Education and Culture

Erasmus Mundus

emQal

UNIVERSIDADE DO ALGARVE

Variable selection optimization for multivariate classification of metabolomics data

By

Leslie Romelia Euceda Wood

Thesis for the degree of European Master in Quality in Analytical Laboratories

Supervisors: Prof. Dr. Bjørn Grung (University of Bergen, Norway)
Prof. Dr. Yizeng Liang (Central South University, PR China)

UNIVERSITAS BERGENSIS

Department of Chemistry,
University of Bergen,
Norway

CENTRAL SOUTH UNIVERSITY
中南大学

Research Center of Modernization
of Traditional Herbal Medicines,
Central South University,
Changsha, PR China

Variable selection optimization for multivariate classification of metabolomics data

By

Leslie Romelia Euceda Wood

Thesis for the degree of European Master in Quality in Analytical Laboratories

Supervisors: Prof. Dr. Bjørn Grung (University of Bergen, Norway)
Prof. Dr. Yizeng Liang (Central South University, PR China)

Department of Chemistry,
University of Bergen,
Norway

Research Center of Modernization
of Traditional Herbal Medicines,
Central South University,
Changsha, PR China

CONTENTS

ACKNOWLEDGEMENTS

First, I would like to thank my supervisor, Prof. Bjørn Grung, for his guidance without which the completion of this thesis would have not been possible. Meetings with you are always interesting and always result in me learning something new, both work related and not. Working with you made me challenge myself to deliver my best effort doing something I was originally not very familiar with. Thank you for your availability and support and also for your help in practical matters related to living in Bergen.

Thanks are also in order for everyone from Professor Yizeng Liang's group in the Central South University of China (CSU), in which a part of this project was carried out. Special thanks to Professor Liang, Xuxia Long, Yonghuang Yun, Wei Fan and Naiping Dong, who made sure that my stay in China was as pleasant as possible, always available to help with anything I needed. Sharing your culture with me contributed greatly in making my experience there unforgettable, and I will always be grateful for that.

Thank you to Jianhua Huang and Yun Yonghuang, for their direction in the work done in CSU, as well as the contact kept afterwards, always patient and taking time to give explanations, clear doubts and interpret results.

I would also like to thank everyone that makes Erasmus Mundus possible, in particular those who were in charge of ensuring the edition of EMQAL I belong to kept its course throughout its entire duration. Special thanks to those involved from our host institution, the University of Algarve, especially Professors Isabel Cavaco and José Paulo Pinheiro, Telma Costa and the fantastic team from the International Relations and Mobility Office for their patience and for doing their best to provide us an adequate learning environment while making us feel at home.

I would also like to express my gratitude to my EMQAL colleagues and friends. I consider that we all contributed to each one's successful completion of this program in one way or another. Special thanks to Débora Mendes and Marko Birkic, without whom China would have never been the same, and to Han Zhu, for guiding us through everything she had already been through in Bergen.

Finally, I would like to thank my family: Gustavo, Miriam, Ana Romelia, Glenda and Allan, for their constant love and support throughout my entire academic process and life in general.

Leslie Romelia Euceda Wood
Bergen, May, 2013

LIST OF ABRREVIATIONS

ADA	American Diabetes Association
ARS	Adaptive reweighted sampling
AUC	Area under curve
BMI	Body mass index
BSTFA	N,O-bis(trimethylsilyl)trifluoroacetamide
CARS	Competitive adaptive reweighted sampling
CART	Classification and regression trees
CHOB	Child obesity dataset
COSS	Conditional synergetic score
CV	Cross validation
EDF	Exponentially decreasing function
EPA	Environmental Protection Agency (USA)
Er	Error
FFA	Free fatty acid
FLVM	Fold latent variable matrix
FLVV	Fold latent variable vector
FN	False negative
FP	False positive
FVIM	Fold variable identity matrix
FVIV	Fold variable identity vector
FVNM	Fold variable number matrix
FVNV	Fold variable number vector
GC	Gas chromatography
IUPAC	International Union of Pure and Applied Chemistry
LDA	Linear discriminant analysis
LV	Latent variable
LVAM	Latent variable accuracy matrix
LVMAM	Latent variable mean accuracy matrix
LVMAV	Latent variable mean accuracy vector

MCC	Mathew’s correlation coefficient
MCS	Monte Carlo sampling
MPA	Model population analysis
MS	Mass spectrometry
m/z	Mass-to-charge ratio
NIH	National Institutes of Health (USA)
NIPALS	Nonlinear iterative partial least squares
NIR	Near infrared spectroscopy
NPE	Normal prediction error
OOB	Out-of-bag
PLS	Partial least squares
PLS-DA	Partial least squares discriminant analysis
POCD	Postoperative cognitive dysfunction dataset
PPE	Permuted prediction error
RF	Random forest
RMSE	Root mean square error
RMSECV	Root mean square error of cross validation
ROC	Receiver Operating Characteristic
TMSC	Trimethylsilyl chloride
TN	True negative
TP	True positive
TS	Training set
T2DM	Type 2 diabetes mellitus dataset
SPA	Subwindow permutation analysis
SR	Selectivity ratio
UV	Ultraviolet
VAM	Variable accuracy matrix
VIC	Variable identity cube
VIM	Variable identity matrix
VIV	Variable identity vector
VNC	Variable number cube

VNM	Variable number matrix
VNV	Variable number vector
VS	Variable selection
WHO	World Health Organization

ABSTRACT

Variable selection is an important step in multivariate calibration in which the number of variables in the independent variable matrix is reduced by eliminating those that are not related to the response. Many methods based on different criteria have been developed for this purpose. Some of them include competitive adaptive reweighted sampling (CARS), subwindow permutation analysis (SPA) and random forest (RF) which can be implemented prior to the construction of both regression and classification models. When applied to metabolomics datasets, variable selection can aid in the discovery of potential biomarkers for a particular disorder.

In this study, the mechanism of the three abovementioned methods described in the literature has been investigated and compared. Their performance when applied to three different metabolomics datasets for multivariate classification was also studied. Although the most favorable method varied for each dataset, model prediction performance was found to improve when variable selection was carried by means of any of the methods. However, because the parameter settings for the methods were set by default for this comparison, an optimization of these is recommended to obtain a more appropriate comparison.

In an attempt to optimize the variable selection stage for the creation of classification models for the three metabolomics datasets of interest, the original CARS algorithm was modified to simultaneously optimize three different parameters. Although promising results were obtained with this modification, a discrepancy was detected in terms of the validation process embedded in the algorithm.

A new variable selection method based on the separate optimization of identity and number of informative variables was developed. However, its implementation did not prove to increase model prediction performance when compared to the results obtained when using the original or modified CARS, or when using all variables in the original dataset. Some of the aspects identified as possible pathways to improve the method's performance were tested, only to be discarded. Further study regarding other untested pathways is needed for the improvement of this method.

1. INTRODUCTION

1.1. OBJECTIVES

The aim of this study was to optimize the variable selection stage prior to the construction of multivariate classification models for three different metabolomics datasets by means of partial least squares discriminant analysis (PLS-DA). To achieve this, the following objectives were defined:

1.1.1. To compare the mechanism of the competitive adaptive reweighted sampling (CARS), subwindow permutation analysis (SPA) and random forest (RF) methods for variable selection as described in the literature.

1.1.2. To perform VS using the three above mentioned methods with standard settings applied to three different metabolomics datasets profiled by gas chromatography (GC)-mass spectrometry (MS).

1.1.3. To examine existing MATLAB scripts for the above methods in detail to verify their mechanism and their accordance to the procedures described in the literature.

1.1.4. To select one of the above methods to modify for improvement by identifying from its algorithm parameters to be optimized.

1.1.5. To establish a strategy and algorithm to simultaneously optimize the identified parameters.

1.1.6. To create a manageable script in MATLAB for the performance of the modified method that allows the user to easily vary the definition of certain input parameters.

1.1.7. To compare the performance of the modified method and the original method applied to the above mentioned three datasets.

1.2. THEORY AND BACKGROUND

Calibration is a widely applied tool in analytical chemistry, without which routine activities as essential as determining the concentration of an analyte in a sample could not be carried out. It is basically a comparison between two sets of numbers, and can be divided into two main types: absolute and relative calibration [1]. Absolute calibration refers to the comparison of a measurement to an accepted standard, for example, the measurement a balance indicates when weighing a standardized mass that is traceable to the international prototype of the kilogram [2]. However, usually in practical quantitative analysis absolute calibration is not really relevant. For example, when using a standard solution of known concentration to calibrate an ultraviolet (UV) visible spectrometer, the purpose is not to obtain a commonly accepted absorbance at a certain wavelength, but to be able to predict the concentration of future samples. This is known as relative calibration, and is how calibration is usually generalized. Martens and Næs define calibration as "the use of empirical data and prior knowledge for determining how to predict unknown quantitative information Y from available measurements X, via some mathematical transfer function" [1].

1.2.1. Univariate versus multivariate calibration

Every calibration model consists of one or more dependent variables, or responses (*y*), one or more independent variables (*x*) and their coefficients (*β*), and an error term (ϵ) which indicates the unexplained variance in the dependent variable [3]. The simplest type of model is the univariate model, in which there is only one independent variable.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

Equation 1.

Equation 1 represents the linear relationship between the *i*th response (*y*) and its corresponding dependent variable (*x*). The parameter β_0 is the intercept, or the value of *y* when *x* is zero [3, 4]; β_1 is the slope and represents the change in *y* for every increase of 1 in x, [3, 4] and ϵ is the error term. The values of β_0 and β_1 provide the best fitting line for a given calibration data set [3].

Although univariate models can be used to solve some analytical problems, they cannot be applied when there is more than one factor or independent variable affecting the response or dependent variable. Multivariate calibration can be applied to solve complex sample analysis problems where univariate analysis comes short. A typical multivariate model can be represented as shown in **Equation 2**.

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_p x_{pi};\ p = number\ of\ variables$$

Equation 2.

The parameters β_0-β_p can be estimated using multiple regression analysis [3, 5, 6]. Multivariate regression is equivalent to univariate analysis in the sense that, when having p number of variables, the values of the parameters or regression coefficients $\beta_0, \beta_1, \beta_2 \ldots \beta_p$ return the best hyperplane from relating the dependent variable (y) and the set of independent variables (X) [3].

1.2.2. Regression versus classification

One of the multivariate statistical techniques that study relationships between numerical variables is regression analysis. Regression was first used as a statistical term by Sir Francis Galton in 1877 [7], who carried out a study about the height of children born from tall parents. During this study, he named the process of predicting one variable from another "regression", because he observed that the children's height tended to move back or "regress" toward the mean height of the population. The term is defined by Gosling as "a statistical technique used to develop a mathematical equation that relates the known variable(s) to the unknown variable" [7]. So, regression analysis is a tool that aids in the estimation of the relationship between the independent and dependent variable(s) through an equation. When there is more than one independent variable involved in the prediction of the dependent one, as in multivariate calibration, the process is termed multiple regression.

Classification or discriminant analysis is a statistical technique that establishes differences between classes of objects by examining sets of multiple variables corresponding to each object [8, 9]. By identifying these differences, the technique allows the assignment of any object to the class it most closely resembles [9].

Because regression aims to predict a response, while in classification, the purpose is to identify the class which a certain object belongs to, the main difference between the two is the type of values that are obtained for the response variable *y*: continuous for the former and class labels for the latter [10, 11].

1.2.3. Partial least squares regression

Partial least squares (PLS) is a linear regression technique that is probably the most commonly used method for multivariate calibration [12]. The linear model that results from PLS (**Equation 3**) consists of a response variable matrix (Y), a descriptive or predictor variable matrix (X), a regression coefficient matrix (β) and a noise or error term (ϵ) of sizes n by m, n by p, p by m and n by m, respectively. For the matrix X, the number of rows (n) is the number of objects and the number of columns (p) is the number of variables [13]. The columns in the response matrix Y represent the number of responses (m) corresponding to the n objects.

$$Y = X\beta + \in$$

Equation 3.

To establish a PLS model, a weight matrix W of size p by c, where c is the number of latent variables (LV), is calculated for X. LVs are orthogonal or non-correlated factors that provide the best predictions and are derived from the original predictive variables [14]. The factor score matrix T (**Equation 4**) of size n by c is then calculated. The columns of W

are weight vectors for each of the *p* columns in X, which are computed in such a way that the covariance between responses and the corresponding scores is maximized [13]. The regression of Y on T is then performed to produce Q, the loadings for Y (**Equation 5**). Finally, the regression coefficients *β* are calculated using Q (**Equation 6**), completing the prediction model (**Equation 3**).

$$T = XW$$

Equation 4.

$$Y = TQ{+}\in$$

Equation 5.

$$\beta = WQ$$

Equation 6.

The loadings for X (*P*) of size *p* by *c* must also be calculated to obtain the unexplained fragment (*F*) of the scores (*T*) in **Equation 7**.

$$X = TP + F$$

Equation 7.

Of the available algorithms that can be used to compute PLS, nonlinear iterative partial least squares (NIPALS) is the standard. From the many variants, the following, detailed by Hill and Lewicki [13], who consider it to be one of the most efficient, assumes that both *X* and *Y* have been transformed to have means of zero. The superscript *T* for a given matrix *X* (X^T) represents the transpose [15] of *X*. For a given vector *y*, its norm or length [16] is denoted with the symbol || *y* ||.

For each LV (*h*), $h = 1, \dots, c$ where the initial values for A, M and C are $A_0 = X^T Y$, $M_0 = X^T X$, $C_0 = I$, and *c* is given,

i. Calculate q_h, the dominant eigenvector of $A_h^T \times A_h$

ii. $w_h = C_h \times A_h \times q_h, w_h/\|w_h\|$, and store w_h into W as a column

iii. $p_h = M_h \times w_h, c_h = w_h^T \times M_h \times w_h, p_h = p_h/c_h$, and store p_h into P as a column

iv. $q_h = A_h^T \times w_h/c_h$, and store q_h into Q as a column

v. $A_{h+1} = A_h - c_h \times p_h \times q_h^T$ and $M_{h+1} = M_h - c_h \times p_h \times p_h^T$

vi. $C_{h+1} = C_h - w_h \times p_h^T$

The unexplained fragment *F* of the scores (**Equation 7**) obtained with the current LV is used as the next *X* to estimate the following LV through steps v and vi of the algorithm. At the end of the iteration or repetition for the last LV, the scores matrix *T* and the regression coefficients β of *Y* on *X* can be calculated according to **Equation 4** and **Equation 6**, respectively [10].

Once a model is built, its prediction performance can be assessed by using an independent test set, that is, a group of objects and their true response values that were not used to build the model. The predicted response for each element in the test set is then compared to the true response to obtain an error. A scheme of the processes of model building using PLS and model validation is shown in **Figure 1**.

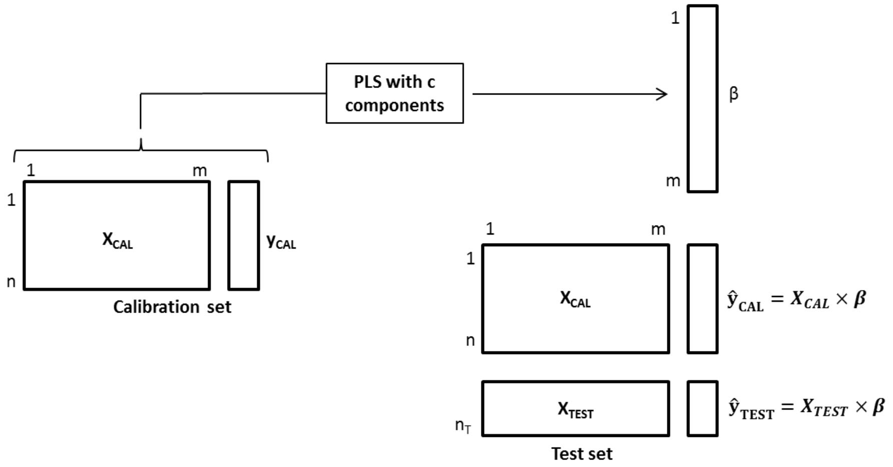


Figure 1. Scheme of PLS predictive model building and validation. The regression coefficient vector β is used to calculate the predicted response of both the calibration set, to evaluate the fitness of the model, and an independent test set, to evaluate the predictive power. (*Taken from* **[12]**)

PLS-DA is a variant of PLS used for classification problems, when the response *y* is categorical [17]. It carries out linear discriminant analysis (LDA) on the score matrix *T* after it has been extracted from the *X* matrix by PLS. LDA can be implemented through Fisher's algorithm, which maximizes the variability between classes in relation to the one within the classes [18].

1.2.4. Cross validation

Defining the best number of LVs, or PLS components, for model building is imperative to avoid underfitting and overfitting. A commonly used technique to accomplish this is cross validation (CV) [12]. It consists of partitioning the dataset into a calibration or training set, from which a model will be built, and a validation or test set, which will be used to assess the model's performance. The partitioning is carried out many times to obtain many different training and test sets, and finally the validation results from all the partitions are averaged.

K-fold cross validation is a type of CV in which the data is divided into K non overlapping groups, or folds, of almost the same size [19]. One of the folds is removed and the rest is used to build a PLS model. The fold, which was removed, is then fitted to the model and the response variable predicted by the model is compared to its true response variable to obtain an error. This procedure is repeated as many times as there are groups, until all of them have been used as a test set only once. The prediction errors for all objects are then combined to obtain an error. This error can be calculated for each number of LVs used to build the final PLS model. The number of LVs used to build the model that achieves the lowest error is the optimal one. A new model is calculated from the entire dataset using the optimal number of components revealed at the end of the CV procedure. An example of 4-fold CV is represented in **Figure 2**.

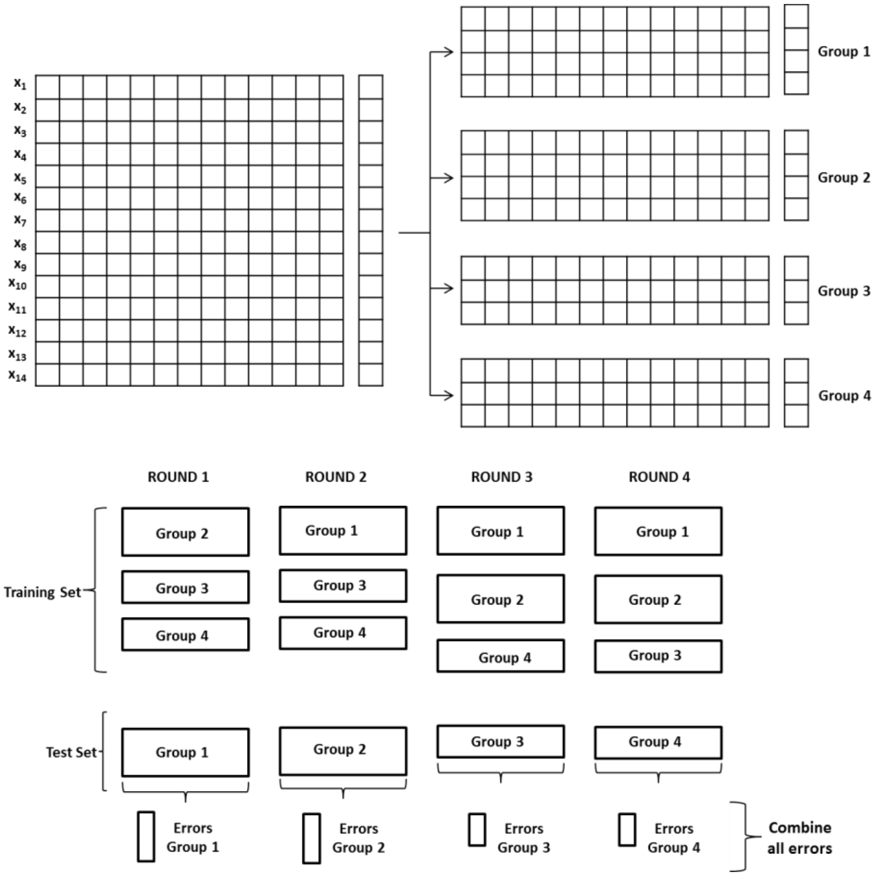


Figure 2. Representation of a 4-fold CV example using a predictive variable matrix *X* of size 14 by 13 and only one response *y* for each of the 14 objects. The objects are partitioned into four different groups which alternate the role of test set in each different CV round. The errors of each group are combined to obtain one final error.

1.2.5. Classification model assessment

Regression and classification differ mainly in the type of values the response variable contains. Since these are continuous for the former, a root mean square error (RMSE) proves to be appropriate to assess model prediction in this case. However, this parameter cannot be applied for classification models as the values recorded in the response are categorical. Other parameters exist as alternative assessment parameters for classification. The ones used in this study are described below. Because this study involved binary classification problems, the following descriptions can be applied to this particular situation, disregarding cases in which more than two classes are to be predicted.

1.2.5.1. Misclassification error

The misclassification error is the total number of incorrectly classified objects, comprising false negatives (FN) and false positives (FP), divided by the total number of classified objects (n) (**Equation 8**) [20].

$$Error = \frac{\sum(\check{y}_i \neq y_i)}{n} = \frac{FN + FP}{TP + FN + FP + TN}, i = 1, 2, \dots, n$$

Equation 8.

1.2.5.2. Accuracy

Subtracting the misclassification error from one generates the model prediction accuracy [20, 21]. This parameter is a measure of how well a model can assign the correct class to an object from unknown or test data [22]. Being the opposite of the misclassification error, it can also be calculated by dividing the number of correctly classified objects, consisting of true negatives (TN) and true positives (TP), by the total number of classified objects (n) (**Equation 9**).

$$Accuracy = \frac{\sum(\check{y}_i = y_i)}{n} = \frac{TN + TP}{TP + FN + FP + TN}, i = 1, 2, \dots, n$$

Equation 9.

1.2.5.3. Sensitivity

Sensitivity is a measure of a model’s ability to correctly classify objects with positive value or of class 1 [23, 24]. Let us consider that, for a given dataset in which the objects represent individuals, class 1 and -1 indicate the presence or absence, respectfully, of a particular disease or condition. A highly sensitive model would produce few false negatives, meaning that most of objects of class 1 would correctly be associated with the condition at issue [24]. This parameter is calculated by dividing the number of correctly classified objects of class 1, or TPs, by the total number of objects of class 1 that were classified, or TPs and FNs (**Equation 10**).

$$Sensitivity = \frac{TP}{TP + FN}$$

Equation 10.

