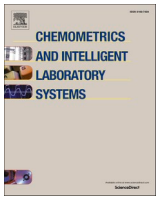
Chemometrics and Intelligent Laboratory Systems 247 (2024) 105096

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics



ProSpecTool: A MATLAB toolbox for spectral preprocessing selection

Jokin Ezenarro *, Daniel Schorn-García, Olga Busto, Ricard Boqué

Universitat Rovira i Virgili. Chemometrics and Sensorics for Analytical Solutions (CHEMOSENS) group, Department of Analytical Chemistry and Organic Chemistry, Campus Sescelades, Edifici N4, C/Marcel·lí Domingo s/n, Tarragona, 43007, Spain

ARTICLE INFO

Keywords:
Partial least squares regression (PLSR)
Automatic
Near-infrared (NIR)
Mid-infrared (MIR)
Raman
UV–Visible

ABSTRACT

This paper introduces the ProSpecTool, a MATLAB toolbox for automated selection of preprocessing methods for obtaining optimal PLS regression models in vibrational spectroscopy. Trial-and-error approaches for preprocessing can be time-consuming, and the success of the process relies on the experience of analysts. The ProSpecTool addresses this challenge by using objective criteria analogous to expert judgment to filter and iterate preprocessing methods based on raw data properties. The toolbox quantifies noise, identifies multiplicative and additive scatter-effects, and selects preprocessing algorithms to correct them. Results demonstrate that the ProSpecTool can produce models that resemble those proposed by experienced analysts based on trial-and-error in terms of performance and robustness, making it a valuable exploratory tool for vibrational spectroscopy practitioners.

1. Introduction

Multivariate data analysis techniques are almost always needed when using vibrational spectroscopy, to get as much information as possible from the data obtained [1]. For instance, Principal Component Analysis (PCA) or Partial Least Squares (PLS) algorithms help the analyst get useful insights about the properties of the data, that *a priori* are not visible in the raw spectra. After collecting the data, the aim of multivariate data preprocessing is to remove unwanted variation in the spectra, such as baseline shifts and scatter-effects [2,3]. Ideally, the preprocessing should only remove unwanted variation, leaving the information of interest unchanged. Unwanted variation is the systematic or random variation present in the descriptor matrix **X**, which is linearly independent of the chemical variation or orthogonal to the response matrix **Y**. These variations introduce modelling problems for the subsequent multivariate projection and regression methods, which may negatively affect model prediction performance or interpretability. Unless there exists a well-established rationale for choosing the appropriate preprocessing steps, trial-and-error approaches are often common practice for deciding which method should be applied for removing or reducing the influence of unwanted variation [4]. This requires prior knowledge of the data analyst of the obtained signal, to evaluate the pre-processed signal or the model performance.

Commercially available software like PLS_Toolbox 9.0 (Eigenvector Research Inc., Manson, USA) can apply all the preprocessing algorithms and their combinations selected by the analyst in an iterative way, and then apply the multivariate data analysis techniques to each of the new datasets generated. The most common way of judging whether a preprocessing method is beneficial for the analytical performance is to compute the prediction uncertainty for an independent test set, the Root Mean Square Error (RMSE), and then select the preprocessing method that gives the lowest RMSE. But as datasets get bigger and the preprocessing and analysis algorithms get more complex, the computation time of iterating all the available options becomes considerably longer and often impractical [5].

Methods that optimise the choice of the preprocessing technique have been proposed using various strategies, such as Design of Experiments [6], orthogonalisation (SPORT) [7], or spectral signal-to-error ratio [5]. Instead, the method proposed in the present article filters the preprocessing algorithms based on the properties of the raw data, creating rules based on criteria that an expert analyst would use but in an automated and objective way, avoiding the need to iterate all options. Using this method, the preprocessing techniques that could improve the characteristics of the raw data are automatically considered and iterated to ideally find the best PLS regression models.

For this purpose, a toolbox to be used in MATLAB (Mathworks Inc., Natick, MA, USA) is presented. The theoretical basis of the applied methodologies is explained, the features and modules are presented, and finally, an illustrative example of how the toolbox works is shown using real data.

* Corresponding author.
E-mail address: jokin.ezenarro@urv.cat (J. Ezenarro).

https://doi.org/10.1016/j.chemolab.2024.105096
Received 22 December 2023; Received in revised form 22 February 2024; Accepted 23 February 2024
Available online 26 February 2024

2. Methodological background

2.1. Partial Least Squares Regression

The spectral measurements (absorbances or intensities registered at different wavelengths) for a set of samples constitute a set of independent variables or predictors that form what is called the **X**-block. The variable that has to be predicted (i.e., a physicochemical property measured by a reference method) is the dependent variable or predictand and constitutes the **Y**-block [8]. Partial Least Squares Regression (PLSR) is one of the most important algorithms in multivariate data analysis to find the correlation between these blocks and building a regression model. This is due to its simplicity, versatility and applicability, as it can fit multiple response variables in a single model and makes interpretation of results more intuitive. This method is based on finding new dimensions (factors) from the original data by maximising the covariance of the **X** and **Y** blocks and using them to build a regression equation [9]. A PLSR model is described as,

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{T} + \mathbf{E} \quad \text{(Eq. 1)}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{T} + \mathbf{F} \quad \text{(Eq. 2)}$$

where **X** and **Y** are the predictor and predictand blocks, respectively; **T** and **U** are projections or score matrices of the **X** and **Y** blocks, respectively; **P** and **Q** are orthogonal loading matrices; and matrices **E** and **F** are the error terms, assumed to be independent and identically distributed random normal variables. Then, a regression vector is needed to correlate the information of **X** and **Y**, which is calculated as:

$$\mathbf{U} = \mathbf{T}\mathbf{B}_{PLS} + \mathbf{H} \quad \text{(Eq. 3)}$$

where $\mathbf{B}_{PLS}$ is the PLS coefficients matrix that relate **X** and **Y** through their scores (**T** and **U**, respectively), and **H** is the residual matrix of the correlation [9]. In the present article, the SIMPLS algorithm is used to compute the described equations [10].

2.2. Spectra preprocessing

There are many algorithms for spectral preprocessing, which serve different purposes, that is, they try to minimise the effect of different types of distortions on the data. To find out what types of algorithms should be taken into consideration, the existence or magnitude of each type of distortion must be determined [2,3,11,12].

2.2.1. Smoothing

Smoothing algorithms intend to maximise the signal-to-noise ratio by keeping only the information related to the sample and removing the noise related to the measurement process. In this article three smoothing algorithms are considered: Gaussian, Savitzky-Golay (SG) and Wavelet denoising.

Gaussian smoothing is a simple yet powerful method for smoothing spectral data. This method assumes that the peaks in a spectrum follow a Gaussian curve, so it approximates each point to the maximum point of a Gaussian peak, weighting the surrounding values with respect to the corresponding height of the peak [13].

Savitzky-Golay smoothing is one of the most used preprocessing methods, which was popularised by Savitzky and Golay [14] and can include a derivation step. In order to find the smoothed value at a central point *i*, a polynomial is fitted to a symmetric window of the raw spectrum. Once the parameters of this polynomial have been calculated, the value of the central point can then be estimated using the obtained equation. This operation is applied sequentially to all points in the spectrum by moving the window. The number of points used to calculate the polynomial (window size) and the degree of the fitted polynomial are both decisions that need to be made.

