
| **B.** | **Build global matrices** | Compute global matrices and $\boldsymbol{X}_p$ filtered which will be used to predict |
| | $\boldsymbol{T}_o = (\boldsymbol{t}_{o1}, \ldots, \boldsymbol{t}_{o(a-1)})$ | The $(n \times (a-1))$ matrix of orthogonal scores |
| | $\boldsymbol{P}_o = (\boldsymbol{p}_{o1}, \ldots, \boldsymbol{p}_{o(a-1)})$ | The $(m \times (a-1))$ matrix of orthogonal loadings |
| | $\boldsymbol{W}_o^{nipals} = (\boldsymbol{w}_{o1} \ldots, \boldsymbol{w}_{o(a-1)})$ | The "deflated" form of the $(m \times (a-1))$ matrix of orthogonal weights |
| | $\boldsymbol{W}_o = \boldsymbol{W}^{nipals}(\boldsymbol{P}_o'\boldsymbol{W}^{nipals})^{-1}$ | The "undeflated" form of the orthogonal weight matrix |
| | $\boldsymbol{X}_o = \boldsymbol{T}_o\boldsymbol{P}_o{}'$ | The $(n \times m)$ orthogonalised $\boldsymbol{X}$ matrix |
| | $\boldsymbol{X}_p = \boldsymbol{X} - \boldsymbol{X}_o$ | The "filtered" $\boldsymbol{X}$ matrix without the part orthogonal to $\boldsymbol{y}$ |
| **C.** | **PLS of $\boldsymbol{y}$ on $\boldsymbol{X}_p$** | Compute a PLS with one component on $\boldsymbol{X}_p$ |
| (1) | $\boldsymbol{w}_p^{nipals} = \boldsymbol{X}_p'\boldsymbol{y}(\boldsymbol{y}'\boldsymbol{y})^{-1}$ | Calculate weight vector $\boldsymbol{w}_p$ from deflated X |
| (2) | $\boldsymbol{w}_p^{nipals} = \boldsymbol{w}_p(\boldsymbol{w}_p{}'\boldsymbol{w}_p)^{-1/2}$ | Normalize $\boldsymbol{w}_p$ |
| (3) | $\boldsymbol{t}_p = \boldsymbol{X}_p\boldsymbol{w}_p$ | Calculate prediction score vector $\boldsymbol{t}_p$ |
| (4) | $\boldsymbol{p}_p = \boldsymbol{X}_p'\boldsymbol{t}_p(\boldsymbol{t}_p'\boldsymbol{t}_p)^{-1}$ | Calculate corresponding loading vector |
| (5) | $\boldsymbol{w}_p = \boldsymbol{w}_p^{nipals}(\boldsymbol{p}_p'\boldsymbol{w}_p^{nipals})^{-1}$ | The "undeflated" form of the predictive weight vector |
| (6) | $\boldsymbol{c} = \boldsymbol{y}'\boldsymbol{t}_p(\boldsymbol{t}_p'\boldsymbol{t}_p)^{-1}$ | Regress $\boldsymbol{y}$ on $\boldsymbol{t}_p$ |
| (7) | $\boldsymbol{b}_p^{opls} = \boldsymbol{w}_p\boldsymbol{c}$ | Derive the regression coefficients |
| (8) | $\boldsymbol{b}^{opls} = (\boldsymbol{I}_m - \boldsymbol{W}_o\boldsymbol{P}_o')\boldsymbol{b}_p^{opls}$ | Derive the corrected regression coefficients to be applied to the unfiltered<br>centred $\boldsymbol{X}$ matrix with the "undeflated" orthogonal weights $\boldsymbol{W}_o$ |

### 3.3.3 OPLS results Interpretation

Loadings and scores can be interpreted similarly to PLS except that they are now separated into $y$-predictive and $y$-orthogonal categories of variations of the data matrix $\boldsymbol{X}$. Indeed, the OPLS-DA can effectively set apart the actual direction that discriminates $\boldsymbol{t}_p$ from the $y$-orthogonal directions $\boldsymbol{T}_{ortho}$, which renders the corresponding vector of loadings $\boldsymbol{p}_p$ straightforward to interpret.

Scores and loadings linked to the predictive variation indicate relationships between the spectra on the basis of this predictive component in the model (scores) as well as the contribution of a spectral descriptor in the construction of this component (loadings).

The vectors of orthogonal loadings aim at identifying the remaining structured variation that is associated with high intra-class variability. The orthogonal variation contained in $\boldsymbol{X}_{ortho}$ can further be decomposed by PCA (Trygg & Wold, 2002) since the scores and loadings are no more related to the dependent variable $\boldsymbol{y}$. The PCA on orthogonal variation gives hints about the identity and potential causes of the encountered structured noise in the data.

As for PLS, the weight vectors $\boldsymbol{w}_p$ and $\boldsymbol{c}$ indicate how the variables are combined together to build up the link between $\boldsymbol{X}$ and $\boldsymbol{y}$. Note that the weight vector $\boldsymbol{w}_p$ of the OPLS-pretreated PLS model is equivalent to the first weight of the initial PLS model since $\boldsymbol{w}_p$ is the projection of the $\boldsymbol{X}$ matrix onto $\boldsymbol{y}$ (if single) and orthogonal columns in $\boldsymbol{X}$ do not influence this projection (Trygg & Wold, 2002).

The regression coefficients vector $\boldsymbol{b}^{opls}$ of size $(m \times 1)$ is again referred as a measure of the influence of a particular variable over the response variable $\boldsymbol{y}$.

Eventually, the scatter plot of both covariance (Eq. 15) and correlation (Eq. 16) between the variables and the unique predictive component $\boldsymbol{t}_p$ which is called the *S-plot*, helps visualising the influence of a specific variable in the model construction.

For $i = 1, \ldots, m$:

$$cov(\boldsymbol{t}_p, \boldsymbol{X}_i) = \frac{\boldsymbol{t}_p' \boldsymbol{X}_i}{n-1} \; ; \tag{15}$$

$$corr(\boldsymbol{t}_p, \boldsymbol{X}_i) = \frac{cov(\boldsymbol{t}_p, \boldsymbol{X}_i)}{s_{t_p} s_{X_i}} \; ; \tag{16}$$

where $s_{t_p}$ and $s_{X_i}$ are the estimated standard deviations of *resp.* $\boldsymbol{t}_p$ and $\boldsymbol{X}_i$.

It is meant to take into account the reliability (correlation) and contribution (covariance) of the spectral variables to the model. Hence, it helps to identify effective descriptors in the class separation problem that are considered as potential biomarkers. Indeed, only focusing on descriptors with high absolute covariance will lead to the selection of biomarkers with a high spectral intensity while selecting only on the basis of correlations will lead to biomarkers with very low concentration and increase the risk of false positives *i.e.* Type I error in the identification of descriptors as potential biomarkers (Wiklund *et al.* , 2008).

### 3.3.4 Biomarkers identification

As in PLS, the absolute value of regression coefficients can be used to rank the variables and select the most related ones to variations in the response vector: $\boldsymbol{B}^{opls-coef} = |\boldsymbol{b}^{opls}|$ and $m^*$ is set arbitrarily and $\mathcal{A}^{opls-coef} = \{i : |b^{opls}|_{(i)} \geq |b^{opls}|_{(m^*)}\}$. The VIP value is also considered for variable ranking in OPLS. It is now measured based on the predictive $X$-scores, $X$-weights and $y$-loadings. the feature scores are $\boldsymbol{B}^{opls-vip} = \boldsymbol{B}^{opls-vip1} = \boldsymbol{v}^{opls}$ and the corresponding decision rules from PLS are transposed here in the context of OPLS: $\mathcal{A}^{opls-vip} = \{i : |v^{opls}|_{(i)} \geq |v^{opls}|_{(m^*)}\}$ and $\mathcal{A}^{opls-vip1} = \{i : |v^{opls}| \geq 1\}$.

## 3.4 Sparse PLS

### 3.4.1 Description

Given its potential to select discriminant variables, strong arguments are in favour of a LASSO-type penalisation (Tibshirani, 1996) to affect the PLS regression coefficients.

The usual purpose of variable selection is to pick a restricted set of strongly meaningful potential biomarkers. It enables a better interpretability of the biological study results and displays biomarkers that are really important to take into consideration. The method is considered as an embedded FS method with the advantage of the number of selected features which is no more dependent of an arbitrary threshold parameter, as usually in (O)PLS.

Another rationale for sparse regression is argued in Chun & Keleş (2010) for high dimensional data. The consistency of PLS estimators have been proven in Naik & Tsai (2000) under normality assumptions on $\boldsymbol{X}$ and $\boldsymbol{y}$, consistency of covariances $cov(\boldsymbol{X}, \boldsymbol{y})$ and $cov(\boldsymbol{X}, \boldsymbol{X})$ and an additional condition (see Chun & Keleş (2010) for a detailed description). The latter requires $m$ to be fixed and much smaller than $n$. When dealing with multivariate metabolomics data in practice, this prerequisite is, however, rarely achieved. Chun & Keleş (2010) extended their result to more common usage situations where the sample size is small compared to the number of variables. They investigated estimators consistency for the case where $m$ is allowed to growth at an appropriate rate. Their finding is that additional conditions are necessary for consistent sample covariance estimates: $m$ should grow much slower than $n$. However, it cannot be met in the common paradigm of large $m$ / small $n$ datasets. Therefore, and despite their ability to resume data information by a few LV, the PLS estimators cannot fully handle the curse of dimensionality of such omics data.

In this context, two Sparse Partial Least Squares (SPLS) algorithms have been developed independently and in parallel by Chun & Keleş (2010) and Lê Cao *et al.* (2008), the latter being integrated into the broader MixOmics collaborative project for multi-omics (Rohart *et al.* , 2017). Both techniques are based on a LASSO regularisation technique on the direction vectors (*i.e.* $\boldsymbol{w}_j$, the $X$-weights).

We decided to apply the algorithm from Chun & Keleş (2010), which is directly based on a penalised version of the original SIMPLS optimisation problem. Besides, their method has been proven to ensure interesting theoretical properties, such as the one of Krylov subspace that ensures the convergence of the algorithm.

### 3.4.2 Statistical methodology

Chun & Keleş (2010) observed that noise variables altered the direction vectors $\boldsymbol{w}_j$, creating a form of inconsistency. Therefore, they decided to impose sparsity on those vectors, rather than directly on the regression coefficients.

The SPLS method is based on the SIMPLS algorithm. The constrained optimisation problem from PLS-SIMPLS (Equs. (7) and (9)) is modified accordingly by imposing an additional $L_1$ sparsity constraint as in LASSO regression (James *et al.* , 2013).

$$\boldsymbol{w}_j = \text{argmax}_{\boldsymbol{w}} \boldsymbol{w}'\boldsymbol{X}'\boldsymbol{y}\boldsymbol{y}'\boldsymbol{X}\boldsymbol{w} \quad \text{s.t.} \quad \boldsymbol{w}_j'\boldsymbol{w}_j = 1\ , \quad \boldsymbol{t}_j'\boldsymbol{t}_k = \boldsymbol{w}_j'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{w}_k = 0 \ \text{ and } \ |\boldsymbol{w}_j|_1 \leq \lambda_1\ ; \tag{17}$$

for $j = 1, \ldots, a$, $k = 1, \ldots, j-1$, and $\lambda_1$ is the thresholding parameter that fine-tunes the amount of sparsity.

Based on a generalisation of the SPCA formulation from Zou *et al.* (2006), this criterion can be reformulated with penalties imposed onto a surrogate direction vector $\boldsymbol{w}_j^*$ rather than on $\boldsymbol{w}_j$ directly, while keeping them close to each other. Under this formulation, it ensures the LASSO property of shrinkage to exact zero:

$$\text{argmin}_{\boldsymbol{w},\boldsymbol{w}^*} -\kappa\boldsymbol{w}'\boldsymbol{M}\boldsymbol{w} + (1-\kappa)(\boldsymbol{w}^* - \boldsymbol{w})'\boldsymbol{M}(\boldsymbol{w}^* - \boldsymbol{w}) + \lambda_1|\boldsymbol{w}^*|_1 + \lambda_2|\boldsymbol{w}^*|_2 \quad \text{s.t.} \quad \boldsymbol{w}'\boldsymbol{w} = 1\ ; \tag{18}$$

The first term maximises the covariance between $\boldsymbol{y}$ and $\boldsymbol{X}$, whereas the second term ensures that $\boldsymbol{w}$ and $\boldsymbol{w}^*$ are close enough to each other. The $L_2$ sparsity constraint (with parameter $\lambda_2$) takes care of potential singularities in $\boldsymbol{M}$, as in Ridge regression (James *et al.* , 2013). $\kappa \in [0, 0.5]$ is a fixed parameter set to avoid local minima issues. In SPLS, $\lambda_2$ is set to $\infty$ and the solution to this optimisation problem has the form of a soft thresholding estimator.

**SPLS algorithm**

For the sake of simplicity, the algorithm is explained here for a single $\boldsymbol{y}$ response. In such case, the solution to Eq. (18) is obtained by soft thresholding of $\boldsymbol{w}$:

$$\boldsymbol{w} = (\tilde{\boldsymbol{s}} - \lambda_1/2)\mathrm{I}(\tilde{\boldsymbol{s}} \geq \lambda_1/2)sign(\tilde{\boldsymbol{s}}) = (\tilde{\boldsymbol{s}} - \lambda_1/2)_+ sign(\tilde{\boldsymbol{s}}) \; ; \tag{19}$$

where

$$(x)_+ = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise;} \end{cases}$$

and $\tilde{\boldsymbol{s}} = \frac{\boldsymbol{X}'\boldsymbol{y}}{\|\boldsymbol{X}'\boldsymbol{y}\|}$, which corresponds to the first direction vector of SIMPLS. This solution is now independent from the meta-parameter $\kappa$.

A form of soft thresholded estimators to derive $\boldsymbol{w}$ that only retains components which are greater than a predefined fraction of the maximum component is proposed by the authors:

$$\boldsymbol{w} = (|\tilde{\boldsymbol{s}}| - \eta \cdot \max_{1 \leq i \leq m} |\tilde{\boldsymbol{s}}_i|)_+ sign(\tilde{\boldsymbol{s}}) \; ; \tag{20}$$

where $\eta \in [0, 1]$ is equivalent to the sparsity parameter $\lambda_1$ from Eq. (19).

Note that in the case of a multivariate response matrix $\boldsymbol{Y}$, The solution to Eq. (18) is obtained by alternatively iterating between solving for $\boldsymbol{w}$ for fixed $\boldsymbol{w}^*$ using Lagrange multipliers, and inversely, solving for $\boldsymbol{w}^*$ for fixed $\boldsymbol{w}$ using the LARS algorithm (Efron *et al.* , 2004).

The described objective function to derive the first direction vector could in theory be used to derive the subsequent ones. However, this straightforward approach does not guaranties the Krylov subsequences structure of vectors in $\boldsymbol{W}$ that will ensure the algorithmic convergence of the sparse solution. To achieve this property, the SPLS fits a PLS regression with the non-zero $\boldsymbol{w}_j$ variables (called active variables) $\boldsymbol{X}_{\mathcal{V}}$ with the covariates contained in subspace $\mathcal{V}$. Hence, $\boldsymbol{W}$ will form a Krylov subsequence on the subspace of active variables.

The full algorithm is described in Table 4.

### 3.4.3 SPLS results interpretation

The interpretation of the outputs produced by SPLS is equivalent to the PLS one since SIMPLS is applied on the set of the active variables to fit the model. The main difference in the results with PLS is the presence of regression coefficients with exact zero values.

### 3.4.4 Biomarkers identification

In SPLS, the regression coefficients are commonly used for FS, and here $\boldsymbol{B}^{spls-coef} = |\boldsymbol{b}^{spls}|$. The decision rule to keep variable $i$ is $\mathcal{A}^{spls-coef} = \{i : |b^{spls}| \neq 0\}$ and the number of features is automatically fixed accordingly.

Table 4: The SPLS algorithm for a single response.

| | | |
|---|---|---|
| **A.** | **Compute SPLS components** | |
| (1) | Initialisation: $\boldsymbol{y}_0 = \boldsymbol{y}_1 = \boldsymbol{y}$ and $\boldsymbol{b}_0^{spls} = \boldsymbol{0}$ | Define initial mean-centred $\boldsymbol{X}$ matrix. Recall that $\boldsymbol{y}$ is me as well here. Set to 0 the original vector of regression coeff |
| (2) | Iterate for $j = 1, \ldots, a$ | |
| (2.1) | $\lvert\tilde{\boldsymbol{s}}_j\rvert = \boldsymbol{X}'\boldsymbol{y}_j / \parallel \boldsymbol{X}'\boldsymbol{y}_j \parallel$ | Calculate and norm the cross-product of $\boldsymbol{X}$ and $\boldsymbol{y}_j$ |
| (2.2) | $\boldsymbol{w}_j = (\lvert\tilde{\boldsymbol{s}}_j\rvert - \eta \cdot \max_{1\leq i\leq m}\lvert\tilde{s}_{ij}\rvert)\boldsymbol{I}(\lvert\tilde{\boldsymbol{s}}_j\rvert \geq \eta \cdot \max_{1\leq i\leq m}\lvert\tilde{\boldsymbol{s}}_i\rvert))sign(\tilde{\boldsymbol{s}}_j)$ | Find the direction vector $\boldsymbol{w}_j$ |
| (2.3) | $\mathcal{V} = \{i : w_{ij} \neq 0\} \cup \{i : b_{i(j-1)} \neq 0\}$ | Update the active variables set with $i = 1, \ldots, m$ |
| (2.4) | Fit the PLS model $\boldsymbol{y}_j = \boldsymbol{X}_{\mathcal{V}}\boldsymbol{\beta}^{pls} + \boldsymbol{e}$ | With PLS-SIMPLS with $j$ LV on the selected variables: $\boldsymbol{X}_{\mathcal{V}}$ $\mathcal{V}\}$. Estimated regression coefficients $\boldsymbol{b}^{pls}$ are used in the n |
| (2.5) | $b_{ij}^{spls} = \begin{cases} b_i^{pls} & \text{if } i \in \mathcal{V} \\ 0 & \text{otherwise;} \end{cases}$ | Update $\boldsymbol{b}_j^{spls}$: unselected variables are zeroed, selected va given by previous PLS regression |
| (2.6) | $\boldsymbol{y}_{j+1} = \boldsymbol{y}_0 - \boldsymbol{X}\boldsymbol{b}_j^{spls}$ | Deflation step over the response, the new response $\boldsymbol{y}_{j+1}$ is t response |

## 3.5 Light Sparse OPLS

### 3.5.1 Description

A recent article from Féraud *et al.* (2017) suggested to combine the previously described OPLS and SPLS into the Light Sparse OPLS (LSOPLS). The idea was to take advantage of these two approaches for biomarkers discovery: sparsity and orthogonal variation filtering. The authors showed that this new approach could lead to improved predictive performances by retaining only a small number of predictors.

