A MATLAB toolbox for multivariate regression coupled with variable selection

Viviana Consonni[a], Giacomo Baccolo[a,b], Fabio Gosetti[a], Roberto Todeschini[a], Davide Ballabio*[a]

[a] Milano Chemometrics and QSAR Research Group, Department of Earth and Environmental Sciences, University of Milano - Bicocca, Milano, Italy

[b] Department of Food Science, University of Copenhagen, Rolighedsvej 30, DK-1958 Frederiksberg C, Denmark.

Corresponding author:
Davide Ballabio: davide.ballabio@unimib.it
Dept. of Earth and Environmental Sciences
University of Milano-Bicocca, P.zza della Scienza, 1 - 20126 Milano, Italy

Abstract

Multivariate regression is a fundamental supervised chemometric approach that defines the relationship between a set of independent variables and a quantitative response. It enables the subsequent prediction of the response for future samples, thus avoiding its experimental measurement. Regression approaches have been widely applied for data analysis in different scientific fields.

In this paper, we describe the regression toolbox for MATLAB, which is a collection of modules for calculating some well-known regression methods: Ordinary Least Squares (OLS), Partial Least Squares (PLS), Principal Component Regression (PCR), Ridge and local regression based on sample similarities, such as Binned Nearest Neighbours (BNN) and *k*-Nearest Neighbours (kNN) regression methods. Moreover, the toolbox includes modules to couple regression approaches with supervised variable selection based on All Subset models, Forward Selection, Genetic Algorithms and Reshaped Sequential Replacement. The toolbox is freely available at the Milano Chemometrics and QSAR Research Group website and provides a graphical user interface (GUI), which allows the calculation in a user-friendly graphical environment.

1. Introduction

Multivariate regression is a major approach of chemometrics [1] and consists in defining a relationship between a set of independent variables and a quantitative response. The response is considered as the interesting element of the system under analysis, such as the concentration of analytes or the biological activity of chemicals. It usually cannot be simply and directly determined, while the variables are generally easily measurable [2]. Therefore, regression models can be used to predict the response of interest for future samples, avoiding thus its experimental measurement.

Regression models can be linear or nonlinear. Linear models can be formulated as a simple equation and directly calculated on the original variables (Ordinary Least Squares regression, OLS) or with a reduced set of intermediate linear combinations of the variables. Latent variables and principal components constitute these combinations in the case of Partial Least Squares regression (PLS) and Principal Component Regression (PCR), respectively, which can enhance modelling on highly correlated variables or datasets with more variables than samples. On the contrary, approaches based on local modelling take advantage of the analogy between samples in terms of both variables and response. These approaches are quite common when dealing with QSAR modelling, where analogy is exploited in terms of molecular structural similarity [3–5].

Some available toolboxes to calculate regression models in MATLAB exist [6,7], as well as specific toolboxes related for example to first, second and third-order multivariate calibration [8–10] or multi-block modelling [11].

This manuscript deals with the presentation of a novel regression toolbox for MATLAB, which is a collection of MATLAB modules to calculate regression models based on different approaches: OLS, PLS, PCR, Ridge and local regression methods (BNN and kNN). Additionally, the toolbox includes four methods (All Subset models, Forward Selection, Genetic Algorithms and Reshaped Sequential Replacement) which can be linked with regression models for the supervised variable selection.

The toolbox is freely available via Internet from the Milano Chemometrics and QSAR Research Group website [12] and alternatively at Zenodo [13]. It provides comprehensive results and diagnostic tools for regression models, which are easily accessible thanks to an easy-to-use graphical user interface (GUI). This enables the user to perform all the required steps of analysis: data loading and pre- processing, univariate data screening, component selection (when required), model calculation and diagnostic, prediction of new samples and variable selection. In the first part of the paper, the theory of regression approaches included in the toolbox is briefly introduced. Then, features of the MATLAB modules and GUI are described and finally an application example on a real spectroscopy dataset is shown.

2. Theoretical background

2.1 Notation

Scalars are represented by italic lower-case characters (e.g. x_{ij}) and vectors by bold lower-case characters (e.g. **x**). Two-dimensional arrays (matrices) are denoted as **X** ($I \times J$), where I is the number of samples (i.e., matrix rows) and J the number of variables included in the model (i.e., matrix columns). The ij-th element of the data matrix **X** (known as model matrix in the regression framework), denoted as x_{ij}, represents the value of the j-th variable for the i-th sample.

2.2 Linear regression

Regression models estimate the mathematical relationship between a set of independent variables, arranged in a data matrix **X** ($I \times J$), and a quantitative independent variable, which is collected in the response column vector **y**.

Linear regression methods estimate this relationship with a linear combination of the independent variables, thus linear models can be formalized as follows:

$$\hat{\mathbf{y}} = \mathbf{Xb} \tag{1}$$

where $\hat{\boldsymbol{y}}$ is the vector containing the estimate of the response **y** and **b** is the vector of the regression coefficients. A brief description of how coefficients are estimated through the regression methods included in the toolbox is given in the following paragraphs. Further details can be found in the literature [2,14].

2.2.1 Ordinary Least Squares (OLS) regression

Ordinary least squares method provides the regression coefficients that minimize the Residual Sum of Squares (RSS) [2,14] and are calculated as follows:

$$\mathbf{b}_{OLS} = \left(\mathbf{X^T X}\right)^{-1} \mathbf{X^T y} \tag{2}$$

OLS is definitely one of the most common and popular regression approaches. However, there are some limitations related to the constraints of the inversion of the information matrix ($\mathbf{X^T X}$): the number I of samples must be higher than the number J of variables and no collinear variables should be present in the data matrix.