The right choice of the window size is crucial, and it is far from trivial to do this correctly. Too small a window will lead to the introduction of large artifacts in the corrected spectra and to a reduced signal-to-noise

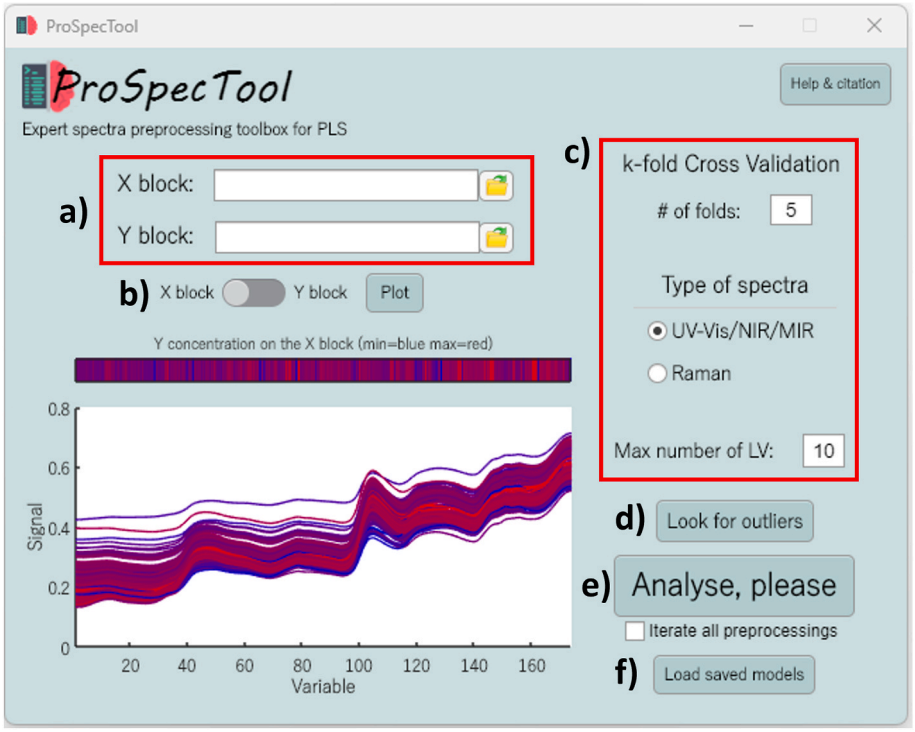


Fig. 1. Main window of the ProSpecTool toolbox. a) Input of **X**- and **Y**-blocks. b) Axis where the imported data can be plotted for a preliminary check. c) Settings of the PLSR models to be built: the number of folds to do in the random subsets cross validation, the maximum number of LVs and the type of data that is being analysed (this will affect the smoothing algorithms that will be tried). d) Button that leads to the outlier detection window. e) Button that leads to the preprocessing analysis. f) If the toolbox has been used before and models have been saved, results can be opened here.

ratio. On the contrary, a too large window may smooth out small peaks that contain relevant information. Additionally, when using the Savitzky-Golay algorithm, the first and last variables corresponding to half of the window-size are lost, as they are usually not extrapolated.

Wavelet denoising is a smoothing algorithm based on the mathematical theory of wavelets [15]. This algorithm decomposes the spectroscopic signal into different frequency bands using a wavelet transform, this is, it divides the signal into a series of wavelet coefficients, which represent the signal's energy at different spectral ranges. The decomposition allows for a more detailed analysis of the signal, as different wavelet coefficients capture different frequency components. Then, the wavelet coefficients are thresholded, which selectively removes coefficients that are considered to be noise-based on their magnitude while retaining the essential signal information. After thresholding, the denoised signal is reconstructed by performing an inverse wavelet transform, which combines the retained wavelet coefficients to recreate the original signal. The reconstructed signal represents an approximation of the underlying noise-free signal.

Wavelet denoising allows for better discrimination between signal and noise components at different scales. It also preserves sharp features in the signal while effectively removing noise, which is particularly beneficial for preserving spectral details in techniques such as Raman spectroscopy [16].

2.2.2. Derivatives

Derivatives have the capability of removing both additive and multiplicative effects in the spectra and have been used in analytical spectroscopy for decades. The first derivative removes only the baseline offset; the second derivative removes both baseline constant and linear trends. In this article two different methods are considered: differentiation and Savitzky-Golay [14].

The most basic method for derivation is the differential or finite differences: the first derivative is estimated as the difference between two subsequent spectral measurement points; the second order derivative is then estimated by calculating the difference between two successive points of the first-order derivative spectrum:

$$x_j^{'} = x_j - x_{j-1} \quad \text{(Eq. 4)}$$

$$x_j^{''} = x_j^{'} - x_{j-1}^{'} = x_{j-1} - 2{\cdot}x_j + x_{j+1} \quad \text{(Eq. 5)}$$

where x'_j denotes the first derivative and x''_j the second derivative at point (wavelength) j. This method is extremely simple, but it must be used with caution as most real measurements have a considerable amount of noise, so a previous smoothing step might be necessary. When this step is added the method is known as Norris-Williams derivation [2].

S-G derivative follows the same process as the S-G smoothing explained in section 2.2.1 with an additional step. After fitting the polynomial to a window around a moving point, the derivative of any order of this function can easily be found analytically and this value is subsequently used as the derivative estimate for this central point. The highest derivative that can be determined depends on the degree of the polynomial used during the fitting (i.e., a third-order polynomial can be used to estimate up to the third-order derivative).

2.2.3. Sample-wise normalisation/scatter correction

Light scattering-effects, temperature changes and other experimental or instrumental variations in the measurement can cause changes in the spectral baseline, which may induce unwanted errors in the subsequent regression models. These baseline shift trends are generally linear, but they can also be curvilinear or higher in specific areas. For example, the dominant feature of NIR diffuse reflectance spectra is the increasing level of the reflectance values over the range 1100–2500 nm. For removing this kind of unwanted variation several algorithms have been proposed, from which the most used ones are considered. Under the name of scatter-correction methods, which are techniques that are designed to reduce the (physical) variability between samples due to scattering effects, we consider three preprocessing concepts: Detrending, Baseline Correction (BC) and Standard Normal variate (SNV). Multiplicative Scatter Correction (MSC) is a widely used method known to introduce a calibration-set dependency, which is not managed in this toolbox. However, in cases of highly varying multiplicative scattering, it remains a valuable option for practitioners to consider [17].

Detrending focuses on removing systematic trends or drifts from the data, which can arise from instrument limitations or environmental factors. It involves polynomial fitting of the data and subtracting the trend component to enhance the visibility of analyte-specific information.

Asymmetric least-squares baseline correction is a specific approach used to correct the baseline when the low-frequency variations are asymmetric or skewed [18]. It is particularly useful when the baseline has a gradual slope or exhibits asymmetry due to factors like instrument drift, solvent effects, or other systematic errors.

The linear slope of the SNV corrected spectrum of a given material is near constant, while the curvature varies with particle size and packing density. Most solids usually show absorbance spectra which are more intense at the longer wavelengths as the combination bands become more probable when the wavelength region approaches the fundamental vibration. SNV preprocessing is probably the most applied method for scatter correction. For a given sample i, the SNV corrected spectrum is calculated as

$$\mathbf{x}_i^* = \frac{\mathbf{x}_i - \bar{x}_i}{s_i} \quad \text{(Eq. 6)}$$

where $\bar{x}_i$ is the mean value of spectrum i ($\mathbf{x}_i$), s_i is standard deviation of the spectrum i, and $\mathbf{x}_i^*$ is the SNV normalised spectrum of sample i. Since SNV does not involve a least squares fitting in the estimation of its parameters, it can be sensitive to noisy values in the spectrum [19].