1.2.5.4. Specificity

Specificity is a measure of a model’s ability to classify objects of negative value or of class -1 [23, 24]. For the example described for sensitivity (**Section 1.2.5.3**), a highly specific model would be one that produces few false positive results. This means that most of the objects of class -1 would correctly be associated with the absence of disease [24]. This parameter is calculated by dividing the number of correctly classified objects of class -1, or TNs, by the total number of objects of class -1 that were classified, or TNs and FPs (**Equation 11**).

$$Specificity = \frac{TN}{TN + FP}$$

Equation 11.

1.2.5.5. Area under curve

The area under the receiver operating characteristic (ROC) curve, or simply area under curve (AUC), is a measure of a model’s ability to discriminate objects of different classes [25]. It plots the rates of correctly classified objects of class 1, or TPs (sensitivity), against the rates of incorrectly classified objects of class -1, or FPs (1-specificity) for an entire range of cut points (**Figure 3**) [25, 26]. AUC values range from 0.5 to 1.0, the latter indicating perfect classification ability (100% sensitivity and specificity) and the former a random choice of class (50% sensitivity and specificity) [27, 28].

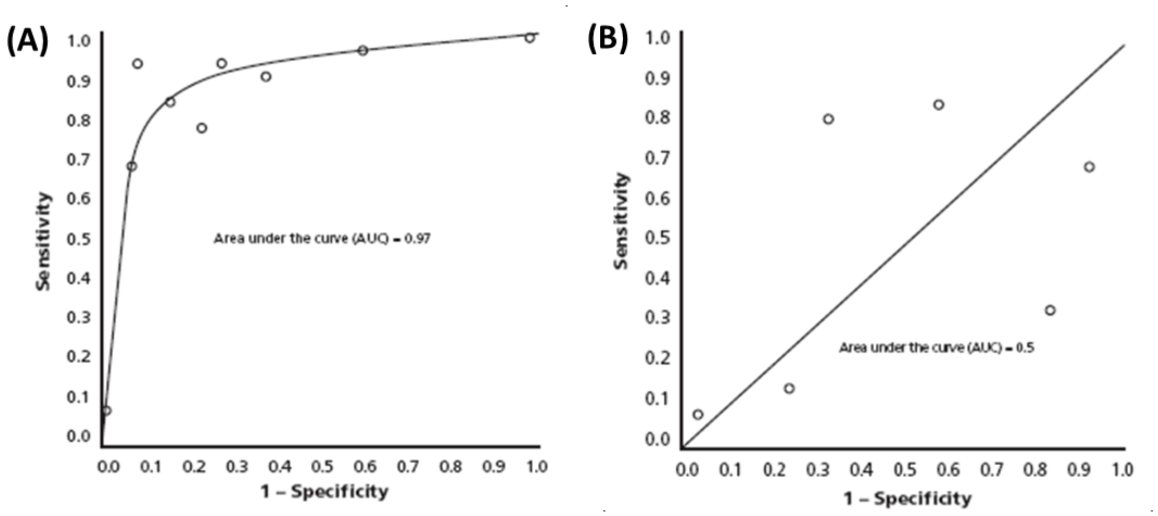


Figure 3. A) ROC curve with AUC close to 1, indicating high discriminatory power and **B)** ROC curve with AUC of 0.5, a diagonal line, indicating no discriminatory power. (*Taken from* **[28]**)

1.2.5.6. Mathew's correlation coefficient

Mathew's correlation coefficient (MCC) is a measure of the correlation between the predicted value and the true response [29]. MCC values range from 1 to -1, indicating perfect positive or negative correlation, respectively. A value of zero indicates orthogonality, or total absence of correlation. MCC is calculated according to **Equation 12**.

$$MCC = \frac{TP \times TN - FN \times FP}{\sqrt{(TP + FN) \times (TP + FP) \times (TN + FP) \times (TN + FN)}}$$

Equation 12.

1.2.6. Variable selection

Variable selection (VS) is the process of reducing the original number of explanatory variables in the dependent variable *X* matrix by discriminating informative variables from the ones that are not related to the response *y* [30]. Some of the reasons why VS is an important step in the calibration process include the following:

a) According to the parsimony principle, also known as Ockham's razor, for a group of competing models that fit a given dataset, the simplest should be considered the best one [31, 32]. In other words, the data should be explained in the simplest way; thus, unnecessary or uninformative variables that do not have any effect on a prediction should be excluded [33].

b) Some variables are not only uninformative, but they are also interfering. That is, they add more noise than relevant information to the model and their inclusion actually makes an analytical prediction worse [33, 34].

c) Cost in terms of time and money can be reduced if irrelevant predictors are not measured [33].

d) The selecting of informative variables can be applied for different purposes such as identifying the most influential factors affecting the quality of a product or the characteristic features of a certain class. In the first instance, a few vital factors are much easier to measure and control in a process industry than all possible process variables [35]. The second case takes place, for example, when classifying metabolomics data.

In a metabolomics dataset, each variable represents a metabolite, and the objects are individuals. In the case of binary classification, there are only two possible responses: 1, indicating the presence of a particular metabolic state, such as a disorder, or -1, indicating its absence. So by selecting informative variables from these types of datasets, a selection of informative metabolites, that can be considered potential biomarkers, is actually taking place [36, 37, 38]. In this way, we can learn about metabolic perturbations that exist in individuals with a disease of interest, and ultimately, determine the pathophysiological mechanisms of the disease, allowing the discovery of new pathways for diagnosis and treatment.

The purpose of VS is to find a subset of variables that produce the smallest errors when used to carry out quantitative analysis or to classify objects into different categories [34]. Many methods have been developed to either identify variables that provide relevant information, eliminate interfering and uninformative variables or both. Three of these methods are described below.

1.2.6.1. Competitive adaptive reweighted sampling (CARS)

CARS is a method originally developed to select informative wavelengths from continuous spectral data, specifically applied for the first time to near infrared spectroscopy (NIR) [39]. It is based on Darwin's evolution principle "survival of the fittest" and is combined with PLS to assess variable importance. It basically consists of a number of iterations involving 1) Monte Carlo sampling (MCS) in object space, 2) VS by means of weights and the exponentially decreasing function (EDF), 3) VS by reweighted sampling of variables selected in the previous step and 4) model building with each subset of selected variables and CV to calculate prediction error.

1.2.6.1.1. Monte Carlo sampling

The first step in the CARS algorithm involves obtaining a sample of objects using the Monte Carlo approach to build a multivariate model. The name "Monte Carlo" was properly attributed after the popular gambling destination, as this sampling method is based on the laws of chance or probability [40]. A sample of objects is selected randomly without replacement, which means that an object that was chosen does not return to the sampling lot and thus, can only be sampled once. The ratio of kept objects for training or model building is usually around 0.80-0.90 [39]. The remaining unsampled objects will not be used during that particular iteration until the fourth step where all objects will be included for CV to obtain a prediction error. This sampling is done to "select variables which are of high adaptability regardless of the variation of training samples" [39]. In other words, the aim is that the method is as robust to the change in samples used for model building as possible. Some of the parameters resulting from the PLS model built from the training sample obtained in this step can be used to calculate an importance score for each variable.

1.2.6.1.2. Two-stage variable selection using EDF

A weight (w) is calculated as an importance score based on the regression coefficient (β) corresponding to each variable (**Equation 13**).

$$w_i = \frac{|\beta_j|}{\sum_{i=1}^{p}|\beta_j|}, j = 1, 2, \dots, p$$

Equation 13.

where p is the number of variables in the original dataset. An alternative importance score to use here is the selectivity ratio (SR), which is the relation between the explained (v_{exp}) and residual (v_{res}) variance for each variable [37, 41] (**Equation 14**).

$$SR_i = v_{exp,i}/v_{res,i}, i = 1, 2, \dots, p$$

Equation 14.

The explained and residual variance should be calculated based on target projection [41], which "reveals the y-relevant variation in the x-variables captured by a multicomponent PLS model on a single latent variable" [41].

The variables with the highest weights or SRs will be kept. The ratio of variables to be kept for each MCS run will vary, and is calculated based on (**Equation 15**):

$$r_i = ae^{-ki}, i = 1, 2, \dots, N$$

Equation 15.

where N is the number of the MCS run or iteration that is taking place and a (**Equation 16**) and k (**Equation 17**) are constants that follow two conditions: I) For the first MCS run, all variables will be selected in this step, II) while in the last run, only two variables will be kept. In this way, the ratio of variables to be kept (r) will decrease for each run (i), exhibiting a behavior that can be graphically observed in **Figure 4**. The decrease is exponential and occurs in two stages: I) rapidly, where the number of chosen variables drops significantly between iterations and II) subtly, where the number of variables kept varies very little in comparison to the previous iteration [39].

$$a = \left(\frac{p}{2}\right)^{1/(N-1)}$$

Equation 16.

$$k = \frac{ln(p/2)}{N-1}$$

Equation 17.

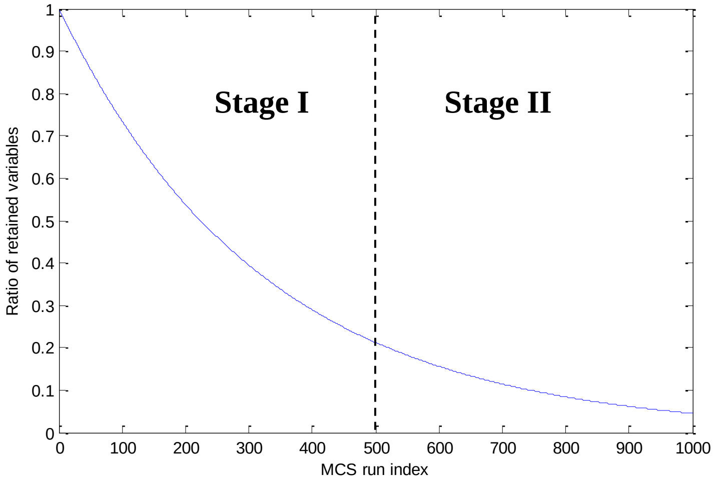


Figure 4. Exponential decrease of the ratio of retained variables in Step 2 of the CARS algorithm for each MCS run. Two stages can be distinguished: I) a rapid decrease in the number of retained variables and II) a more refined selection where the number of kept variables varies very little in comparison to the previous sampling run. (*Taken from* **[39]**)

1.2.6.1.3. Adaptive reweighted sampling

The third step of the CARS algorithm consists of a second variable selection process and is where the evolution principle is applied. Based on their weights or SRs, the higher the importance score, the more fit or competitive a variable is to survive. Variables with lower importance scores are weaker and will be wiped out by the more dominant ones. This process is carried out by the use of adaptive reweighted sampling (ARS), where the higher the importance score assigned to each variable in Step 2, the higher the probability for its corresponding variable to be sampled. In this way, by means of sampling with replacement, in which a variable is selected and then returned or “replaced” to the population which is being sampled [42], the variables with higher scores will be sampled multiple times, while the ones with the lower scores will be completely left out, and thus, eliminated. The variables that were sampled more than once have taken the place of those that were discarded; thus, the resulting vector is the same size as the one containing the variables that were submitted to ARS. Finally, the remaining variables are included only once in the final selected variable subset, regardless of how many times they were resampled, resulting in a variable vector of reduced size.

1.2.6.1.4. CV to evaluate the variable subset

Finally, a PLS model is built considering only the variable subset selected in steps 2 and 3, and an error is obtained using CV. As mentioned before, CV evaluates model prediction by dividing the data into multiple training sets and independent test sets [43]. The objects included in the training and test sets are alternated in such a way that each object is in the test set once and once only [43]. The error obtained will either be a root mean square error of cross validation (RMSECV) or a classification assessment parameter, in the case of regression or classification, respectively.

The four steps above will be repeated for each MCS run or iteration, obtaining an error for each one. The run whose error is the lowest will be considered the optimal one, and the variable subset obtained in that run will be selected as the best combination of variables for predictive purposes. **Figure 5** summarizes the CARS algorithm in a flow chart.

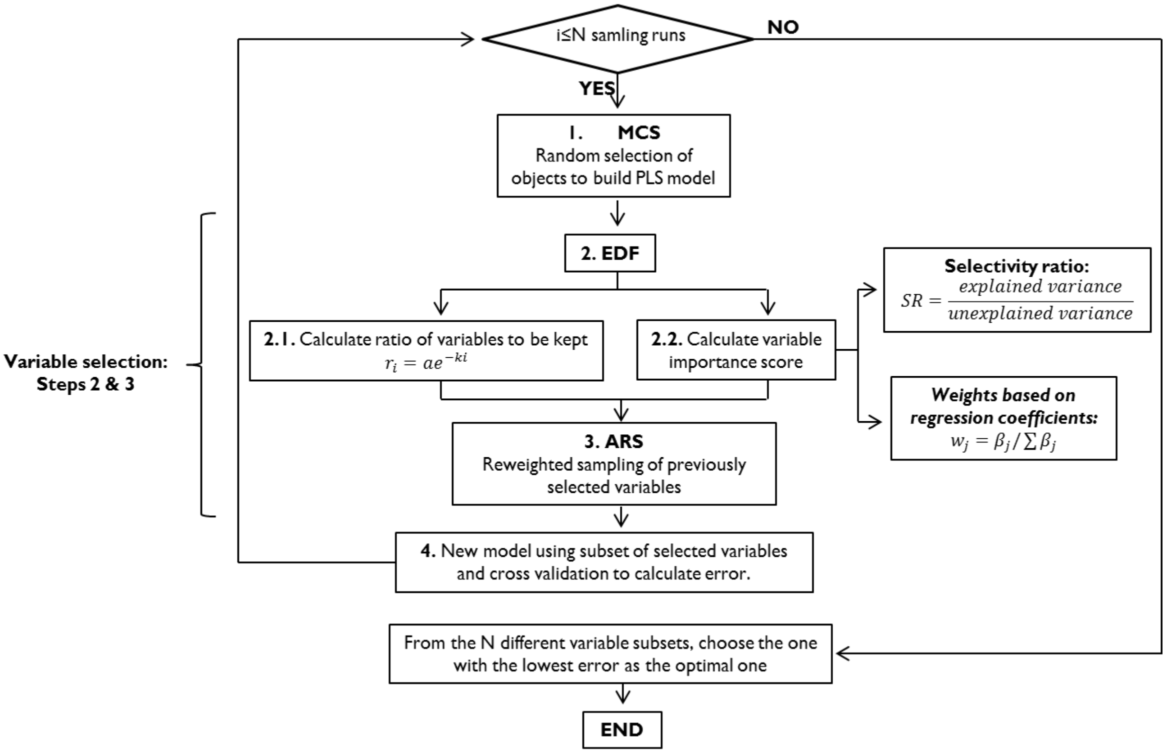

i≤N samling runs
NO
YES
1. MCS
Random selection of objects to build PLS model
2. EDF
2.1. Calculate ratio of variables to be kept $r_i = ae^{-ki}$
2.2. Calculate variable importance score
Selectivity ratio: $SR = \frac{explained\ variance}{unexplained\ variance}$
Weights based on regression coefficients: $w_j = \beta_j / \sum \beta_j$
Variable selection: Steps 2 & 3
3. ARS
Reweighted sampling of previously selected variables
4. New model using subset of selected variables and cross validation to calculate error.
From the N different variable subsets, choose the one with the lowest error as the optimal one
END

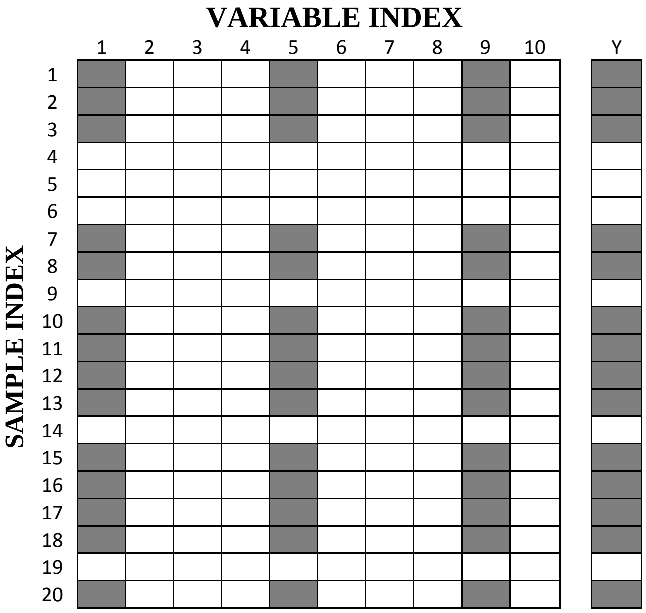
Figure 5. Flow chart of CARS algorithm.

1.2.6.2. Subwindow permutation analysis (SPA)

SPA was developed to be applied to metabolomics datasets for the selection of metabolites that could be informative of the prediction of a clinical outcome, thus considered biomarkers [36]. It is based on the principles of model population analysis (MPA) and like CARS, uses PLS to build a series of submodels. MPA's main principle is to statistically analyze an output of interest of a population of sub-models [44]. In the case of VS, one could analyze the distribution of prediction errors [36]. In summary, the steps to execute SPA are: 1) MCS in object and variable space, 2) PLS submodel building for each sampling run and 3) statistical analysis of the distribution of prediction errors.

1.2.6.2.1. MCS of objects and variables

Unlike CARS, MCS is performed on variables as well as objects for each run (**Figure 6**), resulting in a data subwindow which gives, to some extent, information about the synergetic effect between the variables included in it [36]. This effect refers to the higher performance of the combination of variables when compared to that of the sum of the individual contributions of each one [36].



	1	2	3	4	5	6	7	8	9	10	Y
1	▨				▨				▨		▨
2	▨				▨				▨		▨
3	▨				▨				▨		▨
4											
5											
6											
7	▨				▨				▨		▨
8	▨				▨				▨		▨
9											
10	▨				▨				▨		▨
11	▨				▨				▨		▨
12	▨				▨				▨		▨
13	▨				▨				▨		▨
14											
15	▨				▨				▨		▨
16	▨				▨				▨		▨
17	▨				▨				▨		▨
18	▨				▨				▨		▨
19											
20	▨				▨				▨		▨

Figure 6. Representation of MCS in both object and variable space for a dataset of size 20 X 10, if a ratio of 0.75 objects and a number of 3 variables are retained for each subwindow. The resulting training set would be of size 15 X 3, while the test set would comprise the remaining 5 objects and the same 3 variables. (*Taken from* **[36]***)*

1.2.6.2.2. PLS submodel building

When solving classification problems, PLS-DA can be used to build models with the training sets of each subwindow. CV is employed to choose the optimal number of PLS components.

1.2.6.2.3. Statistical analysis of an output of interest

As mentioned before, for the purpose of VS, a suitable output to analyze is the distribution of the prediction error. For *N* MCS runs, the same number of subwindows will be obtained. However, not all of the *N* subwindows will contain the *j*th variable; so, in order to assess its importance, only the *J* subwindows that contain it should be analyzed.

The *J* submodels obtained from the previous step will be validated using their corresponding *J* test sets, for which two errors will be calculated: a normal prediction error (NPE) and a permuted prediction error (PPE). The difference is that the second one is calculated using the test set after randomly permuting, or giving a random order to the values for the *j*th variable. In this way, the variable of interest is being noised up and so, if it is considered predictive, the prediction error would be expected to increase [45] because the accuracy of the output depends on the specific value of this variable. A *DMEAN* is obtained by subtracting the mean of NPEs from that of PPEs (**Equation 18**). The procedure is illustrated in **Figure 7**.

$$DMEAN_j = MEAN_{PPE,j} - MEAN_{NPE,j}$$

Equation 18.

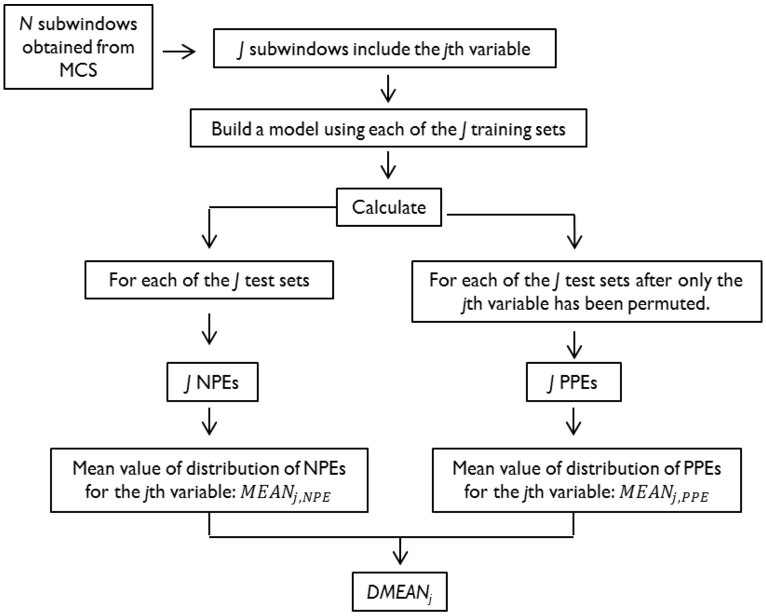

N subwindows obtained from MCS
J subwindows include the *j*th variable
Build a model using each of the *J* training sets
Calculate
For each of the *J* test sets
For each of the *J* test sets after only the *j*th variable has been permuted.
J NPEs
J PPEs
Mean value of distribution of NPEs for the *j*th variable: $MEAN_{j,NPE}$
Mean value of distribution of PPEs for the *j*th variable: $MEAN_{j,PPE}$
$DMEAN_j$

Figure 7. Obtainment of NPEs and PPEs for the calculation of variable importance assessment parameter *DMEAN*.

Each NPE and PPE is dependent of the combination of variables belonging to their corresponding subwindow, hence providing information regarding the interactions between those variables. The whole *J* subwindows encompass the effects that all of the *p*-1 variables have on the variable *j* [36].

The variable selection process consists in I) eliminating all variables with a *DMEAN* lower than zero, II) carrying out Mann-Whitney U-Test to evaluate the significance in the difference between the distributions of both prediction errors, resulting in a ρ-value for each variable, III) variable ranking according to their ρ-value and IV) selecting the variables that comply with a predefined threshold.

The Mann-Whitney U-Test can be considered "the non-parametric equivalent of Student's t-Test" [46] whose use does not require data to be normally distributed. This statistical test checks whether the data of a particular group tends to be larger than that belonging to another group [47].

The ρ-value is inversely proportional to variable importance, and thus, for practical reasons, it can be converted to a conditional synergetic score (COSS) through **Equation 19**. In this way, the score assigned to each variable is directly proportional to its importance, and therefore the acceptance criteria will change, for example, from $\rho \leq 0.01$ to $COSS \geq 2$.

$$COSS = -Log_{10}(\rho)$$

Equation 19.