2.2.2 Ridge Regression (RR)

Unlike OLS, ridge regression (RR) can handle the presence of highly correlated variables. The main idea of RR is to add a k parameter to the diagonal of the matrix ($\mathbf{X}^T\mathbf{X}$) to mitigate the inversion problem due to ill-conditioned data matrices [14]. The RR regression coefficients are calculated by the following equation:

$$\mathbf{b}_{RIDGE} = \left(\mathbf{X}^T\mathbf{X} + k\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y} \quad (3)$$

where $\mathbf{I}$ ($J \times J$) is the identity matrix. The optimal k value can be chosen by a cross-validation protocol, thus selecting the k that minimizes the regression error in prediction.

2.2.3 Principal Component Regression (PCR)

Principal Component Regression integrates Principal Component Analysis (PCA) [15] modelling to OLS regression [14]. PCR coefficients are estimated in the following way:

$$\mathbf{b}_{PCR} = \left(\mathbf{L}\mathbf{\Lambda}^{-1}\mathbf{L}^T\right)\mathbf{X}^T\mathbf{y} \quad (4)$$

where $\mathbf{\Lambda}$ is the matrix of PCA eigenvalues and $\mathbf{L}$ ($J \times M$) is the loading matrix. The optimal number M of principal components to be used in the regression model can be selected by a cross-validation protocol.

2.2.4 Partial Least Squares (PLS) regression

Similarly to PCR, PLS regression method identifies a new set of features, known as Latent Variables (LVs), which are used to estimate the regression coefficients. PLS finds LV directions that describe the largest amount of data variance in X and are most correlated to the response [16]. There are several different algorithms to estimate the PLS coefficients [17]; the regression toolbox implements the NIPALS algorithm [18]:

$$\mathbf{b}_{PLS} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{Q}^T \quad (5)$$

where $\mathbf{P}$ ($J \times M$) and $\mathbf{Q}$ ($1 \times M$) are the loadings for $\mathbf{X}$ and $\mathbf{y}$, respectively, and $\mathbf{W}$ ($J \times M$) collects the PLS weights. As for PCR, the optimal number M of Latent Variables can be estimated in cross-validation.

2.3 Similarity-based regression methods

The *k*-Nearest Neighbours (kNN) approach was originally proposed for supervised classification but later adapted to multivariate regression [10]. The response of a target is estimated as the average of the responses of the k nearest neighbours, that is, the k most similar training samples, which are identified by a selected metric.

The Binned Nearest Neighbours (BNN) approach is quite similar to kNN, with the difference that the prediction is based on a flexible number of neighbours. Similarity intervals (i.e., bins) are defined on the basis of a tuning parameter alpha and the training samples are then distributed into these intervals according to their similarity to the target. All the neighbours falling into the bin with the largest similarity are then considered for the prediction [19].
The optimal number of *k* neighbours or the optimal value of the alpha tuning parameters are selected by cross-validation protocols.

2.4 Variable selection

Variable selection approaches aim to find a subset of variables that can improve the prediction performance and/or simplify the model. This toolbox includes four selection strategies (All Subset models, Forward Selection, Genetic Algorithms and Reshaped Sequential Replacement), which can be coupled with any of the regression methods previously described. For all selection methods, the *RMSECV* and the determination coefficient (R^2_{cv}) in cross-validation are used as the fitness functions.

2.4.1 All Subsets (AS) models

This approach evaluates all the possible combinations of the *J* variables and guarantees that the best subset of variables is found by minimising the regression error in cross-validation (*RMSECV*); however, it is highly computationally consuming and thus it becomes unsuitable for large numbers of variables. In the regression toolbox, this selection method is active only when *J* is lower than or equal to 15.

2.4.2 Forward Selection (FS)

This is a basic and simple approach [20]: starting from an empty set, variables are sequentially added to this set, one at a time; in each iteration, the criterion to select which variable to include is based on the minimisation of *RMSECV*. In this approach, results can be biased by the first variables included in the selected set and, when a high number of variables is present, the application of this method is not suggested.

2.4.3 Genetic Algorithms (GAs)

Genetic Algorithms (GAs) are a popular variable selection approach [21] that is inspired by the natural selection of the individuals of a population during its evolution. In the GA framework, each gene represents a variable and each chromosome (sequence of genes/variables) represents a model. The evolution of the population (the default population consists of 30 chromosomes, that is,

individuals) is determined by two processes: in the crossover step, pairs of chromosomes generate new individuals according to a crossover probability (default 50%), while in the mutation step genes of each chromosome can change according to a mutation probability (default 1%). When a new chromosome has better performance than that of the already existing ones, it enters the population and the worst model is discarded [22].

After evolution, different strategies can be used to select the final subset of variables. In this toolbox, the GA evolution is independently repeated for a fixed number of times (called runs) and the relative occurrence frequency of each variable in the best models is calculated [21]. Then, the user can set a frequency threshold and only the variables with occurrence frequencies higher than the threshold will be further processed by a forward selection or All Subset models procedure [14]. Finally, a list of models (including the best subsets of selected variables) is proposed, from which the user can choose the preferred model.

2.4.4 Reshaped Sequential Replacement (RSR)

The Sequential Replacement (SR) method was originally proposed by Miller [23], based on the idea to replace each variable included in a model of selected size (called seed), one at a time, with all of the remaining variables and see whether a better model is obtained. The initial population is randomly generated, giving constraints on the number of variables (size) for each model. The new seed (i.e., the model with the lowest error in cross-validation) is chosen only after all the variables have been replaced and the obtained models have been compared. The procedure is iterated until no replacement leads to an improvement of the models. The Reshaped Sequential Replacement (RSR) algorithm is based on the same idea, but also implements some functionalities aimed at: a) decreasing the calculation time; b) estimating the prediction ability along with the fitting of the models; c) increasing the probability to converge to the optimal model; d) identifying models that suffer from pathologies, such as overfitting, chance correlation, variable redundancy and collinearity [5,24].