2.2.4. Variable-wise normalisation

Mean Centring (MC) is the most used variable-normalisation method for spectroscopic data: the mean of each data column (variable) is subtracted from all the values in that column to give a data matrix where the mean of each preprocessed variable is zero. However, sometimes autoscaling the variables might provide better results. In autoscaling (variance scaling or column standardisation), after mean centring each variable, the values in each column are divided by the standard deviation of the column, resulting in a matrix where all the columns have mean zero and unit variance. This means that the only information left is related to the correlations among the variables, independently of their magnitude. For a given sample i, an autoscaled spectrum is calculated as

$$x_{i,j}^* = \frac{x_{i,j} - \bar{x}_j}{s_j} \quad \text{(Eq. 7)}$$

where $\bar{x}_j$ is the mean value of spectra at variable j, s_j is standard deviation of the spectra at variable j, and $x_{i,j}^*$ is the value of the autoscaled spectrum of sample i at variable j. By doing this, large, highly variable features decrease in importance and features with low variability become more visible. Autoscaling emphasizes low variability features and this may be especially useful when the parameter to be predicted is related to small spectral bands overlapped by those of major components in the sample. However, enhancing the small contributions also enhances noisy variables so care must be taken using variance scaling. Autoscaling may also difficult the interpretation of the final results [20].

2.3. Model performance evaluation metrics

A plethora of metrics have been proposed to evaluate the performance of multivariate regression models. However, usually more than one metric is calculated and studied to assess the performance of such

models in a more comprehensive way [21–24]. In the present work, four metrics are calculated and shown for each PLSR model, as they provide insights about different properties of the models.

2.3.1. Root Mean Square Error of cross-validation ($RMSE_{CV}$)

Root Mean Square Error (RMSE) is a widely used metric for assessing the performance of regression models, where the goal is to predict continuous numerical values, as it estimates the error of a model. RMSE is calculated as,

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \quad \text{(Eq. 8)}$$

where y_i is the value predicted by the model for the sample with a measured value of $\hat{y}_i$ and N is the number of samples. Squaring the differences ensures that errors are positive and penalises larger errors more heavily than smaller ones, making the metric more sensitive to significant deviations between predictions and actual values [23].

RMSE is an especially valuable tool in cross-validation as it provides a reliable measure of how well the model generalises to new and unseen data, allowing analysts to compare and select the best-performing model for their specific task. In cross-validation, the dataset is divided into several subsets, a model is trained on a combination of these folds and evaluated on the remaining ones, and the process is repeated until all subsets have been used. Then the individual RMSE values are averaged to obtain the $RMSE_{CV}$. A lower $RMSE_{CV}$ indicates better predictive performance, as it represents the average magnitude of the errors made by the model across all the folds.

There are several metrics based on the prediction error that aim to relativise it for an easier comparison between models based on the properties of the **Y**-block, such as the Ratio of Performance to Deviation (RPD) or the Ratio of Error to Range (RER) [25]. However, for the application described in this article, as PLS regression models with exactly the same **Y**-block (and therefore same properties) are compared, there is no need of relativising these metrics and $RMSE_{CV}$ is shown.

2.3.2. Determination coefficient of prediction vs. measured values

The Determination Coefficient (R^2) is a statistical measure used to assess the goodness of fit of a regression model. In the context of prediction versus measured values, R^2 is used to evaluate how well the predictions made by the regression model align with the actual measured values. The determination coefficient is calculated as,

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} \quad \text{(Eq. 9)}$$

where SS_{RES} is the sum of squares of the model's residuals and SS_{TOT} is the total sum of squares of the data (proportional to its variance). It provides an indication of how much variability in the measured values is captured by the model's predictions. It ranges from 0 to 1, a value of 0 indicating that the model fails to explain any variance in the measured data, and a value of 1 meaning that the model perfectly predicts the measured values without any error [23].

However, it is essential to be cautious with this metric, as a high R^2 can be achieved by overfitting the model to the training data, leading to poor performance on new, unseen data. This is why R^2 of cross validation is calculated, in an analogous way to $RMSE_{CV}$, obtaining a metric that provides valuable insights into the predictive ability of the model, to make informed decisions about the performance and generalisation capabilities of the model.

2.3.3. Regression vector noise index

The noise of the Regression Vector (RV) of a PLSR model can be defined as the variability that is not related to the chemical and/or physical properties of the samples that affect the spectra used to build the model. This is reflected in the RV of the model when the model is explaining the variance related to the noise instead of the information of interest. It is easy to see when working with continuous vectors like those obtained from spectra because noise causes the RV to lose its continuity. This noise is an indicator of the level of overfitting of the model and therefore, of its lack of suitability. The Regression Vector Noise Index (NI_{RV}) is calculated as,

$$\mathrm{NI_{RV}} = \frac{\sum |\mathbf{RV} - \mathbf{RV}_{\mathbf{smoothed}}|}{\sum |\mathbf{RV}|} = \frac{\sum |\mathbf{Noise\ Vector}|}{\sum |\mathbf{RV}|} \quad \text{(Eq. 10)}$$

where $\mathbf{RV}_{\mathrm{smoothed}}$ is the **RV** smoothed by using a Gaussian filter, which is an idealisation of how the vector should really be if there was no noise [26]. The residual between the **RV** and $\mathbf{RV}_{\mathrm{smoothed}}$ is calculated, summed up and, finally, relativised by dividing by the summatory of the absolute coefficients of the original **RV** [21].

2.3.4. J-Score

The J-Score is an indicator used to evaluate the performance of a PLSR model in a more global and comprehensive way than just using prediction or cross-validation errors and other supplementary descriptors, plots and other metrics, that the analyst must evaluate. The J-Score is calculated as

$$\mathrm{J-Score} = \left(\frac{\mathrm{RMSE_{CV}}}{s_Y} + 1 - \frac{\mathrm{RMSE_{Cal}}}{\mathrm{RMSE_{CV}}} + \mathrm{NI_{RV}}\right) / 3 \quad \text{(Eq. 11)}$$

where $RMSE_{Cal}$ and $RMSE_{CV}$ are the Root Mean Square Error of Calibration and Cross-Validation, respectively, S_Y is the standard deviation of the reference values (**Y**-block) and NI_{RV} is the Noise Index of the Regression Vector. The first term of Eq. (11) accounts for the prediction error, which is usually the most revised metric; the second term accounts for the overfitting of the model; and the third term accounts for the robustness of the model, based on the noisiness of the regression vector. An arithmetic mean of these terms offers a metric that includes the most important qualities of a regression model: the lower the J-Score, the better the model [21].

This metric can help an analyst compare spectroscopic regression models in an objective and comprehensive way. In the implemented version of the software, the dimensionality (number of LVs) of each model is decided after calculating the J-Score for each one and selecting the one with the absolute minimum value; the models can also be compared using this metric [21]. It is noteworthy that the models fitted to the datasets after different preprocessings may or may not have the same optimal dimensionality, and for selecting the optimal preprocessing method the models with optimal dimensionality have to be compared between them.

3. Main features of the ProSpecTool toolbox for MATLAB

The collection of functions and algorithms included in the toolbox are provided as MATLAB source files, in a folder that needs to be added to the path. The ProSpecTool has no requirements for any other third-party utilities beyond the MATLAB installation and the MATLAB 'Signal Processing Toolbox', 'Wavelet Toolbox' (for Raman) and 'Statistics and Machine Learning Toolbox'. The toolbox was built on MATLAB 2021 (v9.11). The functions are not intended to be called on the MATLAB command window; the graphical user interface (Fig. 1) should be opened by typing "ProSpecTool", which enables the user to perform all the analysis steps in the correct order.