1.2.6.3. Random Forest (RF)

RF is an ensemble method, which combines multiple decision trees to obtain one final prediction [48]. A decision tree is a hierarchical structure consisting of nodes and directed edges which is built by crafting a series of key questions about the attributes of certain data of interest [49]. Three types of nodes make up a decision tree: a root node, which has outgoing edges but no incoming ones; internal nodes, which have both incoming and outgoing edges; and terminal or leaf nodes, which only have incoming edges, and denote a label or prediction. The root and internal nodes, being non-terminal, contain attribute test conditions to separate objects that have different characteristics [49].

To illustrate this, Tan, Steinbach and Kumar [49] present a decision tree for the classification of mammals or non-mammals (**Figure 8**). When an object is run down the tree, the answer to the question "body temperature" will lead to either a follow-up question, or a classification label. In this way, as many follow-up questions will succeed until a final conclusion about the object is made.

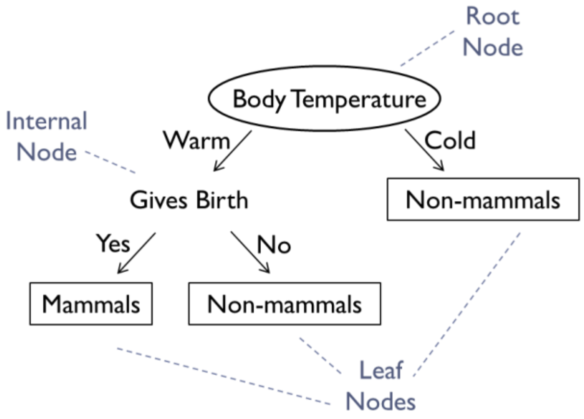


Figure 8. Decision tree classifier for the mammal classification problem. Three types of nodes can be distinguished, where the leaf nodes designate the final outcome or prediction. *(Taken from* **[49]***)*

Although decision trees have the advantages of being able to handle high-dimensional data, ignore unimportant variables and interpret models suitably, their performance is not always satisfactory. The simple decision tree illustrated above (**Figure 8**) fails to correctly classify the monotremes, which are a special group of mammals that lay eggs instead of giving birth [50], such as the platypus. In general, decision trees usually have low prediction accuracies [51], only slightly better than a random choice of class [48]. One of the attempts to improve this has been the use of ensemble methods or combining forecasts, which combine the results of multiple individual models to reach a single prediction [52]. Experimental evidence has shown that ensemble methods are often much more accurate than any single hypothesis [48, 53].

For a given data subset used to build a decision tree, the conditions that separate an object in each of the non-terminal nodes according to its known response (y) will be governed by the "attributes" of each object in the training set, these being represented as a p-dimensional vector of variables associated with each object [48]. Thus, RF can be defined mathematically as an ensemble of B trees $\{T_1(X), \dots, T_B(X)\}$, where $X = \{x_1, \dots, x_p\}$ is a variable vector corresponding to an object whose outcome will be predicted [51]. A total of B predictions will be obtained for each object: $\{\breve{y}_1 = T_1(X), \dots, \breve{y}_B = T_B(X)\}$, one from every tree, all of which will then be combined to produce one final prediction [51]. RF can be used to solve both regression and classification problems, being the final outcome the average of all individual tree predictions for the former or the class obtained by the majority of trees for the latter [51, 52].

1.2.6.3.1. Training algorithm

The following training procedure has been taken from Svetnik et al. [51]. Given data for a set of n objects for training, D = {(X_1, Y_1), …, (X_n, Y_n)}, where X_i, $_{\mathrm{i}}$=1, …, n, is a vector of variables and Y_i is the corresponding prediction for the ith object, the algorithm is as follows:

i. From the training data of n objects, a bootstrap sample is drawn, which is a random sample with replacement of size n. This means that the new sample will have the same number of objects as the original one; it could include some of the original objects more than once, while others will be left out altogether [54]. The selection of an object for the new sample is independent from the previous selection.

ii. For each bootstrap sample, a tree is constructed by choosing the best split at each node, among a randomly selected subset of m_{try} variables, instead of all of them. Here, m_{try} is a tuning parameter that can be chosen as a function of the total number of variables (p). The performance of RF seems to change very little over a wide range of values of m_{try}, except near the extremes: 1 or p. The tree is grown until no further splits are possible, reaching its maximum size, and it is not pruned back.

iii. The above steps are repeated until a sufficient number of trees are grown.

The tree growing algorithm used is CART (Classification and Regression Trees), that builds classification trees according to a splitting rule; the rule that performs the splitting of the training sample into smaller parts [55].

1.2.6.3.2. RF for variable selection

The construction of each decision tree depends on random vectors sampled independently from each other, but with the same distribution for all trees in the forest [48]. This refers to bootstrap sampling [54], and the random vectors sampled are the *p*-sized vectors of variables corresponding to each object in the training set. Thus, the selection of an object for training is independent of the previous one. This means that some objects will be sampled more than once, while others will not be sampled at all [54]. The former will constitute the bootstrap sample, which is the same size of the original dataset [54], with the difference that it contains repeated objects, and will be used for trainng or tree construction. The rest of the objects constitute the out-of-bag (OOB) sample, which is the test set, and will be approxmately one third of the size of the original dataset [51]. According to empirical evidence provided by Breiman [48], having this large test set is almost as accurate as it being the same size as the training set.

Variable importance in RF is carried out by means of the OOB estimates. Due to its complexity, the mechanism of how a group of trees provides a prediction is difficult to interpret. Because it does not produce an explicit model, the relationship between descriptors or variables and the outcome is said to be hidden inside a “black box” whose insides are practically unknowable [51, 56, 57]. To solve this problem, internal OOB estimates can be used to carry out certain measures of variable importance that are available to identify the informative variables [56].

As an approach to measure the importance of the *j*th variable, two measurements of prediction performance are computed for the test set or OOB sample, in a similar way as described in **Section 1.2.6.2.3** as NPEs and PPEs for SPA. Each OOB object is run down its respective tree to obtain a prediction. In addition, a second run is carried out, this time permuting the *j*th variable. At the end of the procedure, each object will have two predictions for each time it constituted the OOB sample for a given tree: a normal prediction and one carried out with the *j*th variable permuted or noised up.

The performance of each prediction must then be measured. As stated by Svetnik et al. [51], in the case of classification problems, the change in prediction accuracy is usually a less sensitive measure than the change in the margin. For multiclass classification problems, margin can be defined as the difference between the proportion of correct class predictions and the maximum proportion of incorrect ones [58]. Svetnik et al. [51] illustrates this by supposing for a given three-class problem that an object of class 1 receives 60, 30 and 10 percent votes of class 1, 2 and 3 respectively. Thus, the margin is equal to $0.6 - max(0.3, 0.1) = 0.3$. In the case of binary classification, the margin is simply the difference between the proportion of correct class predictions and the proportion of incorrect ones. A positive margin indicates a correct class prediction, while a negative one means the opposite [51].

From the margins calculated for normal predictions and predictions with permuted variable j, the means for both, M and M_j, respectively, are calculated. The variable importance is simply the difference between these means (**Equation 20**), where if it is positive, zero or negative, the variable is considered informative, non-informative or interfering, respectively. For regression problems, the RMSE is calculated instead of margins [51].

$$Importance_j = M - M_j$$

Equation 20.

1.2.6.4. Comparison of CARS, SPA and RF

From the literature search carried out, a series of aspects have been identified in which CARS, SPA and RF can be compared.

In general, they are all based on different criteria: CARS on a variable importance score based on parameters obtained from the construction of a PLS model; SPA on the difference in empirical distribution between NPEs and PPEs; and RF on the difference in prediction performance validated on OOB estimates with normal and permuted variable values. Unlike the other methods, which were developed for classification purposes, CARS was originally meant to solve regression problems.

Regarding the selection of objects used for the training procedure, RF uses bootstrap sampling, while the others use MCS. However, during this sampling procedure, both SPA and RF also select a subgroup of variables for training in each run. The original model built in CARS in each run, on the other hand, includes all variables in the dataset.

All of the methods involve a validation stage to generate an error that is used in some way to select the optimal variable subset. SPA and RF calculate a normal error and an error when the values of a certain variable are randomly permuted from an independent test set. CARS carries out CV on the original dataset to obtain an error; thus, because most of the objects are used for training the PLS model in the first step, the test set is not independent.

Finally, the criteria for the selection of variables once the importance scores for each one is known varies between all methods. CARS automatically produces a subset that achieves the lowest error, that was selected by EDF and ARS based on the individual variable importance scores. SPA and RF assign errors to each variable individually, as opposed to doing so to a set of variables as in CARS. However, RF only focuses on the sign of the importance score, designating variables as informative, noninformative or interfering if this is positive, negative or zero, respectively. SPA on the other hand, calculates a ρ-value or COSS for each variable, and defines a threshhold or cutoff value for one of these, or both, as criteria for variable selection. **Table 1** summarizes the similarites and differences between the VS methods at issue.

Table 1. Comparison between the methods of CARS, SPA and RF for VS.

	CARS	SPA	RF
General Criteria:	Regression coefficients or SRs obtained from PLS	Difference in empirical distribution between NPEs and PPEs	OOB estimates to validate performance using normal and permuted variable values
Developed for:	Regression	Classification	Classification
Sampling for training set:	MCS	MCS	Bootstrap sampling
Training set sampling of:	Objects	Objects & variables	Objects & variables
Error(s) generated from validation stage:	CV error	NPE & PPE	NPE & PPE
Validation performed on:	All data	Independent test set	Independent test set (OOB samples)
Criteria for VS:	Subset associated with the lowest error	Variables achieving a ρ-value below or a COSS above a defined threshold	Variables with positive importance score

1.2.7. Instrumentation

Of the available analytical techniques used to generate data from chemical systems prior to multivariate analysis, gas chromatogarphy (GC) coupled with mass spectrometry (MS) is applied in many fields because of its versatility to separate, quantify and identify volatile and semi-volatile organic compounds [59, 60]. It combines the advantages of high degree of separation, or resolution, from GC, and strong identification power, or high sensitivity, from MS [61].

1.2.7.1. Chromatography

The International Union of Pure and Applied Chemistry (IUPAC) defines chromatography as “a physical method of separation in which the components to be separated are distributed between two phases, one of which is stationary (stationary phase) while the other (the mobile phase) moves in a definite direction” [62]. The signal obtained by the chromatographic system is related to the concentration of the separated compounds and is represented graphically in a chromatogram. A chromatogram is a plot of the signal in function of time or of volume of used mobile phase in the form of peaks [63]. It can provide qualitative or quantative information by determining the position of the component's peaks in the time axis or by calculating the area under the peak, respectively [63].

Elution in chromatography is the process in which a sample is dragged by the mobile phase through the stationary phase, which is contained in a chromatographic column. The more affinity a certain compound in the sample has to the composition of the stationary phase, the longer it will be retained in the column, because it will be more difficult for the mobile phase to drag it. If a sample has no affinity to the stationary phase, it will not be retained by it, and will simply move along with the flow of the mobile phase. Thus, the compounds that are less retained by the stationary phase, will flow faster and will exit the column to the detector first. The last compounds detected are the ones that have more affinity to the stationary phase. **Figure 9** shows a scheme of the previous description.

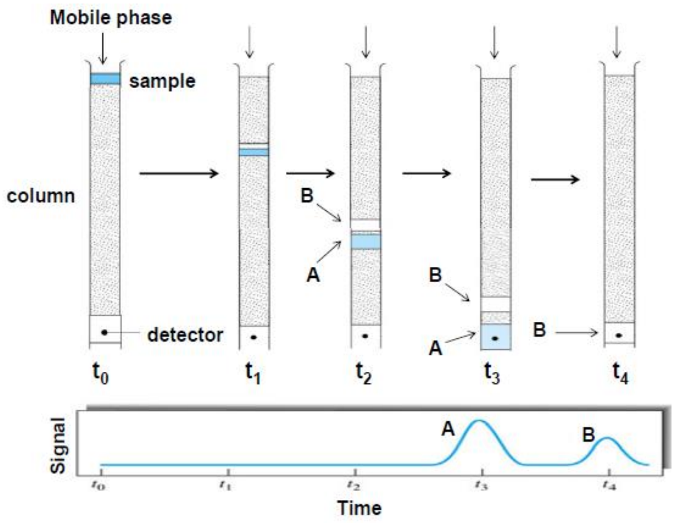


Figure 9. Scheme of the chromatographic elution process. The mobile phase flows continuously through the column, which is packed with the stationary phase. Of the compounds in the sample, *B* has more affinity to the stationary phase, and so is retained longer. Compound *A* exits the column and is detected first (t_3) and *B* follows after a certain time (t_4), thus being separated. (*Taken from* **[64]**)

Compound A (C_A) in **Figure 10** is the unretained species, and so t_0 is the dead time, which gives a measure of the mobile phase migration rate [63]. The retention time t_R for each compound is actually the sum of the dead time with the time the compound was delayed in exiting the column at regular time with the mobile phase (t_s) due to its retention in the stationary phase (**Equation 21**). Knowing the length of the column (L), the average linear velocities of both the mobile phase (**Equation 22**) and each solute or compound separated (**Equation 23**) can be calculated.

$$t_R = t_s + t_0$$

Equation 21.

$$u = \frac{L}{t_0}$$

Equation 22.

$$v = \frac{L}{t_R}$$

Equation 23.

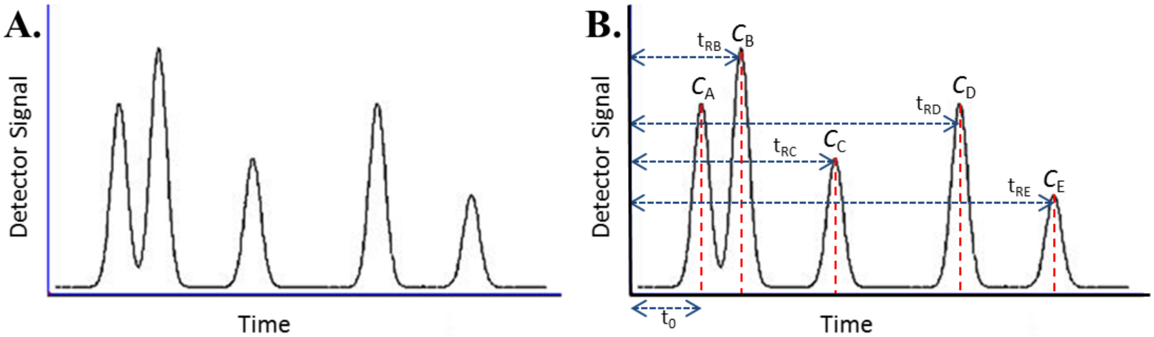


Figure 10. Example of a chromatogram (**A**) and its basic parameters (**B**). The retention time t_R of each compound *C* is the time it takes to travel through the chromatographic column which contains the stationary phase. C_A is an unretained species, and so its elution time is the dead time (t_0).

The different types of chomatography mainly vary in the physical state of the mobile phase. In GC, it is a gas, and the stationary phase is either a solid, (gas-solid chromatography) or a liquid (gas-liquid chromatography), making its interaction with compounds an adsorption, or a partition, respectively. In the latter, the compounds are dissolved in the mobile phase, not just attached to its surface like in the former [65]. The sample is vaporized when it is injected in the column and the first compounds to elute tend to be the ones with lowest boiling point or most volatile [66].

The main parameters that affect the resolution or separation ability in GC are the temperature, the flow rate of the mobile phase or carrier gas, the composition of the stationary phase and the column dimensions [67]. The chromatographic system consists in a) a carrier gas supply with pressure and flow rate regulators, b) an injection system c) a column, d) a detector and e) a read out or recorder system (**Figure 11**).

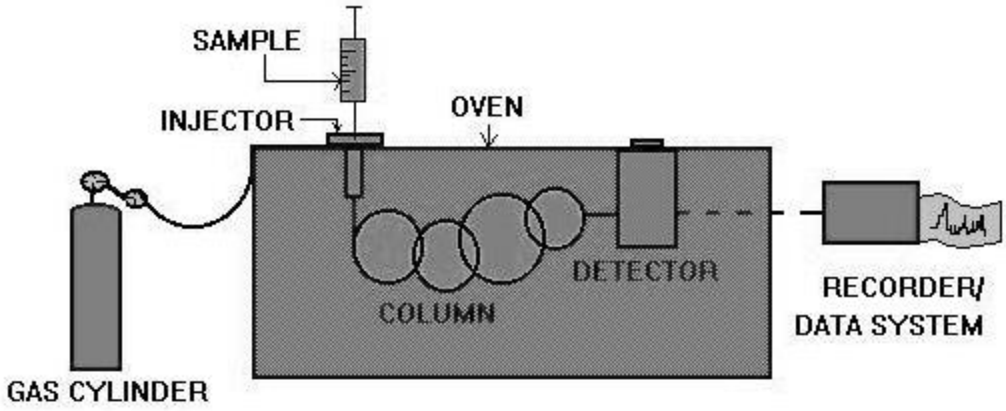


Figure 11. Scheme of a GC system and its components. (*Taken from* **[66]**)

The continuos flow of the carrier gas is carefully controlled, resulting in relatively precise retention times. The sample is injected in liquid or gas phase, and once in the injector, it is vaporized and homgenized with the carrier gas and swept by it into the column. The column is usually a tube wound in a spiral of 1 to over 100 m long [63] and is usually inside an oven with a wide range of temperature settings. After the sample has travelled through the entire column, it passes through the detector and then is dispersed in the atmosphere.

1.2.7.2. Mass spectrometry

Although GC offers advantages such as high resolution, speed and relatively low cost, it usually requires the use of spectroscopy to confirm the identities of the peaks [61, 67]. One of the reasons GC and MS are highly compatible is that both need the sample to be in the gas phase [61]. Instead of the dispersion of the sample into the atmosphere after GC analysis, coupling can be carried out by simply connecting the end of the column to the entrance of the MS system wih a transfer line (**Figure 12**). The vaporization and separation of the components in the sample performed by GC can be considered a "pretreatment" before MS analysis.

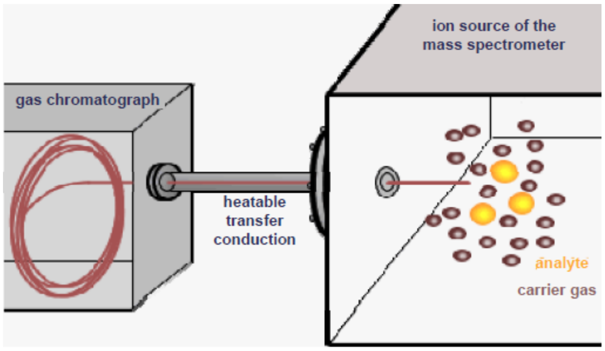


Figure 12. Coupling of a gas chromatograph and a mass spectrometer. (*Taken from* **[68]**)

Fenn et. al. [69] defined MS as "the weighing of individual molecules by tranforming them into ions in vacuo and then measuring the response of their trajectories to electric and magnetic fields or both". After the sample is introduced in the mass spectrometer, three basic operations take place: 1) ionization, 2) separation of the ions based on their mass-to-charge ratio (*m/z*) and 3) counting of the number of ions in each seperated group or measuring the ion current during ion formation [63]. The *m* in *m/z* refers to the atomic mass of the ion while the *z* is its elementary charge. Usually the ions formed have a single charge [63]; thus, the *m/z* in most cases is merely the atomic mass of the ion. The mass spectrometer's response is represented in a plot of relative intensity in function of *m/z* (**Figure 13**).

Although there are many types of mass spectrometers with varying ion sources and mass analyzers, they all consist of the same basic components (**Figure 14**). The ion source transforms the introduced sample into gaseous ions by bombarding it with electrons, photons, ions, molecules or thermal or electric energy. The ions produced, which are usually positive but can also be negative, are then accelerated into the mass analyzer. Here, the energetically charged ions are continuosly detected and sorted according to their *m/z*. Finally, the beam of ions is converted into an electric signal by a transducer to be processed and displayed in a further stage. It is important to note that all the components, with the exception of the last, are maintained in a vacuum, or at a pressure lower than the atmosphere's. The object of this is to reduce the frequency of collisions to ensure the integrity of the ions and electrons produced [63].

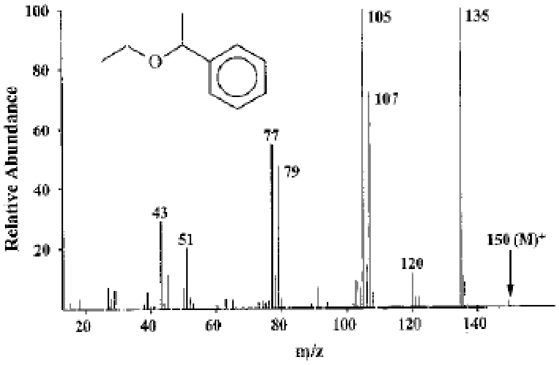


Figure 13. Mass spectra of the compound $C_{10}H_{14}O$ shown. (*Taken from* **[70]**)

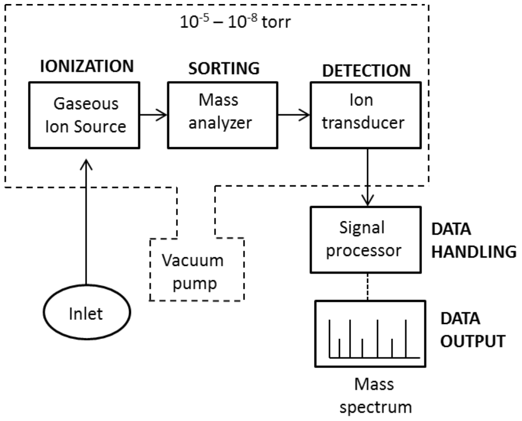


Figure 14. Basic components of a mass spectrometer. (*Taken from* **[63]**)

The resulting mass spectra can be compared with existing spectral libraries until a match is obtained, and thus the compound is identified.

Chromatographic techniques coupled with MS have been recognized as the standard for metabolomic profiling [71, 72]. Of these, the combined advantages of GC-MS mentioned before, as well as the existence of extensive spectral libraries make it an excellent choice for this purpose [71].

2. EXPERIMENTAL

2.1. METABOLOMIC DATASETS

Three different previously available metabolomics datasets profiled using GC-MS were submitted to analysis. For all of them, the values for the independent variable *X* matrix were expressed as ratios of peak area over internal standard peak area, while the dependent variable *y* was a binary response vector.