2.5 Model diagnostics

Depending on the regression method, several diagnostic tools are provided to assess the model quality, evaluate the model assumptions, investigate whether or not there are observations with a large influence on the model and identify critical points, such as the outliers.

The most common dispersion and correlation-based measures to evaluate the overall model quality, that is, the root mean squared error (*RMSE*) and the coefficient of determination R^2, are calculated as:

$$RMSE = \sqrt{\frac{RSS}{I}} = \sqrt{\frac{\sum_{i=1}^{I}\left(y_i - \hat{y}_i\right)^2}{I}} \tag{6}$$

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{I}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{I}\left(y_i - \bar{y}\right)^2} \tag{7}$$

where *RSS* is the residual sum of squares, *TSS* the total sum of squares, y_i and $\hat{y}_i$ are the observed and calculated response, respectively, $\bar{y}$ the average response and *I* the number of samples. These metrics are used also to assess the cross-validation performance (R^2_{cv}, *RMSECV*); in this case, the response $\hat{y}_i$ is predicted when the *i-th* sample is temporarily left out of the training set. When the model predictive ability is evaluated on an external test, the following metrics are calculated [25]:

$$RMSEP = \sqrt{\frac{PRESS}{I_{ext}}} = \sqrt{\frac{\sum_{i=1}^{I_{ext}}\left(y_i - \hat{y}_i\right)^2}{I_{ext}}} \tag{8}$$

$$Q^2_{\mathrm{F3}} = 1 - \frac{PRESS/I_{ext}}{TSS/I} = 1 - \frac{\dfrac{\sum_{i=1}^{I_{ext}}\left(y_i - \hat{y}_i\right)^2}{I_{ext}}}{\dfrac{\sum_{i=1}^{I}\left(y_i - \bar{y}\right)^2}{I}} \tag{9}$$

where *PRESS* is the predicted residual sum of squares, $\hat{y}_i$ is the predicted response for the *i-th* external test sample and I_{ext} the number of external test samples. Q^2_{F3} is a standardized measure that is particularly well-suited for comparing the external predictivity of different models developed on the same training dataset. However, this metric is sensitive to the training data variation and, thus, its usage for comparing the predictive ability of models fitted to different data is inappropriate and should be avoided, due to the risk of drawing misleading or unreliable conclusions [26] .

To identify the most influential samples, that is, those samples that change the model parameters, thus, having a large influence on the fit of the model, the leverage (h_{ii}) can be calculated as the following [27]:

$$h_{ii} = \mathbf{x}_i^{\mathrm{T}} \left(\mathbf{X}^{\mathrm{T}} \mathbf{X} \right)^{-1} \mathbf{x}_i \quad (10)$$

where $\mathbf{x}_i$ ($1 \times J$) is the independent variable vector of the *i-th* sample. The leverage indicates somehow the distance of a sample from the centre of the model space, thus, high values of leverage are likely associated potential influential samples and/or samples that are out of the model applicability domain. Traditionally, when dealing with methods based on components, anomalous samples can be detected also by means of Hotelling's T^2 (i.e., the sum of the normalized squared scores), which is a measure of the variation of each sample within the model [15]:

$$T_i^2 = \sum_{m=1}^{M} \frac{t_{im}^2}{\lambda_m} \quad (11)$$

where t_{im} and λ_m are the score of the *i*-th sample and the eigenvalue on the *m*-th component, respectively.

The *Q* statistic indicates how well each sample conforms to the model, it being a function of the sample residuals:

$$Q_i = \mathbf{e}_i \mathbf{e}_i^{\mathrm{T}} \quad (12)$$

where $\mathbf{e}_i$ is the *i*-th row of the residual matrix **E**:

$$\mathbf{X} = \mathbf{T}\mathbf{L}^{\mathrm{T}} + \mathbf{E} \quad (13)$$

T is the score matrix and **L** the loading matrix. Both Hotelling's T^2 and *Q* residuals can be associated to upper confidence limits [15] and contributions for each specific sample. These can be useful when inspecting a potential outlier since they indicate which variables caused the sample to have high values for Hotelling's T^2 and Q residuals [28].

Finally, to identify outliers, which are defined as the samples that have a large residual (i.e., the observed value for the sample is very different from that predicted by the regression model), the toolbox provides ordinary and standardised (r_i') residuals of samples, which can be compared to thresholds equal to 1, 2 or 3 units of standard deviation. Standardized residuals are calculated as the following [27]:

$$r_i' = \frac{y_i - \hat{y}_i}{s\sqrt{1 - h_{ii}}} \quad (14)$$

where y_i and $\hat{y}_i$ are the observed and calculated response for the *i-th* sample, respectively; h_{ii} is the leverage of the *i-th* sample and *s* is the standard error of the estimate, which is calculated as the square root of $s^2 = RSS/(I - J)$.

3. Structure and main features of the toolbox

The toolbox consists of a collection of functions provided as MATLAB source files, with no requirements for any other third party's utilities beyond the MATLAB installation. Calculation and validation of regression models can be performed by the MATLAB command window or a graphical user interface.

3.1 Input data

Data must be arranged in a numerical matrix with dimensions $I \times J$, where I is the number of samples and J the number of independent variables, while the quantitative response y must be uploaded as a separate numerical column vector ($I \times 1$), where the *i-th* element of this vector represents the response value of the *i-th* sample. The sample and variable labels can be defined into two different additional string arrays.

3.2 Calculation by the command line

Regression models can be calculated and internally cross-validated in the MATLAB command window by means of the functions that are listed and described in Table 1. The output of these functions is a structure array, where the results are collected together with all the selected options for fitting and/or validating the model.