3.1. Input data

The **X**-block data must be structured as a numerical matrix with dimensions $I \times J$, where I is the number of samples and J the number of variables (wavelengths, wavenumbers, …). The **Y**-block data must be structured as a column numerical vector ($I \times 1$), where the element i of this vector represents the predictand value of the sample number i. If

replicates are present in the dataset, they should be averaged before starting the analysis, as a random cross-validation with replicates would offer over-optimistic results. Results can also be affected by other structures in the data, such as different classes and populations, which the models may overfit or under-represent. These situations must be addressed by the analyst before using the present toolbox.

3.2. Looking for outliers

For choosing the best preprocessing strategy, the input dataset should not contain any outstanding spectral outlier, nor an **Y**-block outlier. Including outliers may cause the 'best performing' models to be those with the preprocessing that bring the outliers closer to the sample population, leading to suboptimal models. And even if it is uncommon, the different preprocessing could also cause for some samples to become spectral outliers or to outstand more. Because of this, it is highly recommended to check for outliers in the final models.

If the user is not sure there are no outliers in the raw dataset, the "Look for outliers" option should be used in the main window, opening a new window (Fig. 2) showing the descriptive plots of a PLSR model fitted to the raw data. In this window the user can select the outlier samples based on different plots and statistical criteria (Hotelling T^2, Q residuals and/or score values). Once the outliers have been selected and removed the analysis of the data can be started.

3.3. Flowchart of the automatic spectra preprocessing iteration

The flowchart described in Fig. 3 is followed by the ProSpecTool for an objective analysis of the dataset, this is, to decide which preprocessing algorithms should be tried and then iterating all the sensible combinations.

3.3.1. Smoothing

The Gaussian and Savitzky-Golay (S-G) smoothing algorithms consider a moving window of spectral points to make a prediction on the central point, the size of the window (among other parameters) must be optimised. Usually this would be done by the analyst based on experience and trial-and-error, as the optimal number of points to consider depends on the type of spectroscopy, the spectral resolution, or the sample characteristics (shape of the spectrum). In the present work, to decide how many points to use for the smoothing process and consider all of the mentioned parameters, a new algorithm has been implemented: for each dataset, an averaged spectrum is calculated, then the first derivative (gap derivative) is applied to the average spectrum. The distance between the zeros is obtained (maximum and minimum points of the original spectra) and the mean distance between correlative zeros is calculated to obtain an indicator of the number of points that should be used in the smoothing algorithm, based on the method proposed by Ezenarro et al. (2023) [21].

Even if this is not necessarily the optimal number of points, it will be around this number, so the range of points to iterate is reduced significantly. This process ensures that in the smoothing process the real peaks of the spectrum will not be removed but the noise will be reduced.

3.3.2. Derivative

The datasets that have been already derived using the S-G method, will not be modified. The datasets that have not been derived will be duplicated and derived by first and second order differentiation. The preprocessed datasets obtained from this step will be added to the results of the previous steps.

3.3.3. Sample-wise normalisation

As there are many ways of normalising the spectrum of a sample

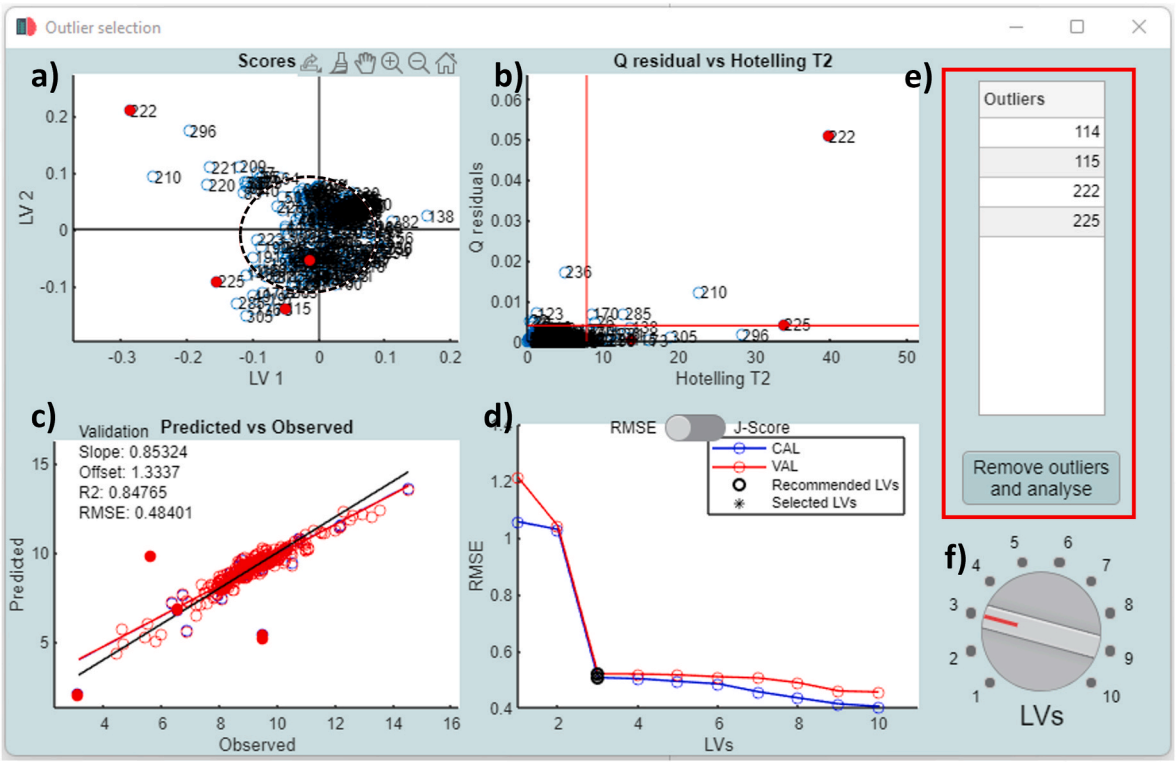


Fig. 2. Outlier selection window of the ProSpecTool toolbox, where a PLSR model is fitted to the raw data. a) Scores of LV 2 vs LV 1 with a confidence ellipse of 95 %. b) Q residuals vs. Hotelling's T^2 with 95 % confidence limits. c) Predicted values by the model vs the observed (reference) values of the **Y**-block. d) $RMSE_{Cal}$ and $RMSE_{CV}$ values of the model with different number of LVs; alternatively, it can be changed to J-Score values of the model with different number of LVs. e) Index of the samples selected as outliers in section b), which are filled in red in sections a), b) and c); and the option to start the preprocessing analysis without these samples. f) Number of LVs of the current model that can be changed if the user thinks it is necessary, the suggested/default number is selected based on the J-Score curve [21]. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