### 3.5.2 Statistical methodology

Similarly to OPLS, LSOPLS is composed of two subsequent stages: an orthogonalisation and a modelling step where only predictive variability is retained to explain the dependent variable. The orthogonalisation step is derived from some of the original OPLS algorithm and the modelling step consists is applying SPLS on the previously filtered out predictive matrix.

Note that for predictive purposes, the coefficients $\boldsymbol{\beta}_p^{lsopls}$ need to be corrected into $\boldsymbol{\beta}^{lsopls}$ in order to be applied directly to the original predictive matrix $\boldsymbol{X}$, as in OPLS. The uncorrected coefficients $\boldsymbol{\beta}_p^{lsopls}$ are still used for variable selection, since those coefficients are the sparse ones, contrarily to $\boldsymbol{\beta}_p^{lsopls}$.

### 3.5.3 LSOPLS results interpretation

The interpretation results are similar to the ones of the previous approaches, since the produced outputs are from an SPLS model that is described previously.

### 3.5.4 Biomarkers identification

The uncorrected regression coefficients constitute the LSOPLS feature scores $\boldsymbol{B}^{lsopls-coef} = \lvert\boldsymbol{b}_p^{lsopls}\rvert$. As for SPLS, the subset of selected variables is: $\mathcal{A}^{lsopls-coef} = \{i : \lvert b^{lsopls}\rvert \neq 0\}$.

## 3.6 Summary of the feature selection procedures

In Table 6 is summarised $\boldsymbol{B}$'s content and the decision rule to selected variables as discriminant features.

Table 5: The LSOPLS algorithm for a single response.

| | | |
|---|---|---|
| **A.** | **Orthogonalisation step** | Apply steps (1) and (2) from OPLS algorithm with $(a-1)$ orthogonal components and mean-centred $\boldsymbol{X}$ and $\boldsymbol{y}$. |
| **B.** | **Build global matrices** | Build the following global matrices: $\boldsymbol{P}_o$, $\boldsymbol{W}_o$ and $\boldsymbol{X}_p$ derived from Step A |
| **C.** | **SPLS** | Apply SPLS on the filtered matrix $\boldsymbol{X}_p$ |
| (1) | Fit the SPLS model $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}_p^{spls} + \boldsymbol{e}$ | With SPLS-SIMPLS where $\boldsymbol{X} = \boldsymbol{X}_p$ and for 1 PrC. |
| (2) | $\boldsymbol{b}_p^{lsopls} = \boldsymbol{b}_p^{spls}$ | Define the sparse LSOPLS coefficients as the ones of the SPLS model above |
| (2) | $\boldsymbol{b}^{lsopls} = (\boldsymbol{I}_m - \boldsymbol{W}_o P_o')\boldsymbol{b}_p^{lsopls}$ | Derive the non sparse regression coefficients to be applied to unfiltered $\boldsymbol{X}$ |

Table 6: Summary of the elements in $\boldsymbol{B}$ (Index) for $i = 1, \ldots, m$; the decision rule to include the variable in $\mathcal{A}$ and the methods abbreviation in this work.

| **Underlying model** | **Feature score element** | **Decision rule for $i \in \mathcal{A}$** | **FS method abbrev.** |
|---|---|---|---|
| $t$-test | $\lvert\delta\rvert_{(i)}$ | $i \leq m^*$ | t-stat |
| | $\lvert\delta\rvert_i$ | $\rho_{i,fdr} \leq 0.05$ | t-fdr |
| PLS | $\lvert b^{pls}\rvert_{(i)}$ | $i \leq m^*$ | pls-coef |
| | $v_{(i)}$ | $i \leq m^*$ | pls-vip |
| | $v_i$ | $v_i \geq 1$ | pls-vip1 |
| OPLS | $\lvert b^{opls}\rvert_{(i)}$ | $i \leq m^*$ | opls-coef |
| | $v_{(i)}$ | $i \leq m^*$ | opls-vip |
| | $v_i$ | $v_i \geq 1$ | opls-vip1 |
| SPLS | $\lvert b^{spls}\rvert_i$ | $\lvert b^{spls}\rvert_i \neq 0$ | spls-coef |
| LSOPLS | $\lvert b^{lsopls}\rvert_i$ | $\lvert b^{lsopls}\rvert_i \neq 0$ | lsopls-coef |

The decision rule depends either on an arbitrary threshold or on a predefined one. When arbitrary, $m^*$ is set to the number of true features, *i.e.* $m^* = 46$.

# 4 Metaparameter tuning

Except for multiple t-tests, all supervised methods have meta-parameters to be tuned in order to fit – without overfitting – a parsimonious model to the data at hand, to ensure predictive power and a good model generalisation. For PLS and SPLS, it corresponds to the number of PrC. Whereas, for OPLS and LSOPLS, it corresponds to the number of orthogonal LV, since only one predictive LV is considered in their case. In addition, sparse methods (SPLS and LSOPLS) have a second meta-parameter to adjust: the degree of sparsity $\eta$.

In practice, due to the usually restricted number of data samples, their tuning is achieved by cross-validation (CV) on a *training set*. This training set represents a certain percentage of the total dataset (*e.g.* 80 %), the remaining part being defined as the *test set*. Note that in practice, with a limited number of observations, the training set often represents 100% of the total dataset and no test set is defined. The training set is then divided into $k$ subgroups of data and a model is built based on samples from $k-1$ groups. The model then predicts the remaining samples that were not used in the model construction and evaluates the prediction performances based on a predefined criterion. The most frequent CV segmentation is the $k$-folds CV with $k = 10$. It divides at random the training dataset into 10 different folds and is usually preferred to others to prevent a possible overfit. Note that if $k = n$, the technique is called Leave-One-Out CV.

The final predictive performances are then evaluated on the basis of the test set that was left aside, whose response is now predicted by the adjusted model on the training set with the chosen meta-parameters.

In the regression context where PLS-derived methods have been developed, the Root Mean Square Error (RMSE) that estimates the error of prediction is an usual tuning criterion to evaluate predictive performances (Wehrens, 2011). Its name becomes RMSECV if the criterion is used in CV or RMSEP if it is used to estimate the prediction errors of an independent test set.

$$RMSE = \sqrt{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2/n}\ ; \tag{21}$$

where $\hat{y}_i$ is the predicted response of observation $i$.

Other well known derived criteria are the Predictive Residual Sum of Squares $PRESS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ and $Q2 = 1 - \frac{PRESS}{SST}$ where $SST$ stands for Total Sum of Squares and $SST = \sum_{i=1}^{n} y_i$, with mean-centred $\boldsymbol{y}$. They can be both used in CV or validation as well and are all equivalent.

In the context of classification problems, other criteria such as the discriminant Q2 (Westerhuis *et al.* , 2008) or the Area Under the ROC Curve (AUC) may seem more appropriate to evaluate classification performances. According to our own previous trials, meta-parameter tuning were slightly better handled by the AUC (which will be described latter in this paper), which, in a sense, is the more natural way to evaluate the classification fit. Note that an exception is made to decide on the number of orthogonal components. In this case, the RMSE is used as suggested by Trygg & Wold (2002). Though this issue is beyond the scope of this work, finding an appropriate criterion in the classification context remains an open question and would deserve to be investigated further.

# 5 Application of the methods to a sub-sample

In order to complete the methods description, they have been applied to one sub-sample (with $n = 200$) from the analysed semi-artificial dataset and the results are presented here. 10-fold CV has been applied to decide on the MP that needed optimisation. The chosen parameters are presented in Table 7. One can see that the total number of components is in the same order of magnitude for PLS and OPLS. SPLS and OPLS have much different MPs. In order to discuss the MP selection for sparse methods that have

both the number of components and the sparsity, one can inspect the two graphs presented in Figure 4. They illustrate the AUC and the number of selected features $m^*$ given fixed MPs for SPLS. The AUC increase through the MPS is not even approaching monotonicity as it is the case for univariate MP selection. Highest values are met at various combinations of $\{a, \eta\}$. Though AUC is an objective and adapted criterion, the wining MPs combinaison is plural and heterogeneous, and so may be the resulting sets of selected features. On the other hand, the impact of the MPs' choice on the size of the feature subset is very clear: a larger sparsity constraint or a lower number of components logically leads to a smaller set of selected variables.

Table 7: Selected meta-parameters for supervised classifiers.

| | **PLS** | **OPLS** | **SPLS** | **LSOPLS** |
|---|---|---|---|---|
| # PrC | 7 | 1 | 11 | 1 |
| # OC | - | 6 | - | 12 |
| Sparsity ($\eta$) | - | - | 0.8 | 0.25 |

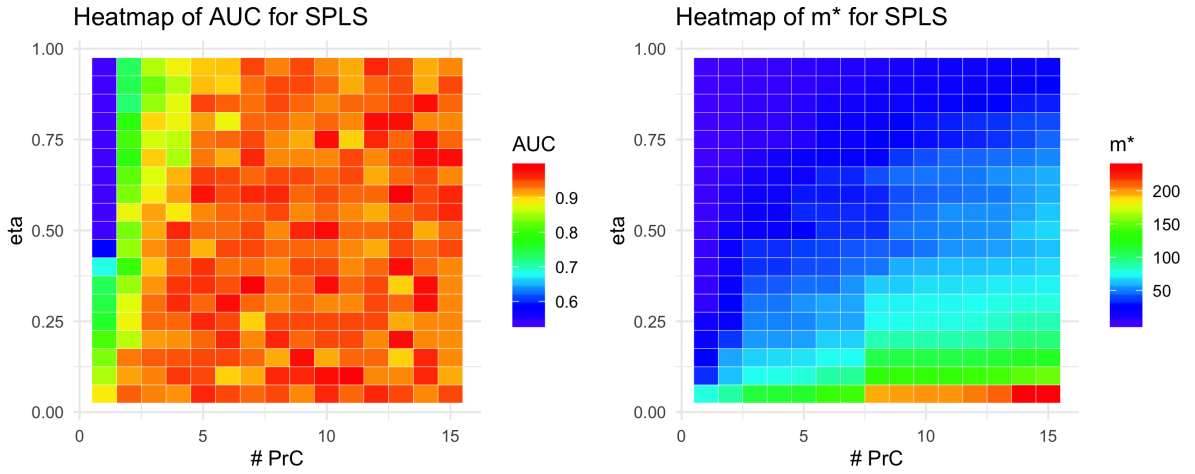


Figure 4: Heatmaps of AUC and $m^*$ obtained by 10-fold CV for SPLS, and given the values of the MPs: the number of LV $a$ and sparsity $\eta$.

Scores and loadings plots are traditionally interpreted to visually assess the links between: the samples, the original variables, and each other, in a reduced space of a few components (PrC or OC). Scores plot enables to see if structured information or clusters emerge from those samples *i.e.* if they are well separated according to their class label, or to detect another structure in the data unrelated to the response of interest. It also is particularly useful to detect outliers. These plots are displayed in Figure 5 for all the studied multivariate methods, plus PCA. The latter was not able to discriminate along PC1-2 the two classes of spectra. The four supervised methods were mainly able to recover the group structure from the data along the first component (LV1). However, the observations are only well separated by the combination of LV1 and LV2 for (S)PLS methods. Indeed, methods with orthogonal filters ((LS)OPLS) only need one component to discriminate between the two classes. Moreover, the visualisation of the orthogonal components in (LS)OPLS is informative: it helps to identify other structured information than the class membership, to understand if the samples were appropriately prepared, acquired and pre-processed before their actual data analysis.

Since discrimination between the spectra is mainly driven by the first PrC, the main loading of interest will also be the first one, and we will now concentrate on this one. Figure 6 represents the loadings from the corresponding first component for each method. Their magnitude represents the importance of each original variable in the construction of the latent variables. Only the amplitude is important here, as well as the relative direction within a loading, and not the overall direction of that loading. The sign of the latter is set arbitrarily during components' construction. For all methods, higher loading values are present on the right side of the spectral window where half of the true biomarkers are located. The first two axes in PCA is not built on the discriminant variables. PLS and OPLS have a few high values, representing the true biomarkers, and a lot of smaller intensities, representing false discoveries ; whereas SPLS and LSOPLS have fewer low values.

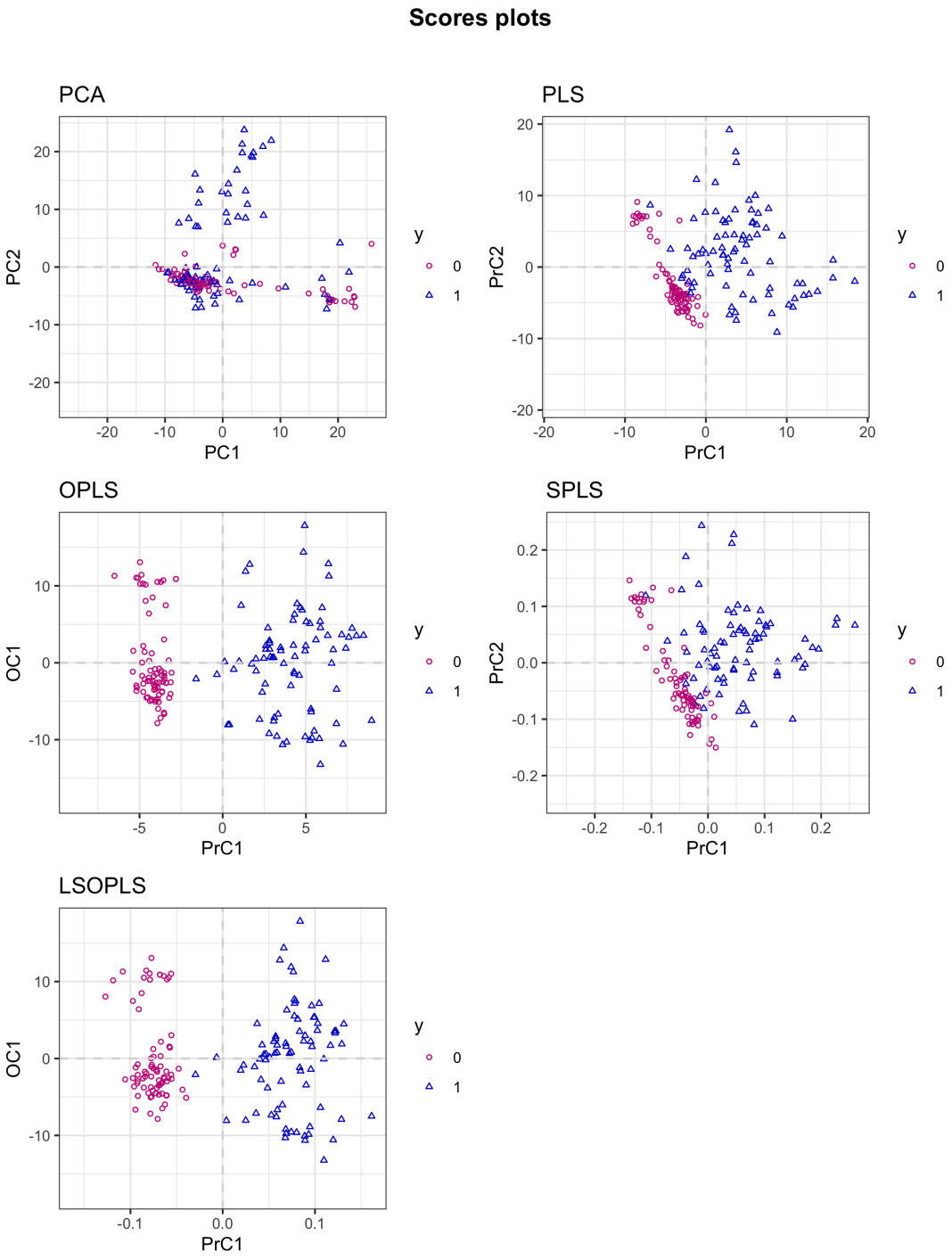


Figure 5: Scores plots. The first two scores are represented for each method. For OPLS and LSO-PLS, since only one predictive component is retained, the second axis represents the first orthogonal component.

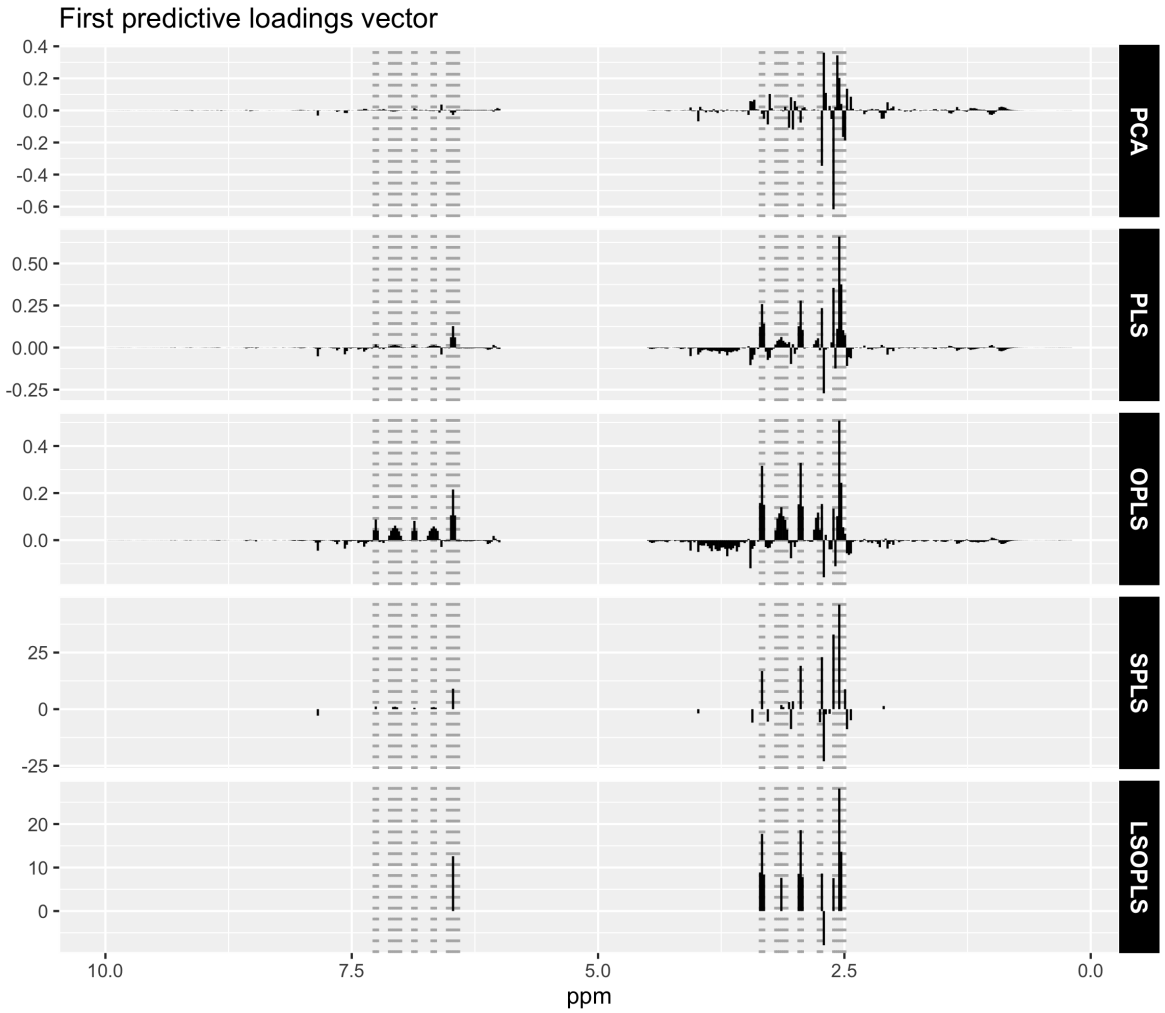


Figure 6: Loadings plots. The first loadings vector is represented for each method.