When selecting cross-validation, the user must define the number of cancellation groups and the way samples are distributed in these groups (i.e., venetian blinds or contiguous blocks). The difference between the two approaches is depicted in the following example: let a dataset be comprised of 6 samples, divided in 3 cross-validation groups (each constituted by 2 samples):

1. the venetian blinds approach allocates samples in the following groups: [1,0,0,1,0,0], [0,1,0,0,1,0], and [0,0,1,0,0,1].
2. the contiguous blocks approach splits the samples in the following way: [1,1,0,0,0,0], [0,0,1,1,0,0] and [0,0,0,0,1,1].

The choice of the most suitable type of cross-validation depends on how samples are organized into the dataset. The other two available validation protocols are bootstrap with resampling and Montecarlo random sampling (20% of samples in the evaluation set) [29]; in these cases, the user must define the number of iterations.

Pre-processing methods can be applied both on rows (standard normal variate, multiplicative scatter correction, first and second derivative, Savitzky-Golay smoothing) and columns (centering, unit variance, autoscaling, 0-1 range scaling) of the data matrix.

If the model is calculated by biased methods, ad-hoc routines allow to estimate the optimal complexity, that is, the optimal number of significant components to be retained (for PLS and PCR) or the optimal value of the tuning parameters, such as the Ridge parameter *k*, the number of neighbours or the alpha value for local regression based on *k*-Nearest Neighbours (kNN) and Binned Nearest Neighbours (BNN), respectively. These routines provide the values of R^2_{cv} and *RMSECV* as a function of the number of components or values of tuning parameters, from which the user can select the optimal setting for the model calculation.

Finally, unknown or test samples can be predicted with a fitted model by ad-hoc routines (Table 1), which return a structure array containing the predicted responses and additional diagnostics (depending on the regression approach), such as Hotelling's T^2, Q residuals, and leverages.

In order to carry out variable selection, the following routines are available: ga_selection, forward_selection, rsr_selection, allsubset_selection. All settings for variable selection can be defined by means of ad-hoc functions (ga_options, forward_options, rsr_options, allsubset_options).

3.3 Graphical User Interface (GUI)

The graphical interface (GUI) of the toolbox enables the user to perform all the steps in a friendly way. Typing *reg_gui* in the MATLAB prompt, the main window of the GUI opens (Figure 1). The menu *File* enables the user to upload the data matrix (together with sample and variable labels, if available) and the quantitative response vector, which can be selected both from the MATLAB workspace or MATLAB files.

The menu *View* includes basic graphical tools that can be used for the visualization and initial inspection of the data. Savitzky-Golay smoothing can be applied and profiles of samples can be plotted both with the raw and scaled data, allowing the user to analyse the variable distributions and select the most adequate data pre-processing. Boxplots, histograms and bi-variate plots can be used to further explore the variable distributions, the presence of anomalous samples and the correlation between independent variables and/or variables and response.

The menu *Calculate* allows selecting the regression method (i.e., Ordinary Least Squares, Partial Least Squares, Principal Component Regression, Ridge Regression and Local Regression), as well as to couple regression methods with variable selection. To implement biased methods, specific approaches based on validation protocols (e.g., cross-validation, bootstrap with resampling and Montecarlo random sampling) are available to select the optimal model complexity, that is, the

significant number of latent variables and components for PLS and PCR, respectively, the *k* parameter for Ridge regression, the number of neighbours for kNN and the alpha value for BNN.
After calculation, the model can be saved in the MATLAB workspace. The model parameters and results are collected as the fields of a unique MATLAB structure, as described in Table 1. A saved model can later be re-loaded to be used for further analyses; for instance, by the menu *Predict*, to calculate the predicted responses of new samples. If the experimental responses of these new samples are known and uploaded by the menu *File*, then the toolbox also calculates the regression measures to assess the quality of predictions on the new set of samples.

4. Illustrative example

The following paragraphs describe an example of the application of the regression toolbox for MATLAB to highlight the main characteristics of the toolbox. In particular, we calibrate a PLS model to predict the content of the active substance in pharmaceutical tablets on the basis of their spectroscopic profiles [30]. The dataset consists of 310 pharmaceutical tablets (samples); each tablet is described by the NIR transmittance spectrum in the range 7400-10500 cm^{-1} (404 variables) and the content of the active substance (% w/w), which was experimentally determined by means of HPLC. A detailed description of the sampling and the data structure is available in the original study [30].
This dataset was randomly divided into a calibration set of 243 samples and a test set of 67 samples. Following the original study, the NIR spectra have been pre-processed using second derivatives and all the calibration analyses have been then performed on these pre-processed data, which are provided in the toolbox as MATLAB data file (tablet.mat).

4.1 Calculation and validation of the regression model

In the menu *Calculate*, after clicking on Partial Least Squares, the window 'PLS settings' enables the selection of the number of latent variables (LVs), the type of data pre-processing and validation protocol. When calibrating a PLS model, the first mandatory step is to determine the optimal number of LVs to be included in the model. To this end, we set the cross-validation procedure, based on the venetian blinds and 4 cross-validation groups, and the mean centering as the data pre-processing (Figure 2a). Then, clicking on the button 'optimal LV', the toolbox automatically returns the plots of R^2_{cv} and *RMSECV* as a function of the number of Latent Variables, as shown in Figure 2b. We can thus select 3 LVs for the subsequent model calibration, which is executed through the button 'calculate'. This illustrative example was run on a personal computer with an Intel Core(TM)

processor running at 2.60 GHz using 16 GB of RAM, running Windows version 10. In this framework, consider that the computation of PLS required fractions of seconds (0.005 sec.), while cross validation only 0.2 seconds.