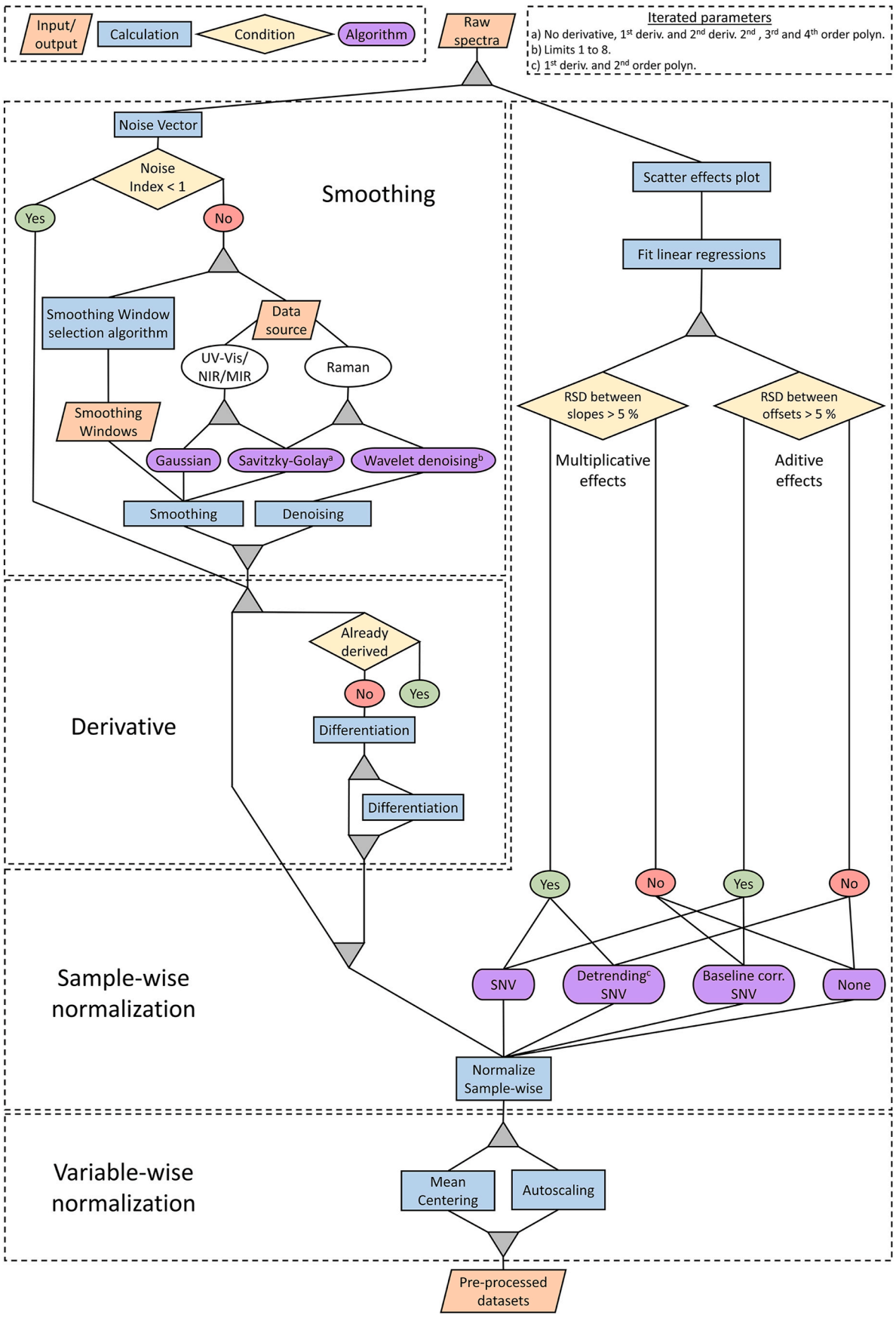


Fig. 3. Flowchart showing the process of choosing and applying the preprocessing algorithms that are considered that could improve the subsequent models.

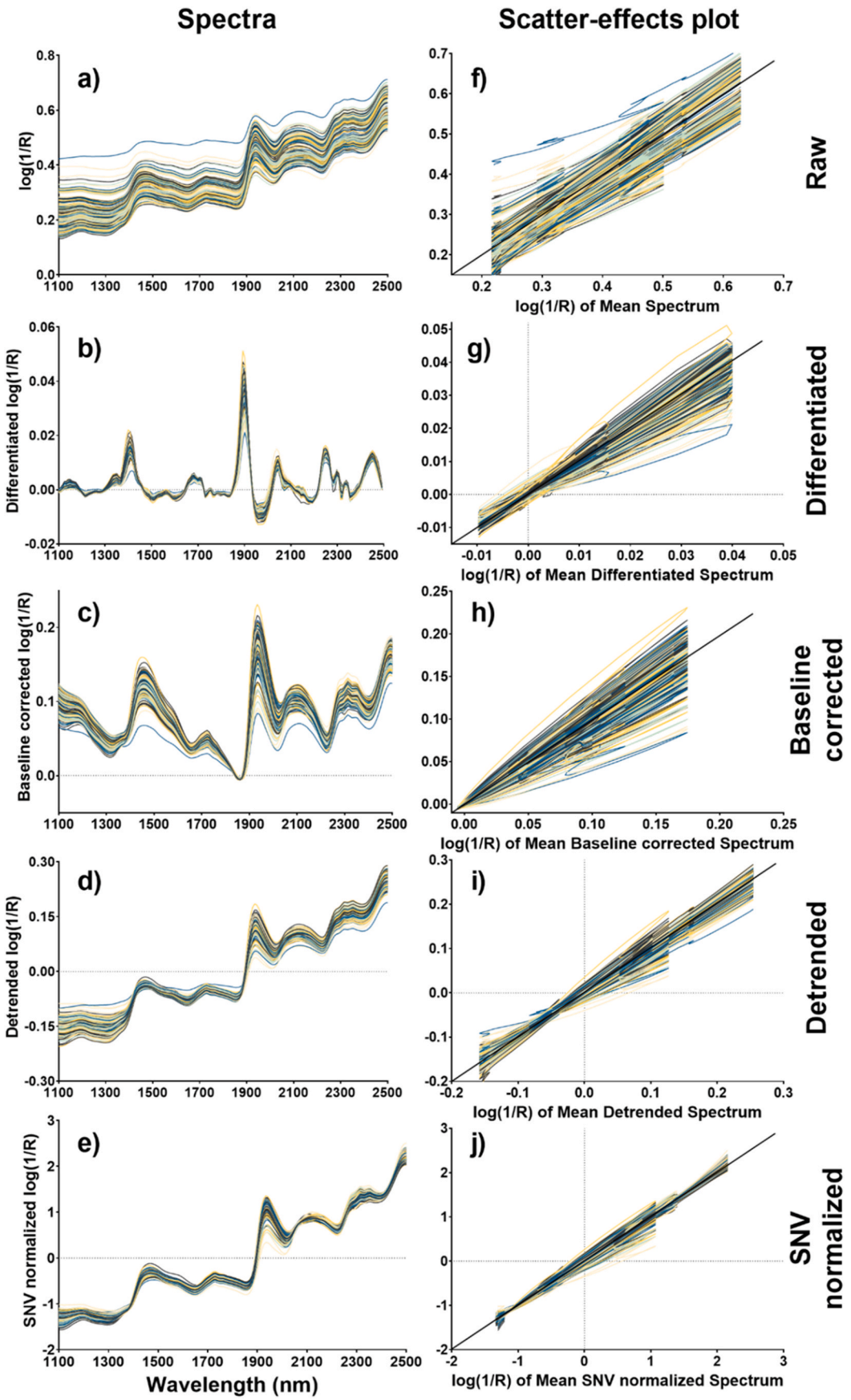


Fig. 4. Example of a NIR spectra dataset (Ruisánchez et al. (2002) [28]) with different preprocessings and their corresponding scatter-effects plot: a) and f) Raw; b) and g) Differentiated; c) and h) Baseline corrected; d) and i) Detrended; and e) and j) SNV normalised.

depending on the properties of the measurement method, the decision of which algorithms to iterate has been objectivised using quantitative values based on the scatter-effects plot. This plot is a widely used tool for the detection of multiplicative and additive scatter-effects as it helps the analyst to visualise this kind of effects on the spectral dataset [27]. This plot is built by representing each spectrum against the average spectrum of the dataset (raw or preprocessed), as it can be seen in Fig. 4, and can show in a visual way the structure of the scattering present in the spectra of a dataset.