2.1.1. Type 2 diabetes mellitus dataset (T2DM)

The *X* matrix in T2DM contains the free fatty acids (FFAs) profiles of 45 type 2 diabetes mellitus patients and 45 healthy controls (size 90 by 21) as obtained by Tan et al. [73]. Diagnosis was based on the criteria of the American Diabetes Association (ADA) [74]. The subjects' overnight fasting plasma samples were obtained from the Xiangya Hospital of Hunan in Changsha, China. All the patients had at least one month of treatment through diet and athletic activities. The controls were from the same city as the patients, but not blood related.

Immediately after collection, each sample was submitted to centrifugation prior to storage with anticoagulant at -80°C. Sample preparation was carried out according to the procedure described by Yi et al. [75], in which hexane is used for double extraction of lipids obtaining methyl esters of esterified fatty acids (EFA) in the first extraction and of FFAs in the second. Instrumental analysis was carried out with a Shimadzu GC2010A (Kyoto, Japan) gas chromatographer coupled to a GCMS-QP2010 single quadrupole mass spectrometer (Compaq Pro Linear data system, class 5 K software). The GC-MS conditions are summarized in **Table 2**.

Table 2. Summary of GC-MS conditions used by Tan et al. for the acquisition of T2DM **[73]**.

GC							
Sample Introduction	**Carrier Gas**	**Injector Temperature**	**Column**				
			Type	**Internal Diameter**	**Length**	**Film Thickness**	**Temperature program**
1:10 split ratio 1.0 µL sample	Helium Flow rate: 1.0 mL/min	250°C	DB-23	0.25 mm	30 m	0.25 µm	70-150°C, 20°C/min 150-180°C, 6°C/min 180-220°C, 20°C/min 150-180°C, 6°C/min, then held for 9 min
MS							
Ionization voltage	**Ion source temperature**	**Full scan mode mass ranges**	**Velocity**				
70 eV	200°C	30-450 amu	0.2 s/scan				

The National Institute of Standards and Technology (NIST02) spectral library was implemented for the identification of FFAs. Chemometric resolution methods were used to solve overlapping peaks, as described in [73].

2.1.2. Postoperative cognitive dysfunction dataset (POCD)

The *X* matrix in POCD contains the metabolic profiles of 24 female Sprague Dawley rats after isoflurane anesthesia: 12 diagnosed with POCD and 12 healthy (size 24 by 44) as obtained by Zhang et al. [38]. The subjects were kept under controlled conditions of light and humidity, but free access to food and water for a week prior to the experiments. Since POCD involves loss in one or more components of mental capacity after induction of anesthesia [76], diagnosis was based on the successful or unfavorable completion of the y-maze ethology test [77, 78] to evaluate cognitive function 24 hours after anesthesia. The rats were purchased from Hunan Agricultural University in Changsha, China. The plasma samples were separated from blood through centrifugation and stored at -80°C.

The sample preparation procedure performed by Zhang et al. is described in their work [38] and involves protein precipitation using methanol and vortex and centrifugation to obtain a supernatant that is later evaporated dry. After being reconstituted with methoxyamine hydrochloride solution and incubated at 70°C, the mixture is derivatized with N,O-bis(trimethylsilyl)trifluoroacetamide (BSTFA) and incubated at 70°C. Instrumental analysis was carried out with a Shimadzu GCMS-QP2010 gas chromatographer-quadrupole mass spectrometer (Kyoto, Japan). The GC-MS conditions are summarized in **Table 4**.

Table 3. Summary of GC-MS conditions used by Zhang et al. for the acquisition of POCD **[38]**.

GC						
Sample Introduction	**Carrier Gas**	**Column**				
		Type	**Internal Diameter**	**Length**	**Film Thickness**	**Temperature program**
1:10 split ratio 1.0 µL sample	Helium Flow rate: 1.0 mL/min	DB-5ms	0.25 mm	30 m	0.25 µm	70°C for 4 min 70-300°C, 8°C/min, then held for 3 min
MS						
Ionization voltage	**Ion source temperature**	**Interface temperature**	**Full scan m/z ranges**	**Velocity**	**Detector voltage**	
70 eV	200°C	250°C	35-800 amu	0.2 s/scan	0.9 kV	

Of the over 100 obtained peaks, only the ones with signal-to-noise ratio higher than 10 were kept. Metabolite identification and quantification was carried out with the aid of NIST mass spectral library search and chemometric methods for peak resolution as described in [38].

2.1.3. Child obesity dataset (CHOB)

The *X* matrix in CHOB contains the metabolic profiles of 29 prepubertal children: 16 diagnosed as obese and 13 as overweight (size 29 by 30) as obtained by Zeng et al. [79]. The diagnosis was based on the children’s body mass index (BMI) to categorize them as overweight or obese. The subjects’ blood plasma samples were obtained from the Xiangya Hospital of Central South University in Changsha, China and stored at -80°C.

Sample preparation was carried out according to the procedure described by Zeng et al. [80], which involves protein precipitation with acetonitrile and vortex and centrifugation to obtain a supernatant that is later evaporated dry. After being reconstituted with hexane, the sample is derivatized with a mixture of BSTFA and trimethylsilyl chloride (TMSC) at 70°C. Instrumental analysis was carried out with a Shimadzu GCMS-QP2010 gas chromatographer-quadrupole mass spectrometer (Kyoto, Japan). The GC-MS conditions are summarized in **Table 4**.

Table 4. Summary of GC-MS conditions used by Zeng et al. for the acquisition of CHOB **[79]**.

GC							
Sample Introduction	**Carrier Gas**	**Injector Temperature**	**Column**				
			Type	**Internal Diameter**	**Length**	**Film Thickness**	**Temperature program**
1:10 split ratio 1.0 µL sample	Helium Flow rate: 1.0 mL/min	280°C	DB-1	0.25 mm	30 m	0.25 µm	100°C for 10 min 100-300°C, 8°C/min, then held for 2 min

MS						
Ionization voltage	**Ion source temperature**	**Full scan m/z ranges**	**Velocity**	**Detector voltage**	**Solvent cut time**	**Data acquisition start time**
70 eV	200°C	35-800 amu	0.2 s/scan	0.9 kV	3.5 min	4.0 min

The NIST/Environmental Protection Agency (EPA)/National Institute of Health (NIH) (NIST05) mass spectral library was implemented for the identification of metabolites.

2.2. DATA ANALYSIS

All data analysis procedures and programming, with the exception of outlier detection, were performed in MATLAB, version 7.11.0 (R2010b), Copyright 1984-2020, The MathWorks Inc., Natick, MA, USA. In addition, the pre-coded MATLAB scripts used for the implementation of algorithms are available freely at [81].

2.2.1. Comparison of VS methods

As a preliminary stage, the performance of CARS, SPA and RF applied to the described datasets was compared. The inputs for each script were set by default for all datasets.

2.2.1.1. CARS

The script carsplsda.m [81] was used for the implementation of CARS. The maximum number of LVs, or PLS components to extract, fold for CV and number of MCS were fixed to 5, 5 and 1000, respectively. Centering, which is subtracting the mean of all elements in a column from each element [13], was used as pretreatment for both X and y. At this stage, the variable importance score was set to be assigned according to the regression coefficients.

2.2.1.2. SPA

The script used to run SPA was spa.m [81]. The maximum number of PLS components for CV, fold for CV, number of MCS runs, ratio of training samples to total samples and number of variables to be sampled in each MCS run were fixed to 5, 5, 1000, 0.8 and 10, respectively. Centering was used as the data pretreatment method. The ρ value of 0.05 was predefined as a proper threshold for statistical significance [82, 83].

2.2.1.3. RF

The script rfvs.m was coded based on the TreeBagger MATLAB function to carry out variable selection using RF [84]. The script indicates the storage of the OOB estimates for variable importance obtained from this function. It allows the construction of more than one forest and averages the variable importance scores obtained for each one. These means are then plotted for each variable in a bar graph, which is an output of the script. The greater the value of the result of a given variable, the more important it is. In this way, variables with positive or negative importance are considered informative or interfering, respectively, while variables with importance equal to zero are considered non-informative.

The number of iterations or forests was set to 10. To choose the number of trees *N*, to be used, trial runs were carried out for each data set with 1000 and 1500 defined as *N*. Five hundred was defined as a suitable number of trees based on the plot of OOB classification error against the number of trees grown. A relatively stable trend of the OOB classification error is achieved well below this number for all datasets (**Figure 15**). However, there is a variation in stability when the number of trees grown (*N*) is changed. The impact *N* has on the result is not really understood and is beyond the scope of this study.

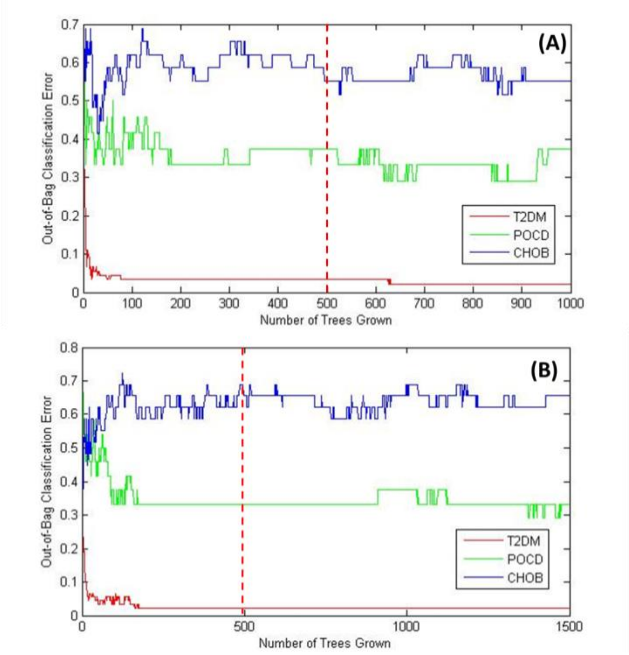


Figure 15. OOB error classification rate per number of trees when using **A)** *N*=1000 and **B)** *N*=1500. A relative stable trend in the OOB error classification rate is achieved well below 500 trees in all datasets.

2.2.1.4. PLS-DA model building

The datasets containing only the selected variables resulting from the previous methods were submitted to PLS-DA through the implementation of the script plslda.m [81]. Centering was used as the data pretreatment method. The script was run six times, alternating the number of PLS components to extract from two through seven, and selecting the one that produced the most accurate result.

To build models for the original datasets without VS, the script plsldacv.m [81] was used. This performs PLS-DA with K-fold CV to determine the optimal number of PLS components. Centering was chosen as the data pretreatment method and the fold was set to 5. The maximum number of PLS components to extract was set to 7.

The outputs of the scripts used for PLS-DA modeling with or without CV include the prediction assessment parameters: 1) misclassification error, 2) selectivity, 3) specificity, 4) AUC and 5) MCC. These parameters were taken into account in this stage.

The parameter settings for each variable selection and model building algorithm are summarized in **Table 5**.

Table 5. Summary of parameter settings for the comparison of VS methods.

	CARS for VS	**SPA for VS**	**RF for VS**	**PLS-DA model building after VS**	**PLS-DA model building with CV without VS**
SCRIPT	carsplslda.m	spa.m	rfvs.m	plslda.m	plsldacv.m
Number of LVs	5	-	-	-	-
Fold for CV	5	5	-	2-7	-
Number of MCS	1000	1000	-	-	-
Data pretreatment method	centering	centering	-	centering	centering
Variable importance according to	regression coefficients	-	-	-	-
Maximum LVs for CV	-	5	-	-	7
Ratio of training objects	-	0.80	-	-	-
Variables sampled during MCS	-	10	-	-	-
ρ cutoff value	-	0.05	-	-	-
Number of forests	-	-	10	-	-
Number of trees	-	-	500	-	-

2.2.2. Optimization of VS Method

2.2.2.1. Outlier detection

Score plots were built for the detection of outliers in each dataset using Sirius, Version 8.1, Copyright 1995-09, Pattern Recognition Systems AS, Bergen, Norway. PLS-DA was applied to obtain the scores, which are a representation of the objects in the new PLS coordinate system [85]. They were then projected on a plot of one PLS component against another. Because the first component has the highest percentage of variance explained in *X* and *y*, which successively decreases for each new one, the plot of the first two was given more consideration for the determination of outliers. As part of the usual pretreatment before building PLS models [13], the elements in both the response variable *y* and the predictor variables *X* were standardized prior to the abovementioned procedures to uniform the variables' standard deviation; however, normalization was not carried out because the data is expressed as a peak area ratio of each object over the internal standard.

2.2.2.2. Analysis of CARS algorithm

The script for CARS was examined in detail using MATLAB's debugging mode. From this stage on, the variable importance score was based on the SRs rather than the regression coefficients, because the values of the latter might be affected by the amount of orthogonal variation in the independent variable matrix *X* [41, 86]. The default setting for the ratio of objects to be kept in each MCS was found to be 0.95, with which, for datasets with few objects like the ones used, very few objects are left out of the training set. In 1000 MCS, many repeated training sets are bound to result, which basically makes the attempt of the first step of CARS to ensure its robustness in the variation of training samples quite insignificant.

To observe the effect of the training set size on the variation in resulting errors in each MCS, plots of CV error against MCS run were built. In addition, the unbalance of classes in training sets, that is, the inclusion of a much higher proportion of objects of one class than another, was investigated. The tests were performed on the POCD dataset. **Table 6** shows the different proportions of objects to be sampled in the first step of CARS that were tested.

Table 6. POCD training sets employed for the construction of CV error vs. MCS run plots.

Plot Code	Training Set Selection Description
TS1	0.70 sampled randomly
TS2	0.80 sampled randomly
TS3	0.90 sampled randomly
TS4	All objects of class 1 (12) and 7 of class -1 sampled randomly
TS5	All objects of class -1 (12) and 7 of class 1 sampled randomly
TS6	Equally distributed: 10 of Class 1 and 9 of Class -1 for the first 500 runs; 9 of Class 1 and 10 of Class -1 for the rest of the runs, all sampled randomly.
TS7	All objects (no sampling)

In addition to the training set issue, another irregularity in the original CARS algorithm was identified. The number of PLS components used to calculate the variable importance scores for steps 2 and 3 of CARS is arbitrarily defined, not optimized with CV as for the second PLS model built in step 4. The final selected variable subset depends on the importance scores calculated from this first model; thus, it is important to alternate the number of PLS components here as well, to determine which one produces a better model.

2.2.2.3. Modification of CARS algorithm

It was proposed that the purpose of the MCS in step 1 of obtaining results independent of the composition of the training set could be achieved using K-fold CV by removing one of the folds for training alternately until all folds were excluded once for model building.

In addition to this, three main parameters were identified for simultaneous optimization in the CARS algorithm:

i. Number of PLS components used for SR calculation in step 2
ii. Number and identity of variables to keep
iii. Number of PLS components used for CV in step 4

With all this in mind the original CARS algorithm was modified to consist of four main loops as described below.

2.2.2.3.1. Top loop: K-Fold CV

The original dataset is divided into *K* folds or groups. For the iterations one through *K*, the group number corresponding to the iteration number is excluded from the training set.

2.2.2.3.2. Outer loop: PLS components for SR calculation

For each training set, *c* different PLS-DA models are built using one through *c* PLS components. The SR vector resulting from the models are the variable importance scores.

2.2.2.3.3. Middle loop: EDF-ARS run for VS

For each SR vector, *N* different VS runs are carried out. Each run involves EDF and ARS and will produce a subset of selected variables. Note that due to the random component in the ARS stage of the VS process that creates a casual variation in the number of variables chosen, it is not satisfied that the highest VS run will select the fewest variables.

2.2.2.3.4. Inner loop: PLS components for CV

A model is built using each of the *N* subsets of variables. CV is carried out to optimize PLS components extracted to build the model. Thus, the training set is then divided into *K* subfolds, resulting in *K* subtraining sets and their corresponding test sets consisting of the group excluded from the subtraining set. The model built with the subtraining set is used to make predictions on the test set. The predicted class ($\check{y}$) is compared to the true class (y) of the *i*th object. A misclassification error is then calculated for each model.

After all loops have been completed, the total number of CV errors obtained will be the product of multiplying the number of folds (*K*), number of LVs for SR calculation (c_{SR}), number of VS runs (*N*) and number of LVs for model building (c_{PLSDA}) (**Equation 24**). The optimal model will be the one with the smallest CV error. Because we are dealing with classification problems, the error is not a value of how correct or incorrect a prediction is, like in regression problems. Any positive number obtained as a prediction is automatically converted to 1 (class 1). The same applies to negative numbers, converted to -1 (class -1). Thus, it is very common that the same error value will be repeated for different models. If more than one model achieves the lowest error, than the optimal one is the one that includes the lowest number of variables. If more than one model achieves the lowest error and uses the lowest number of variables, then the one that uses that least number of LVs for model building will be selected, and so on for the number of LVs for SR calculation. In accordance with the principle of parsimony, the simplest model with the lowest error and lowest number of variables is our target [31, 32]. The variable subset used to build this optimal model is the final subset of informative variables. **Figure 16** shows a scheme of the four loops in the algorithm of the modified CARS method.

$$Total\ Errors = K \times c_{SR} \times N \times c_{PLSDA}$$

Equation 24.

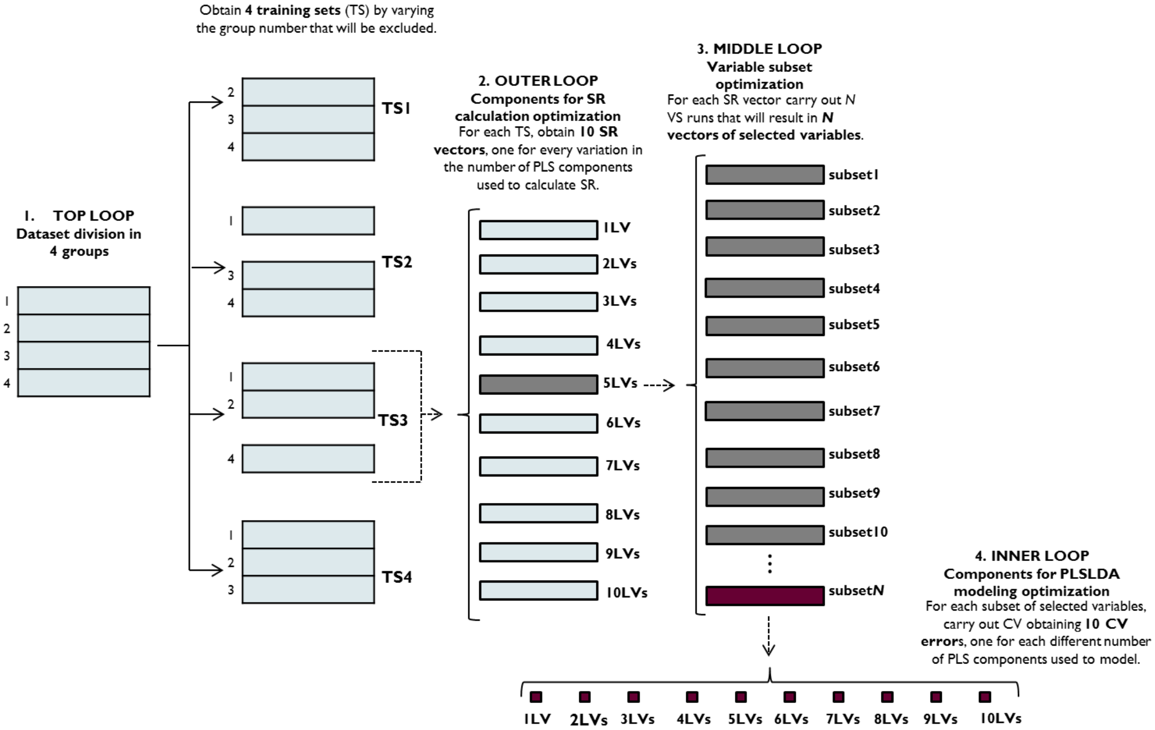

Obtain 4 training sets (TS) by varying the group number that will be excluded.
1. TOP LOOP
Dataset division in 4 groups
1
2
3
4
TS1
TS2
TS3
TS4
2. OUTER LOOP
Components for SR calculation optimization
For each TS, obtain 10 SR vectors, one for every variation in the number of PLS components used to calculate SR.
1LV
2LVs
3LVs
4LVs
5LVs
6LVs
7LVs
8LVs
9LVs
10LVs
3. MIDDLE LOOP
Variable subset optimization
For each SR vector carry out N VS runs that will result in N vectors of selected variables.
subset1
subset2
subset3
subset4
subset5
subset6
subset7
subset8
subset9
subset10
subsetN
4. INNER LOOP
Components for PLSLDA modeling optimization
For each subset of selected variables, carry out CV obtaining 10 CV errors, one for each different number of PLS components used to model.
1LV
2LVs
3LVs
4LVs
5LVs
6LVs
7LVs
8LVs
9LVs
10LVs

Figure 16. Scheme of the four loops of the modified CARS algorithm when using 4-fold CV, *N* VS runs and 10 maximum number of PLS components for SR calculation and model building.

2.2.2.4. New CARS-based method for variable identity and number optimization (VINO)

In addition to the modification of the CARS algorithm, VINO was proposed as a new VS method. The algorithm involves a separate optimization of the identity and the number of variables that should be considered important. In addition, the number of PLS components for both SR calculation and model building are optimized. In the previously described modified method (**Section 2.2.2.3**) the fold left out for training is simply discarded, not used as a test set for CV; thus, CV is carried out on the same data that was used for training. This does not ensure a proper prediction for future samples but just for this particular training data. To solve this, in the new method the fold that was excluded was used as a test set for the validation of the models built. A detailed description of the VINO algorithm can be found below.

2.2.2.4.1. Stage 1: Obtainment of LV accuracy matrix

2.2.2.4.1.1. Top loop: components for SR calculation optimization

A total of *c* maximum number of LVs for SR calculation is defined. The number will vary from one through *c* for each top loop iteration.

2.2.2.4.1.2. Outer loop: K-fold CV

For each of the *c* different possible number of LVs for SR calculation, the data is divided into *K* groups or folds. This will produce *K* training sets by alternating the fold that is excluded. The fold that is excluded from a training set is its corresponding test set.

2.2.2.4.1.3. Middle loop: EDF-ARS runs for VS

For each of the *K* test sets, a model is built using the number of LVs defined for the corresponding top loop iteration. The model will produce an SR vector, which is used to carry out *N* VS runs in two stages: EDF and ARS. This will result in *N* subsets of selected variables for each training set.

2.2.2.4.1.4. Inner loop: components for PLS-DA model building

For each of the *N* variable subsets, *c* models are built by varying the number of LVs for PLS-DA from one through *c*. Each model is validated using the test set that was left out from the training set used in the corresponding outer loop iteration. This will produce *c* misclassification errors for each of the *N* variable subsets in a size *N* by *c* LV error matrix. Each element of this matrix is subtracted from one to produce an LV accuracy matrix (LVAM) of the same size. **Figure 17** shows a scheme of the steps leading to the obtainment of the LVAM.