Once the model has been calculated, the main window is updated with basic information on the model (number of LVs, data pre-processing, explained variance in X), the fitting (R^2 = 0.928, *RMSE* = 0.3430) and cross-validation (R^2_{cv} = 0.922, *RMSECV* = 0.3560) performance. To further validate the calibrated PLS model on the test set, we first upload the test data matrix and response vector by the menu *File* and then select the 'predict samples' option in the menu *Predict*. At this stage, the fitted model calculates the predictions for the test samples and the corresponding predictivity metrics, which are updated in the main window (Q^2_{F3} = 0.913, *RMSEP* = 0.3758). Finally, both the model and the predictions for the test set can be saved into the MATLAB workspace as structures (Table 1) by the options 'save model' and 'save pred' in the menu *File*.

4.2 Model diagnostics

To further analyse the calibrated model and get more insight into the obtained predictions, one can exploit the diagnostic plots that can be easily generated by the menu *Result* of the toolbox GUI. In particular, in the case of PLS, explained and cumulative variances can be plotted as a function of the retained latent variables (Figure 3a) and samples and variables can be analysed through the options 'scores' and 'coefficients', respectively. By selecting the option 'score', a new window opens (Figure 3b), which allows the generation of several diagnostic plots.

For example, we might be interested in visualising calculated against experimental response (Figure 3b); in this plot, training samples have been coloured in a greyscale according to their experimental response and test samples in red. If necessary, also the sample labels (i.e., the identification numbers or other user-defined labels) can be visualized near the corresponding points. This plot allows a quick look at the model performance; in our case, none of the samples seems to be severely biased.

Other plots can be generated by selecting different combinations of the following metrics: leverages, residuals, standardised residuals, samples scores, Hotelling's T^2 and Q residuals. The plot of residuals vs. experimental responses can be useful for checking the assumption of linearity and homoscedasticity. The assumption of linearity is usually considered as fulfilled if the standardized residuals are in the range ±2 (95% confidence levels), the homoscedasticity assumption holds if there is no specific pattern in the residuals (Figure 4a). As outliers may or may not be influential points, it is important to understand why these samples have extreme behaviour. The influence plot (Figure 4b), where Hotelling's T^2 are plotted against Q residuals, can be useful to rapidly identify anomalous samples within the model. In particular, Hotelling's T^2 identifies samples

with a large distance from the origin of the model space, while Q residuals indicate how well each sample conforms to the Latent Variable space and thus it is a measure of the difference between a sample and its projection into the retained Latent Variables. The confidence limits, represented by the red lines, can help the identification of anomalous samples.

For example, sample S218 appears anomalous in the influence plot (Figure 4b) and thus requires further investigation. To this end, the user can select this specific sample by clicking the "view sample" button and positioning the mouse cursor on its point in the plot; in this way, the toolbox opens a new plot showing the profiles of the raw or scaled variables together with its Hotelling's T^2 and Q contributions (Figure 5a). These plots can help to evaluate which variables are responsible for the anomalous sample behavior. Therefore, sample S218 results anomalous mainly on two spectral windows (8000-8400 and 9500-10500), where its Hotelling's T^2 contributions are moderately larger than the confidence threshold. On the contrary, all of the samples located near the axes origin of the influence plot (bottom left part of the plot) are associated with significantly lower Hotelling's T^2 and Q residual, e.g. sample S119 (highlighted in Figure 5b). From the comparison of the raw data profiles, it is apparent that S218 has different signals in the two spectral regions mentioned above.

Beside the diagnostic plots based on samples, the GUI also provides simple tools to evaluate the influence and weight of each variable on the model. An interactive window similar to that used to analyse samples can be activated through the option 'coefficients' in the menu *Results*. When dealing with PLS, this window enables the user to create plots to visualize the model coefficients, variable loadings and weights. For example, it might be interesting to look at the standardised regression coefficients (Figure 6a), where the spectral region from 8800 cm^{-1} to 8900 cm^{-1}, which includes the original variables with the largest absolute values of the standardized coefficients, appears as the most relevant to determine the concentration of the active substance, coherently with the results of the original study [30]. Then, we can use the bivariate plots provided in the toolbox, through the option 'univariate/bivariate plots' in the menu *View*, to deeper evaluate the relationships of single wavenumbers with the experimental response. For instance, the plot of the concentration of active substance versus NIR transmittance at 8802 cm^{-1} (i.e., one of the wavenumbers mostly related to the response) clearly shows the existence of an inverse correlation between this variable and the concentration of the active substance (Figure 6b).

4.3 Variable selection

All the regression methods implemented in the toolbox can be coupled with the following supervised variable selection procedures: All Subset models, Forward Selection (FS), Genetic Algorithms (GAs) and Reshaped Sequential Replacement (RSR). For each variable selection method, different

parameter settings are made available in a dedicated window that can be activated by the option 'Variable selection' in the menu *Calculate*.

When dealing with GAs or FS selection coupled with PLS, PCR or Ridge regression, the user can decide to divide variables in contiguous windows (intervals). Variables included in the same window are treated as one input in the selection process. This approach has been demonstrated to improve the selection outcome when dealing with highly correlated data, such as spectra.

The original study [30] of the Tablet dataset showed that the variable selection improved the model performance or at least reduced the model complexity, lowering the number of variables. In effect, looking at the Figure 6a, one can see that there are several wavelengths associated to very small or null regression coefficients, that is, they do not contribute to the model.

To run the variable selection in the toolbox, initially we select PLS as the regression method and set the specific options on data pre-processing and validation protocol (mean centering, venetian blinds cross-validation with 4 cv-groups), then we select Genetic Algorithms as the selection method and define the specific parameter settings: 100 runs (i.e., how many times the GA is iteratively repeated), the number of windows equal to 100 and the maximum number of windows in the model equal to 10. Since the Tablet dataset has 404 original variables, selecting 100 windows means that each window includes 4 adjacent variables.