Values of both the difference in means per class and the t-statistic per variables are compared in Figure 7. While the difference in means is clearly pointing at the true biomarkers, the t-statistic has in addition a series of high values for all the uninformative and noise without signal variables. This is due to the very low standard deviation of those descriptors, directly associated to their very low mean values. This result highlight the main limitation of using t-tests for feature selection with wide and short data tables, as it definitely leads to a large amount of false discoveries on such data.

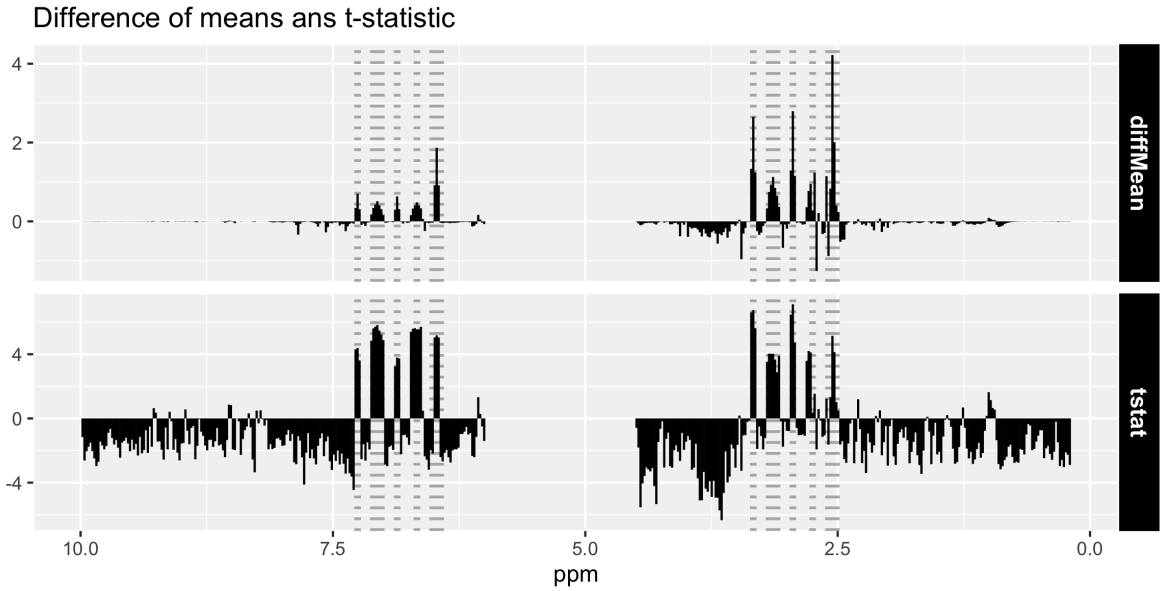


Figure 7: Plots of the difference of the descriptors mean values per class and the t-statistic value. The vertical dotted lines represent the location of the true features.

The regression coefficients and VIP values of the supervised classifiers are of direct interest here since they summarise by themselves the global importance of variables to discriminate between the two classes of spectra and are used later on in the generic evaluation study for feature selection.

The regression coefficients are presented on the same ordinate axis in Figure 8. Recall that, based on the regression coefficients, SPLS and LSOPLS select the non-null values as potential features of interest whereas for PLS and OPLS, an arbitrary threshold is set and descriptors with the higher absolute values than that threshold are selected. For LSOPLS, the regression coefficients are the sparse uncorrected ones $\boldsymbol{b}_p^{lsopls}$ that will be used for feature selection. Based on a binary response vector, positive coefficients indicate which descriptors have a higher intensity in the presence of the alteration ($\boldsymbol{y} = 1$), while negative coefficients indicate that those descriptors tend to have a lower intensity in that case. The following comments can be made after analysing this figure. Logically, sparse methods have a restricted amount of non-null regression coefficients. As expected from the loadings plot, SPLS and LSOPLS are leading to sparser solutions. PLS and OPLS have very similar regression coefficients, as a consequence to the $X$-weights equality (*cf.* Section 3.3.3) in the case of a single response vector. They have good performances in the identification of features from the noise-free region since true features coefficients are higher in this area. Compared to LSOPLS, SPLS is better at recovering correlated features from a same biomarker. Finally, LSOPLS has the sparser solution but could not find all the peaks' locations.

Figure 9 represents the coefficient path plot for sparse methods along the iterative recruitment of a new model components (PrC or OC). For each new PrC in SPLS, there are new variables selected. The already enrolled variables tend to have a higher regression coefficient value as well. In LSOPLS, the effect of subtracting an OC to $\boldsymbol{X}$ has almost no influence on the selection subset, though a slight increase in the coefficients value is observed, especially for the first OCs.

Finally, VIPs are also considered for feature selection and the threshold of 1 is illustrated in Figure 10. Note that VIP values are always positive. They indicate which descriptor in important in the class discrimination but tell nothing about the direction of the varying intensities in the presence of the alteration. Based on this threshold, one can see that PLS will select all the peaks, which is not the case for OPLS. This threshold of 1 discards most of the false discoveries but also some of the true features,

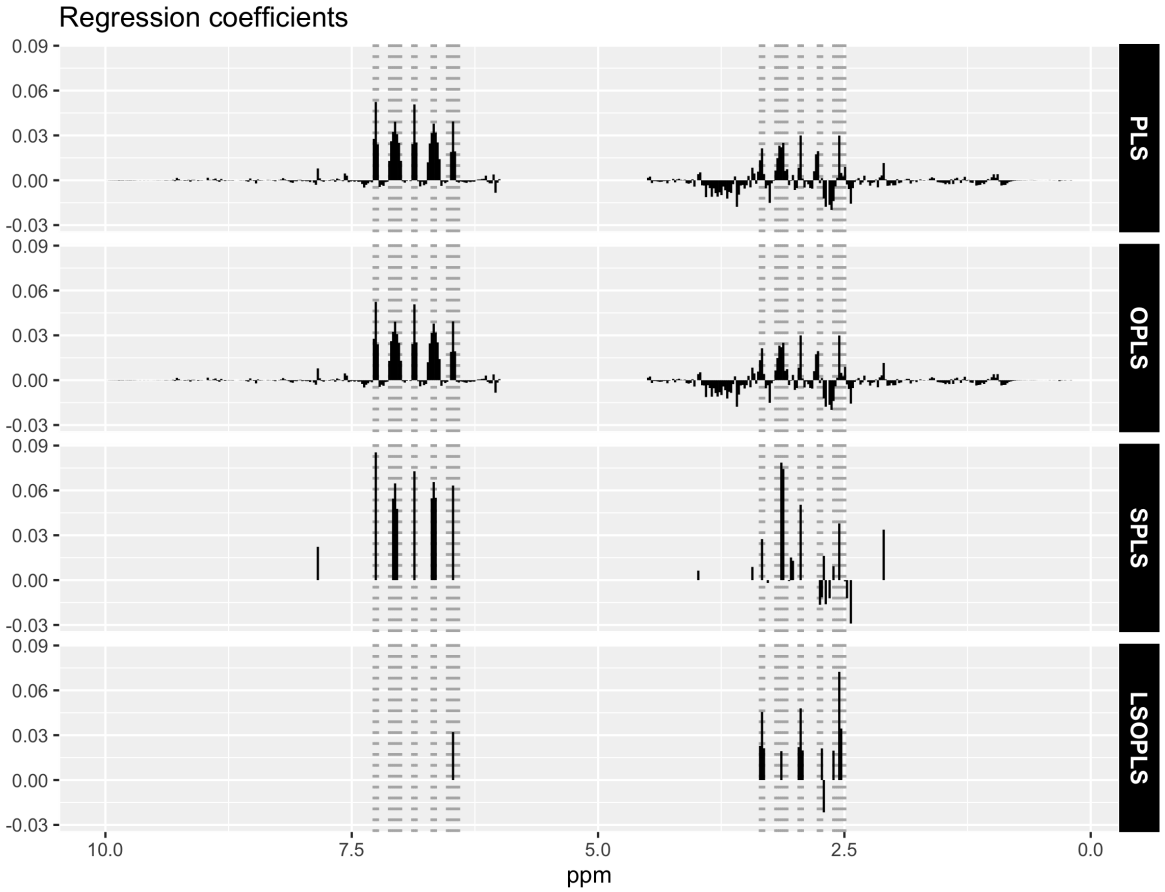


Figure 8: Plots of the regression coefficients for all the methods. The vertical dotted lines represent the location of the true features.

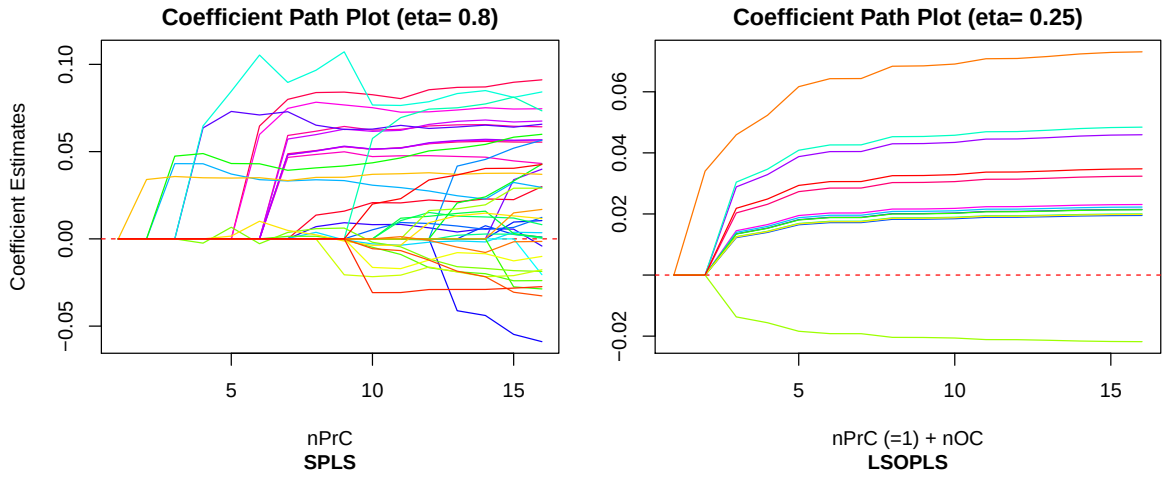


Figure 9: Coefficient path plot of SPLS and LSOPLS. The plot illustrates the variation in the estimated coefficients as a function of the number of LV, with $\eta$ fixed at its optimised level by CV.

especially for OPLS. Compared to regression coefficients, VIPs seem to lead to a better feature recovery in the noisy part of the spectra.

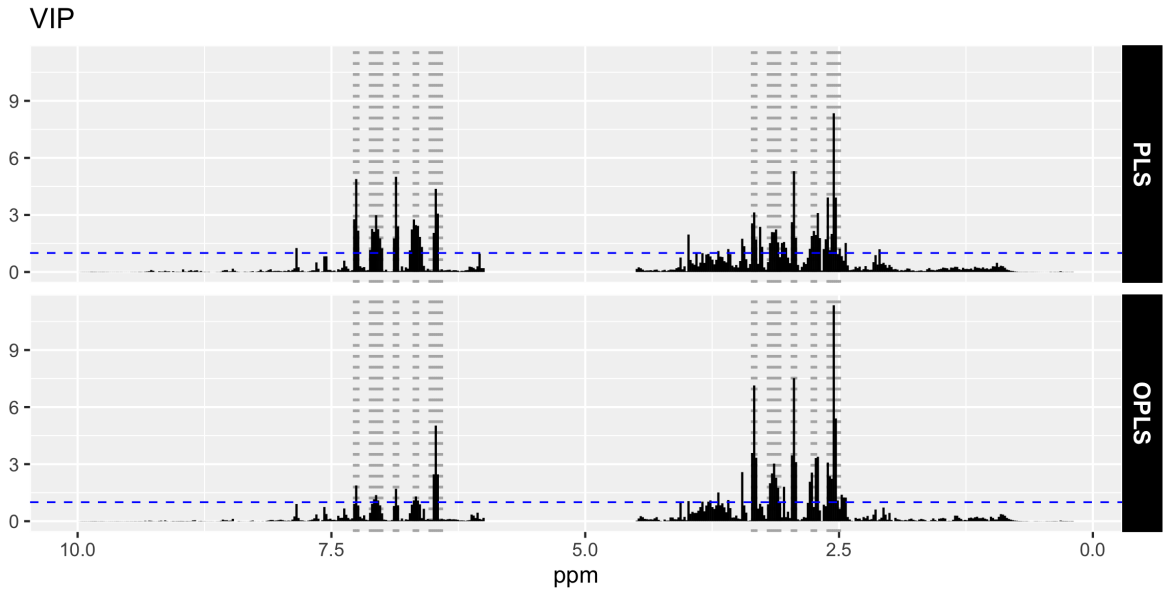


Figure 10: Plots of the VIPs for PLS and OPLS. The vertical dotted lines represent the location of the true features and the horizontal dashed blue line represents the threshold of 1 for VIP values.

# 6 Evaluation of the classifiers performances

## 6.1 Introduction

The classifiers performances are now systematically evaluated on a large bunch of resampled sub-datasets from the semi-artificial database described above. The 5 biomarker selection methods are then applied to these sub-datasets. This approach is close to the spirit of the ensemble feature selection techniques (Abeel *et al.* , 2009) that consists in the combination of a large set of different models, gathered into a global and more robust one. Those sub-models are based on resampled sub-samples of the original dataset.

The main aims of this study are to:

- avoid misleading results due to a particular draw of one sample.
- study which method more likely discovers all true features – *i.e.* potentially correlated features.
- analyse the sensitivity and the FDR of features selection for each method.
- test the robustness of each method to different realistic sample sizes, in particular to a small sample size for $n = 50$.
- measure selected features and meta-parameters stabilities across resampling.

Note that methods comparison will be principally conducted on the $n = 200$ sample size dataset. Nevertheless, specific variations for a lower sample size ($n = 60$) will be highlighted and all the corresponding results are available in Appendix A.

To achieve these above-mentioned goals, a double loop evaluation scheme was implemented for all methods but the t-test, with resampling and $k$-fold CV. This validation pattern is illustrated in Figure 11. The outer loop consists in resampling[5] the raw spectral matrix $R$ times (here, $R = 50$). An inherent incentive for resampling is the ability to measure the variability due to data sampling. Each resampled sample of size $n = n_{train} + n_{test}$ is split into a training and a test sets in the *outer loop* and each training set is again split in a series of training and test sets within an *inner loop* during $k$-fold CV. The outer loop is used to assess the predictive models performances while the *inner loop* decides on the optimal MPs for trained models.

The results are produced and discussed in the following order or appearance: MPs stability, classification accuracy, global assessment of feature selection accuracy, visual inspection of the selected features on the spectral scale, and feature stability. Some measurements are based on the known alterations' true location, but others are "unsupervised" and could be used as *proxi* to evaluate the true features recovery in experimental datasets, where true location is unknown (*e.g.* by measuring the predictive performances or the selected features stability).

## 6.2 MPs stability

Deciding on the meta-parameters is inherent to the training of multivariate classifiers. They should be set wisely to avoid an overfit of the model to the data and ensure a parsimonious classifier. The variability in the choice of these MPs give insights into the algorithmic stability of the methods. Indeed, spectra sampled from a same population should lead to similar MPs, otherwise the algorithm is highly influenced by small changes in the data and is therefore considered unstable. The MPs variability will also influence the stability of the features selection process, and especially for procedures with automatic threshold, where the number of retained features directly depends on those MPs. Note that t-tests do not have MPs to tune as they are not classifiers *per se*. Recall that MP are on the one side, the number of predictive (PLS and SPLS) or orthogonal (OPLS and LSOPLS) components and, on the other side, the degree of sparsity $\eta$ for SPLS and LSOPLS.

[5] Resampling consists here in a random draw of the spectral samples and their associated class with replacements.