If there are no scatter-effects the lines will be parallel and overlapping, only the peaks of the spectra getting out of the line (Fig. 4j). If there are additive effects the lines will be parallel but not overlapping (Fig. 4g); if there are multiplicative effects the lines will be overlapping at some point but not parallel (Fig. 4h and i); and if both types of effects are present the lines will not be parallel nor overlapping (Fig. 4f).

But in order to categorically evaluate the existence of the mentioned scatter-effects, they must be quantified. In the present paper, the above-mentioned scatter-effects plot has been used to achieve this: linear regression has been applied to each spectrum represented in the plot (Fig. 4f–j), obtaining a slope and an offset for each sample relative to the

Preprocessed PLS models

Model	Smoothing	S. Window	Pol. Order	Derivative	Normalisation	N. Pol. Order	Limit	Autoscale	Sugg. LVs (J-Score)	J-Score	RMSE-CV	R2-CV	RV Noise
1	No			0	No			No	3	0.2380	0.3718	0.9155	5.4223
2	No			0	SNV			No	4	0.2402	0.2950	0.9474	14.6721
3	Gaussian	7		0	SNV			No	4	0.1967	0.2986	0.9463	10.5416
4	Gaussian	9		0	SNV			No	4	0.1834	0.2962	0.9455	9.0242
5	Gaussian	11		0	SNV			No	5	0.1647	0.2827	0.9510	9.6240
6	Gaussian	13		0	SNV			No	5	0.1535	0.2999	0.9451	6.6864
7	Gaussian	15		0	SNV			No	5	0.1452	0.2983	0.9453	5.8349
8	S-G	7	2	0	SNV			No	4	0.2195	0.2797	0.9518	15.2428
9	S-G	9	2	0	SNV			No	5	0.2027	0.2812	0.9517	13.8527
10	S-G	11	2	0	SNV			No	4	0.1899	0.2816	0.9519	12.1329
11	S-G	13	2	0	SNV			No	5	0.1762	0.2886	0.9505	9.2872
12	S-G	15	2	0	SNV			No	4	0.1621	0.2902	0.9492	7.9696
13	S-G	7	3	0	SNV			No	4	0.2229	0.2932	0.9480	13.3115
14	S-G	9	3	0	SNV			No	4	0.2059	0.2799	0.9521	13.8527
15	S-G	11	3	0	SNV			No	4	0.1875	0.2824	0.9521	12.1329
16	S-G	13	3	0	SNV			No	5	0.1757	0.2855	0.9505	9.2872

Fig. 5. Table describing the PLSR models obtained from the datasets preprocessed in diverse ways. The first eight columns describe the preprocessing applied to the spectra and index the new datasets. The following column shows the suggested number of LVs for each dataset based on the J-Score curve, the dimensionality that is used to build the model and obtain the descriptive values of the following columns: J-Score value, Root Mean Square Error of Cross Validation, Determination coefficient of the Cross Validation predicted vs. observed values, and Noise Index of the Regression Vector (RV).

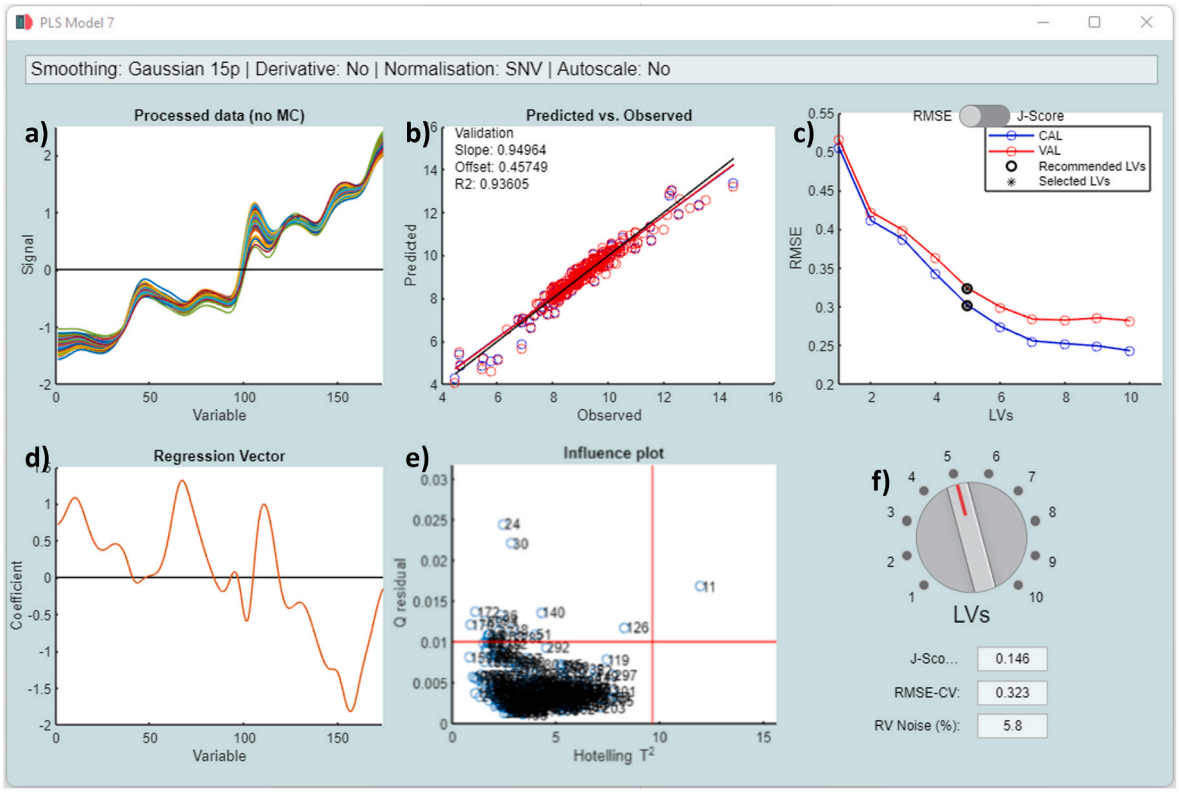


Fig. 6. Example of window of the descriptive plots of one of the PLSR models. a) Plot of the spectra used to build the model, after preprocessing (without the variable-wise normalisation step for better interpretability). b) Predicted vs observed (reference) values of the **Y**-block. d) $RMSE_{Cal}$ and $RMSE_{CV}$ values of the model with different number of LVs; alternatively, it can be changed to J-Score values of the model with different number of LVs. d) Regression Vector of the model. e) Q residuals vs. Hotelling's T^2 with 95 % confidence limits. f) Number of LVs of the current model, which can be changed if the user thinks it is necessary (the suggested/default number is selected based on the J-Score curve).

mean of the samples, in an analogous way to the first part of the MSC method (Fig. S1) [17].

For the determination of the magnitude of the multiplicative scatter-effects, the Relative Standard Deviation (RSD) between the obtained slopes is calculated, a normalised indicator that shows how different the slopes are. Instead, for the determination of the magnitude of additive effects, as the mean value is around zero by definition, the Standard Deviation (s) is normalised by the average signal intensity of the average spectrum of the dataset. A critical value of 5 % has been set to categorise both effects as present or not present and consequently preprocess (or not) the spectra using sample-wise normalisation methods.

In addition, as an example, in Fig. 4 it can be seen how a spectral outlier (blue spectrum, sample 222) behaves differently depending on the type of outlier and the applied preprocessing method. As explained in section 3.2, this can affect the resulting models and outliers should be removed before starting the preprocessing analysis.