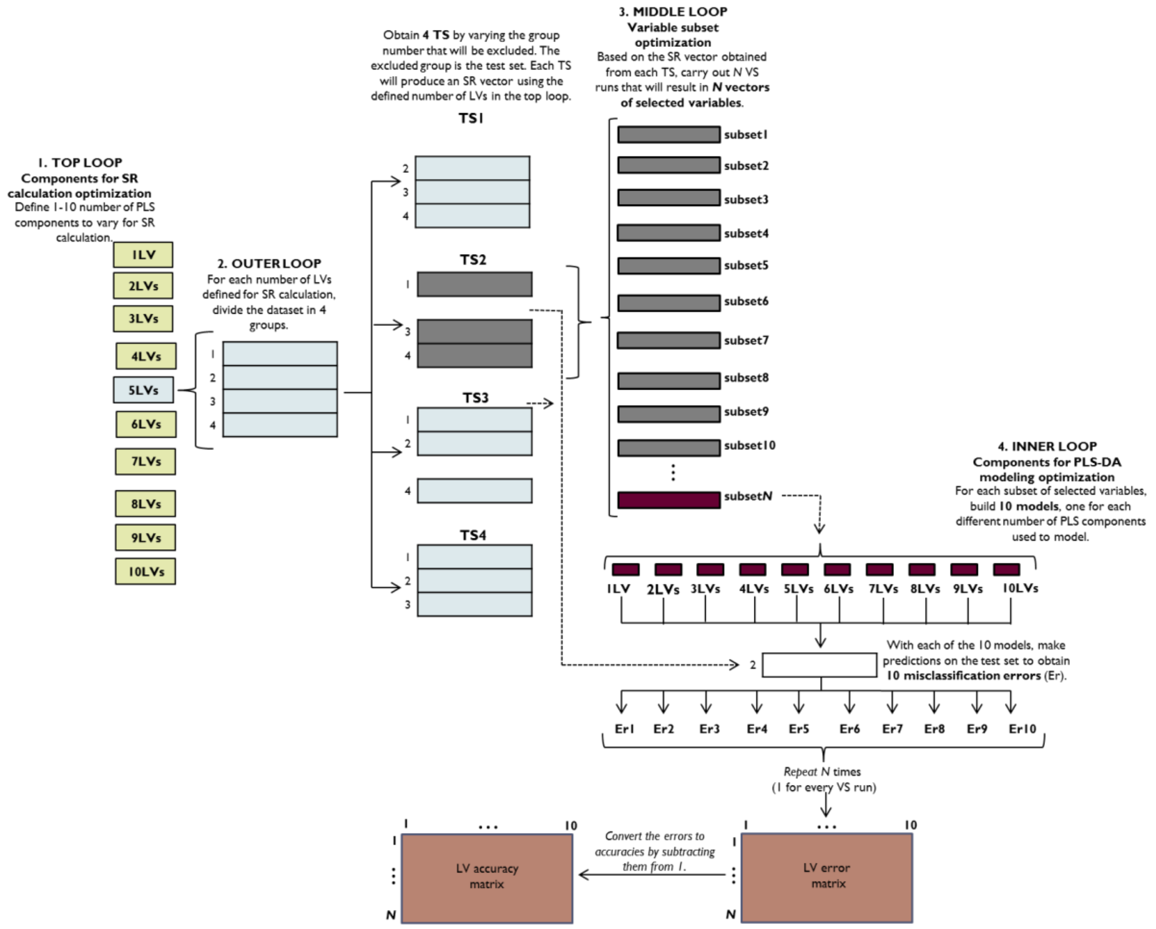

3. MIDDLE LOOP
Variable subset optimization
Based on the SR vector obtained from each TS, carry out N VS runs that will result in N vectors of selected variables.
Obtain 4 TS by varying the group number that will be excluded. The excluded group is the test set. Each TS will produce an SR vector using the defined number of LVs in the top loop.
TS1
1. TOP LOOP
Components for SR calculation optimization
Define 1-10 number of PLS components to vary for SR calculation.
1LV
2LVs
3LVs
4LVs
5LVs
6LVs
7LVs
8LVs
9LVs
10LVs
2. OUTER LOOP
For each number of LVs defined for SR calculation, divide the dataset in 4 groups.
1
2
3
4
TS2
TS3
TS4
subset1
subset2
subset3
subset4
subset5
subset6
subset7
subset8
subset9
subset10
subsetN
4. INNER LOOP
Components for PLS-DA modeling optimization
For each subset of selected variables, build 10 models, one for each different number of PLS components used to model.
1LV
2LVs
3LVs
4LVs
5LVs
6LVs
7LVs
8LVs
9LVs
10LVs
With each of the 10 models, make predictions on the test set to obtain 10 misclassification errors (Er).
Er1
Er2
Er3
Er4
Er5
Er6
Er7
Er8
Er9
Er10
Repeat N times
(1 for every VS run)
1
...
10
N
LV accuracy matrix
Convert the errors to accuracies by subtracting them from 1.
LV error matrix

Figure 17. Scheme of stage 1 of VINO when using 10 maximum LVs for SR calculation and model building, four folds for CV and *N* VS runs. The output of interest is the LV accuracy matrix.

2.2.2.4.2. Stage 2: Obtainment of variable number cube, variable identity cube and latent variable mean accuracy matrix

From the LVAM, a LV mean accuracy vector (LVMAV) of size one by *c* is obtained. Each element in this vector contains the mean value of the accuracies of the *N* models built using the variables selected in the corresponding VS runs for each one through *c* LVs for PLS-DA. It is important to note that it will not be possible to use all of the *c* amounts of LVs to build a model for every VS run selected subset. For example, in the last VS run where no more than two variables can be selected, LVs three through *c* will not be used because there is no room for dimensionality reduction. Thus, the mean for each number of LVs for modeling is obtained by summing the accuracies of only the VS runs where that amount of LVs could be used and then dividing by the total of those runs, which can be *N* or lower (**Equation 25**). This vector gives information on the effect that using a certain number of LVs for modeling has on the prediction performance.

$$Accuracy_{h,LVMAV} = \frac{\sum Accuracies\ of\ VS\ runs\ using\ h\ LVs\ for\ modeling}{Number\ of\ VS\ runs\ where\ modeling\ with\ h\ LVs\ is\ possible}, h = 1,2,\dots,c$$

Equation 25.

A LVMAV is obtained for each of the *K* training sets. Combining all of them will result in a latent variable mean accuracy matrix (LVMAM) of size *K* by c.

In addition to LVMAV, other data arrays result from LVAM. For every column in LVAM corresponding to a number of LVs used for modeling, a variable accuracy matrix (VAM) with rows representing VS runs (*N*) and columns representing each variable (p) is obtained. The accuracy reported for the first VS run is recorded in the first row of VAM, but only in the columns of the variables that were selected during that VS run. The rest of the elements in that row are given a negative value. This is repeated for each of the *N* VS runs, completing the whole VAM.

From VAM, a variable number vector (VNV) of size p by one is obtained. Each element of this vector contains an accuracy value assigned to every possible amount of variables that can be selected in each VS run. For each amount, the accuracies of the VS runs that selected it are summed and then divided by the total of those runs (**Equation 26**). This vector gives information of the performance when choosing a certain number of variables, regardless of which ones they are.

$$Accuracy_{j,VNV} = \frac{\sum Accuracies\ of\ VS\ runs\ that\ selected\ j\ number\ of\ variables}{Number\ of\ VS\ runs\ that\ selected\ j\ number\ of\ variables}, j = 1,2,\dots,p$$

Equation 26.

In addition to VNV, a variable identity vector (VIV) of size one by p is obtained from VAM. This vector gives information of the effect that including a particular variable has on the prediction performance, regardless of how many other variables it is combined with. For each variable, the accuracies of the VS runs that selected it are summed and then divided by the total of those runs (**Equation 27**). Each element of this vector contains an accuracy value assigned to each variable.

$$Accuracy_{j,VIV} = \frac{\sum Accuracies\ of\ VS\ runs\ that\ selected\ variable\ j}{Number\ of\ VS\ runs\ that\ selected\ variable\ j}, j = 1,2, \dots, p$$

Equation 27.

Since a total of c VAMs, one for each PLS component used for modeling, will be obtained from each LVAM, the same number of VNVs and VIVs will result for each training set. Combining the VNVs and the VIVs will produce a variable number matrix (VNM) and a variable identity matrix (VIM), respectively. Uniting the K VNMs and the K VIMs obtained for each training set will generate two three dimensional data arrays: variable number cube (VNC) and variable identity cube (VIC), respectively.

Figure 18 shows a scheme of stage 2 of VINO, showing the previously described processes that take place to obtain LVMAM, VNC and VIC.

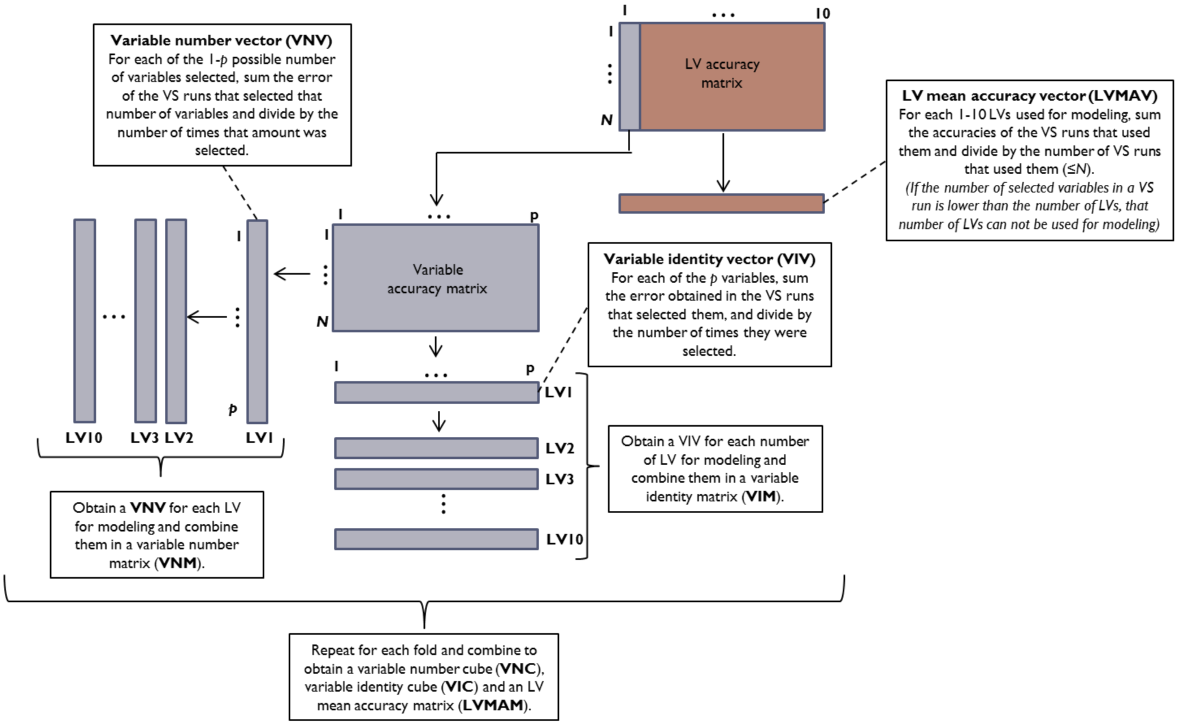

1
...
10
1
⋮
N
LV accuracy matrix
LV mean accuracy vector (LVMAV)
For each 1-10 LVs used for modeling, sum the accuracies of the VS runs that used them and divide by the number of VS runs that used them (≤N).
(If the number of selected variables in a VS run is lower than the number of LVs, that number of LVs can not be used for modeling)
Variable number vector (VNV)
For each of the 1-p possible number of variables selected, sum the error of the VS runs that selected that number of variables and divide by the number of times that amount was selected.
1
...
p
1
⋮
N
Variable accuracy matrix
Variable identity vector (VIV)
For each of the p variables, sum the error obtained in the VS runs that selected them, and divide by the number of times they were selected.
1
⋮
p
...
LV10
LV3
LV2
LV1
1
...
p
LV1
LV2
LV3
⋮
LV10
Obtain a VIV for each number of LV for modeling and combine them in a variable identity matrix (VIM).
Obtain a VNV for each LV for modeling and combine them in a variable number matrix (VNM).
Repeat for each fold and combine to obtain a variable number cube (VNC), variable identity cube (VIC) and an LV mean accuracy matrix (LVMAM).

Figure 18. Scheme of stage 2 of VINO when using 10 maximum LVs for SR calculation and model building, four folds for CV and *N* VS runs. The outputs of interest are VNC, VIC and LVMAM.

2.2.2.4.3. Stage 3: Obtainment of fold variable number matrix, fold variable identity matrix and fold latent variable vector.

The values for each training set in LVMAM, VNC and VIC are combined by way of a mean value. For LVMAM, this means that the first dimension, or the rows, will be averaged to obtain a fold latent variable vector (FLVV) of size one by *c*. For VNC and VIC, the third dimension will be averaged producing a fold variable number matrix (FVNM) and a fold variable identity matrix (FVIM), respectively. Averaging the third dimension consists in obtaining the mean of all the elements in the same row and column. The sizes of FVNM and FVIM are *p* by *c* and *c* by *p*, respectively. **Figure 19** shows a scheme of the training set averaging third stage of VINO.

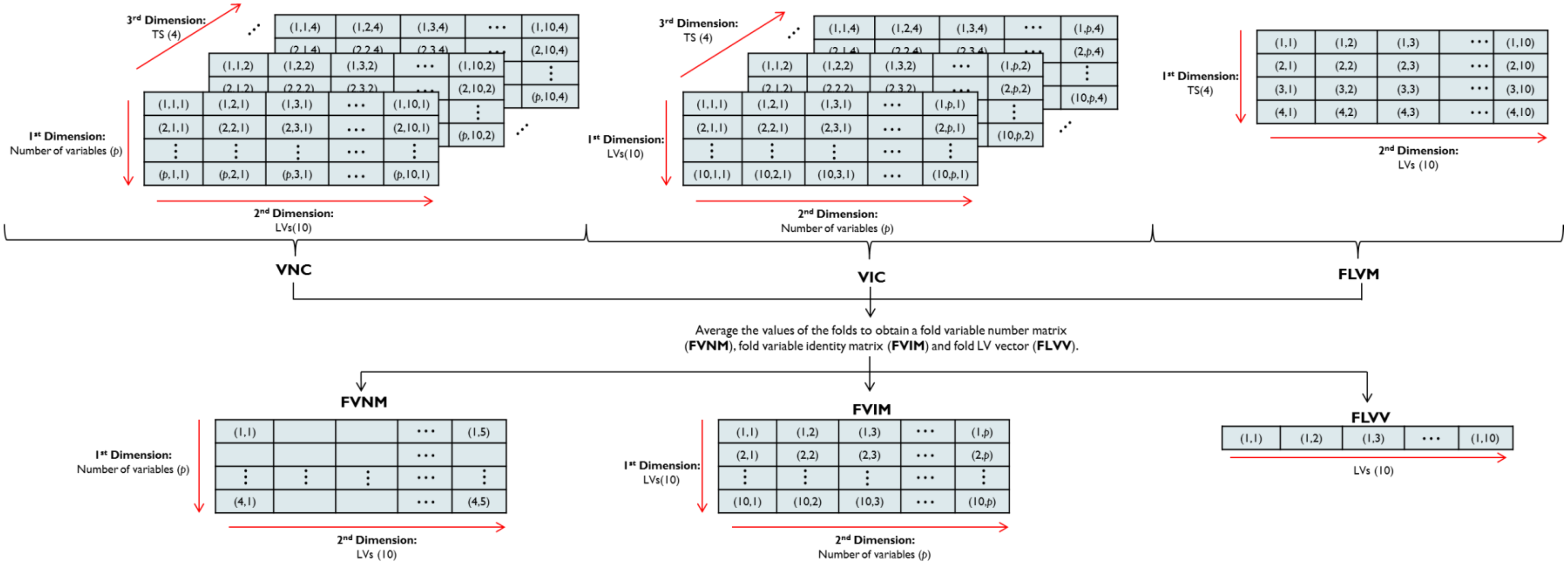

3rd Dimension: TS (4)
1st Dimension: Number of variables (p)
2nd Dimension: LVs(10)
(1,1,1) (1,2,1) (1,3,1) ··· (1,10,1)
(2,1,1) (2,2,1) (2,3,1) ··· (2,10,1)
(p,1,1) (p,2,1) (p,3,1) ··· (p,10,1)
(1,1,2) (1,2,2) (1,3,2) ··· (1,10,2)
(2,10,2)
(p,10,2)
(1,1,4) (1,2,4) (1,3,4) ··· (1,10,4)
(2,10,4)
(p,10,4)
VNC
3rd Dimension: TS (4)
1st Dimension: LVs(10)
2nd Dimension: Number of variables (p)
(1,1,1) (1,2,1) (1,3,1) ··· (1,p,1)
(2,1,1) (2,2,1) (2,3,1) ··· (2,p,1)
(10,1,1) (10,2,1) (10,3,1) ··· (10,p,1)
(1,1,2) (1,2,2) (1,3,2) ··· (1,p,2)
(2,p,2)
(10,p,2)
(1,1,4) (1,2,4) (1,3,4) ··· (1,p,4)
(2,p,4)
(10,p,4)
VIC
1st Dimension: TS(4)
2nd Dimension: LVs (10)
(1,1) (1,2) (1,3) ··· (1,10)
(2,1) (2,2) (2,3) ··· (2,10)
(3,1) (3,2) (3,3) ··· (3,10)
(4,1) (4,2) (4,3) ··· (4,10)
FLVM
Average the values of the folds to obtain a fold variable number matrix (FVNM), fold variable identity matrix (FVIM) and fold LV vector (FLVV).
FVNM
1st Dimension: Number of variables (p)
2nd Dimension: LVs (10)
(1,1) ··· (1,5)
(4,1) ··· (4,5)
FVIM
1st Dimension: LVs(10)
2nd Dimension: Number of variables (p)
(1,1) (1,2) (1,3) ··· (1,p)
(2,1) (2,2) (2,3) ··· (2,p)
(10,1) (10,2) (10,3) ··· (10,p)
FLVV
(1,1) (1,2) (1,3) ··· (1,10)
LVs (10)

Figure 19. Scheme of stage 3 of VINO when using 10 maximum LVs for SR calculation and model building, four folds for CV and *N* VS runs. The outputs of interest are FVNM, FVIM and FLVV.

2.2.2.4.4. Stage 4: Optimization of variable identity and number and of PLS components for SR calculation and PLS-DA modeling

From the FLVV, the number of LVs that exhibit the highest accuracy value is chosen as the optimal number of PLS components for modeling. If more than one number of LV achieves this value, the lowest number of LVs is chosen, to obtain a simpler model [31, 32].

The vectors in FVNM and FVIM corresponding to the optimal number of PLS components for modeling are chosen as the fold variable number vector (FVNV) and the fold variable identity vector (FVIV), respectively.

The FVIV is sorted in descending order so that the variables with highest accuracy are ranked first. The FVNV will aid in the specification of the cutoff value for the sorted FVIV. The number of variables to be kept (*u*) will be the one corresponding to the highest accuracy value in FVNV. If there is more than one number of variables with this value, then the lowest number of variables is selected, to obtain a simpler model [31, 32]. The optimal variable subset will contain the first *u* variables in the sorted FVIV.

The entire process to obtain an optimal number of PLS components for modeling and an optimal variable subset is repeated for each one through *c* LVs for SR calculation. A list of highest accuracy values obtained for each optimal number of PLS-DA LVs corresponding to every amount of LVs used for SR calculation is generated. The amount of LVs for SR calculation to which the highest value in this list belongs to is the optimal one. If more than one amount of LVs for SR calculation is associated with this accuracy value, the least amount of LVs is chosen.

The optimal number of PLS components for modeling and optimal variables subset are the ones corresponding to the optimal number of PLS components for SR calculation. **Figure 20** shows a scheme of the final stage of VINO in which the final outputs are four parameters which have been optimized: 1) number of PLS components for SR calculation,

2) number of PLS components for PLS-DA modeling, 3) identity of variables and 4) number of variables.