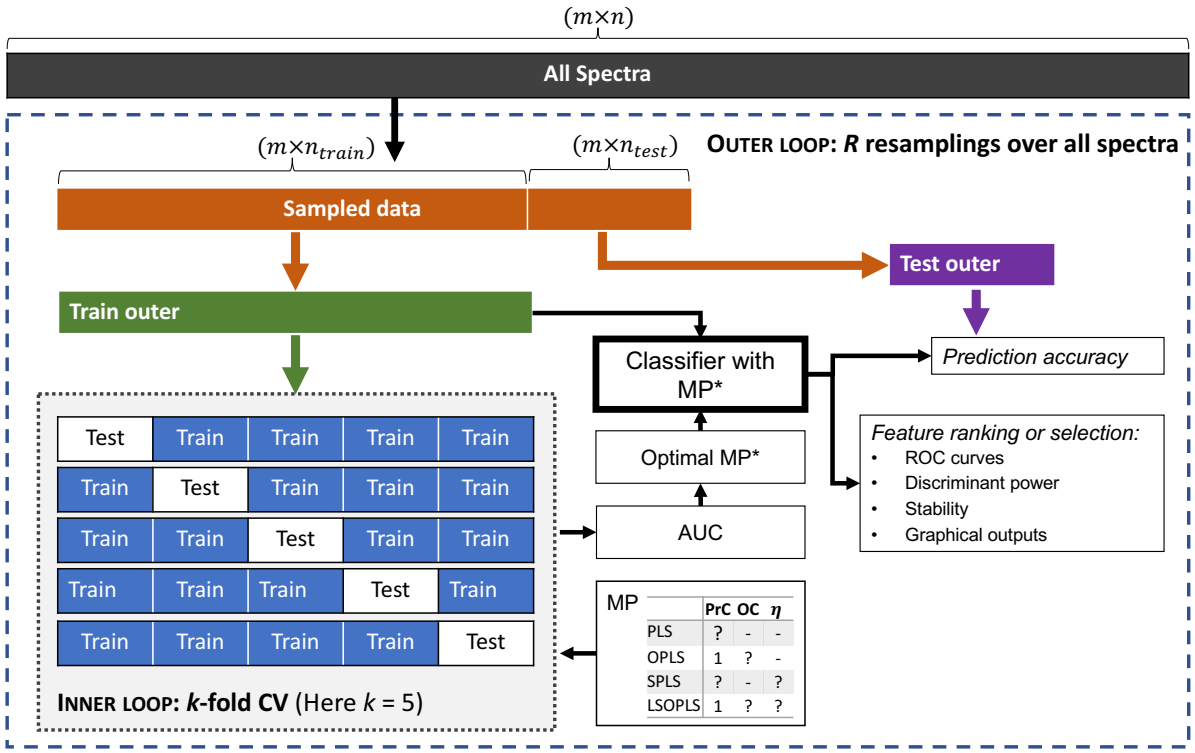


Figure 11: Double loop validation scheme applied to all the methods. The outer loop consists in $R = 50$ resampled subsamples and the inner loop in a 10-fold CV (*rem.*: no inner loop for t-tests).

From Figure 12, it can be seen that the distribution of those MPs linked to the number of components is equivalent between the classifiers, with median values around 8 LVs. LSOPLS is the less stable algorithm to determine the number of LVs.

With regards to the sparsity parameter, SPLS has a larger median sparsity degree than LSOPLS. This could be explained by the prior orthogonal variation filtering in LSOPLS that already removes the uninformative variation from the data matrix, hence the filtering of uninformative descriptors is less important. This parameter is also less stable for SPLS. As illustrated for SPLS in Section 5 (Figure 4), a wide variety of possible combinations of LVs and sparsity is indeed leading to high AUC values for sparse classifiers. Hence, their MPs' stability will be affected accordingly. Since the size of the selection set is directly influenced by the amount of sparsity, this can be seen as a weakness of such sparse methods having multiple parameters to be optimised.

With a lower sample size (*cf.* Figure 22), MPs tend to be more spread and less stable across the sub-samples. The median number of OC is lowered, which is compensated by a higher degree of sparsity for LSOPLS. For SPLS, a decrease in its sparsity median value can also be observed.

## 6.3 Size of the selection sets for automatic threshold methods

Graphical result in Figure 13 represents the size of the actual selection sets retained for automatic threshold methods. T-fdr has the highest median value around 100 of selected features, namely the double of the true features set size. Compared to PLS methods, OPLS ones tend to select less features. LSOPLS is the sparsest method with the lowest size of the selection set among all those methods. SPLS has a median value of 50 features selected but this amount is highly variable across the resampled sub-samples, as expected from the MPs variability results. In contrast, Vip1 methods have the lowest variability in the number of selected features. In the case of a reduced sample size (Figure 23), and except for t-fdr that is becoming much more conservative, results are approximately the same.

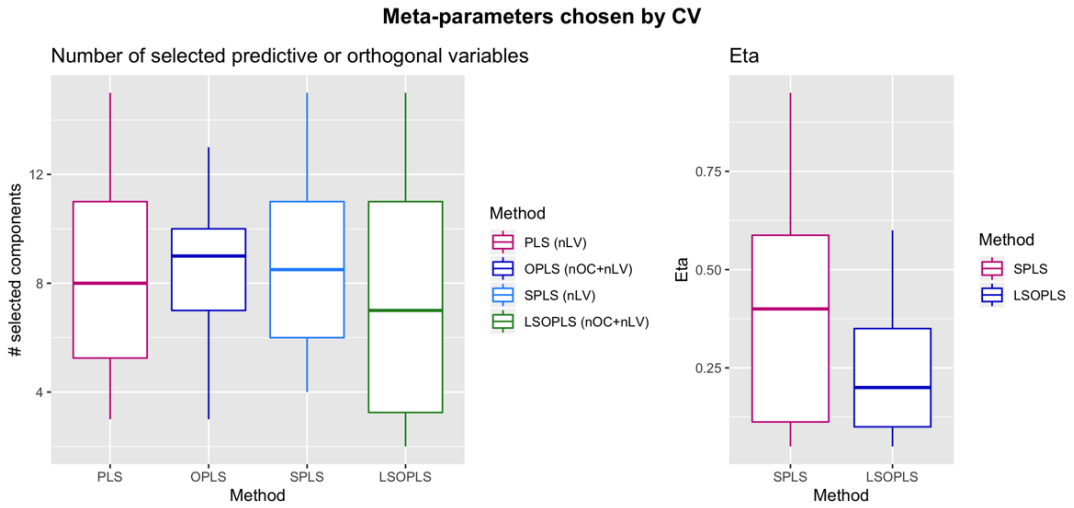


Figure 12: Distribution of MPs chosen by the inner-loop CV ($n = 200$).

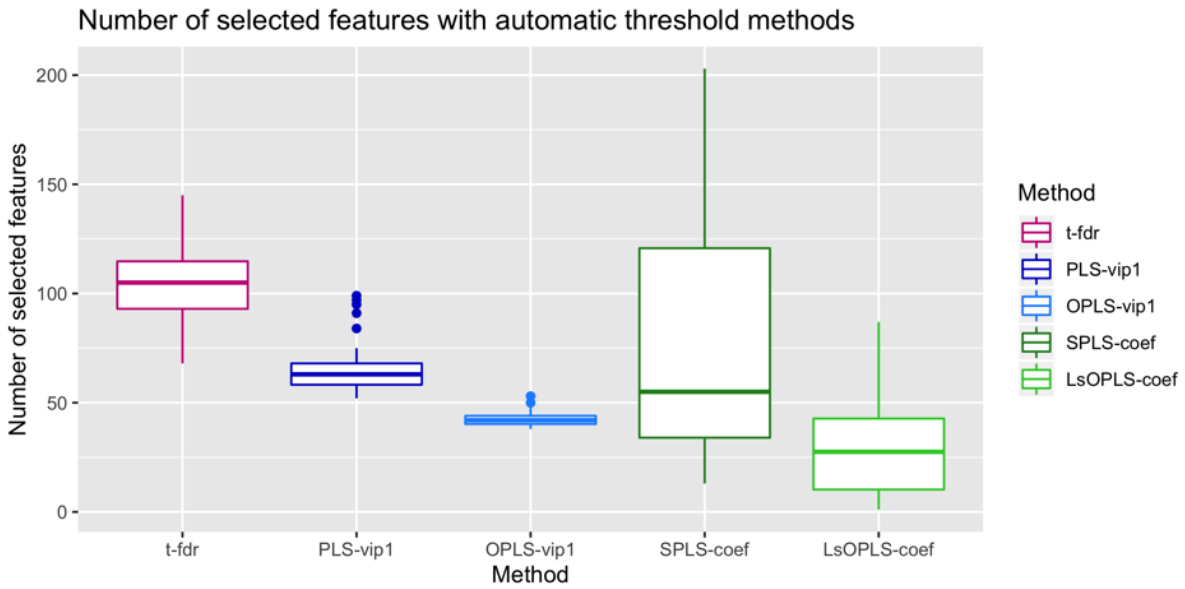


Figure 13: Number of selected features for automatic threshold methods ($n = 200$).

## 6.4 ROC curves and classification accuracies

Features and classification accuracies are evaluated in this section on the basis of indices derived from a binary confusion table.

### 6.4.1 Confusion matrix indices

In binary classification, a series of performance indices can be derived from a confusion matrix, based on *true* and *predicted* states *e.g.* of a patient (see Table 8).

Table 8: Confusion matrix for binary classification.

| | | **True class** | |
|---|---|---|---|
| | | Positive | Negative |
| **Predicted class** | Positive | True Positives (TP) | False Positives (FP) |
| | Negative | False Negatives (FN) | True Negatives (TN) |

From this table, the *confidence in classification* is measured with the *sensitivity* – or number of actual positives that are correctly identified – $TP/(TP+FN)$ and the *specificity* – or number of actual negatives that are correctly identified – $TN/(TN+FP)$. The *predictive ability* of the classifier is measured with the *Positive Predictive Value* (PPV or *precision*): $TP/(TP+FP)$ and the *Negative Predictive Value* ($NPV$): $TN/(TN+FN)$. Another measure of interest is the opposite of PPV, the *False Discovery Rate* (FDR): $1-PPV$.

Different aggregated measures that are independent of the group size *i.e.* can be used even in the case of unbalanced group sizes are:

- The Average predictive ability: $(PPV+NPV)/2$
- The Average confidence in classification: $(sensitivity + specificity)/2$
- The F1 score (harmonic mean of PPV and sensitivity) focusing on the correct classification of true positive features: $2.\frac{PPV \times sensitivity}{PPV + sensitivity} = \frac{2TP}{2TP+FP+FN}$

### 6.4.2 Application to classification accuracy

The ROC curves (Fawcett, 2006) are an appropriate tool to visualise and evaluate the behaviour of classifiers with a varying parameter. They are based on two specific confusion matrix indices: the sensitivity and the FDR. Generally, ROC curves are designed to evaluate a binary classifier with the discrimination threshold that varies along the curve or compare several classifiers between each other.

The space of a ROC curve (illustrated in Figure 14) extends from (0,0) with no positive classification to (1,1) representing an unconditional positive classification. The perfect classification occurs at D (0,1). Overall, a point located in a Northwest position compared to another one is better (B compared to C). Furthermore, classifiers appearing at the lower left-hand side of the graph are conservative: they make few false discoveries errors and need strong evidence to classify as positive - whereas classifiers at the upper right-hand side of the graph tend to have high false discoveries by classifying as positive even if only little evidence exists. Besides, the diagonal line where *sensitivity* = FDR represents a random class allocation, and for a classifier appearing in the lower triangle (E), the class allocation should be inverted to perform better than a random guessing.

Another aggregated scalar, the Area Under the Curve (AUC) can be used for model comparison (Fawcett, 2006). Its value is always between 0 and 1 and a classifier should have at least an AUC of more than 0.5.

Note that all these indices are averaged over the outer loop.

#### Classification accuracy results

Classification performances depicted in Figure 15 (for $n = 200$ and $n = 60$) and in Table 9 are measured at the outer loop level. They are calculated by applying the classifier with the optimal MPs on an

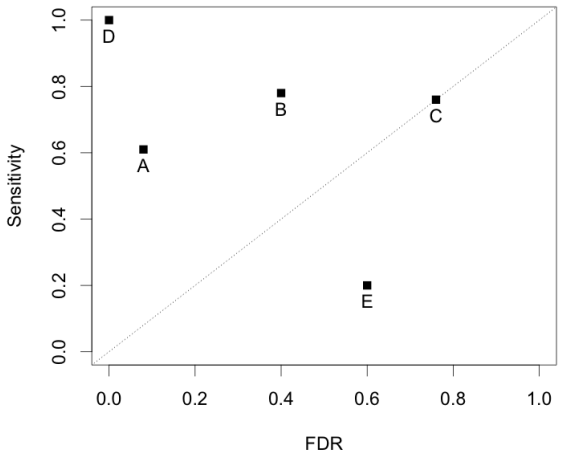


Figure 14: The ROC space.

independent test set in order to avoid an over-optimistic fit (Varmuza & Filzmoser, 2016). Classifiers perform quite well as expected by dataset construction with clear and distinct artificial alterations. Therefore, classification performances here are of little help to rate the classifying methods between themselves. It should be noted that, although very good, orthogonal methods performances are reduced when compared to the other classifiers. For all the methods, performances are lowered when the sample size decreases (*cf.* Figure 24) since less samples are available to train the models.

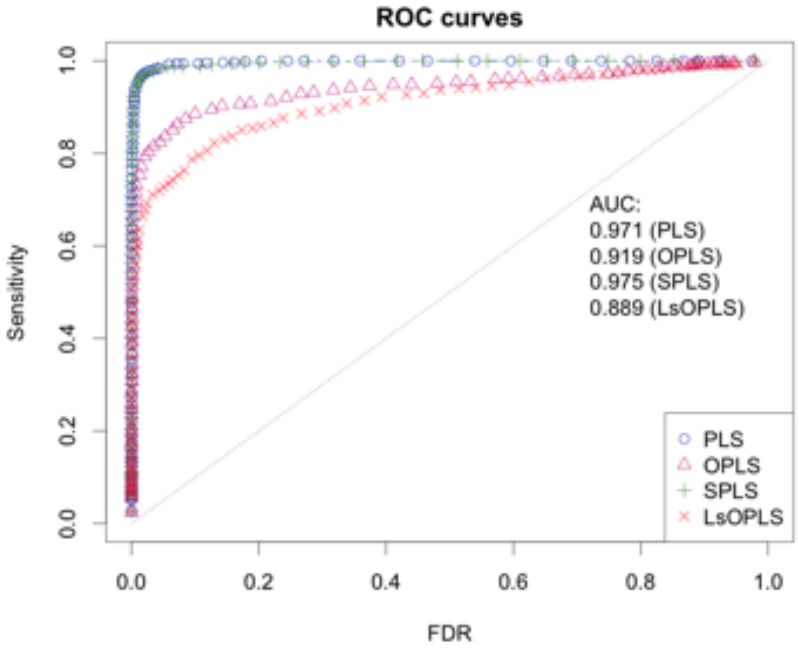


Figure 15: ROC curves for classification accuracy ($n = 200$).

## 6.5 General assessment of features accuracy

A global assessment of features accuracy is presented in this section, before inspecting in more details the recovery of biomarkers by spectral area in the next one.

Table 9: Classification performances: mean values of sensitivity, FDR, average confidence in classification, average predictive ability and F1 score at best threshold ($n = 200$).

| | **AUC** | **Sensitivity** | **FDR** | **Average confidence in classification** | **Average predictive ability** | **F1 score** |
|---|---|---|---|---|---|---|
| PLS | 0.97 | 0.97 | 0.02 | 0.98 | 0.98 | 0.98 |
| OPLS | 0.92 | 0.88 | 0.08 | 0.9 | 0.9 | 0.89 |
| SPLS | 0.98 | 0.98 | 0.03 | 0.97 | 0.98 | 0.97 |
| LsOPLS | 0.89 | 0.82 | 0.13 | 0.85 | 0.87 | 0.84 |

### 6.5.1 Use of ROC curves to evaluate features accuracy

Since the exact location of the true features is known in advance, the ROC curves principles can also be applied to evaluate each method's ability to discover the true features, as in Rousseau (2011) or Christin *et al.* (2013).

Each ROC curve is assigned to a classifier. Along these curves, two different parameters impacting features accuracy are varied: either the number of selected features $m^* = 1, \ldots, m$ when an arbitrary threshold is to be decided or the sparsity level $\eta = 0.05, \ldots, 0.95$ for sparse classifiers. Note that in this case, the curves can be non-monotonic, as should be expected for their regular use. here, it is important to decipher how the *true* features are classified, rather than how are classified the non-discriminant ones. To this end, we focus on indices where TP and FN are present such as in sensitivity, FDR, PPV or F1. Because the number of potential biomarkers detected will affect the range of subsequent research experiments (*e.g.* purification processes of those biomarkers), classifiers with a lower number of false discoveries (or higher precision) are preferred over classifiers with a higher sensitivity (Christin *et al.* , 2013). However, given the fact that different spectral descriptors represent a single metabolite and that metabolites are correlated, one may also be interested to keep all the discriminant features, even if they are highly related to each other. In this case all the relevant information is preserved and one would like to have a higher sensitivity. In ROC curves, the sensitivity and the FDR are illustrated.

The indices are as previously averaged over the outer resampling loop.

#### ROC curves for feature accuracy results

Figure 16 represents the ROC curves with a different varying parameter given the type of method applied, and for $n = 200$.

With regards methods with a varying threshold $m^*$, all curves are nearly monotonic, except for the t-stat. The more interesting part of the graph is located on the left side of it, where only a restricted set of potential features are selected.

At the beginning of the curves, coefficient methods outperform vip methods. They further have the higher AUC among all the arbitrary threshold methods. T-stat is poorly performing at the start of the curve as its ratio sensitivity/FDR is very low, but outperforms vip methods at some point. OPLS-coef has the highest sensitivity and lowest FDR when selecting a lower number of potential features. For a short threshold value window, t-stat has the best sensitivity/specificity trade-off. However, PLS-vip is quickly outperforming the other methods at some point. OPLS-vip performs less well than the other (O)PLS derived methods.

The number of selected variables at best thresholds ranges from 43 to 57, corresponding to approximatively 10% (8.6 - 11.4) of the total number of spectral variables. At their best threshold, t-stat has surprisingly the best sensitivity/FDR trade-off. It can also be observed that the average number of selected features or threshold across $R$ with the VIP threshold of 1 (blue dot on the graph) is either above or below the best measured mean threshold respectively for PLS-vip and OPLS-vip.

Among those methods, no one is outperforming the others over the entire curve. Nevertheless, coefficient methods outrun others for a restricted amount of selected features, which is often the case in the omics sciences.

In the case of a lower sample size (*cf.* Figure 25), t-stat is performing extremely poorly, and PLS-vip do also especially suffer from the lowered sample size.