At the end of the evolution, an interactive window for variable selection (Figure 7) appears; here, a bar plot shows the frequency of selection of each variable (or variable window), that is, the relative occurrence frequency of the variable in the best models of all GA runs (100 in our case). The most frequent windows, which are coloured in red in the bar plot, include those spectral variables that were previously identified as the most significant on the basis of regression coefficients (Figure 6a). Then, the user can select a frequency threshold (e.g., 0.25) and all the variables (or windows) with frequency higher than this threshold are used for a subsequent refinement based on a forward or all subsets selection. We decide to proceed with the all subset selection since we have only 3 frequent windows. The All Subsets method evaluates all the possible combinations of the most frequently selected variables (windows) and the best combinations are finally provided together with their specific details (size, labels of included variables, *RMSE*, *RMSECV*, R^2 and R^2_{cv}, suggested number of LVs).

The user can finally select its favourite solution. In our case study, we select a model with satisfactory performance (R^2: 0.931 and R^2_{cv}: 0.929), which is based on 3 windows that include a total of 12 variables (wavelengths from 7522 cm^{-1} to 7545 cm^{-1} and from 8818 cm^{-1} to 8872 cm^{-1}). In conclusion, variable selection was able to identify a significantly simpler model, still maintaining the regression performance as high as that achieved on the entire range of wavenumbers. Finally, the model based on the selected variables can be further investigated and validated with the available tools explained

in the previous sections. Note that, when predicting external or new samples, the subset of variables to be considered to run the model is automatically selected by the toolbox.

5. Independent testing

Prof. José Manuel Amigo, at the Faculty of Science and Technology, University of the Basque Country (Bilbao, Spain), informed that he has tested the toolbox and found that it appears to function as the Authors described.

6. Conclusion

The regression toolbox for MATLAB is a collection of modules for calculating multivariate regression methods: Ordinary Least Squares, Partial Least Squares, Principal Component Regression, Ridge and local regression methods based on sample similarities, such as Binned Nearest Neighbours and *k*-Nearest Neighbours regression approaches. Moreover, All Subset models, Forward Selection, Genetic Algorithms and Reshaped Sequential Replacement can be coupled with regression methods to select proper subsets of variables.

The toolbox is freely available via internet [12,13] and provides comprehensive results and diagnostic tools for regression models, which are easily accessible by an easy-to-use graphical user interface (GUI). All the steps of regression analysis can be performed in a user-friendly way: data loading and pre- processing, univariate data screening, component selection (when required), model calculation, validation and diagnostics, prediction of new samples and variable selection.

This GUI can represent a valuable tool, since it helps users, especially beginners of chemometrics or MATLAB, to better handle input and output of multivariate regression approaches, provided that a basic knowledge on the underlying regression methods is essential to correctly calibrate models and interpret the results. This toolbox will be constantly maintained and updated. Moreover, since MATLAB will probably stop soon the support of GUIs, we are planning to convert it in a MATLAB app or compile and distribute the toolbox as executable file.

Figure captions

Figure 1. Main graphical interface of the regression toolbox for MATLAB

Figure 2. Tablet dataset: a) different settings for PLS calculation; b) plots for optimal LV number selection: R^2cv and RMSECV as a function of the number of latent variables in the PLS model.

Figure 3. a) PLS explained variance per latent variable; b) interactive graphical interface for results visualization: experimental vs predicted response plot; training samples are coloured in a greyscale according to their experimental response, from white (minimum) to black (maximum), while test samples are coloured in red. The red diagonal represents the ideal perfect fit line.

Figure 4. a) plot of experimental responses versus standardized residuals; b) influence plot: Hotelling's T^2 versus Q residuals. Training samples are coloured in a greyscale according to their experimental response, from white (minimum) to black (maximum), while test samples are coloured in red. The red lines represent the confidence limits.

Figure 5. Variable profiles of the raw data, Hotelling's T^2 contributions and Q contributions of the sample S218 with anomalous behaviour (a) in comparison with the sample S119 (b) that shows a standard behaviour. Red lines represent the confidence limits.

Figure 6. a) standardised regression coefficients as a function of variables; b) plot of experimental response versus transmittance at 8802 cm-1. Samples are coloured in a greyscale according to response, from white (minimum) to black (maximum).

Figure 7. Interactive window for variable selection. Bar plot of the relative occurrence frequency of the variable windows in the best models of all GA runs.

7. References

[1] M.A. Nemeth, R.G. Brereton, Multivariate Pattern Recognition in Chemometrics, Illustrated by Case Studies (Data Handling in Science and Technology, Vol. 9), Elsevier, 1995. https://doi.org/10.2307/1269162.

[2] K. Varmuza, P. Filzmoser, Introduction to Multivariate Statistical Analysis in Chemometrics, CRC press, 2016. https://doi.org/10.1201/9781420059496.

[3] G. Klopman, S.E. Stuart, Multiple computer-automated structure evaluation study of aquatic toxicity. III. Vibrio Fischeri, Environ. Toxicol. Chem. 22 (2003) 466–472. https://doi.org/10.1897/1551-5028(2003)022<0466:MCASES>2.0.CO;2.

[4] M. Cassotti, D. Ballabio, R. Todeschini, V. Consonni, A similarity-based QSAR model for predicting acute toxicity towards the fathead minnow (Pimephales promelas), SAR QSAR Environ. Res. 26 (2015) 217–243. https://doi.org/10.1080/1062936X.2015.1018938.

[5] M. Cassotti, F. Grisoni, R. Todeschini, Reshaped Sequential Replacement algorithm: An efficient approach to variable selection, Chemom. Intell. Lab. Syst. 133 (2014) 136–148. https://doi.org/10.1016/j.chemolab.2014.01.011.

[6] M. Daszykowski, S. Serneels, K. Kaczmarek, P. Van Espen, C. Croux, B. Walczak, TOMCAT: A MATLAB toolbox for multivariate calibration techniques, Chemom. Intell. Lab. Syst. 85 (2007) 269–277. https://doi.org/https://doi.org/10.1016/j.chemolab.2006.03.006.