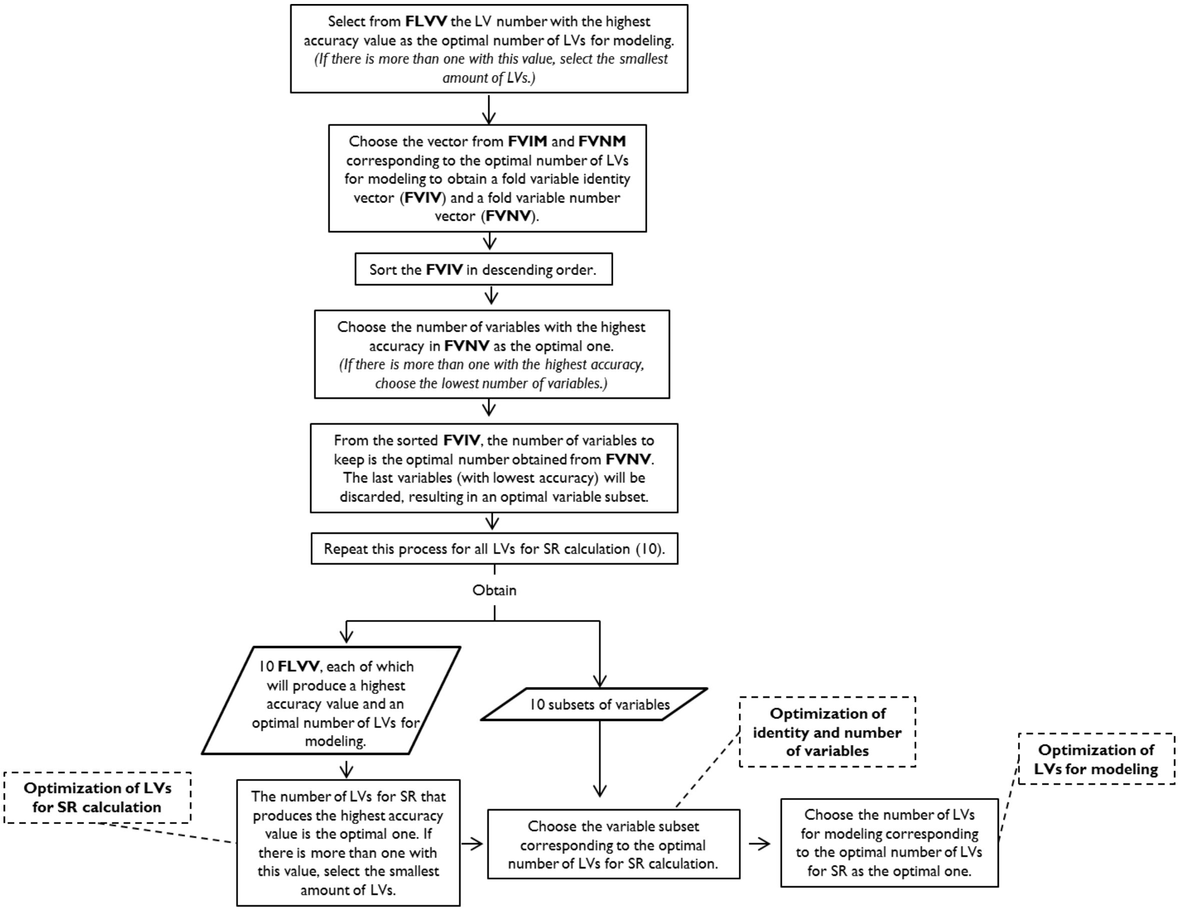

Select from **FLVV** the LV number with the highest accuracy value as the optimal number of LVs for modeling. *(If there is more than one with this value, select the smallest amount of LVs.)*
Choose the vector from **FVIM** and **FVNM** corresponding to the optimal number of LVs for modeling to obtain a fold variable identity vector (**FVIV**) and a fold variable number vector (**FVNV**).
Sort the **FVIV** in descending order.
Choose the number of variables with the highest accuracy in **FVNV** as the optimal one. *(If there is more than one with the highest accuracy, choose the lowest number of variables.)*
From the sorted **FVIV**, the number of variables to keep is the optimal number obtained from **FVNV**. The last variables (with lowest accuracy) will be discarded, resulting in an optimal variable subset.
Repeat this process for all LVs for SR calculation (10).
Obtain
10 **FLVV**, each of which will produce a highest accuracy value and an optimal number of LVs for modeling.
10 subsets of variables
Optimization of identity and number of variables
Optimization of LVs for modeling
Optimization of LVs for SR calculation
The number of LVs for SR that produces the highest accuracy value is the optimal one. If there is more than one with this value, select the smallest amount of LVs.
Choose the variable subset corresponding to the optimal number of LVs for SR calculation.
Choose the number of LVs for modeling corresponding to the optimal number of LVs for SR as the optimal one.

Figure 20. Scheme of stage 4 of VINO when using 10 maximum LVs for SR calculation and model building, four folds for CV and *N* VS runs. At the end of the algorithm four parameters will have been optimized: 1) LVs for SR calculation, 2) LVs for PLS-DA modeling and 3) identity and 4) number of variable in the form of a selected variable subset.

2.2.2.5. Comparison of original CARS performance with that of modified CARS and the new VINO method

The parameter settings to perform VS were constant when executing either CARS, modified CARS or VINO (**Table 7**). Because the number of objects in all datasets with outliers removed was divisible by four, with zero remainder for T2DM and POCD and one remainder for CHOB, the number of folds (*K*) was switched from five to four. This was done in an attempt to keep the number of objects in each fold as equal as possible. The maximum number of LVs for PLS-DA modeling (*c*) was set to 10. In the case of CARS, the number of LVs for SR calculation is defined as *c*, not varied from one through *c* like in the modified CARS and VINO. One thousand was established as a sufficient number of MCS or VS runs, as the case may be. Centering was kept as the data pretreatment method and, as mentioned before, the variable importance was calculated according to SRs.

The procedures were carried out five times for each method to have an idea of how stable their performance was in terms of repeatability.

Table 7. Parameter settings for the comparison of CARS, modified CARS and VINO.

Fold for CV	**Number of MCS/VS runs**	**Data pretreatment method**	**Variable importance according to**	**Maximum LVs for SR calculation* and model building**
4	1000	centering	SR	10

**In the case of CARS, the LVs for SR calculation is defined as 10, not varied from 1-10.*

Once the VS procedure was completed, modeling was carried out using the same plsldacv.m script described in **Section 2.2.1.4**. The number of maximum PLS components for CV was set to 10. The prediction assessment parameters mentioned in **Section 2.2.1.4** were taken into account to evaluate the VS procedure; however, because the misclassification error encompasses more information than the rest of the assessment parameters, more emphasis was placed on it.

3. RESULTS AND DISCUSSION

3.1. COMPARISON OF VS METHODS

The results of the performance of the methods of CARS, SPA and RF for VS in the preliminary stage are summarized in **Table 8**.

Table 8. Classification assessment results after carrying out VS with CARS, SPA and RF for T2DM, POCD and CHOB. The results of the VS method displaying the best performance have been highlighted for each dataset.

Datasets	Variable Selection Method	Prediction Assessment Parameters (%)				Variables chosen
		Error	AUC	Sensitivity	Specificity	
T2DM	None	0.0333	0.9746	0.9778	0.9556	All (21)
	CARS	0.0222	0.9751	0.9778	0.9778	5, 9, 11, 14, 16, 18, 20
	SPA	0.0000	0.9778	1.0000	1.0000	2, 4, 5, 6, 7, 8, 9, 10, 11, 14, 15, 17, 18, 20, 21
	RF	0.0111	0.9743	0.9778	1.0000	1, 2, 3, 4, 5, 6, 8, 9, 10,11, 12, 13, 14, 15
POCD	None	0.3103	0.6635	0.7500	0.6154	All (44)
	CARS	0.1250	0.8438	0.8333	0.9167	21, 22, 29, 33, 35
	SPA	0.3333	0.7431	0.6667	0.6667	8,11
	RF	0.0833	0.8750	0.9167	0.9167	5, 6, 9, 11, 13, 14, 26, 27, 28, 30, 31, 32, 35, 36, 39, 41, 42
CHOB	None	0.3103	0.6947	0.6875	0.6154	All (30)
	CARS	0.1724	0.8438	0.7500	0.9231	1, 2, 4, 5, 7, 9, 10, 14, 15, 19, 23, 24, 26, 27, 28, 30
	SPA	0.2069	0.7909	0.8125	0.7692	5, 23, 26
	RF	0.2759	0.8077	0.7500	0.8462	1, 8, 10, 12, 13, 14, 26

In general, T2DM appears to be the most stable dataset exhibiting the most ideal results for all VS methods, even when no VS is carried out. CHOB seems to be the least stable. This does not come as a surprise due to the fact that the dissimilarities in the profiles of the two groups being classified, obese and overweight, should be much more subtle then the ones in the profiles of an obese and healthy group, for example.

For every dataset, the results tend to improve when implementing any VS method. However, the method providing the best results varies from dataset to dataset: SPA, RF and CARS perform better on T2DM, POCD and CHOB, respectively.

Something to note is that considering many prediction assessment parameters makes interpretation complicated. Not all parameters will exhibit the best value for the same method. In **Table 8** when comparing the results for POCD when using all variables and the ones after VS with SPA, in terms of specificity, the latter is better; however, regarding the misclassification error, it is the contrary.

A possible alternative to deal with this without disregarding any of the parameters of interest is combing all of them to a single response [87]. A multi response optimization approach involving the termed “desirability function” has been implemented to solve problems involving industrial quality control, analytical techniques and pharmaceutical formulation, to name a few [87, 88, 89, 90, 91]. This approach is described in detail by Derringer and Suich [87] and consists in transforming each response into desirability values which are then combined through a geometric mean to obtain a global desirability value. This value will range between zero and one and will represent the overall assessment of the combined responses, being there a more favorable balance as it increases. This would produce a single response for comparison of VS methods if it were to be applied to the resulting prediction assessment parameters.

Note that, since in this stage the parameters for VS were given standard values, a more significant comparison would require their optimization for every method and dataset.

3.2. OPTIMIZATION OF VS METHOD

3.2.1. Outlier detection

The PLS-DA score plots for the three datasets of interest built in Sirius (© 1995-09) are presented in **Figure 21**. The percentage of explained variance by each LV for the independent variable *X* and the dependent variable *y* is shown in their axes labels. The ellipses are constructed using Hotelling's T^2 test [92], and indicate the limits out of which outliers will fall [93].

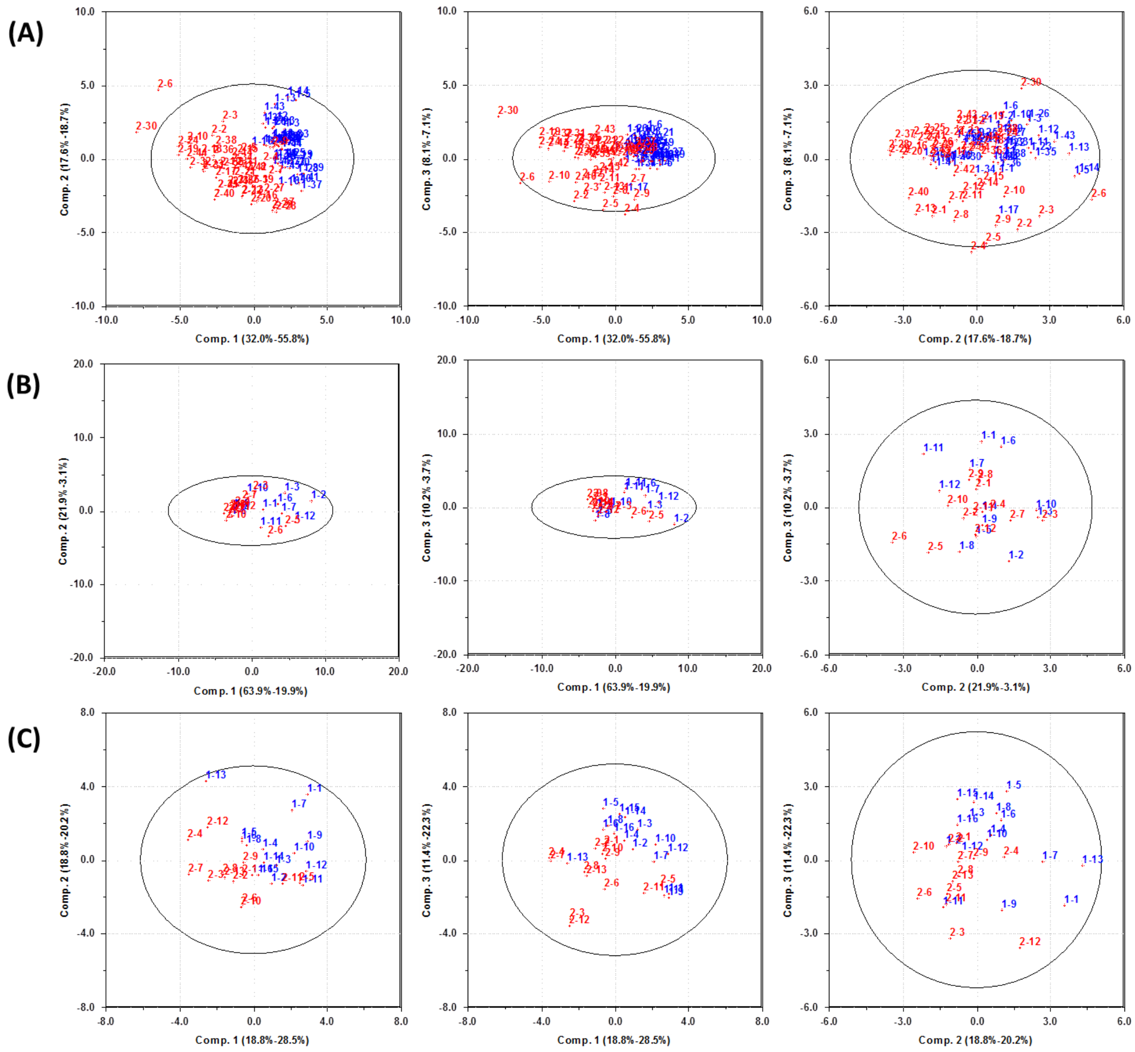


Figure 21. PLS-DA score plots of the first three PLS components of A) T2DM, B) POCD and C) CHOB. Objects of class 1 are labeled in blue, while class -1 markers are red. The variance explained in *X* and *y*, respectively, is stated beside the component number on the axes labels.

The resulting variance explained is in accordance to the known fact that it decreases with each new LV [94]. Because of this, the plot of component one against two was given more consideration for the selection of outliers. All objects in POCD and CHOB lie within the established limits. There appears to be an irregularity in relation to the first LV for the sixth and thirtieth object of class -1 in T2DM that is repeated in the second plot for both and in the third for the former. These two objects were eliminated from T2DM before carrying out the following stages. The fourth object of class -1 is deviated in relation to the third LV in the second and third plot; however, because the percentage of variance that this component explains for both X and y, 8.1 and 7.1, respectively, is quite small, this object was not considered an outlier.

Regarding the separation of classes, the differences seem to be less marked in CHOB, as was expected due to the similarity in adverse metabolic effects that obesity and overweight cause [95, 96].

3.2.2. Analysis of CARS algorithm

A problem that has been identified in the performance of CARS is the lack of consistency in its results [39]. The optimization of this method was chosen as a starting point to improve VS in our metabolomics datasets.

One of the findings during the analysis of the CARS MATLAB script used is that the number of LVs for SR calculation in the second step is simply set arbitrarily, not optimized like for the number of LVs for modeling in the final step. The need for this optimization was established in addition to that already existing for number of LVs for modeling and variable subset.

Another observation is that the ratio of objects selected during MCS in the first stage was set to 0.95. This value does not allow much variation of the training set composition when the number of objects in a dataset is small. For example, when applying this ratio to POCD, which is the dataset with the smallest amount of objects (24), only one of them will be left out alternately during MCS. This leaves only 24 training set possibilities for 1000

MCS runs; thus, they will be repeated more than once. This is not realistic enough to ensure robustness of the method with regards to the variation of objects used for training, and therefore eliminates the entire purpose of the MCS.

It was also found that the CARS script was coded in such a way that, independently of the number of objects kept for training, the proportion of class1/class-1 was maintained. For example, for the original T2DM, without outliers removed, having the same amount of objects from each class (45), the number of objects sampled from each one of them is 43, to obtain a total of 86 objects for training, 0.95 of the total number of objects (90). However, not every dataset contains an equal amount of objects for each class. T2DM with outliers removed, for instance, has two less objects for class -1, while for CHOB, it is three less objects for that class.

To evaluate the effect that sampling different proportions of classes, as well as different ratios for training in the MCS stage has on CARS's VS performance, plots of CV errors per MCS runs were constructed for POCD using different training sets as detailed in **Table 6** of **Section 2.2.2.2**. (**Figure 22**).

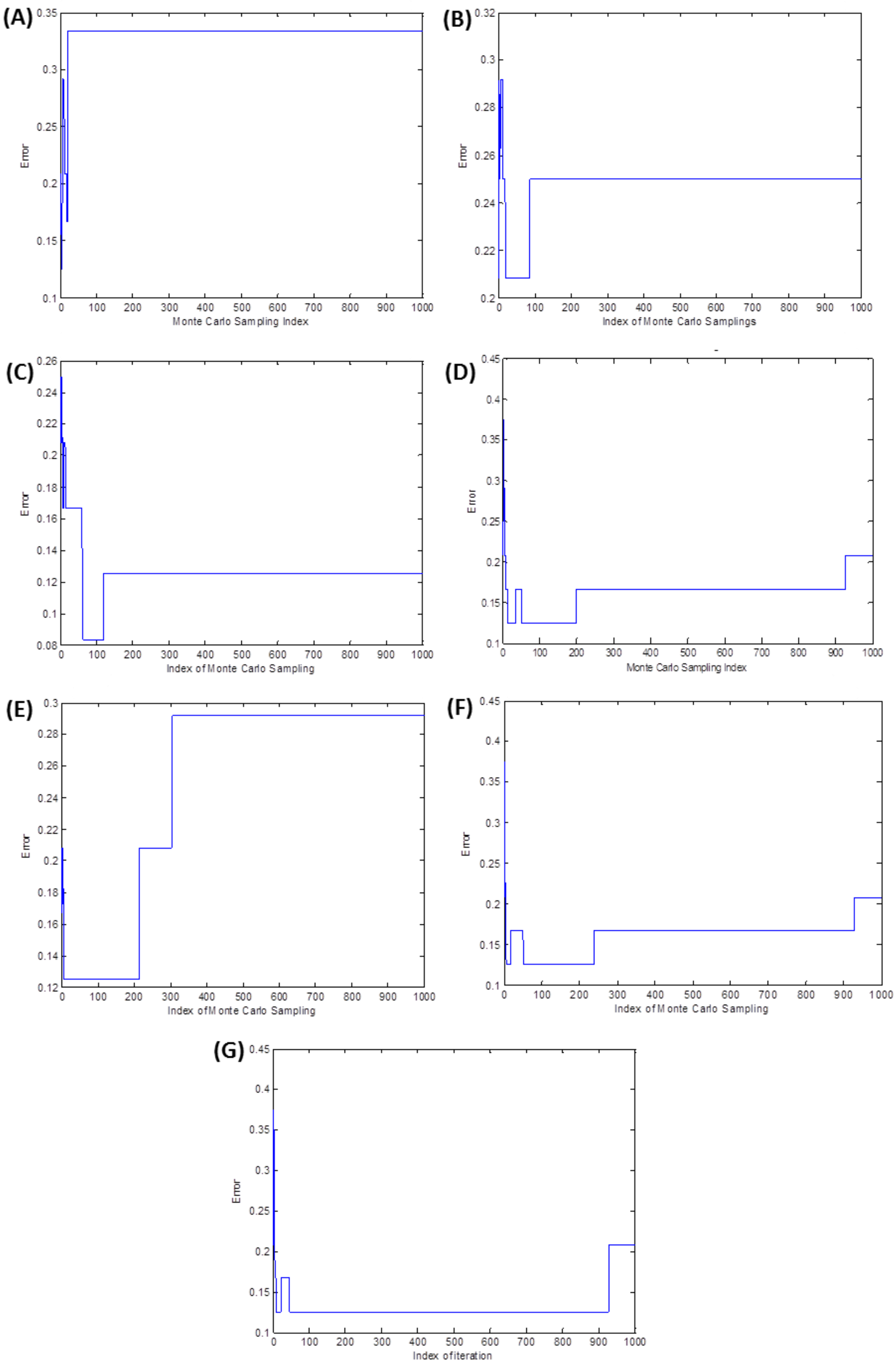


Figure 22. Plots of CV error per MCS run for the POCD dataset for training sets A) TS1, B) TS2, C) TS3, D) TS4, E) TS5, F) TS6 and G) TS7.

Because the error taken into account is a misclassification error, which is just a count of objects that were incorrectly classified divided by the total number of classified objects [20], multiple MCS runs will achieve the lowest error value. In the case of POCD, there are 24 objects to be classified, so there are basically 24 error possibilities (**Equation 28**), being the lowest 0.042 and the highest 1. This is illustrated in the following example for POCD in which the number of total objects in a dataset, *n*, which will be used to obtain a CV error in the final stage of CARS, is 24.

$$Possible\ Error_i = \frac{i}{n}, i = 1, 2, 3, \dots n$$

Equation 28.

i. $Possible\ Error = (1/n), (2/n), (3/n), \dots, (n/n = 1)$
ii. $Possible\ Error = (1/24), (2/24), (3/24), \dots, (24/24)$
iii. $Possible\ Error = (0.042), (0.083), (0.0125), \dots, (1)$

This being explained, the MCS run with lowest error and highest index was selected as the optimal one when using every training set. However, in every plot (**Figure 22**), there is some oscillation during the first MCS runs, followed by a constant error that only varies until the last 100 runs or less. This may be due to the few possible errors, training set combinations or both, that are bound to be repeated in so many MCS runs (*N*=1000). This suggests that the algorithm could be improved by implementing a new strategy to vary the objects used for training.

Another inference obtained by the plots in **Figure 22** is that, in some cases, the lowest error is achieved by chance. In plot A, for example, the lowest error value (0.1250) is achieved in the fourth MCS run, sixteen runs before it stabilizes to a constant value of 0.3333. This situation in which the lowest error value is achieved before a stable error trend is established also occurs in plots B through H. This random component affects the stability of the method.

One could think that, because the number of retained variables in the EDF stage of CARS decreases for each MCS run, in addition to the random component of the ARS stage, repeating a training set between runs doesn’t necessarily mean that the same variable subset, and therefore the same error, will be obtained. However, a different impression is given from plot G. Because for its construction no MCS was carried out, all objects are being included for training in every VS run; there is no variation in the training set. An error value of 0.1250 is maintained from runs 46 through 927 indicating that although there is a random component in the VS process through ARS, it does not have a significant effect on the subset of variables chosen. It can be considered an “educated” random choice, as variables with higher importance scores have a higher probability of being sampled. The uninformative variables will probably be left out anyways independently of the ratio of variables kept defined in the EDF stage. The increase in the error toward the final runs is probably due to the fact that at this point, the maximum number of selected variables is lower than that of actual informative variables.

It is important to keep in mind that CARS was originally developed for continuous data, specifically NIR spectra [39], for which the purpose is regression instead of classification. Thus, a RMSE is used to evaluate prediction performance as opposed to a misclassification error [23]. This gives way to infinite error possibilities, which leaves little or no room for repetition. This is an important difference between classification and regression that must be noted. Another difference is that a misclassification error does not indicate how correct or incorrect an error is; it is an absolute “yes” or “no”.

3.2.3. Modification of CARS algorithm

In an attempt to improve the strategy to ensure the robustness of the method in regards to the variation of the training set, the CARS algorithm was modified to divide the data into *K* folds as equal as possible to alternately leave out for training. The modified algorithm also involved the simultaneous optimization of LVs for both SR calculation and PLS-DA modeling in addition to that of the variable subset. The results of five independent runs of the modified and original CARS for T2DM, POCD and CHOB are shown in **Table 9**, **Table 10**, **Table 11**, respectively.

Table 9. Prediction performance results from T2DM when applying modified and original CARS five times and when including all variables (no VS).