With regards to sparse methods with their sparsity level that varies along the curve, LSOPLS has a lower sensitivity than SPLS at equal $\eta$, and optimal $\eta$ is lower for LSOPLS ($\eta = 0.05$) than for SPLS ($\eta = 0.3$), as expected from the previous results. For higher levels of sparsity, LSOPLS outperforms SPLS: for a same sensitivity rate, it will recover less false discoveries. SPLS has the advantage to be able to select more true features, but at the expense of more false discoveries as well. Beware that these averaged performances are very poor for LSOPLS with $\eta > 0.35$, thus LSOPLS should not be applied in general with a large amount of sparsity to ensure an acceptable features accuracy. SPLS performances are also affected by a too large $\eta$ (here $\eta > 0.7$). Although performances are globally lowered in the case of lower sample sizes (*cf.* Figure 25), methods comparison leads to the same observations as above.

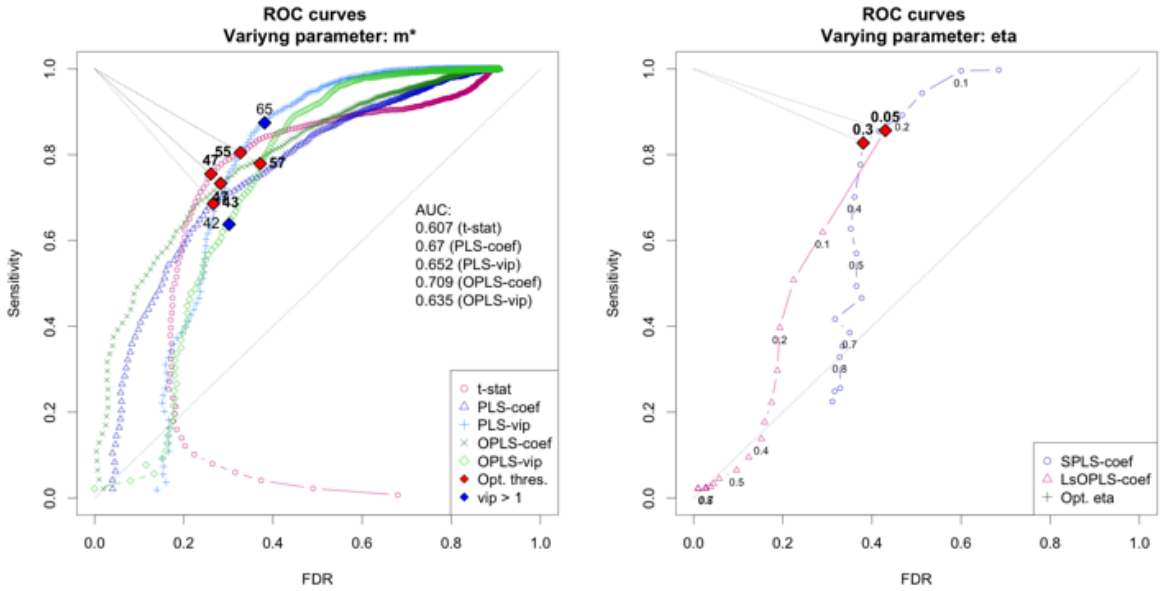


Figure 16: ROC curves for feature accuracy (methods with an arbitrary threshold or sparse methods, $n = 200$). Varying parameters are either the selection threshold $m^* = 1, \ldots, m$ or the sparsity parameter $\eta = 0.05, \ldots, 0.95$ depending on the classifier. Optimal thresholds are represented in red.

### 6.5.2 Number of true features selected

Figure 17 presents the number of correct features selected across the 50 resampled sub-samples with values ranging from 0 (no correct feature selected) to 46 (all the true features were recovered). Note that all the methods cannot be directly compared to each other since the threshold for a part of them is arbitrarily chosen.

Methods with an arbitrary threshold have their median value varying between 30 (OPLS-vip) and 35 (t-stat). T-stat has the highest median number of correct identifications. However, from previous results, it has been seen that it is at the cost of much more false discoveries. For some of the resampled samples, PLS had a lower rate of recovery. Otherwise, performances do not vary a lot across the sub-samples.

For methods with an automatic threshold, t-fdr is also recovering a high amount of true features (but still with a large amount of false discoveries), closely followed by PLS-vip1. OPLS-vip1 has a less important recovery rate. Finally, SPLS and LSOPLS have a much more scattered number of TP and their sensitivity can be above or below other methods' ones. This is partly due to those sparse classifiers' construction itself that will directly decide on the size of the selection set, which that could be much lower than 46.

Table 10 complements the results from the previous graph for the automatic threshold classifiers. In this table are represented ROC indices derived from the selection sets of those classifiers that were trained with optimal MPs, and averaged across the 50 sub-samples. T-fdr leads to a higher recovery rate of true features but its FDR is highly inflated as well. The F1 score, which summarises the trade-off between these two indices, is best for PLS-vip1. Both vip1 methods outperform sparse ones. It can be noted that LSOPLS has the lowest FDR but is not able to recover more than half of the true features, which is due

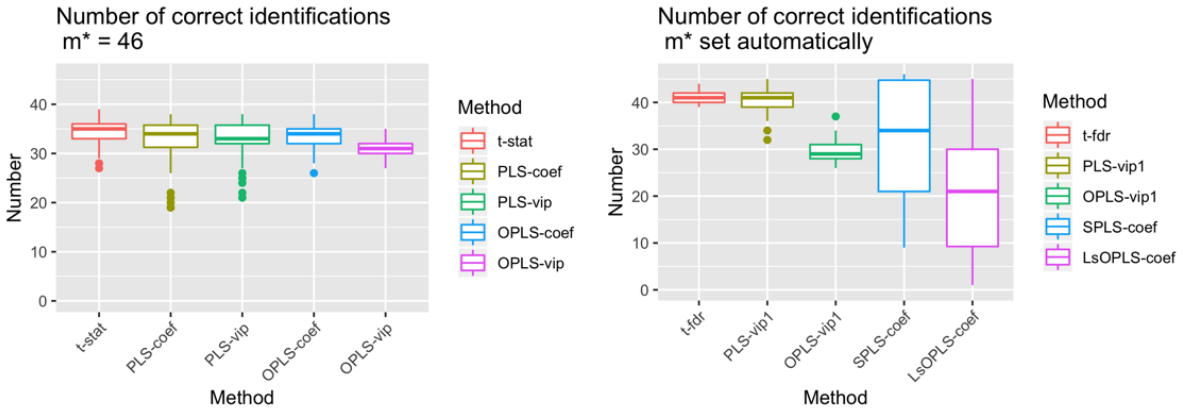


Figure 17: Number of correct features selection. Note that for arbitrary threshold methods, it corresponds to the number of selected features for the 1:46th ranked features ($n = 200$).

to its lower number of selected ones. In the lower sample size case (*cf.* Table 13), t-fdr only recovers a very small amount of features. Based on the F1 score, LSOPLS is more affected by the size drop whereas other methods are moderately affected.

Table 10: Mean ROC indices from the actual selection sets based on the optimal MP (automatic threshold methods) for features accuracy ($n = 200$).

| | **Sensitivity** | **FDR** | **F1** |
|---|---|---|---|
| t-fdr | 89.35 | 59.97 | 55.28 |
| PLS-vip1 | 87.65 | 36.92 | 73.36 |
| OPLS-vip1 | 64.65 | 30.23 | 67.11 |
| SPLS-coef | 70.91 | 45.5 | 61.63 |
| LsOPLS-coef | 44.22 | 23.55 | 56.03 |

## 6.6 Selected features inspection on the spectral scale

This section proposes several graphical results to illustrate, on the entire spectral scale or by independent biomarker, the feature selection achieved by the different methods. Once more, for methods needing to set a threshold $m^*$, one arbitrarily decides to keep the total number of generated true features (46) as the number of variables to select based on their features scores.

In this first graph (Figure 18), features scores $\boldsymbol{B}$ are averaged over the 50 data sub-samples ($n = 200$). Here, the coefficient intensities can be directly compared to each other. The main interest of this graph lies in the representation of the selection threshold for each method. This threshold is either: values different from 0 (t-fdr, SPLS-coef and LSOPLS-coef) or superior to the horizontal dotted lines for the remaining ones.

T-stat has high negative mean intensities for almost all non-discriminant descriptors. As a consequence, it leads to a larger amount of false discoveries. With the threshold fixed at $m^* = 46$ (blue line), few false discoveries are still observed, although much less than for t-fdr.

Among the coefficient methods, PLS and OPLS share approximatively the same profile and have much more small intensities, corresponding to noise only, across the spectrum than their sparse counterparts. They have higher mean feature scores in the noiseless area than in the noisy part, to the contrary of vip methods. SPLS is also able to well recover biomarkers from the noise-free side of the spectrum but has much lower scores on the noisy part of it. It still has a lot of small non-null intensities, especially on the right side. LSOPLS has the lowest mean scores for the noise-free features across all coefficient methods. Finally, the threshold of 1 applied to vip methods is too high for OPLS-vip to detect biomarkers in the noise-free area.

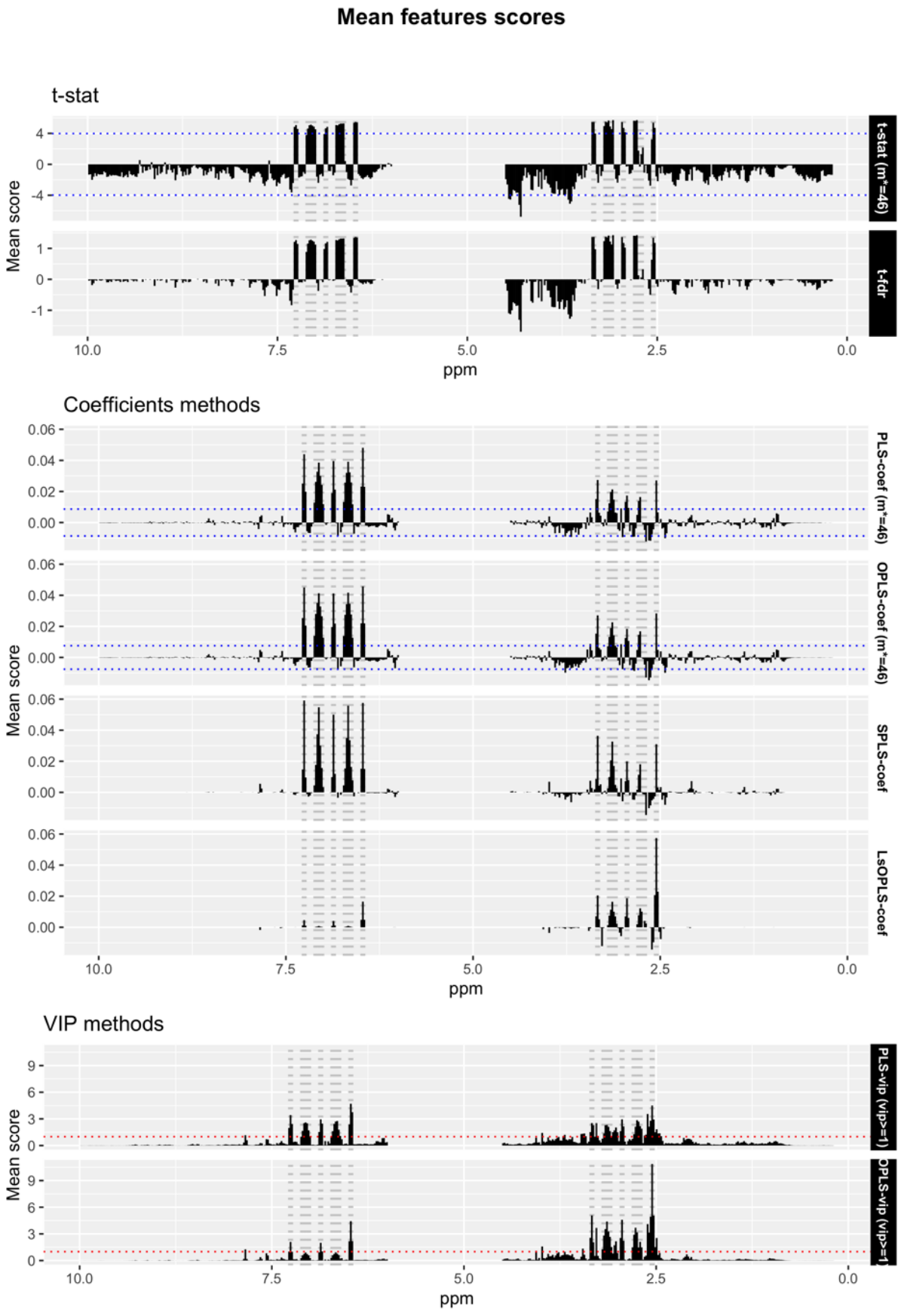


Figure 18: Mean features scores $\boldsymbol{B}$. Blue lines set the arbitrary threshold, it corresponds to the selection of 46 features ($n = 200$) and red lines represent the VIP$\geq$ 1. For sparse methods, recall that all coefficients $\neq 0$ are selected.

An interesting way to present the results is to draw the frequencies of TP as positive intensities and FP as negative ones (Figure 19, $n = 200$). Note that in this second graph, all the methods cannot be directly compared to each other since the threshold for a part of them is arbitrarily chosen. As observed previously, both approaches based on the t-statistic have a high amount of FP but do have the advantage to detect all the independent biomarkers. Vip methods are better at recovering all relevant features in the noisy part of the spectra, where alteration intensities are also higher, but have more false discoveries that (O)PLS coefficients. On the contrary, the latter have a higher sensitivity in the left side of the spectra, *i.e.* in the noise-free area, where spectral intensities are lower.

With regards to the automatic threshold techniques, vip1 methods do, as previously noted, better recover true features in the noisy part of the spectra. PLS-vip1 has the highest TP frequencies among those methods, with a limited amount of false discoveries. SPLS and LSOPLS are surprisingly leading to a lot of false discoveries. Combined with the mean scores figure above, this can be explained by the fact that those algorithms are still selecting variables with very low – still non-null – intensities, and the sparsity parameter is somewhat under-estimated. Furthermore, the frequencies of TP are not identical among the descriptors of a same biomarker. This means that those sparse techniques will generally select only a subset of features from a same biomarker as they represent redundant information. In all logic such classifiers are better suited for minimal-optimal problems. When the sample size is lowered (*cf.* Figure 28), t-fdr is affected with a clearly reduced TP rate while other methods' results are mainly unaffected.

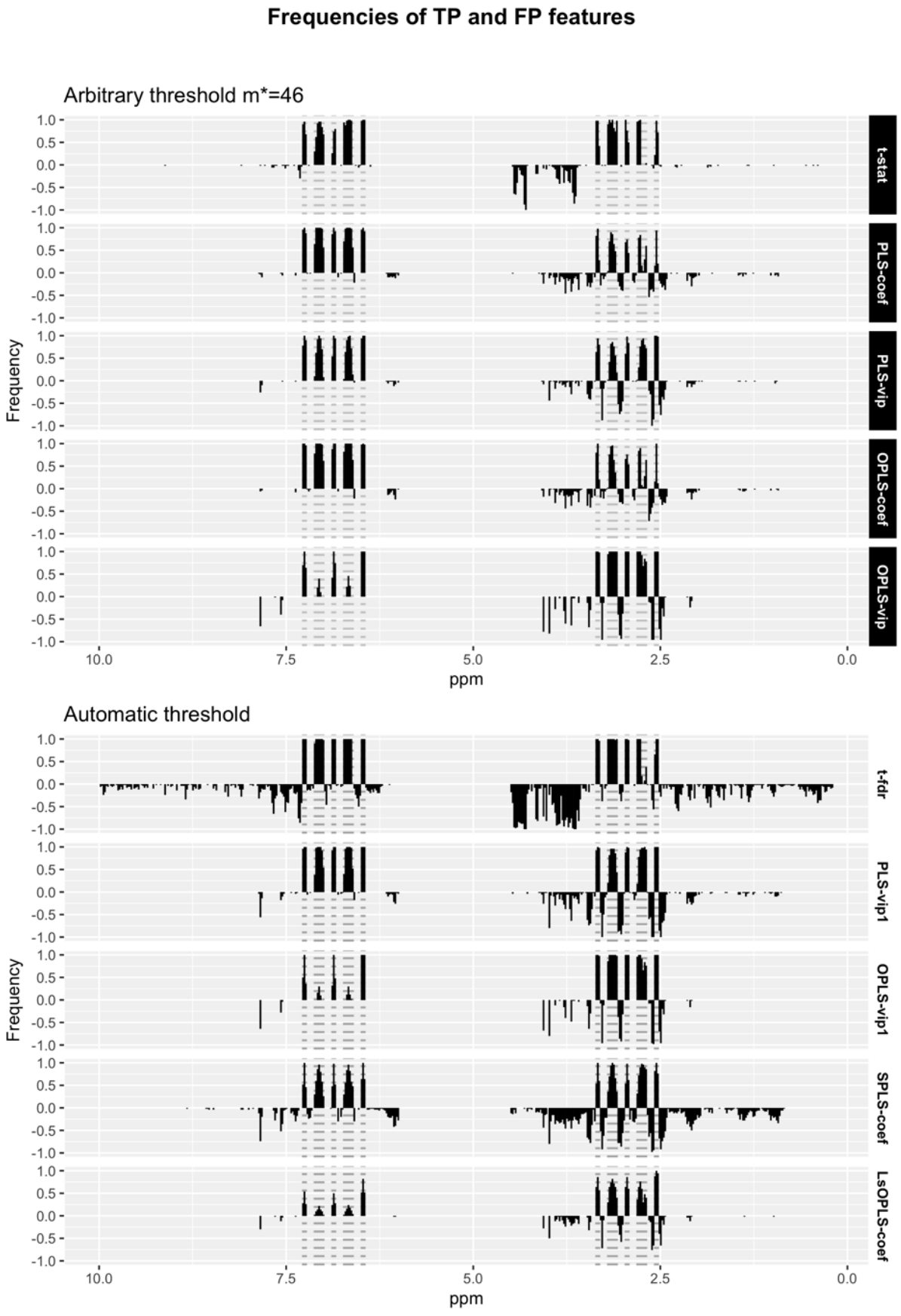


Figure 19: Frequencies of TP and FP. Note that for methods with an arbitrary threshold, it corresponds to frequencies of selected features for the 1:46th ranked features ($n = 200$).