[7] H.D. Li, Q.S. Xu, Y.Z. Liang, libPLS: An integrated library for partial least squares regression and linear discriminant analysis, Chemom. Intell. Lab. Syst. 176 (2018) 34–43. https://doi.org/10.1016/j.chemolab.2018.03.003.

[8] A.C. Olivieri, H.C. Goicoechea, F.A. Iñón, MVC1: An integrated MatLab toolbox for first-order multivariate calibration, Chemom. Intell. Lab. Syst. 73 (2004) 189–197. https://doi.org/10.1016/j.chemolab.2004.03.004.

[9] A.C. Olivieri, H.L. Wu, R.Q. Yu, MVC2: A MATLAB graphical interface toolbox for second-order multivariate calibration, Chemom. Intell. Lab. Syst. 96 (2009) 246–251. https://doi.org/10.1016/j.chemolab.2009.02.005.

[10] A.C. Olivieri, H.L. Wu, R.Q. Yu, MVC3: A MATLAB graphical interface toolbox for third-order multivariate calibration, Chemom. Intell. Lab. Syst. 116 (2012) 9–16. https://doi.org/10.1016/j.chemolab.2012.03.018.

[11] P. Mishra, J.M. Roger, D.N. Rutledge, A. Biancolillo, F. Marini, A. Nordon, D. Jouan-Rimbaud-Bouveresse, MBA-GUI: A chemometric graphical user interface for multi-block data visualisation, regression, classification, variable selection and automated pre-processing, Chemom. Intell. Lab. Syst. 205 (2020) 104139. https://doi.org/10.1016/j.chemolab.2020.104139.

[12] http://michem.unimib.it/download/matlab-toolboxes/regression-toolbox-for-matlab/

[13] http://doi.org/10.5281/zenodo. 4663192

[14] D. Ruppert, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Science & Business Media, 2004. https://doi.org/10.1198/jasa.2004.s339.

[15] R. Bro, A.K. Smilde, Principal component analysis, Anal. Methods. 6 (2014) 2812–2831. https://doi.org/10.1039/c3ay41907j.

[16] S. Wold, M. Sjöström, L. Eriksson, PLS-regression: A basic tool of chemometrics, Chemom.

Intell. Lab. Syst. 58 (2001) 109–130. https://doi.org/10.1016/S0169-7439(01)00155-1.

[17] M. Andersson, A comparison of nine PLS1 algorithms, J. Chemom. 23 (2009) 518–529. https://doi.org/10.1002/cem.1248.

[18] I.S. Helland, On the structure of partial least squares regression, Commun. Stat. - Simul. Comput. 17 (1988) 581–607. https://doi.org/10.1080/03610918808812681.

[19] R. Todeschini, D. Ballabio, M. Cassotti, V. Consonni, N3 and BNN: Two New Similarity Based Classification Methods in Comparison with Other Classifiers, J. Chem. Inf. Model. 55 (2015) 2365–2374. https://doi.org/10.1021/acs.jcim.5b00326.

[20] I. Guyon, A. Elisseeff, An Introduction to Variable and Feature Selection, J. Mach. Learn. Res. 3 (2003) 1157–1182. https://doi.org/10.1016/j.aca.2011.07.027.

[21] R. Leardi, A. Lupiáñez González, Genetic algorithms applied to feature selection in PLS regression: How and when to use them, Chemom. Intell. Lab. Syst. 41 (1998) 195–207. https://doi.org/10.1016/S0169-7439(98)00051-3.

[22] R. Todeschini, V. Consonni, A. Mauri, M. Pavan, MobyDigs: Software for regression and classification models by genetic algorithms, Data Handl. Sci. Technol. 23 (2003) 141–167. https://doi.org/10.1016/S0922-3487(03)23005-7.

[23] A.J. Miller, Selection of Subsets of Regression Variables, J. R. Stat. Soc. Ser. A. 147 (1984) 389. https://doi.org/10.2307/2981576.

[24] F. Grisoni, M. Cassotti, R. Todeschini, Reshaped sequential replacement for variable selection in QSPR: Comparison with other reference methods, J. Chemom. 28 (2014) 249–259. https://doi.org/10.1002/cem.2603.

[25] V. Consonni, D. Ballabio, R. Todeschini, Comments on the definition of the Q2 parameter for QSAR validation, J. Chem. Inf. Model. 49 (2009) 1669–1678. https://doi.org/10.1021/ci900115y.

[26] V. Consonni, R. Todeschini, D. Ballabio, F. Grisoni, On the Misleading Use of QF32 for QSAR Model Comparison, Mol. Inform. 38 (2019) 1800029. https://doi.org/10.1002/minf.201800029.