VS Method	Prediction Assessment Parameters					Selected Variables
	Error	Sensitivity	Specificity	AUC	MCC	
None	0.0341	0.9333	1.0000	0.9682	0.9341	(All) 1-21
Modified CARS	0.0114	0.9778	1.0000	0.9695	0.9775	5, 8, 11, 18
	0.0114	0.9778	1.0000	0.9695	0.9775	5, 8, 11, 18
	0.0114	0.9778	1.0000	0.9695	0.9775	5, 8, 11, 18
	0.0114	0.9778	1.0000	0.9695	0.9775	5, 8, 11, 18
	0.0114	0.9778	1.0000	0.9695	0.9775	5, 8, 11, 18
CARS	0.0341	0.9556	0.9767	0.9693	0.9321	9, 11, 18
	0.0227	0.9778	0.9767	0.9690	0.9545	8, 11, 18
	0.0341	0.9556	0.9767	0.9693	0.9321	9, 11, 18
	0.0227	0.9778	0.9767	0.9690	0.9545	8, 11, 18
	0.0341	0.9333	1.0000	0.9682	0.9341	1-21

The overall results for the modified CARS when applied to T2DM surpass those of CARS (**Table 9**). The modified method achieved the highest values for every prediction assessment parameter. In addition, those values, as well as the number and identity of variables selected, is unchanged for all five runs, indicating great stability.

The minor variability in results from different independent CARS runs reflects its previously reported instability [39]. However, the most disconcerting result is that it selected all variables in the fifth run. It also achieved the same error as when no VS was performed in runs two and three, although it was never lower. This is also considered a successful optimization for the first four CARS runs, as the model has been simplified by including fewer variables without decreasing the prediction performance [31, 32].

Table 10. Prediction performance results from POCD when applying modified and original CARS five times and when including all variables (no VS).

VS Method	Prediction Assessment Parameters					Selected Variables
	Error	Sensitivity	Specificity	AUC	MCC	
None	0.1667	0.8333	0.8333	0.7465	0.6667	1-44
Modified CARS	0.0833	1.0000	0.8333	0.8056	0.8452	9, 20, 22, 26, 30, 31, 32, 34, 35, 36, 37, 43
	0.0833	0.9167	0.9167	0.8611	0.8333	2, 6, 9, 11, 14, 19, 20, 22, 24, 26, 28, 30, 31, 32, 34, 35, 36, 41, 44
	0.0833	0.9167	0.9167	0.8264	0.8333	4, 6, 8, 9, 11, 20, 22, 26, 29, 30, 31, 32, 41
	0.1250	0.8333	0.9167	0.8194	0.7526	19, 22, 27, 31, 32, 35
	0.1250	0.9167	0.8333	0.8264	0.7526	6 20 22 28 31 32 34 35
CARS	0.1667	0.8333	0.8333	0.7951	0.6667	31, 32
	0.1667	0.8333	0.8333	0.8333	0.6667	6, 22, 31, 32
	0.1667	0.8333	0.8333	0.7465	0.6667	1-44
	0.1667	0.8333	0.8333	0.7465	0.6667	1-44
	0.1667	0.9167	0.8333	0.7535	0.7526	4, 9, 12, 13, 18, 20, 22, 24, 26, 27, 28, 29, 30, 31, 32, 34, 35, 39, 41, 42, 43

As with the previous dataset, the results obtained for POCD reflect a better VS performance by the modified CARS when compared to the original. Although modified CARS's results are less stable for this dataset than those for T2DM, the superiority of all its prediction assessment parameters over those of original CARS and when no VS is carried out is maintained. However, the apparent variability in number and identity of variables is not satisfactory.

CARS's stability improved in regards to its prediction assessment parameter values, but decreased for the selected variable subsets. The number of variables selected varies from two to all, the latter resulting for one more run in this dataset than in the previous one. The prediction performance parameter values for the three runs that did not select all variables are quite similar than those obtained with the absence of VS. However, the fact that for these runs the number of variables considered is reduced without affecting the prediction performance, makes it a successful optimization, as mentioned before.

Nevertheless, as in the modified CARS, there is a large variability in the selected variable subsets.

Table 11. Prediction performance results from CHOB when applying modified and original CARS five times and when including all variables (no VS).

VS Method	Prediction Assessment Parameters					Selected Variables
	Error	Sensitivity	Specificity	AUC	MCC	
None	0.2759	0.6875	0.7692	0.7067	0.4546	1-30
Modified CARS	0.3750	0.5000	0.7500	0.5521	0.2582	5, 12, 15, 16, 17, 19, 20, 23, 26, 27, 30
	0.3750	0.5000	0.7500	0.6354	0,2582	5, 8, 10, 11, 17, 27, 29
	0.4583	0.5000	0.5833	0.5035	0.0836	5, 8, 11, 12, 13, 17, 21, 23, 28, 29
	0.5417	0.4167	0.5000	0.4861	0.0836	1, 5, 8, 11, 12, 13, 17, 23
	0.5000	0.5000	0.5000	0.4896	0.0000	5, 8, 11, 12, 13, 17, 23
CARS	0.2759	0.6875	0.7692	0.7043	0.4546	5, 8, 12, 16, 17, 20, 23, 26, 28
	0.2759	0.7500	0.6923	0.7115	0.4423	5, 10, 12, 17, 23
	0.2759	0.6875	0.7692	0.7067	0.4546	1-30
	0.2759	0.6875	0.7692	0.7067	0.4546	1-30
	0.2759	0.7500	0.6923	0.6226	0.4423	10, 12, 23, 26

Unlike the previous datasets, the results for modified CARS when applied to CHOB (**Table 11**) decline in regards to the original CARS and when including all variables. There appears to be a higher consistency in variables selected than in POCD, but the results for all prediction assessment parameters are not only unstable, but also indicate a much poorer performance.

CARS's consistency in the error values is improved to 100 per cent; however, the instability in selected variable subsets persists, as well as two occurrences when all variables are included. In general, the prediction performance when applying CARS is the same than when no VS is performed, and much better than the modified CARS.

As observed for the previous preliminary stage when three different VS methods were compared (**Section 3.1**), from the three datasets, T2DM appears to be the most stable while CHOB, the least. This indicates that the differences in metabolic profiles between diabetic and healthy individuals are much more marked than those of obese and overweight prepubertal children [95, 96, 97].

3.2.4. New method VINO

In spite of the rather promising results obtained from the modified CARS, the validation carried out in the final stage is performed on the same data used for training. A proper validation must be carried out on an independent test set [98, 99]. This was corrected in the new method VINO. The simultaneous optimization of the three parameters performed by modified CARS was also incorporated in the new method. The results of applying VINO on T2DM, POCD and CHOB five times are shown in **Table 12**, **Table 13** and **Table 14**, respectively. For each dataset, the results of running CARS five times, previously reported for comparison with modified CARS (**Section 3.2.3**) are also presented.

Table 12. Prediction performance results of VINO when applied to T2DM five times compared with five runs of CARS and when including all variables (no VS).

VS	Prediction Assessment Parameters					Selected Variables
Method	Error	Sensitivity	Specificity	AUC	MCC	
None	0.0341	0.9333	1.0000	0.9682	0.9341	(All) 1-21
VINO	0.0341	0.9778	0.9535	0.9672	0.9320	2, 5, 8, 9, 11, 12, 16, 18
	0.0227	0.9778	0.9767	0.9649	0.9545	2, 5, 8, 9, 11, 16, 17, 18
	0.0277	0.9778	0.9767	0.9649	0.9545	2, 5, 8, 9, 11, 16, 17, 18
	0.0341	0.9778	0.9535	0.9682	0.9320	2, 5, 6, 8, 9, 11, 16, 18
	0.0341	0.9778	0.9535	0.9682	0.9320	2, 5, 6, 8, 9, 11, 16, 18
CARS	0.0341	0.9556	0.9767	0.9693	0.9321	9, 11, 18
	0.0227	0.9778	0.9767	0.9690	0.9545	8, 11, 18
	0.0341	0.9556	0.9767	0.9693	0.9321	9, 11, 18
	0.0227	0.9778	0.9767	0.9690	0.9545	8, 11, 18
	0.0341	0.9333	1.0000	0.9682	0.9341	1-21

The prediction performance for T2DM when using both VS methods, as well as when no method is applied, appears to be quite similar when considering the assessment parameters (**Table 12**). There appears to be little variability in this respect for both CARS and VINO, as well for the number and identity of variables selected, with the exception of the last run for the former. However, CARS selects fewer variables while maintaining the prediction performance, which makes it the better option.

Table 13. Prediction performance results of VINO when applied to POCD five times compared with five runs of CARS and when including all variables (no VS).

VS Method	Prediction Assessment Parameters					Selected Variables
	Error	**Sensitivity**	**Specificity**	**AUC**	**MCC**	
None	0.1667	0.8333	0.8333	0.7465	0.6667	1-44
VINO	0.1667	0.8333	0.8333	0.7986	0.6667	9, 22, 26, 31, 35
	0.2083	0.7500	0.8333	0.8056	0.5854	1, 3, 5, 6, 9, 14, 15, 22, 26, 28, 29, 30, 31, 32, 33, 35, 37, 39, 40, 41, 42, 43, 44
	0.2917	0.5833	0.8333	0.5208	0.4303	1, 6, 9, 22, 26, 28, 30, 31, 32, 35, 40, 41, 42, 43, 44
	0.4167	0.5833	0.5833	0.5451	0.1667	4, 5, 34, 36, 39, 43
	0.2917	0.5833	0.8333	0.6146	0.4304	3, 6, 11, 25, 41, 43
CARS	0.1667	0.8333	0.8333	0.7951	0.6667	31, 32
	0.1667	0.8333	0.8333	0.8333	0.6667	6, 22, 31, 32
	0.1667	0.8333	0.8333	0.7465	0.6667	1-44
	0.1667	0.8333	0.8333	0.7465	0.6667	1-44
	0.1667	0.9167	0.8333	0.7535	0.7526	4, 9, 12, 13, 18, 20, 22, 24, 26, 27, 28, 29, 30, 31, 32, 34, 35, 39, 41, 42, 43

The prediction performance for POCD (**Table 13**) drops considerably when using VINO, being the first run an exception. The values for the assessment parameters are not only unstable, but also indicate a low prediction performance. There is also lack of consistency in the identity and number of variables selected.

Table 14. Prediction performance results of VINO when applied to CHOB five times compared with five runs of CARS and when including all variables (no VS).

VS Method	Prediction Assessment Parameters					Selected Variables
	Error	Sensitivity	Specificity	AUC	MCC	
None	0.2759	0.6875	0.7692	0.7067	0.4546	1-30
VINO	0.2759	0.6875	0.7692	0.7067	0.2759	1, 2, 7, 8, 9, 10, 12, 17, 18, 20, 21, 24, 25, 26, 27, 28, 29
	0.4483	0.6250	0.4615	0.5577	0.4483	1, 2, 7, 9, 10, 12, 15, 16, 18, 20, 21, 24, 25, 26, 27, 28, 29
	0.4483	0.6250	0.4615	0.5577	0.4483	1, 2, 5, 7, 9, 10, 12, 16, 18, 20, 21, 24, 25, 26, 27, 28, 29
	0.4138	0.6250	0.5385	0.5962	0.4138	1, 2, 5, 9, 12, 15, 18, 20, 21, 22, 23, 24, 26, 27, 28, 29
	0.2759	0.7500	0.6923	0.7163	0.2759	2, 5, 9, 12, 16, 18, 20, 21, 22, 23, 24, 26, 27, 29
CARS	0.2759	0.6875	0.7692	0.7043	0.4546	5, 8, 12, 16, 17, 20, 23, 26, 28
	0.2759	0.7500	0.6923	0.7115	0.4423	5, 10, 12, 17, 23
	0.2759	0.6875	0.7692	0.7067	0.4546	1-30
	0.2759	0.6875	0.7692	0.7067	0.4546	1-30
	0.2759	0.7500	0.6923	0.6226	0.4423	10, 12, 23, 26

Like for POCD, there is a decrease in the prediction performance when using VINO to select variables from CHOB (**Table 14**) when comparing it to that when using CARS and when no VS is carried out. Although the error values appear to be more stable for this dataset than for POCD, the instability of selected variable subsets persists.

Despite VINO's poor performance in selecting important variables from these datasets, its ability to discard interfering variables was yet to be verified. For this purpose, new variable subsets were generated by removing from the original datasets the variables that were never selected during any of the five runs of VINO. The results are shown in **Table 15**.

Table 15. Prediction performance results for T2DM, POCD and CHOB in three instances: 1) with no VS, 2) when eliminating the variables that weren't selected in any of the previously reported five runs of VINO and 3) in the five previously reported runs of VINO.

DATASET	VS Method	Prediction Assessment Parameters					Selected Variables
		Error	Sensitivity	Specificity	AUC	MCC	
T2DM	None	0.1667	0.8333	0.8333	0.7465	0.6667	1-44
	VINO Elim.	0.0341	0.9778	0.9535	0.9618	0.9320	2, 5, 6, 8, 9, 11, 12, 16, 17, 18
	VINO	0.0341	0.9778	0.9535	0.9672	0.9320	2, 5, 8, 9, 11, 12, 16, 18
		0.0227	0.9778	0.9767	0.9649	0.9545	2, 5, 8, 9, 11, 16, 17, 18
		0.0277	0.9778	0.9767	0.9649	0.9545	2, 5, 8, 9, 11, 16, 17, 18
		0.0341	0.9778	0.9535	0.9682	0.9320	2, 5, 6, 8, 9, 11, 16, 18
		0.0341	0.9778	0.9535	0.9682	0.9320	2 5 6 8 9 11 16 18
POCD	None	0.1667	0.8333	0.8333	0.7465	0.6667	1-44
	VINO Elim.	0.1250	0.9167	0.8333	0.8611	0.7526	2, 4, 6, 8, 9, 11, 14, 19, 20, 22, 24, 26, 27, 28, 29, 30, 31, 32, 34, 35, 36, 37, 41, 44
	VINO	0.1667	0.8333	0.8333	0.7986	0.6667	9, 22, 26, 31, 35
		0.2083	0.7500	0.8333	0.8056	0.5854	1, 3, 5, 6, 9, 14, 15, 22, 26, 28, 29, 30, 31, 32, 33, 35, 37, 39, 40, 41, 42, 43, 44
		0.2917	0.5833	0.8333	0.5208	0.4303	1, 6, 9, 22, 26, 28, 30, 31, 32, 35, 40, 41, 42, 43, 44
		0.4167	0.5833	0.5833	0.5451	0.1667	4, 5, 34, 36, 39, 43
		0.2917	0.5833	0.8333	0.6146	0.4304	3, 6, 11, 25, 41, 43
CHOB	None	0.2759	0.6875	0.7692	0.7067	0.4546	1-30
	VINO Elim.	0.3448	0.5625	0.7692	0.7476	0.3350	1, 5, 8, 10, 11, 12, 13, 15, 16, 17, 19, 20, 21, 23, 26, 27, 28, 29, 30
	VINO	0.4483	0.6250	0.4615	0.5577	0.0874	1, 2, 7, 8, 9, 10, 12, 17, 18, 20, 21, 24, 25, 26, 27, 28, 29
		0.4483	0.6250	0.4615	0.5577	0.0874	1, 2, 7, 9, 10, 12, 15, 16, 18, 20, 21, 24, 25, 26, 27, 28, 29
		0.4138	0.6250	0.5385	0.5962	0.1635	1, 2, 5, 7, 9, 10, 12, 16, 18, 20, 21, 24, 25, 26, 27, 28, 29
		0.2759	0.7500	0.6923	0.7163	0.4423	1, 2, 5, 9, 12, 15, 18, 20, 21, 22, 23, 24, 26, 27, 28, 29
		0.3103	0.6875	0.6923	0.6683	0.3780	2, 5, 9, 12, 16, 18, 20, 21, 22, 23, 24, 26, 27, 29

The minor improvement in the results for POCD and CHOB suggest that VINO's ability to discard interfering variables is better than selecting informative ones. However, there appears to be no change in the results for T2DM. This aspect is one that can be investigated further and could lead to a change in the focal point of the VINO algorithm for its improvement.

In light of VINO's poor VS performance, the possibility that the mean did not provide a proper representation to designate an accurate identity importance value for each of the *p* variables was proposed. If this was the case, it could be replaced by the median, for instance. To discard this hypothesis, normal probability plots of the accuracies resulting from the 1000 MCS runs when using optimal LVs for SR calculation and PLS-DA model building were constructed for each of the *p* variables. **Figure 21** shows the plots that were identified as the most asymmetrical.

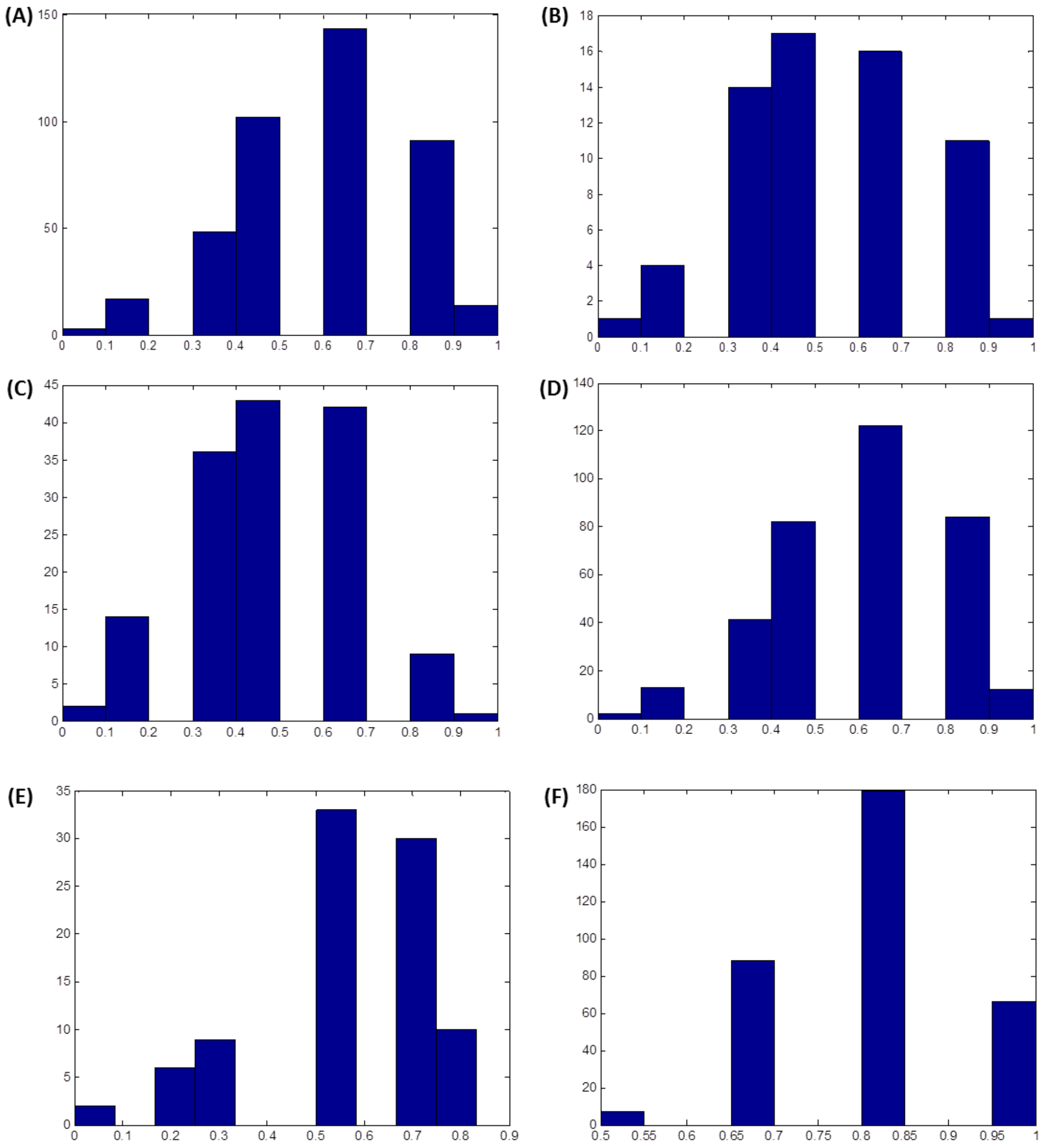


Figure 23. POCD normal probability plots of accuracy values designated in the 1000 MCS runs for A) variable 6; fold 2, B) variable 24; fold 2, C) variable 26; fold 2, D) variable 30; fold 2, E) variable 32; fold 2 and F) variable 41; fold 4.

The accuracy values do not appear to lean to any side of the mean in a manner that might be considered unusual in any of the plots. This indicates that the mean is a suitable measure to represent the combined data.

4. CONCLUSION

The theoretical mechanism of three variable selection methods as well as their performance when applied to three different metabolomics datasets using arbitrarily defined settings was compared. Key similarities and differences were identified among the methods, and they were all found to base variable importance according to different criteria. Although the method achieving the best assessment values varied with each dataset, for every one of them the prediction performance was improved when VS was carried out, whatever the method, as opposed to using all the original variables. This indicates that VS does indeed improve a model's prediction performance. In addition, CARS was not found to perform worse than the other VS methods, even though it was developed to solve regression problems. This suggests that this method is indeed applicable for classification as well.

In regards to the datasets used, T2DM exhibited the most stable behavior while CHOB displayed the least. This indicates a more marked difference between classes in the former than in the latter. Considering that the metabolic differences between diabetic and healthy individuals are bound to be larger than those between obese and overweight children, this outcome is quite realistic.

Important differences between regression and classification were established, the most relevant being the type of values assigned to the response variable: continuous for the former and categorical for the latter. This leads to different ways of validating prediction performance for each one. The parameter on which this study was based on, the misclassification error, does not provide information about how correct or incorrect a prediction is. This gives reason to doubt whether basing the importance score assigned to each variable on this prediction assessment parameter is appropriate.

The original CARS algorithm was modified to simultaneously optimize the PLS components used for SR calculation and modeling as well as the subset of informative variables. Although the results seem to be quite acceptable, the fact that model validation was carried out on the same objects used for training led to the rethinking of the method's approach.

VINO was proposed as a new VS method based on the separate optimization of identity and number of informative variables. In addition, this method also involves the optimization of the three parameters considered in the modified CARS. However, its implementation did not prove to increase model prediction performance when compared to the results obtained when using the original or modified CARS, or when using all variables in the original dataset. Some of the aspects identified as possible pathways to improve VINO's performance were tested, only to be discarded. Further study regarding other untested pathways is needed for the improvement of this method.

5. FUTURE WORK

In order to obtain a more accurate comparison of the VS methods included in this study, the optimization of their parameters, which were set by default here, should be optimized prior to their implementation.

Seeing that there are many parameters that assess prediction performance, instead of choosing one to analyze individually, combining the responses into one overall value is an alternative that can be implemented in future studies. The desirability function is a possible approach that applies this strategy.

The final variable subset in original and modified CARS, as well as in VINO is selected based on a misclassification error. Because this parameter does not indicate the degree of correctness of a prediction, the use of another measure that produces continuous values for the evaluation of prediction performance involved in the algorithms of the abovementioned methods can be tested.