The last graphs for this section (Figure 20, $n = 200$) are a summarised representation of the correct identification percentage by independent biomarkers for each method. It is measured by biomarker as either (A) the number of TP divided by the number of total features in this biomarker or (B) as the mean value of a binary-coded vector with 1 = at least one feature from this biomarker is found and 0 = none of the features from this biomarker is selected. Therefore, methods that are able to recover all the correlated features from a same biomarker will have a higher % of correct identification and are in favour of the *all-relevant problem*, whereas being able to recover at least one feature per independent biomarker represents the solution to the *minimal-optimal problem*. Recall that Biomarkers 1-3 are located in the noise-free area while Biomarkers 4-6 are situated in the noisy area. Here again, methods with arbitrary and automatic thresholds cannot be compared to each other.

In the A-scenario, and for methods based on an arbitrary threshold, no one method is outperforming the others. Regression coefficients are better at recovering true features located in the noise-free area but are much less efficients regarding biomarkers in the noisy area. The opposite is true for vip methods though they still maintain a high true discovery rate for Biom 3. For both biomarker locations, OPLS methods have a better percentage of identification than PLS ones. T-stat has mid-way performances across all biomarkers.

With regards to the automatic threshold methods, t-fdr can recover the largest amount of total discriminant features, but again at the expense of a high FDR. It is closely followed by PLS-vip1 that shows a satisfying recovery rate of at least 75% for every biomarker. In comparison, OPLS-vip1 could recover biomarkers in the noisy area with higher intensities but is affected by lower intensities as it was not able to find all features in the noise-free area. Sparse methods logically selected less features by biomarker. As observed for other methods, SPLS better recovered features in the noise-free area. For smaller sample sizes (*cf.* Figure 29), the average rate of identification is lowered for all methods and they have a similar behaviour, except for methods derived from the t-statistic.

In the B-scenario (the *minimal-optimal problem*), all arbitrary threshold methods (except for OPLS-vip and Biom 3) were able to recover at least one feature from each independent biomarker across sub-samples. For methods with an automatic threshold, vip1 methods and SPLS also recovered $\simeq 100\%$ of biomarkers. LSOPLS for its part was not able to discover at least one discriminant feature for all resampled sub-samples, and especially in the noise-free area. This is probably linked to the small amount of selected variables by this method. Again, when the sample size drops, performances are lowered but their behaviour remains equivalent (except fo t-stat and t-fdr).

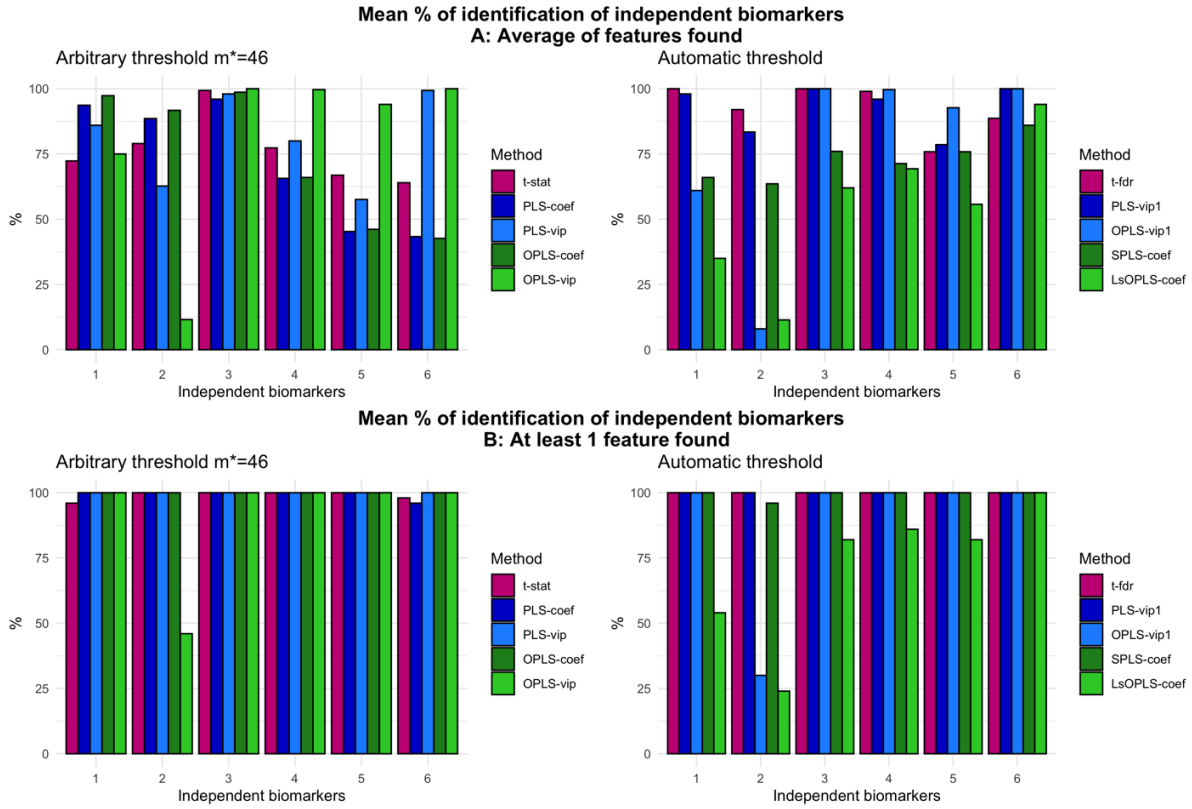


Figure 20: Mean number of identification of independent biomarkers per method (A) calculated as the average of features found for each biomarker and (B) where at least 1 feature found for each biomarker. Note that for arbitrary threshold methods, it corresponds to frequencies of selected features for the 1:46th ranked features ($n = 200$).

## 6.7 Feature selection stability

Feature selection stability is essential to achieve reproducible research: having a stable set of features is as much important as identifying relevant ones (Nogueira *et al.* , 2017). The features ranking/selection reproducibility across subsets of data is also a major concern for the classifiers evaluation and this topic starts to gain more importance in the feature selection context (read for example Abeel *et al.* (2009), and also (Christin *et al.* , 2013; Rinaudo *et al.* , 2016) in metabolomics applications).

Indeed it is also important for the rankings or selection sets to be similar across the resampled subsamples for every feature selection technique. In a way, it evaluates the robustness of the classification algorithms, given the data characteristics (noise, dimensionality, etc.). A procedure is considered unstable if a small change in the dataset leads to a very different ranking or subset selection. If the true features are unknown, which is the case for experimental datasets, feature stability could be a good indicator of the results reliability. Although not used to that purpose, such stability criterion could also presumably be used to select the sparsity level of sparse algorithms, such as LASSO-type regressions (Meinshausen & Bühlmann, 2010).

The selected stability measures are based on the work of Nogueira *et al.* (2017) where the authors compare existing measures to a new one, based on a statistical point of view. Indeed, there is plenty of existing robustness criteria. Their measurement is not trivial and desired properties must be met to adequately quantify the features stability.

The criterion suggested by Nogueira *et al.* (2017) meets the expected properties of such estimator: it is fully defined, is strictly monotonic, is bounded reaches its maximum value *iff* all features across the features subsets are identical, it has correction for chance. It further has the ability to compare subsets of features with different sizes.

Consider a binary matrix $\boldsymbol{D}$ of size $(m \times R)$ where each element $d_{ij}$ corresponds to the selection of a given feature $i = 1, \ldots, m$ in the resampled samples $r = 1, \ldots, R$, where

$$d_{ir} = \begin{cases} 1 & \text{if } i \in \mathcal{A}_r \\ 0 & \text{otherwise.} \end{cases}$$

Let $\Phi(\boldsymbol{D})$ be the stability function with input the binary matrix $\boldsymbol{D}$ and as output a percentage between 0 and 100%:

$$\Phi(\boldsymbol{D}) = 1 - \frac{\frac{1}{m}\sum_{i=1}^{m} s_i^2}{\mathbb{E}[\frac{1}{m}\sum_{i=1}^{m} s_i^2 | H_0]} = 1 - \frac{\frac{1}{m}\sum_{i=1}^{m} s_i^2}{\frac{\bar{m}^*}{m}(1 - \frac{\bar{m}^*}{m})} \; ; \tag{22}$$

where $s_i^2 = \frac{R}{R-1} f_i(1 - f_i)$ is the unbiased sample variance of the selection of the $i^{th}$ feature with $f_i$ the observed frequency of selection of the $i^{th}$ feature: $f_i = \frac{1}{R}\sum_{r=1}^{R} d_{ir}$. Note that this formula is derived from a Bernoulli distribution. $\bar{m}^* = \frac{1}{R}\sum_{i=1}^{m}\sum_{r=1}^{R} d_{ir} = \sum_{i=1}^{m} f_i$ is the mean number of selected features across all selection sets $\mathcal{A}$.

Stability results are presented in Figure 21. The first graph concerns methods with an arbitrary threshold where $m^*$ varies along the x-axis. The second graph shows the feature selection stability for sparse methods, with their degree of sparsity ($\eta$) varying along the x-axis.

For methods with an arbitrary threshold, this stability is at its lowest at the start of the curve. It then greatly increases when the selection sets are including the first 1:50 features. Then, they decrease to reach an approximatively constant rate. Vip methods achieve the best stability rate among all those methods and OPLS-vip has the best stability at lower values of the threshold. Coefficient methods have equivalent performances. The feature selection stability is greatly varying for t-stat. After reaching its best value around $m^* = 50$, it considerably drops to become the less stable method. From this graph, we can say that feature selection stability is greatly depending on the number of features selected. Those results can be analysed in parallel with the ROC curves from Figure 16. For vip and t-stat methods, the observed peak corresponds to the optimal trade-off on the ROC curve. However, for coefficient methods the correspondance is less clear.

With regards to the sparse methods, feature selection stability is slightly decreasing with an increase in the degree of sparsity for SPLS while it has a V-shape for LSOPLS. The latter has his feature selection stability more affected by the level of sparsity. Finally, note that stability is in average higher for vip methods than for sparse methods. In Table 14 are given the actual selection sets from other methods with an automatic threshold. OPLS-vip1 has the highest stability, a result that corroborates with the previous graph.

In the case of a reduced sample size (*cf.* Figure 30), curves keep the same shapes. Except for t-stat that is greatly affected, their stability is slightly diminished. Their relative performances are maintained as well.

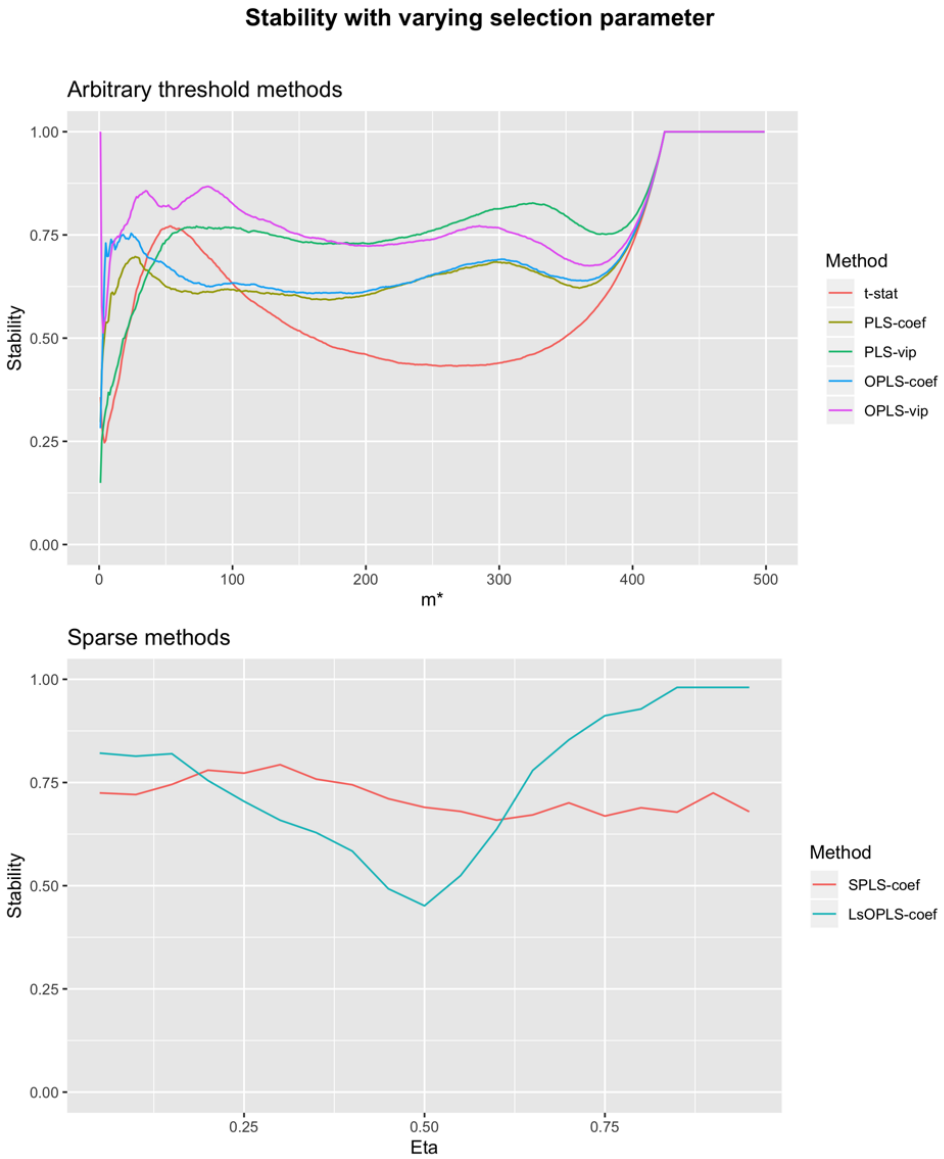


Figure 21: Features Stability across the number of selected features for methods with an arbitrary threshold and across $\eta$ for sparse methods ($n = 200$).

# 7 Conclusions

The success of biomarker discovery in the omics sciences is heavily dependent on the multivariate method(s) used to extract the features but also on the data characteristics. In this paper, classic

Table 11: Feature selection stability for the other automatic threshold methods ($n = 200$).

| t-fdr | PLS-vip1 | OPLS-vip1 |
|---|---|---|
| 60.98 | 75.77 | 82.32 |

supervised approaches – t-test, (O)PLS and their sparse counterparts – have been compared on the Semi-Artificial Database, in order to mimic real experimental conditions and with known in advance biomarkers location. In this study, emphasis has been placed to (1) use numerous resampled sub-datasets to average the methods performances (2) train appropriately the supervised models to avoid their overfit to the data and (3) compare the feature selection techniques with several evaluation measurements, which, for part of them, takes into account the prior information from the true features location.

Both approaches derived from the t-statistic led to a high amount of false discoveries. By reducing the arbitrary threshold in t-stat, most of these could be avoided. In contrast, t-fdr selected a large amount of FP due to the low standard deviation of noise descriptors, even in the case of a small FDR rate that was fixed at 0.05. Still, the advantage of this approach is that it recovers all correlated features. It could therefore be used in the all discriminant approach, provided that the FDR can truly be controlled or with a low threshold. However, we found that performances of both methods were also highly sensitive to a reduced sample size, which is a more realistic experimental condition.

With regards to methods based on (O)PLS, coefficient and vip methods have the disadvantage that no clear threshold is defined, though it indeed affects the feature selection stability and accuracy. In this study, the ratio TP/FDR and feature selection stability were maximum around 50 features selected, namely around 10% of the total number of variables in the data. Selecting a lower number leads to a much lower sensibility and stability whereas, selecting more features decreases specificity. The fixed threshold of 1 for vip methods enabled to have a good feature accuracy (F1 score) for PLS but this threshold was too high for OPLS to recover discriminant features with lower intensities. Nevertheless, both vip1 methods outperformed sparse classifiers to detect all the discriminant features. These non-sparse multivariate methods do have each their advantages and drawbacks. vip methods were better at recovering biomarkers in the noisy area with higher features intensities, whereas coefficients-based approaches were in general better at recovering biomarkers having lower intensities in the noise-free area. Models with an orthogonal filter (OPLS and LSOPLS) seemed indeed to be more sensible to the peaks intensity: they could not well recover lower intensity biomarkers in the noise-free area.

We saw that the degree of sparsity also affected feature selection stability and accuracy. Here can be concluded to the importance of a fine tuning of the MPs for such embedded sparse approaches. Those methods should perhaps rely on another criterion to decide on the optimal combination of the number of LV and the sparsity degree since multiple solutions were equivalent in terms of the AUC. Further investigations on the interaction between MPs should be led, as they directly impact features accuracy and stability. Indeed, it seems that sparse methods have the potential to achieve better performances but would need a better training criterion, based *e.g.* on the desired features subset size or its stability. Due to its prior removal of orthogonal variation, LSOPLS has an optimal sparsity parameter that tends to be lower than SPLS. Results also showed that applying SPLS, but more importantly LSOPLS, with a large degree of sparsity is counterproductive as their feature selection sensibility and specificity will drop considerably. Surprisingly, still a lot of FP could be observed for such sparse classifiers. Since the degree of sparsity should not be increased excessively, one solution could be to slightly increase the 0 threshold applied to the regression coefficients for feature selection. Furthermore, the frequencies of selected TP were not identical among the descriptors of a same biomarker. This means that those sparse techniques will generally select only a subset of those features since they represent redundant information. In all logic such classifiers are better suited for minimal-optimal problems.

Results were mainly conducted on the larger sample size. When only a reduced sample size is available, overall performances are logically reduced, but their relative performances were maintained. Nevertheless, some of the studied methods were more robust while others did suffer more: t-stat and t-fdr had their performances affected with a significant drop of true features recovery. LSOPLS was also less robust to the sample size decrease than the other related methods.

In the case of our resampled dataset, PLS-vip1 had the best overall performances regarding all aspects of this study, and even in the low sample size case. But other methods were sometimes better with regard to the various investigated evaluation measurements. Yet, this result is truly dependent on the dataset at hand and its characteristics (level of noise, number of true discriminant features, number of samples, intensities of those features, etc.). Therefore PLS-vip1 may not be always the best method to detect potential biomarkers.

Other methodological aspects that were not investigated are the feature selection methods combination to take advantage of their different assets. This could be adequate since here, no method was always superior to the rest. Other sparse classifiers, such as the group lasso, could also take into account the feature membership to each particular biomarker. With regards to the problem of not selecting all the correlated biomarkers, a solution could be to apply the bagging principle that originates from Random Forest, by resampling in the feature space (T. K. Ho, 1998). Finally, more realistic datasets (*e.g.* a spiked experimental dataset), could be applied with our methodology to inspect the replicability of our results.