3.3.4. Variable-wise normalisation

As there is no definitive evidence whether spectral data should or should not be autoscaled in every case, and no easy test can be performed to decide if the dataset should be autoscaled, both autoscaling and mean centring have been iterated for every preprocessing strategy.

3.4. PLSR model selection

When the iteration of the automatically selected preprocessing methods is finished and a PLSR model is fitted for each of the newly created datasets, a table will appear describing the models (Fig. 5).

The results can be ordered by any of the performance-evaluation parameters or filtered using the filtering tool that opens together with the table (Fig. S2). The models can be filtered based on the preprocessing, the descriptive properties or an automatic filter based on the J-Score to show only the best-performing models.

The user can inspect a specific model more in detail by clicking the index of this model (blue column in Fig. 5), which will make descriptive figures of that model appear in a new window (Fig. 6). It is encouraged to investigate the most adequate models after sorting or filtering, to determine why they work better than others or what they have in common, better understanding the preprocessing methods selected as optimal.

4. Illustrative example: selection of optimal preprocessing

Even if the toolbox has been tested using various datasets, the operation of the algorithm is illustrated and validated in this article by using the Forages dataset as an example, which is a real dataset for prediction described and used by Ruisánchez et al. (2002) [28]. This dataset consists of NIR spectra of 305 forage samples recorded under the same conditions by a specialised laboratory. Additionally, two parameters were measured: moisture at 103–105 °C and crude protein content. The original authors distributed the dataset to six participants, all of whom had previous knowledge and experience on multivariate calibration. The objective was to determine how different the results from several laboratories were when each one chose the most suitable preprocessing method, multivariate calibration method and software.

This dataset has been analysed using the ProSpecTool toolbox twice, one for moisture prediction and the other for protein prediction. The report of the dataset description showed that the spectra have a non-negligible amount of noise and have additive and multiplicative scatter-effects, as it can be deduced from Fig. 4f. Therefore, the iterated preprocessing steps with the ProSpecTool toolbox were the following: Gaussian smoothing, S-G smoothing and derivatives, differentiation, SNV normalisation, autoscaling and MC.

For humidity, the samples 114, 115 and 222 were detected as outliers and the analysis was conducted after removing them for all datasets. From the 134 PLSR models generated, the top one by J-Score was the one with Gaussian smoothing with a window of fifteen points, SNV normalisation and MC. This model was compared (Table 1) with the models proposed by the laboratories in the original paper for predicting humidity in forages.

As can be seen in Table 1, even though the model proposed by the ProSpecTool does not have the lowest error in cross-validation, it does have the lowest J-Score. This is, it is the most parsimonious model, maintaining a good prediction performance and a high robustness. Even if this does not mean that it is the absolute best model for the desired application, as this must be judged by the user, it is a model comparable to the best model proposed by experienced analysts. The user should decide the optimal preprocessing by exploring the top models (ordered by J-Score, RMSE or any desired property) based on their overall characteristics and it is advisable to validate the best candidates more thoroughly outside the toolbox, as it is an exploratory tool.

For protein content, the samples 170, 222, 225 and 236 were detected as outliers and the analysis was conducted after removing them for all datasets. The model built using the spectra preprocessed with Gaussian smoothing with a 15-point window, first derivative by differentiation, SNV normalisation and MC was selected as the optimal model by the J-Score criterion. This model was compared (Table 2) with the models proposed by the laboratories in the original paper for predicting protein content in forages.

Similar to the humidity prediction models, in Table 2 it can be seen that the model proposed by the ProSpecTool for protein content prediction is the most parsimonious one, with only 5 LVs and the lowest J-Score, maintaining a good prediction performance even if it does not have the lowest error in cross-validation. However, there might be other models that achieve similar results with less preprocessing steps, so it is up to the user to explore the best models and decide which is the most

Table 1
Comparison between the PLSR model for predicting humidity obtained by the ProSpecTool toolbox and the models obtained by the laboratories in the original paper, sorted by the best J-Score. For describing the models, the values of $RMSE_{CV}$, the ratio of $RMSE_{Cal}$ to $RMSE_{CV}$ and the Noise Index of the Regression Vector are included in the table.

	Preprocessing	LVs	J-Score	$RMSE_{CV}$ (%)	$1-\frac{RMSE_{Cal}}{RMSE_{CV}}$	NI_{RV}
ProSpecTool	Gaussian 15p SNV MC	3	0.12	0.39	0.02	0.05
Lab 4 & Lab 6	MC	3	0.13	0.39	0.03	0.07
Lab 2	SNV MC	6	0.17	0,30	0.08	0.20
Lab 5	S-G 11p 3rd ord. 2nd deriv. MC	9	0.18	0.27	0.14	0.20
Lab 3	S-G 7p 2nd ord. 2nd deriv. MC	3	0.18	0.39	0.03	0.20
Lab 1	S-G 7p 2nd ord. 2nd deriv. MC	7	0.20	0.28	0.08	0.25

Table 2
Comparison between the PLSR model for predicting protein content obtained by the ProSpecTool toolbox and the models obtained by laboratories in the original paper, sorted by the best J-Score. For describing the models, the values of $RMSE_{CV}$, the ratio of $RMSE_{Cal}$ to $RMSE_{CV}$ and the Noise Index of the Regression Vector, are included in the table.

	Preprocessing	LVs	J-Score	$RMSE_{CV}$ (%)	$1-\frac{RMSE_{Cal}}{RMSE_{CV}}$	NI_{RV}
ProSpecTool	Gaussian 15p Differentiation SNV MC	5	0.18	0.78	0.05	0.16
Lab 4	MC	6	0.20	0.96	0.05	0.18
Lab 5	S-G 11p 3rd ord. 2nd deriv. MC	9	0.20	0.69	0.10	0.22
Lab 1 & Lab 3	S-G 7p 2nd ord. 2nd deriv. MC	8	0.22	0.71	0.13	0.24
Lab 2	SNV MC	7	0.22	0.83	0.08	0.25
Lab 6	MC	8	0.22	0.86	0.07	0.27

appropriate one.

5. Independent testing

Prof. José Manuel Amigo, from the Department of Chemistry, University of the Basque Country (UPV/EHU), Spain, informed that he has tested the software and reported that it appears to work as the authors described.

6. Conclusions

A new MATLAB toolbox has been made available to help practitioners decide the best preprocessing methodology for the analysed spectral data, in order to obtain optimal PLS regression models. This toolbox quantifies the noise of the data, the multiplicative and additive scatter-effects and decides which preprocessing algorithms should be iterated to correct the spectra, providing the user with best-performing models to choose the optimal preprocessing technique for their data.

In the example shown, the resulting best models provided by the ProSpecTool (according to the J-Score) have proven to perform similarly to the best models proposed by experienced analysts. The toolbox has shown the capability to provide robust and parsimonious models that can compete with those obtained by experience driven trial-and-error, and in considerably less time than what a trial-and-error approach requires. Therefore, the ProSpecTool can be proposed as a tool that can help practitioners explore different preprocessing methods and obtain optimal regression models, considering that the final models should be more extensively validated and that ultimate conclusions and interpretation of the models are to be made by the analyst.

Funding

Grant PID2019-104269RR-C33 funded by MCIN/AEI/10.13039/501100011033. Grant URV Martí i Franqués –Banco Santander (2021PMF-BS-12). Chemometrics and Sensorics for Analytical Solutions (CHEMOSENS, ref.2021 SGR 00705, Departament de Recerca i Universitats, Generalitat de Catalunya).