In this study, VINO was applied to metabolomics datasets to solve classification problems. It would also be interesting to evaluate its performance on datasets with continuous response variables on which regression will be performed.

6. REFERENCES

[1] H. Martens and T. Næs, "Multivariate Calibration", John Wiley & Sons Ltd., ISBN 0 471 90979 3, 1989.

[2] "The Kilogram - Unit of Mass," Physikalisch-Technische Bundesanstalt, 14 December 2011. [Online]. Available: http://www.ptb.de/cms/en/fachabteilungen/abt1/fb-11/ag-1110/the-kilogram-unit-of-mass.html. [Accessed 3 April 2013].

[3] "Handout 3 - Regression Analysis," Iowa State University Department of Economics, [Online]. Available: http://www2.econ.iastate.edu/classes/econ321/rosburg/Econ%20321%20-%20Spring%202010/Handouts/Handout%203%20-%20Regression%20Analysis%20(Spring%202010).pdf. [Accessed 3 April 2013].

[4] S. Waner, "1.3 Linear Functions and Models, Part A: Basics: Slope and Intercept," 2010. [Online]. Available: http://www.zweigmedia.com/RealWorld/tutorialsf0/frames1_3.html. [Accessed 3 April 2013].

[5] J. Wooldridge, "Introductory Econometrics: A Modern Approach, Fifth Edition", Mason, OH, USA: South-Western, Cengage Learning, ISBN-13 978-1-111-53104-1, 2013.

[6] R. Brant, "Multiple Regression," The University of British Columbia, 24 March 2004. [Online]. Available: http://www.stat.ubc.ca/~rollin/teach/643w04/lec/node15.html. [Accessed 3 April 2013].

[7] J. Gosling, "Introductory Statistics: A comprehensive, self-paced step-by-step statistics course for tertiary students", Glebe, NSW, Australia: Pascal Press, ISBN 1 86441 015 9, 1995.

[8] G. McLachlan, "Discriminant Analysis and Statistical Pattern Recognition", Hoboken, NJ, USA: John Wiley & Sons, Inc., ISBN 0-471-61531-5, 1992.

[9] W. Klecka, "Discriminant Analysis", Newbury Park, CA, USA: Sage Publications, Inc., ISBN 0-8039-1491-1, 1980.

[10] R. Burns and R. Burns, "Business Research and Statistics using SPSS: Additional Chapters. Chapter 25: Discriminant Analysis," SAGE Publications, 2008. [Online]. Available: http://www.uk.sagepub.com/burns/website%20material/Chapter%2025%20-%20Discriminant%20Analysis.pdf. [Accessed 8 April 2013].

[11] "Popular Decision Tree: Classification and Regression Trees (C&RT)," StatSoft Inc., 2013. [Online]. Available: http://www.statsoft.com/textbook/classification-and-regression-trees/?button=1. [Accessed 8 April 2013].

[12] Y. Liang and L. Yi, "The Basis of Chemometrics", Shanghai, China: East China University of Science and Technology Press, ISBN 978-7-5628-2871-6, 2010.

[13] T. Hill and P. Lewicki, "Statistics: Methods and Applications, A Comprehensive Reference for Science Industry and Data Mining", Tulsa, OK, USA: StatSoft, Inc., ISBN 1-884233-59-7, 2006.

[14] H. Abdi, "Partial least squares regression and projection on latent structure regression (PLS Regression)," 2010. [Online]. Available: www.wiley.com/wires/compstats. [Accessed 9 April 2013].

[15] A. Sharma, "Text Book of Matrix", New Delhi, India: Discovery Publishing House, ISBN 81-7141-894-5, 2004.

[16] M. King and N. Mody, "Numerical and Statistical Methods for Bioengineering: Applications in MATLAB", New York, NY, USA: Cambridge University Press, ISBN 978-0-521-87158-7, 2011.

[17] C. Cerrano-Sinca and B.Gutierrez-Nieto, "Partial Least Square Discriminant Analysis for bankrupcy prediction", Decision Support Systems, Vol. 54, No. 3, p. 1245-1255, 2013.

[18] "ChemModLab Documentation: PLS-LDA," NCSU, 2008. [Online]. Available: http://eccr.stat.ncsu.edu/ChemModLab/PLS-LDA.pdf. [Accessed 9 April 2013].

[19] A. Izenman, "Modern Multivariate Statistical Techniques: Regression, Classification and Manifold Learning", New York, NY, USA: Springer Science+Business Media, ISBN 978-0-387-78188-4, 2008.

[20] M. Gönen, "Analyzing Receiver Operating Characteristic Curves with SAS", Cary, NC, USA: ASA Institute Inc., ISBN 978-1-59994-298-8, 2007.

[21] K. Sarma, "Predictive Modeling with SAS® Enterprise Miner™: Practical Solutions for Business Applications", Cary, NC, USA: SAS Institute Inc., ISBN 978-1-59047-703-8, 2007.

[22] N. Gopalan and B. Sivaselvan, "Data Mining: Techniques and Trends", New Delhi, India: PHI Learning Private Limited, ISBN 978-81-203-3812-8, 2009.

[23] R. Wehrens, "Chemometrics with R, Multivariate Data Analysis in the Natural Sciences and Life Sciences", Berlin, Germany: Springer, ISBN 978-3-642-17840-5, 2011.

[24] "Disease Screening-Statistics Teaching Tools," New York State Department of Helath, 1999. [Online]. Available: http://www.health.ny.gov/diseases/chronic/discreen.htm. [Accessed 29 April 2013].

[25] D. J. Hosmer, S. Lemeshow and R. Sturdivant, "Applied Logistic Regression, Third Edition", Hoboken, NJ, USA: John Wiley & Sons, Inc., ISBN 978-0-470-58247-3, 2013.

[26] C. Cortes and M. Mohri, "AUC Optimization vs. Error Rate Minimization", Florham

Park, NJ, USA: AT&T Labs Research, http://books.nips.cc/papers/files/nips16/NIPS2003_AA40.pdf.

[27] S. Sayad, "Model Evaluation-Classification," [Online]. Available: http://www.saedsayad.com/model_evaluation_c.htm. [Accessed 29 April 2013].

[28] J. Fan, S. Upadhye and A. Worster, "Understanding operating receiver characteristic (ROC) curves", Canadian Journal of Emergency Medicine, Vol. 8, No. 1, p. 19-20, 2006.

[29] X. Hu (Ed.) and Y. Pan (Ed.), "Knowledge Discovery in Bioinformatics: Techniques, Methods and Applications", Hoboken, NJ, USA: John Wiley & Sons, Inc., ISBN 978-0-471-77796-0, 2007.

[30] K. Sarma, "Variable Selection and Transformation of Variables in SAS® Enterprise Miner™ 5.2," 2007. [Online]. Available: http://www.nesug.org/proceedings/nesug07/sa/sa17.pdf. [Accessed 3 April 2013].

[31] M. Forster, "Parsimony and Simplicity," University of Wisconsin-Madison: Department of Philosophy, January 1998. [Online]. Available: http://philosophy.wisc.edu/forster/220/simplicity.html. [Accessed 3 April 2013].

[32] J. Braithwaite, "Occam's Razor: The Principle of Parsimony," Academia, 2007. [Online]. Available: http://www.academia.edu/1742741/Occams_Razor_The_principle_of_Parsimony. [Accessed 3 April 2013].

[33] J. Faraway, "Practical Regression and Anova using R," The Johns Hopkins University, 2002. [Online]. Available: http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf. [Accessed 3 April 2013].

[34] Z. Xiaobo, Z. Jiewen, M. Povey, M. Holmes and M. Hanpin, "Variables selection methods in near-infrared spectroscopy", Analytica Chimica Acta, Vol. 667, p. 14-32,

2010.

[35] I. Chong and C. Jun, "Performance of some variable selection methods when multicollinearity is present", Chemometrics and Intelligent Laboratory Systems, Vol. 78, No. 1-2, p. 103-112, 2005.

[36] H. Li, M. Zeng, B. Tan, Y. Liang, Q. Xu and D. Cao, "Recipe for revealing informative metabolites based on model population analysis", Springer Science+ Business Media, Vol. 6, No. 3, p. 353-361, 2010.

[37] T. Rajalahti, R. Arneberg, F. Berven, K. Mhyr, R. Ulvik and O. Kvalheim, "Biomarker discovery in mass spectral profiles by means of selectivity ratio plot", Chemometrics and Intelligent Laboratory Systems, Vol. 95, p. 35-48, 2009.

[38] W. Zhang, L. Zhang, H. Li, Y. Liang, R. Hu, N. Liang, W. Fan, D. Cao, L. Yi and J. Xia, "GC-MS Based Serum Metabolomic Analysis of Isoflurane-Induced Postoperative Cognitive Disfunctional Rats: Biomarker Screening and Insight into Possible Pathogenesis", Springer-Verlag, Vol. 75, p. 799-808, 2012.

[39] H. Li, Y. Liang, Q. Xu and D. Cao, "Key wavelengths screening using competitive adaptive reweighted sampling method for multivariate calibration", Analytica Chimica Acta, Vol. 648, p. 77-84, 2009.

[40] E. Anderson, "Monte Carlo Methods and Importance Sampling," Lecture Notes from Stat 578C Statistical Genetics, 1999. [Online]. Available: http://ib.berkeley.edu/labs/slatkin/eriq/classes/guest_lect/mc_lecture_notes.pdf. [Accessed 8 March 2013].

[41] O. M. Kvalheim, "Interpretation of partial least squares regression models by means of target projection and selectivity ratio plots", Journal of Chemometrics, Vol. 24, p. 496-504, 2010.

[42] D. Chapman, "Selecting Unrestricted and Simple Random with Replacement Samples Using Base SAS® and PROC SURVEYSELECT", SAS Global Forum 2012.

Statistics and Data Analysis, Vol. 347, 2012.

[43] P. Refaeilzadeh, L. Tang and H. Liu, "Cross Validation," in *Encyclopedia of Database Systems*, Springer Science+Business Media, 2009.

[44] H. Li, Y. Liang, Q. Xu and D. Cao, "Model population analysis for variable selection", Journal of Chemometrics, John Wiley & Sons, Ltd, Vol. 24, p. 418-423, 2010.

[45] H. Ishwaran, "Variable importance in binary regression trees and forests", Electronic Journal of Statistics, Vol. 1, p. 519-537, 2007.

[46] T. MacFarland, "Mann-Whitney U-Test," 1998. [Online]. Available: http://www.nyx.net/~tmacfarl/STAT_TUT/mann_whi.ssi. [Accessed 27 November 2012].

[47] R. Shier, "Statistics: 2.3 The Mann-Whitney U Test," Mathematics Learning Support Centre, 2004. [Online]. Available: http://mlsc.lboro.ac.uk/resources/statistics/Mannwhitney.pdf. [Accessed 27 November 2012].

[48] L. Breiman, "Random Forests", Berkeley, CA, USA: Statistics Department, University of California, 2001.

[49] P. Tan, M. Steinbach and V. Kumar, "Introduction to Data Mining, Pearson International Edition", Boston, MA, U.S.A.: Addison Wesley-Longman Inc., ISBN 0321321367, 2006.

[50] L. Klappenbach, "Monotremes," About.com, [Online]. Available: http://animals.about.com/od/monotremes/p/monotremes.htm. [Accessed 15 May 2013].

[51] V. Svetnik, A. Liaw, C. Tong, J. Culberson, R. Sheridan and B. Feuston, "Random Forest: A Classification and Regression Tool for Compound Classification and QSAR

Modeling", Journal of Chemometrics Informatics Computational Sciences, Vol. 43, No. 6, p. 1947-1958, 2003.

[52] W. Tong, H. Hong, H. Fang, Q. Xie and R. Perkins, "Decision Forest: Combining the Predictions of Multiple Independent Decision Tree Models", Journal of Chemical Information and Computer Sciences, Vol. 43, No. 2, p. 525-531, 2003.

[53] T. Dietterich, "Ensemble Learning," in *The Handbook of Brain Theory and Neural Networks, Second Edition*, Cambridge, MA, USA, The MIT Press, 2002.

[54] P. Cugnet, "Confidence Interval Estimation for Distribution Systems Power Consumption by using the Bootstrap Method", Master thesis, Blacksburg, VA, USA: Virigina Polytechnic Institute and State University, 1997.

[55] R. Timofeev, "Classification and Regression Trees (CART): Theory and Applications", Master Thesis, Berlin, Germany: Humboldt University of Berlin, 2004.

[56] M. Sandri and P. Zuccolotto, "Variable Selection Using Random Forests", Brescia, Italy: Universita di Brescia.

[57] L. Breiman, "Wald Lecture II: Looking Inside the Black Box," [Online]. Available: http://oz.berkeley.edu/users/breiman/wald2002-2.pdf. [Accessed 13 March 2013].

[58] P. Bartlett, Y. Freund, W. Lee and R. Schapire, "Boosting the margin: a new explanation for the effectiveness of voting methods", Annals of Statistics, Vol. 26, No. 5, p. 1651-1686, 1998.

[59] "Gas Chromatography-Mass Spectrometry," Koninklijke Philips Electronics N.V., 2009. [Online]. Available: http://www.innovationservices.philips.com/sites/default/files/materials-analysis-gcms.pdf. [Accessed 10 April 2013].

[60] "Technologies-Gas Chromatography/Mass Spectrometry (GC/MS)," Smiths Detection, 2011. [Online]. Available: http://www.smithsdetection.com/GC_MS.php.

[Accessed 10 April 2013].

[61] R. Hites, "Gas Chromatography Mass Spectrometry," in *Handbook of Instrumental Techniques for Analytical Chemistry*, Prentice Hall, PTR, 1997.

[62] L. Ettre, "Nomenclature for Chromatography (IUPAC Recommendations 1993)", Pure and Applied Chemistry, Vol. 65, No. 4, 1993.

[63] D. Skoog, J. Holler and S. Crouch, "Principles of Intrumental Analysis, Sixth Edition", Thomson Brooks/Cole, ISBN-13 978-0-495-01201-6, 2007.

[64] J. Da Silva, "Introduction to Chromatographic Techniques" Lecture, University of Algarve, Faro, Portugal: Erasmus Mundus Master in Quality in Analytical Laboratories, 7-10 November, 2011.

[65] K. Yip, "Types of Chromatography," Rensselaer Polytechnical Institute (RPI), 1997. [Online]. Available: http://www.rpi.edu/dept/chem-eng/Biotech-Environ/CHROMO/be_types.htm. [Accessed 12 April 2013].

[66] "Gas Chromatography," NMSU Board of Regents, 2012. [Online]. Available: http://web.nmsu.edu/~kburke/Instrumentation/GC.html. [Accessed 12 April 2013].

[67] J. Da Silva, "Gas Chromatography" Lecture, University of Algarve, Faro, Portugal: Erasmus Mundus Master in Quality in Analytical Laboratories, 19-22 December, 2011.

[68] Clariant Analytical Services, "Technical Sheet 011: GC/MS Coupling-(Gaschromatography-mass spectrometry)," [Online]. Available: http://www.clariant.com/C125702E0040AEE5/vwLookupDownloads/CLAS_TS_011_1e.pdf/$FILE/CLAS_TS_011_1e.pdf. [Accessed 20 April 2013].

[69] J. Fenn, M. Mann, C. Meng, S. Wong and C. Whitehouse, "Electrospray ionization for mass spectrometry of large biomolecules", Science, Vol. 246, No. 4926, p. 64-71, 1989.

[70] J. Da Silva, "Mass Spectrometry" Lecture, University of Algarve, Faro, Portugal: Erasmus Mundus Master in Quality in Analytical Laboratories, 13-16 February, 2011.

[71] K. Spagou, G. Theodoridis, I. Wilson, N. Raikos, P. Greaves, R. Edwards, B. Nolan and M. Klapa, "A GC-MS metabolic profiling study of plasma samples of mice and low- and high-fat diets", Journal of Chromatography B, Vol. 879, No. 17-18, p. 1467-1475, 2011.

[72] Y. Sakamoto, K. Nakagawa, S.Kawana, N. Lingga, H. L. Chin, H. Miyagawa and T. Ogura, "Profiling of Japanese Green Tea Metabolites by GC-MS, GC/MS Technical Report No. 1, GC/MS Metabolomics & Life Sceince Project," Shimadzu Corporation, Tokyo, Japan, 2010.

[73] B. Tan, Y. Liang, L. Yi, H. Li, Z. Zhou, X. Ji and J. Deng, "Identification of free fatty acids profiling of type 2 diabetes mellitus and exploring possible biomarkers by GC-MS coupled with chemometrics", Springer Science+Business Media, Vol. 6, p. 219-228, 2009.

[74] The Expert Committee on the Diagnosis and Classification of Diabetes Mellitus, "Report of the Expert Committee on the Diagnosis and Classification of Diabetes Mellitus", Diabetes Care, Vol. 20, No. 7, p. 1183-1197, 1997.

[75] L. Yi, J. He, Y. Liang, D. Yuan, H. Gao and H. Zhou, "Simultaneously quantitative measurement of comprehensive profiles of esterified and non-esterified fatty acid in plasma of type 2 diabetic patients", Chemistry and Physics of Lipids, Vol. 150, No. 2, p. 204-216, 2007.

[76] J. Hudetz, Z. Iqbal, S. Gandhi, K. Patterson, T. Hyde, D. Reddy, A. Hudetz and D. Warltier, "Postoperative Cognitive Dysfunction in Older Patients with a History of Alcohol Abuse", Anesthesiology, Vol. 106, No. 3, 2007.

[77] J. Crawley, "What's Wrong With My Mouse?: Behavioral Phenotyping of Transgenic and Knockout Mice, Second Edition", Hoboken, NJ, USA: John Wiley & Sons, Inc.,

ISBN 978-0-471-47192-9, 2007.

[78] "Cognition-Spontaneous Alternation," PsychoGenics, [Online]. Available: http://www.psychogenics.com/spontaneousalternation.html. [Accessed 15 May 2013].

[79] M. Zeng, Y. Liang, H. Li, M. Wang, B. Wang, X. Chen, N. Zhou, D. Cao and J. Wu, "Plasma metabolic fingerprinting of childhood obesity by GC/MS in conjunction with multivariate statistical analysis", Journal of Pharmaceutical and Biomedical Analysis, Vol. 52, p. 265-272, 2010.

[80] M. Zeng, Z. Che, Y. Liang, B. Wang, X. Chen, H. Li, J. Deng and Z. Zhou, "GC-MS Based Plasma Metabolic Profiling of Type 2 Diabetes Mellitus", Chromatographia, Vol. 69, p. 941-948, 2009.

[81] "spa2010, Subwindow Permutation Analysis for identifying informative variables in OMICS study," Google Project Housing, [Online]. Available: http://code.google.com/p/spa2010/downloads/list. [Accessed 17 September 2012].

[82] M. Cowles and C. Davies, "On the Origins of the .05 Level of Statistical Significance", American Psychologist, Vol. 37, No.5, p. 553-558, 1982.

[83] G. Dallal, "Why P=0.05?," Jerry Dallal, 2012. [Online]. Available: http://www.jerrydallal.com/LHSP/p05.htm. [Accessed 17 April 2013].

[84] "TreeBagger," The MathWorks, Inc., [Online]. Available: http://www.mathworks.se/help/stats/treebagger.html. [Accessed 17 April 2013].

[85] "Model Building: Plotting Scores," Eigenvector Research Inc., 2012. [Online]. Available: http://wiki.eigenvector.com/index.php?title=Model_Building:_Plotting_Scores. [Accessed 18 April 2013].

[86] J. Trygg and S. Wold, "Orthogonal projections to latent structures (O-PLS)", Journal of Chemometrics, Vol. 16, p. 119-128, 2002.

[87] G. Derringer and R. Suich, "Simultaneous Optimization of Several Response Variables", Journal of Quality Technology, Vol. 12, No. 4, p. 214-219, 1980.

[88] P. Gatza and R. McMillan, "The Use of Experimental Desgin and Computerized Data Analysis in Elastomer Development Studies", Cincinnati, OH, USA: Division of Rubber Chemistry, American Chemical Society Fall Meeting, Paper No. 6, 1972.

[89] M. Hadjmohammadi and S. Nazari, "Optimization of Separation of Flavonoids using Experimental Design and Derringer's Desirability Function", Journal of Liquid Chromatography & Related Technologies, Vol. 36, No. 7, p. 943-957, 2013.

[90] Z. Li, B. Cho and B. Melloy, "Quality by Design Studies on Multi-response Pharmaceutical Formulation Modeling and Optimization", Journal of Pharmaceutical Innovation, Vol. 8, No. 1, p. 28-44, 2013.

[91] E. J. Harrington, "The Desirability Function", Industrial Quality Control, Vol. 21, No. 10, p. 494-498, 1965.

[92] M. Bilodeau and D. Brenner, "Theory of Multivariate Statistics", New York, NY, USA: Springer-Verlag, ISBN 0-387-98739-9, 1999.

[93] Pattern Recognition Systems AS, "New Features in Sirius 6.5," Pattern Recognition Systems AS, [Online]. Available: http://www.stjapan.co.jp/053_prs/Brochure/New_Features_Sirius_6_5.pdf. [Accessed 16 May 2013].

[94] D. Livingstone, "A Practical Guide to Scientific Data Analysis", Chichester, West Sussex, UK: John Wiley & Sons, Ltd., ISBN 978-0-470-85153-1, 2009.

[95] "Global Health Observatory (GHO), Obesity," World Health Organization (WHO), 2013. [Online]. Available: http://www.who.int/gho/ncd/risk_factors/obesity_text/en/. [Accessed 25 April 2013].

[96] B. Bresnahan and E. Saab, "Obesity," Medical College of Wisconsin, 2013. [Online].

Available:
http://www.mcw.edu/Nephrology/ClinicalServices/OverweightorObese.htm.
[Accessed 25 April 2013].

[97] "Diabetes and Metabolism," Diabetes.co.uk-The Global Diabetes Community, [Online]. Available: http://www.diabetes.co.uk/diabetes-and-metabolism.html. [Accessed 26 April 2013].

[98] M. Kittleson, R. Irizarri, B. Heidecker and J. Hare, "Chapter 12: Transcriptomics: Translation of Global Expression Analysis to Genomic Medicine," in *Genomic and Personalized Medicine*, San Diego, CA, USA, Elsevier Inc., ISBN 978-012-370888-5, 2009.

[99] P. Verhagen, "Case Studies in Archaelogical Predictive Modelling", Leiden, Netherlands: Leiden University Press, ISBN 978 90 8728 007 9, 2007.