# A Other results with $n = 60$

## A.1 MPs stability

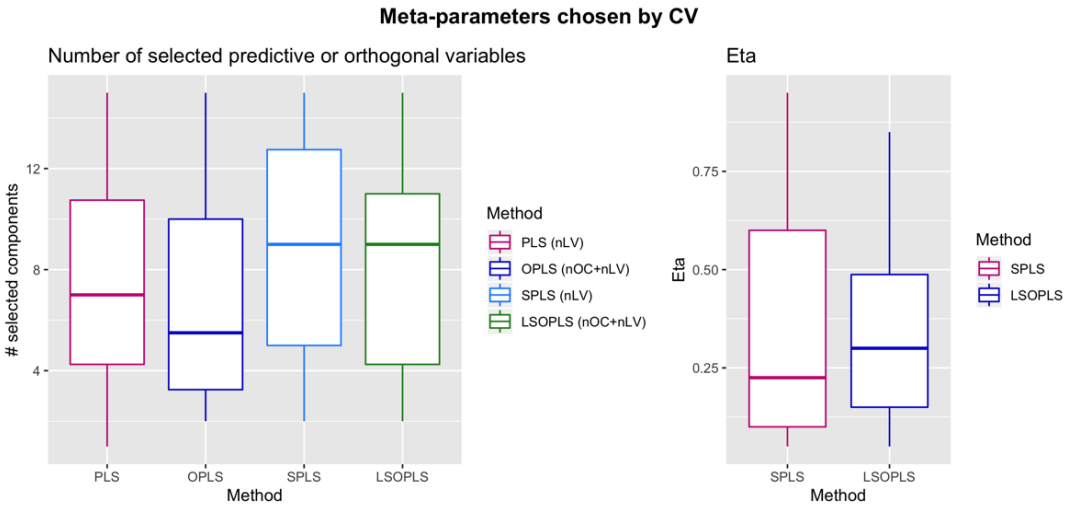


Figure 22: Distribution of MPs chosen by the inner-loop CV ($n = 200$).

## A.2 Size of the selection sets for automatic threshold methods

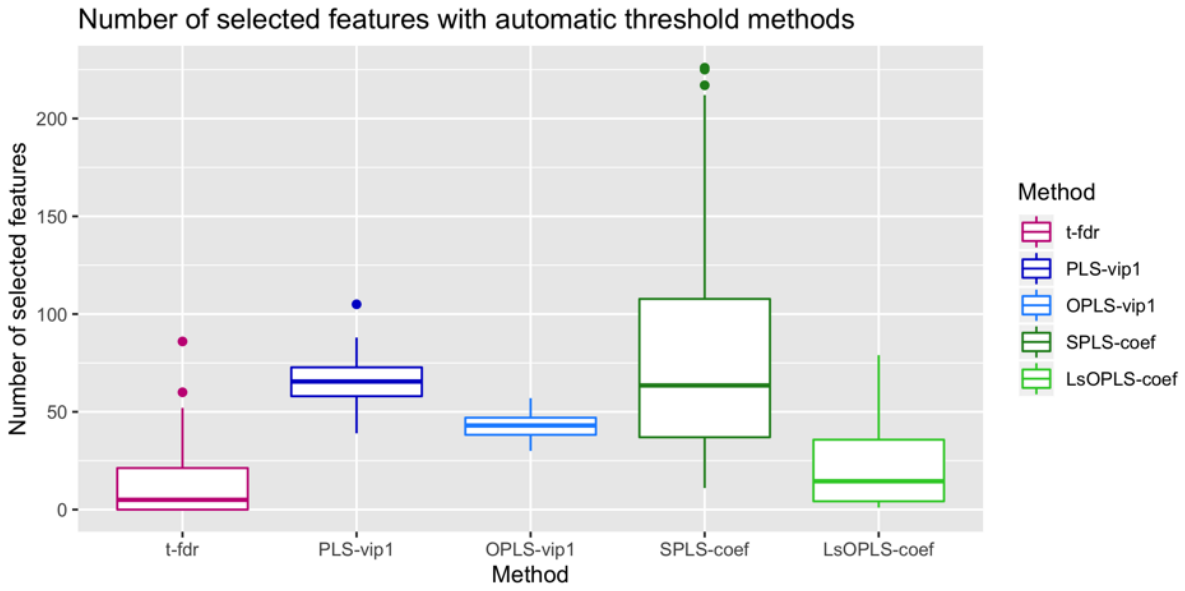


Figure 23: Number of selected features for automatic threshold methods ($n = 60$).

## A.3 Classification accuracy

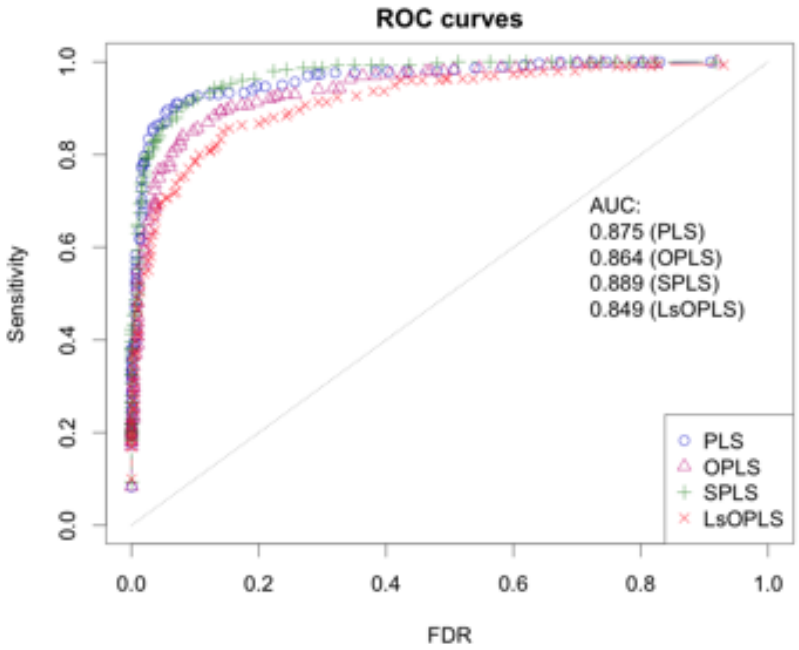


Figure 24: ROC curves for classification accuracy ($n = 60$).

Table 12: Classification performances: mean values of sensitivity, FDR, average confidence in classification, average predictive ability and F1 score at best threshold ($n = 60$).

| | **AUC** | **Sensitivity** | **FDR** | **Average confidence in classification** | **Average predictive ability** | **F1 score** |
|---|---|---|---|---|---|---|
| PLS | 0.88 | 0.91 | 0.07 | 0.92 | 0.93 | 0.92 |
| OPLS | 0.86 | 0.88 | 0.13 | 0.88 | 0.9 | 0.88 |
| SPLS | 0.89 | 0.91 | 0.09 | 0.91 | 0.93 | 0.91 |
| LsOPLS | 0.85 | 0.85 | 0.14 | 0.85 | 0.87 | 0.85 |

## A.4 General assessment of features accuracy

Table 13: Mean ROC indices from the actual selection sets based on the optimal MP (automatic threshold methods) for features accuracy ($n = 60$).

| | **Sensitivity** | **FDR** | **F1** |
|---|---|---|---|
| t-fdr | 10.22 | 72.4 | 14.91 |
| PLS-vip1 | 79.13 | 43.42 | 65.98 |
| OPLS-vip1 | 59.65 | 36.35 | 61.59 |
| SPLS-coef | 69.26 | 52.29 | 56.5 |
| LsOPLS-coef | 31.87 | 23.92 | 44.92 |

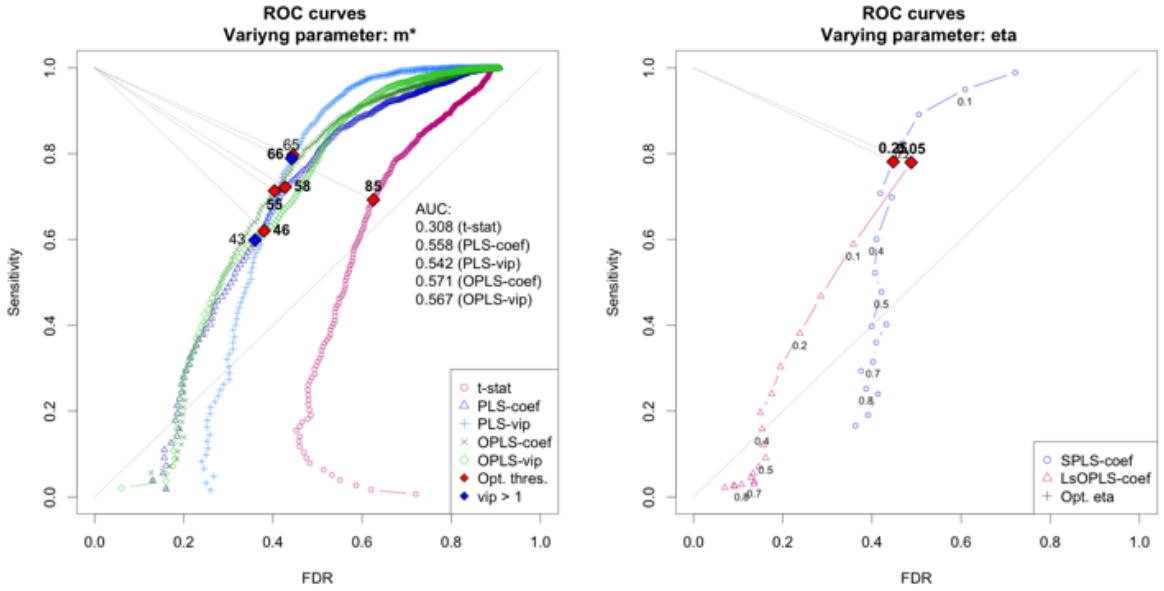


Figure 25: ROC curves for feature accuracy (methods with an arbitrary threshold or sparse methods, $n = 60$). Varying parameters are either the selection threshold $m^* = 1, \ldots, m$ or the sparsity parameter $\eta = 0.05, \ldots, 0.95$ depending on the classifier. Optimal thresholds are represented in red.

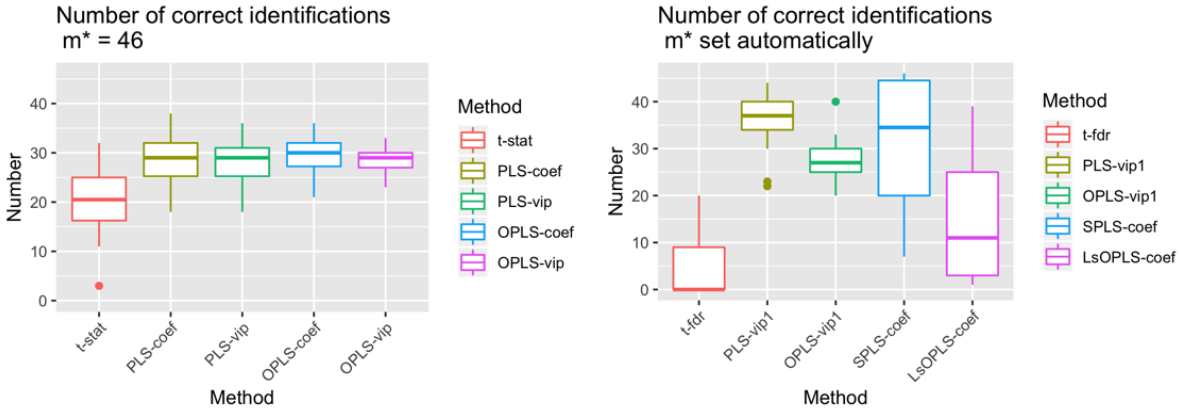


Figure 26: Number of correct features selection. Note that for arbitrary threshold methods, it corresponds to the number of selected features for the 1:46th ranked features ($n = 60$).

## A.5 Selected features inspection on the spectral scale

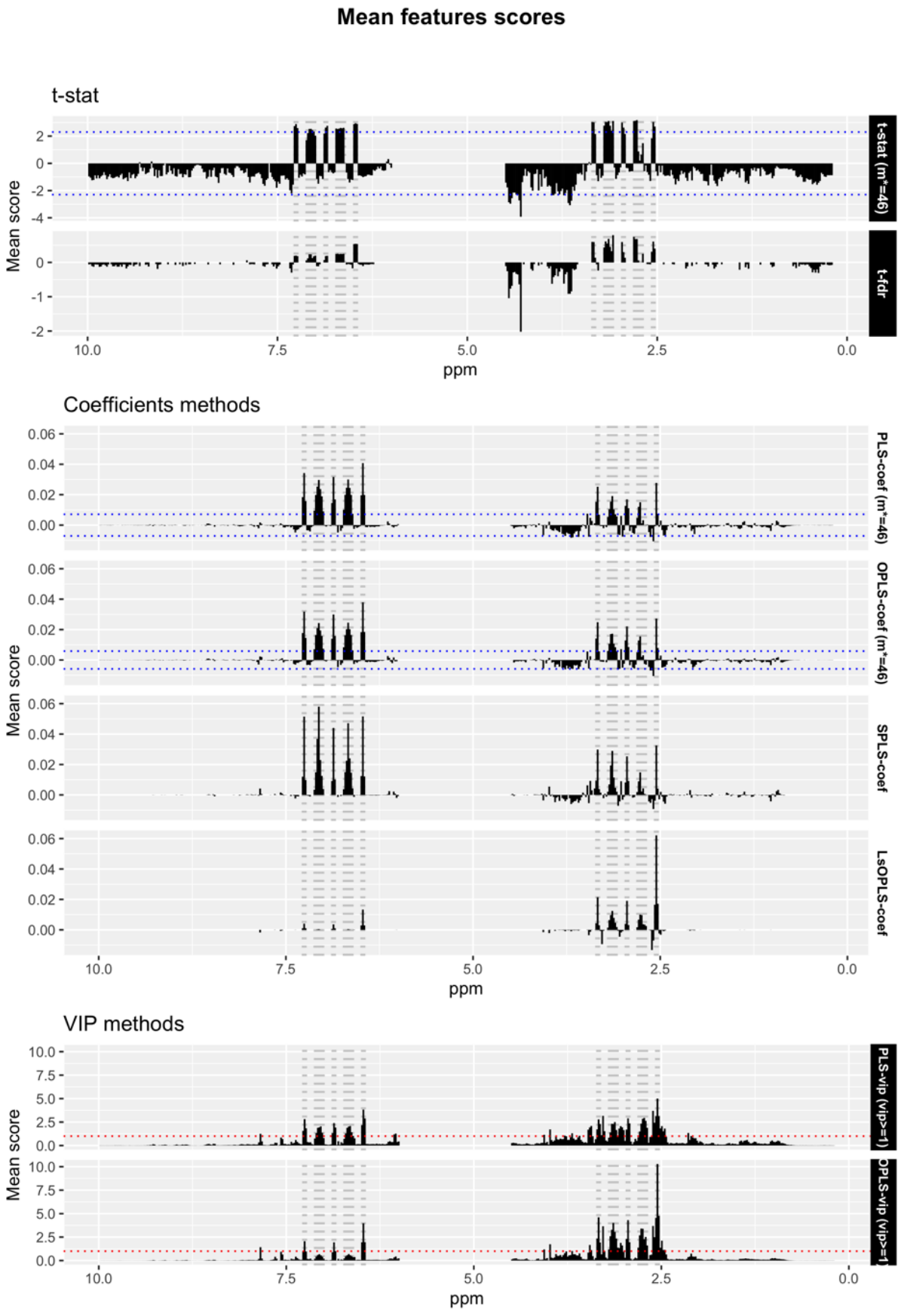


Figure 27: Mean features scores $\boldsymbol{B}$. Blue lines set the arbitrary threshold, it corresponds to the selection of 46 features ($n = 60$) and red lines represent the VIP$\geq 1$. For sparse methods, recall that all coefficients $\neq 0$ are selected.

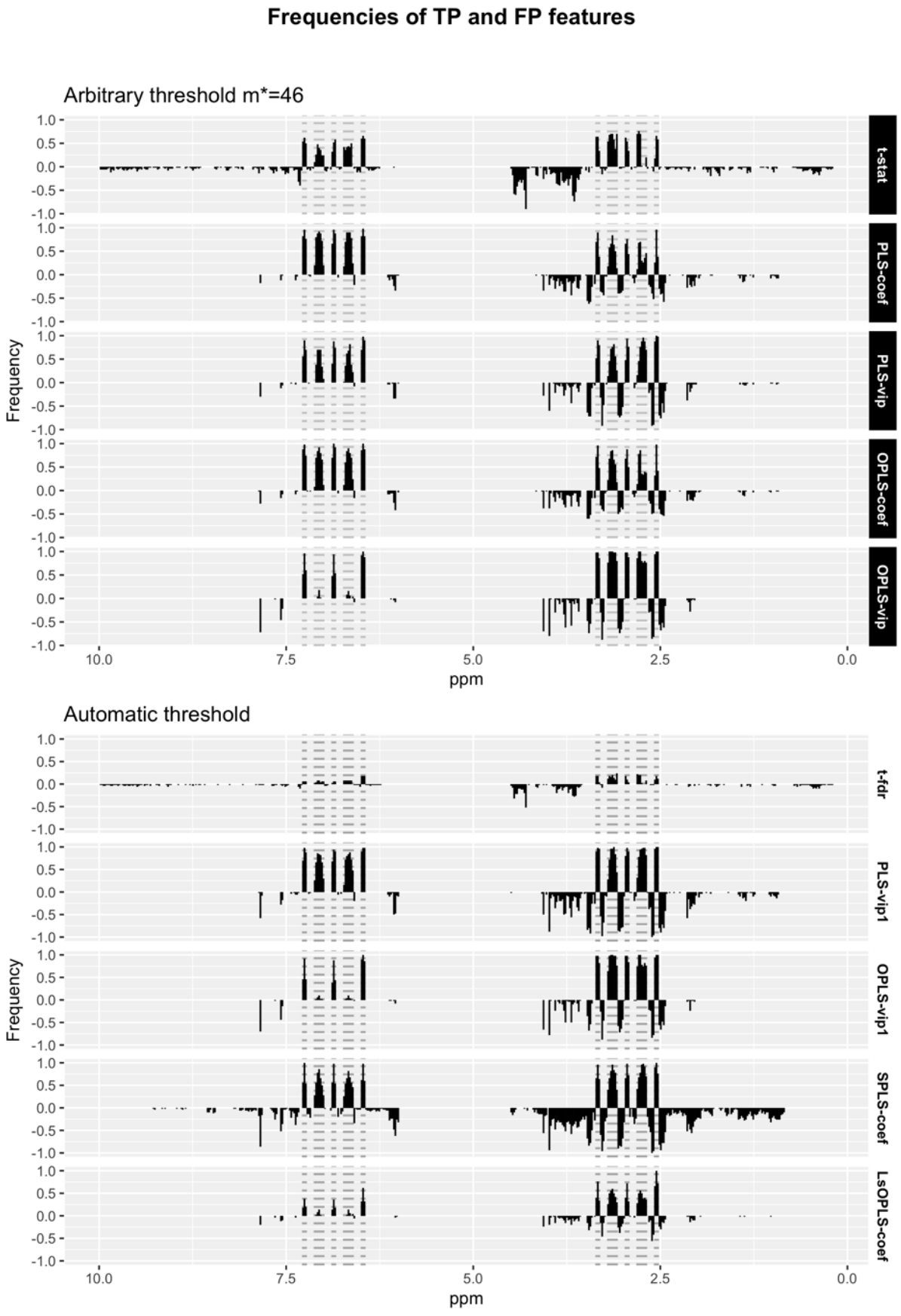


Figure 28: Frequencies of TP and FP. Note that for methods with arbitrary threshold, it corresponds to frequencies of selected features for the 1:46th ranked features ($n = 60$).