CRediT authorship contribution statement

Jokin Ezenarro: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Daniel Schorn-García:** Writing – review & editing, Writing – original draft, Supervision, Investigation. **Olga Busto:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Ricard Boqué:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The ProSpecTool software will be freely available (for research purposes) under request to the authors.

During the preparation of this work the authors used ChatGPT 3.5 in order to edit text and improve readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2024.105096.

References

[1] P. Geladi, Chemometrics in spectroscopy. Part 1. Classical chemometrics, Spectrochim. Acta Part B At. Spectrosc. 58 (2003) 767–782, https://doi.org/10.1016/S0584-8547(03)00037-5.
[2] Å. Rinnan, F. Van Den Berg, S.B. Engelsen, Review of the most common pre-processing techniques for near-infrared spectra, Trends Anal. Chem. 28 (2009) 1201–1222, https://doi.org/10.1016/j.trac.2009.07.007.
[3] B. Dayananda, S. Owen, A. Kolobaric, J. Chapman, D. Cozzolino, Pre-processing applied to instrumental data in analytical Chemistry: a brief review of the methods and examples, Crit. Rev. Anal. Chem. (2023), https://doi.org/10.1080/10408347.2023.2199864.
[4] H. Jonsson, J. Gabrielsson, Evaluation of preprocessing methods, in: S.D. Brown, B. Walczak, R. Tauler (Eds.), Comprehensive Chemometrics, Elsevier, 2009, pp. 199–206.
[5] E.T.S. Skibsted, H.F.M. Boelens, J.A. Westerhuis, D.T. Witte, A.K. Smilde, New indicator for optimal preprocessing and wavelength selection of near-infrared spectra, Appl. Spectrosc. 58 (2004) 264–271, https://doi.org/10.1366/000370204322886591.
[6] J. Gerretzen, E. Szymańska, J.J. Jansen, J. Bart, H.J. van Manen, E.R. van den Heuvel, L.M.C. Buydens, Simple and effective way for data preprocessing selection based on Design of Experiments, Anal. Chem. 87 (2015) 12096–12103, https://doi.org/10.1021/acs.analchem.5b02832.
[7] J.M. Roger, A. Biancolillo, F. Marini, Sequential preprocessing through ORThogonalization (SPORT) and its application to near infrared spectroscopy, Chemometr. Intell. Lab. Syst. 199 (2020), https://doi.org/10.1016/j.chemolab.2020.103975.

[8] J.M. Andrade-Garda, R. Boqué-Martí, J. Ferré-Baldrich, A. Carlosena-Zubieta, Partial least-squares regression, in: J.M. Andrade-Garda (Ed.), Basic Chemometric Techniques in Atomic Spectroscopy, The Royal Society of Chemistry, Cambridge, 2013, pp. 181–243. http://www.ncbi.nlm.nih.gov/pubmed/25654500. (Accessed 11 October 2022).
[9] P. Geladi, B.R. Kowalski, Partial least-squares regression: a tutorial, Anal. Chim. Acta 185 (1986) 1–17.
[10] S. de Jong, SIMPLS: an Alternative Approach Squares Regression to Partial Least, 1993.
[11] J. Engel, J. Gerretzen, E. Szymańska, J.J. Jansen, G. Downey, L. Blanchet, L.M. C. Buydens, Breaking with trends in pre-processing? TrAC, Trends Anal. Chem. 50 (2013) 96–106, https://doi.org/10.1016/j.trac.2013.04.015.
[12] P. Oliveri, C. Malegori, R. Simonetti, M. Casale, The impact of signal pre-processing on the final interpretation of analytical outcomes - a tutorial, Anal. Chim. Acta 1058 (2018) 9–17, https://doi.org/10.1016/j.aca.2018.10.055.
[13] S. Särkkä, J. Sarmavuori, Gaussian filtering and smoothing for continuous-discrete dynamic systems, Signal Process. 93 (2013) 500–510, https://doi.org/10.1016/j.sigpro.2012.09.002.
[14] A. Savitzky, M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Z. Physiol. Chem. 40 (1951) 1832. https://pubs.acs.org/sharingguidelines. (Accessed 20 September 2022).
[15] A. Antoniadis, G. Oppenheim, Wavelets and Statistics, Springer, New York, 1995, https://doi.org/10.1007/978-1-4612-2544-7. New York, NY.
[16] F. Ehrentreich, L. Summchen, Spike removal and denoising of Raman spectra by wavelet transform methods, Anal. Chem. 73 (2001) 4364–4373, https://doi.org/10.1021/AC0013756.
[17] M.S. Dhanoa, S.J. Lister, R. Sanderson, R.J. Barnes, The link between multiplicative scatter correction (MSC) and standard normal variate (SNV) transformations of NIR spectra, J. Near Infrared Spectrosc. 2 (1994) 43–47, https://doi.org/10.1255/jnirs.30.
[18] P.H.C. Eilers, A perfect smoother, Anal. Chem. 75 (2003) 3631–3636, https://doi.org/10.1021/AC034173T.
[19] R.J. Barnes, M.S. Dhanoa, S.J. Lister, Standard Normal Variate Transformation and De-trending of Near-Infrared Diffuse Reflectance Spectra, 1989.
[20] M. Zeaiter, D. Rutledge, Preprocessing methods, in: Comprehensive Chemometrics, Elsevier, 2009, pp. 121–231, https://doi.org/10.1016/B978-044452701-1.00074-0.
[21] J. Ezenarro, D. Schorn-García, L. Aceña, M. Mestres, O. Busto, R. Boqué, J-Score, A new joint parameter for PLSR model performance evaluation of spectroscopic data, Chemometr. Intell. Lab. Syst. 240 (2023) 104883, https://doi.org/10.1016/J.CHEMOLAB.2023.104883.
[22] A.C. Olivieri, N.M. Faber, Validation and error, in: Comprehensive Chemometrics, Elsevier, 2009, pp. 91–120, https://doi.org/10.1016/B978-044452701-1.00073-9.
[23] F. Westad, F. Marini, Validation of chemometric models - a tutorial, Anal. Chim. Acta 893 (2015) 14–24, https://doi.org/10.1016/j.aca.2015.06.056.
[24] E. Lopez, J. Etxebarria-Elezgarai, J.M. Amigo, A. Seifert, The importance of choosing a proper validation strategy in predictive models. A tutorial with real examples, Anal. Chim. Acta 1275 (2023) 341532, https://doi.org/10.1016/J.ACA.2023.341532.
[25] T. Fearn, Assessing calibrations: SEP, RPD, RER and R2, NIR News 13 (2002) 12–13, https://doi.org/10.1255/nirn.689.
[26] C. Liu, S.X. Yang, X. Li, L. Xu, L. Deng, Noise Level Penalizing Robust Gaussian Process Regression for NIR Spectroscopy Quantitative Analysis, vol. 201, Chemometrics and Intelligent Laboratory Systems, 2020, https://doi.org/10.1016/j.chemolab.2020.104014.
[27] K.H. Esbensen, Brad Swarbrick, Multivariate Data Analysis, sixth ed., Camo, Oslo, 2018.
[28] I. Ruisánchez, F.X. Rius, S. Maspoch, J. Coello, T. Azzouz, R. Tauler, L. Sarabia, M. C. Ortiz, J.A. Fernández, D. Massart, A. Puigdomènech F, C. García, Preliminary results of an interlaboratory study of chemometric software and methods on NIR data. Predicting the content of crude protein and water in forages, Chemometr. Intell. Lab. Syst. 63 (2002) 93–105, https://doi.org/10.1016/S0169-7439(02)00039-4.