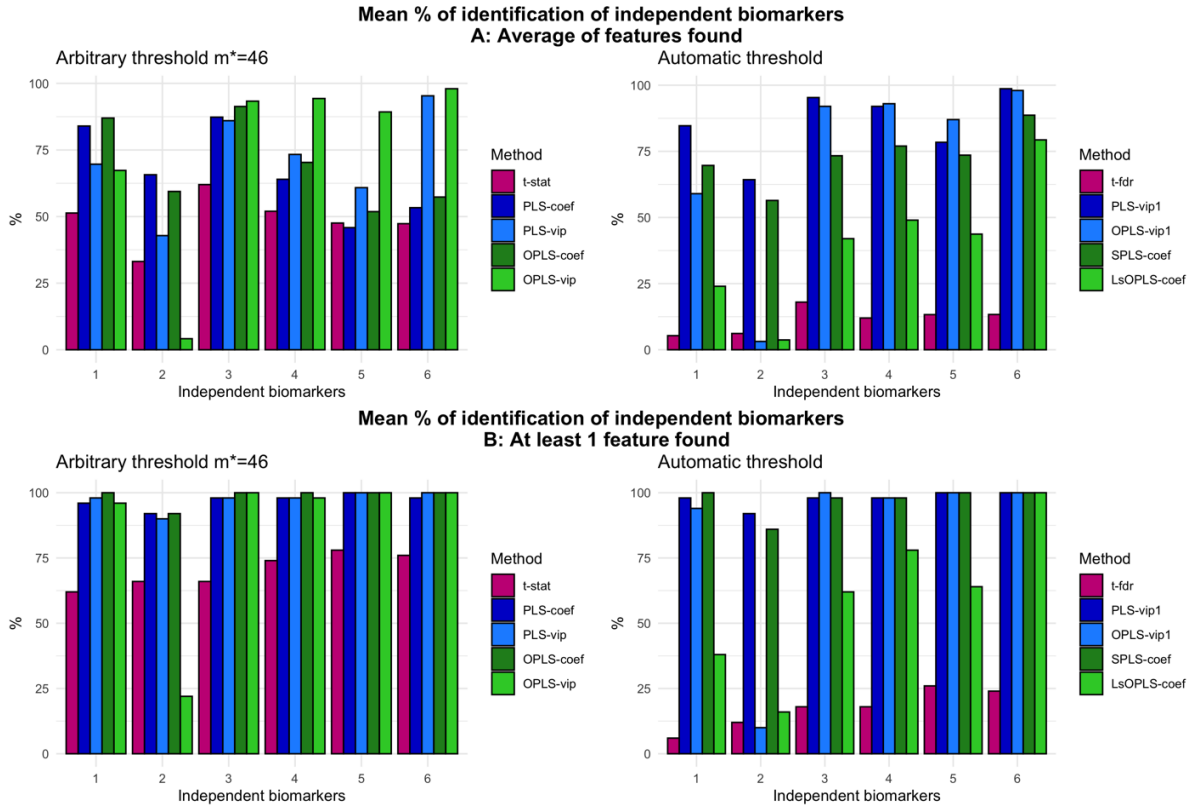


Figure 29: Mean number of identification of independent biomarkers per method (A) calculated as the average of features found for each biomarker and (B) where at least 1 feature found for each biomarker. Note that for arbitrary threshold methods, it corresponds to frequencies of selected features for the 1:46th ranked features ($n = 60$).

## A.6 Feature stability

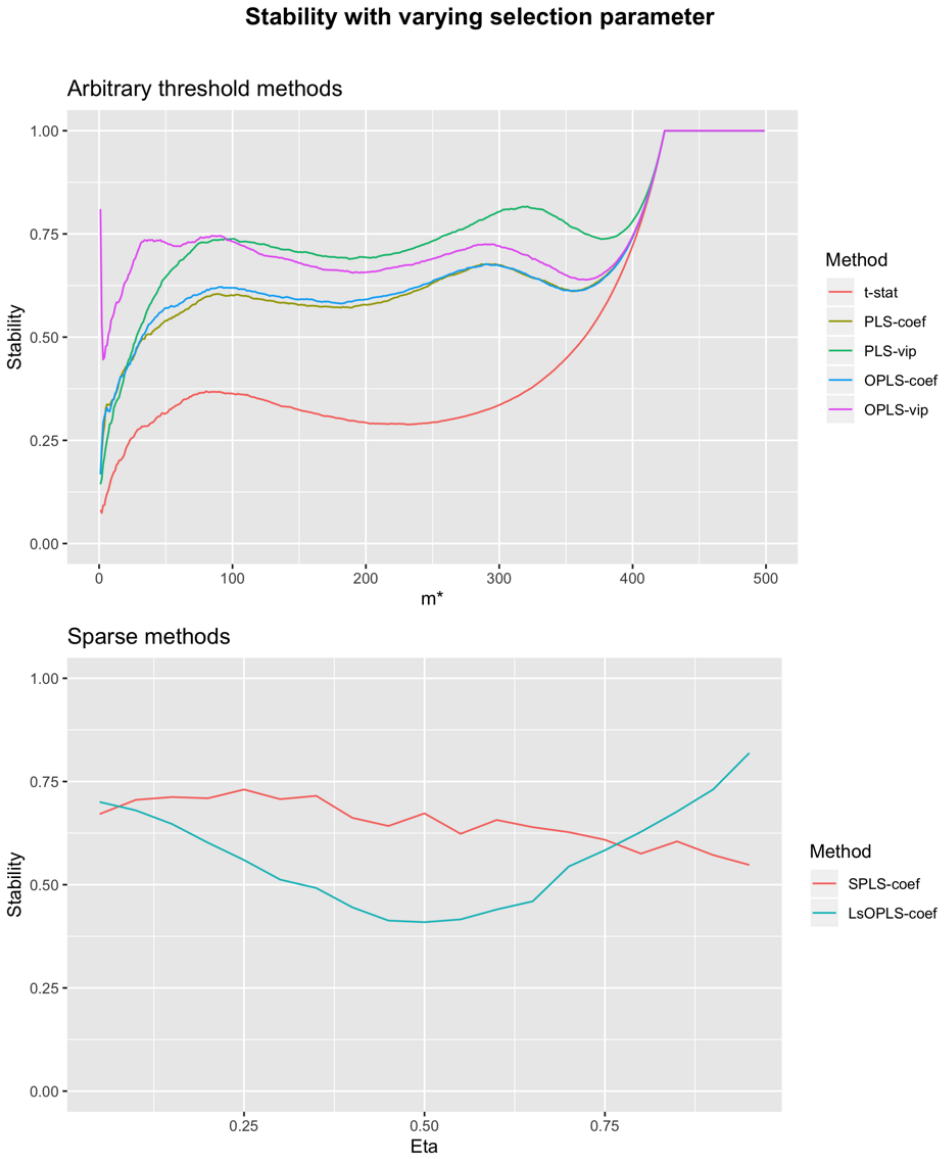


Figure 30: Features Stability across the number of selected features for methods with an arbitrary threshold and across $\eta$ for sparse methods ($n = 60$).

Table 14: Feature selection stability for the other automatic threshold methods ($n = 60$).

| t-fdr | PLS-vip1 | OPLS-vip1 |
|---|---|---|
| 9.142 | 67.94 | 72.42 |

# References


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

Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., & Saeys, Y. 2009. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, **26**(3), 392–398.

Andersson, M. 2009. A comparison of nine PLS1 algorithms. *Journal of Chemometrics*, **23**(10), 518–529.

Antti, H., Ebbels, T., Keun, H. C., Bollard, M. E., Beckonert, O., Lindon, J. C., Nicholson, J. K., & Holmes, E. 2004. Statistical experimental design and partial least squares regression analysis of biofluid metabonomic NMR and clinical chemistry data for screening of adverse drug effects. *Chemometrics and intelligent laboratory systems*, **73**(1), 139–149.

Barker, M., & Rayens, W. 2003. Partial least squares for discrimination. *Journal of chemometrics*, **17**(3), 166–173.

Benjamini, Y., & Yekutieli, D. 2001. The Control of the False Discovery Rate in Multiple Testing under Dependency. *The Annals of Statistics*, **29**(4), 1165–1188.

Boulesteix, A.-L., & Strimmer, K. 2006. Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. *Briefings in Bioinformatics*, **8**(1), 32–44.

Bylesjö, M., Rantalainen, M., Cloarec, O., Nicholson, J. K., Holmes, E., & Trygg, J. 2006. OPLS discriminant analysis: combining the strengths of PLS-DA and SIMCA classification. *Journal of Chemometrics*, **20**(8-10), 341–351.

Christin, C., Hoefsloot, H. C. J., Smilde, A. K., Hoekman, B., Suits, F., Bischoff, R., & Horvatovich, P. 2013. A Critical Assessment of Feature Selection Methods for Biomarker Discovery in Clinical Proteomics. *Molecular & Cellular Proteomics*, **12**(1), 263–276.

Chun, H., & Keleş, S. 2010. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(1), 3–25.

de Jong, S. 1993. SIMPLS: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, **18**(3), 251–263.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., *et al.* . 2004. Least angle regression. *The Annals of statistics*, **32**(2), 407–499.

Farrés, Mi., Platikanov, S., Tsakovski, S., & Tauler, R. 2015. Comparison of the variable importance in projection (VIP) and of the selectivity ratio (SR) methods for variable selection and interpretation. *Journal of Chemometrics*, **29**(10), 528–536.

Fawcett, T. 2006. An introduction to ROC analysis. *Pattern recognition letters*, **27**(8), 861–874.

Féraud, B., Munaut, C., Martin, M., Verleysen, M., & Govaerts, B. 2017. Combining strong sparsity and competitive predictive power with the L-sOPLS approach for biomarker discovery in metabolomics. *Metabolomics*, **13**(11), 130.

Geladi, P. 1988. Notes on the history and nature of partial least squares (PLS) modelling. *Journal of Chemometrics*, **2**(4), 231–246.

Geladi, P., & Kowalski, B. R. 1986. Partial least-squares regression: a tutorial. *Analytica chimica acta*, **185**, 1–17.

Grissa, D., Pétéra, M., Brandolini, M., Napoli, A., Comte, B., & Pujos-Guillot, E. 2016. Feature Selection Methods for Early Predictive Biomarker Discovery Using Untargeted Metabolomic Data. *Frontiers in Molecular Biosciences*, **3**, 30.

Guyon, I., & Elisseeff, A. 2003. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.*, **3**(Mar.), 1157–1182.

Indahl, U. G., Liland, K. H., & Næs, T. 2009. Canonical partial least squares - a unified PLS approach to classification and regression problems. *Journal of Chemometrics*, **23**(9), 495–504.

Ipsen, I. C. F., & Meyer, C. D. 1998. The Idea Behind Krylov Methods. *The American Mathematical Monthly*, **105**(10), 889–899.

James, G., Witten, D., Hastie, T., & Tibshirani, R. 2013. *An introduction to statistical learning*. Vol. 112. Springer.

Jeanmougin, M., de Reynies, A., Marisa, L., Paccard, C., Nuel, G., & Guedj, M. 2010. Should We Abandon the t-Test in the Analysis of Gene Expression Microarray Data: A Comparison of Variance Modeling Strategies. *PLOS ONE*, **5**(9), 1–9.

Krämer, N. 2007. An overview on the shrinkage properties of partial least squares regression. *Computational Statistics*, **22**(2), 249–273.

Kuyuk, S., & Ercan, I. 2017. Commonly used statistical methods for detecting differential gene expression in microarray experiments. *Biostatistics Epidemiol Int J*, **1**(1), 00001.

Lê Cao, K.-A., Rossouw, D., Robert-Granié, C., & Besse, P. 2008. A sparse PLS for variable selection when integrating omics data. *Statistical applications in genetics and molecular biology*, **7**(1).

Lindgren, F., Geladi, P., & Wold, S. 1993. The kernel algorithm for PLS. *Journal of Chemometrics*, **7**(1), 45–59.

Lindon, J. C., Nicholson, J. K., Holmes, E., Antti, H., Bollard, M. E., Keun, H., Beckonert, O., Ebbels, T. M., Reily, M. D., Robertson, D., *et al.* . 2003. Contemporary issues in toxicology the role of metabonomics in toxicology and its evaluation by the COMET project. *Toxicology and Applied Pharmacology*, **187**(3), 137–146.

Mehmood, T., & Ahmed, B. 2016. The diversity in the applications of partial least squares: an overview. *Journal of Chemometrics*, **30**(1), 4–17.

Mehmood, T., Liland, K. H., Snipen, L., & Sæbø, S. 2012. A review of variable selection methods in Partial Least Squares Regression. *Chemometrics and Intelligent Laboratory Systems*, **118**, 62 – 69.

Meinshausen, N., & Bühlmann, P. 2010. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(4), 417–473.

Naik, P., & Tsai, C.-L. 2000. Partial Least Squares Estimator for Single-Index Models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **62**(4), 763–771.

Nogueira, S., Sechidis, K., & Brown, G. 2017. On the Stability of Feature Selection Algorithms. *Journal of Machine Learning Research*, **18**, 174–1.

Poirion, O. B., Zhu, X., Ching, T., & Garmire, L. 2016. Single-Cell Transcriptomics Bioinformatics and Computational Challenges. *Frontiers in Genetics*, **7**, 163.

Rinaudo, P., Boudah, S., Junot, C., & Thévenot, E. A. 2016. biosigner: A New Method for the Discovery of Significant Molecular Signatures from Omics Data. *Frontiers in Molecular Biosciences*, **3**, 26.

Rohart, F., Gautier, B., Singh, A., & Lê Cao, K.-A. 2017. mixOmics: an R package for 'omics feature selection and multiple data integration. *bioRxiv*.

Rosipal, R., & Krämer, N. 2006. Overview and Recent Advances in Partial Least Squares. *Pages 34–51 of:* Saunders, C., Grobelnik, M., Gunn, S., & Shawe-Taylor, J. (eds), *Subspace, Latent Structure and Feature Selection*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Rousseau, R. 2011. *Statistical contributrion to the analysis of metabonomic data in* $^1$H-NMR *spectroscopy*. Ph.D. thesis, Louvain-la-Neuve: UCL.

Rousseau, R., Govaerts, B., Verleysen, M., & Boulanger, B. 2008. Comparison of some chemometric tools for metabonomics biomarker identification. *Chemometrics and Intelligent Laboratory Systems*, **91**(1), 54 – 66. Selected papers presented at the Chemometrics Congress " CHIMIOMETRIE 2006" Paris, France, 30 November - 1 December 2006.

Shi, L., Westerhuis, J. A., Rosén, J., Landberg, R., & Brunius, C. 2018. Variable selection and validation in multivariate modelling. *Bioinformatics*, **35**(6), 972–980.

Sjöström, M., Wold, S., Lindberg, W., Persson, J.-Å., & Martens, H. 1983. A multivariate calibration problem in analytical chemistry solved by partial least-squares models in latent variables. *Analytica Chimica Acta*, **150**, 61 – 70.

Svensson, O., Kourti, T., & MacGregor, J. F. 2002. An investigation of orthogonal signal correction algorithms and their characteristics. *Journal of Chemometrics*, **16**(4), 176–188.

T. K. Ho. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(8), 832–844.

ter Braak, C. J. F., & de Jong, S. 1998. The objective function of partial least squares regression. *Journal of Chemometrics*, **12**(1), 41–54.

Tibshirani, R. 1996. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, **58**(1), 267–288.

Trygg, J., & Wold, S. 2002. Orthogonal projections to latent structures (O-PLS). *Journal of Chemometrics*, **16**(3), 119–128.

Varmuza, K., & Filzmoser, P. 2016. *Introduction to multivariate statistical analysis in chemometrics*. CRC press.

Wehrens, R. 2011. *Chemometrics with R: multivariate data analysis in the natural sciences and life sciences*. Springer Science & Business Media.

Westerhuis, J. A., van Velzen, E. J. J., Hoefsloot, H. C. J., & Smilde, A. K. 2008. Discriminant Q2 (DQ2) for improved discrimination in PLSDA models. *Metabolomics*, **4**(4), 293–296.

Wiklund, S., Johansson, E., Sjöström, L., Mellerowicz, E. J, Edlund, U., Shockcor, J. P, Gottfries, J., Moritz, T., & Trygg, J. 2008. Visualization of GC/TOF-MS-based metabolomics data for identification of biochemically interesting compounds using OPLS class models. *Analytical Chemistry*, **80**(1), 115–122.

Wold, S., Antti, H., Lindgren, F., & Öhman, J. 1998. Orthogonal signal correction of near-infrared spectra. *Chemometrics and Intelligent Laboratory Systems*, **44**(1), 175–185.

Wold, S., Sjöström, M., & Eriksson, L. 2001. PLS-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, **58**(2), 109–130.

Zou, H., Hastie, T., & Tibshirani, R. 2006. Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, **15**(2), 265–286.