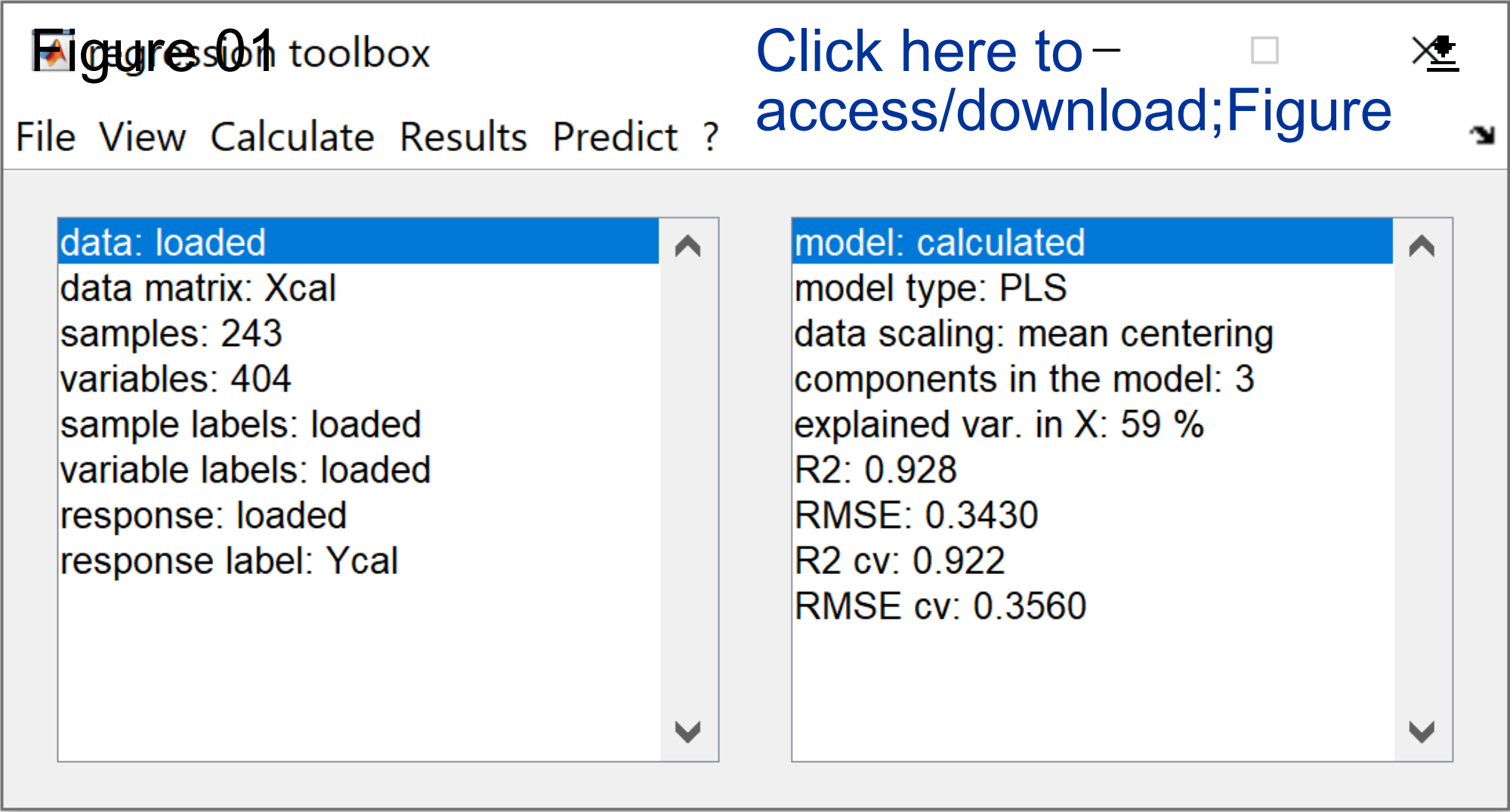
[27] A. Robinson, R.D. Cook, S. Weisberg, Residuals and Influence in Regression., New York: Chapman and Hall, 1984. https://doi.org/10.2307/2981746.

[28] A.K. Conlin, E.B. Martin, A.J. Morris, Confidence limits for contribution plots, J. Chemom. 14 (2000) 725–736. https://doi.org/https://doi.org/10.1002/1099-128X(200009/12)14:5/6<725::AID-CEM611>3.0.CO;2-8.

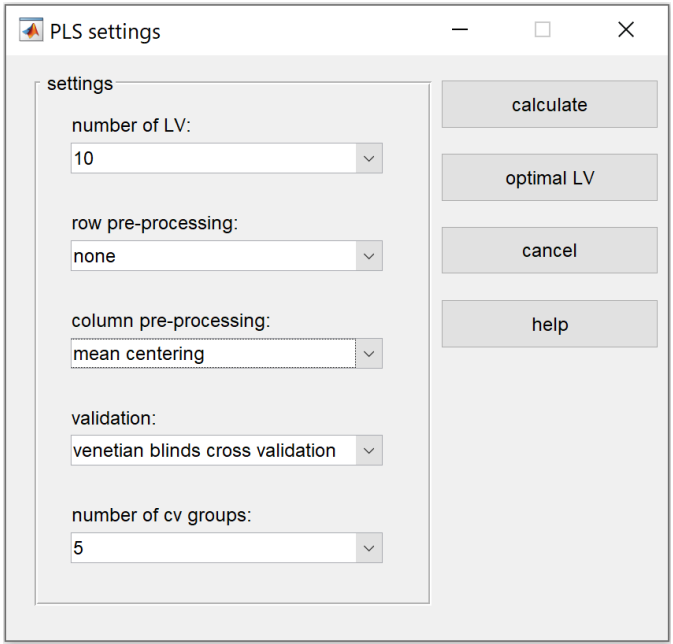
[29] P. Filzmoser, B. Liebmann, K. Varmuza, Repeated double cross validation, J. Chemom. 23 (2009) 160–171. https://doi.org/10.1002/cem.1225.

[30] M. Dyrby, S.B. Engelsen, L. Nørgaard, M. Bruhn, L. Lundsberg-Nielsen, Chemometric quantitation of the active substance (containing C≡N) in a pharmaceutical tablet using near-infrared (NIR) transmittance and NIR FT-Raman spectra, Appl. Spectrosc. 56 (2002) 579–585. https://doi.org/10.1366/0003702021955358.

Figure 01


regression toolbox
Click here to access/download;Figure
File View Calculate Results Predict ?
data: loaded
data matrix: Xcal
samples: 243
variables: 404
sample labels: loaded
variable labels: loaded
response: loaded
response label: Ycal
model: calculated
model type: PLS
data scaling: mean centering
components in the model: 3
explained var. in X: 59 %
R2: 0.928
RMSE: 0.3430
R2 cv: 0.922
RMSE cv: 0.3560

a) Figure 02



b) Click here to access/download;Figure;fig2.pdf

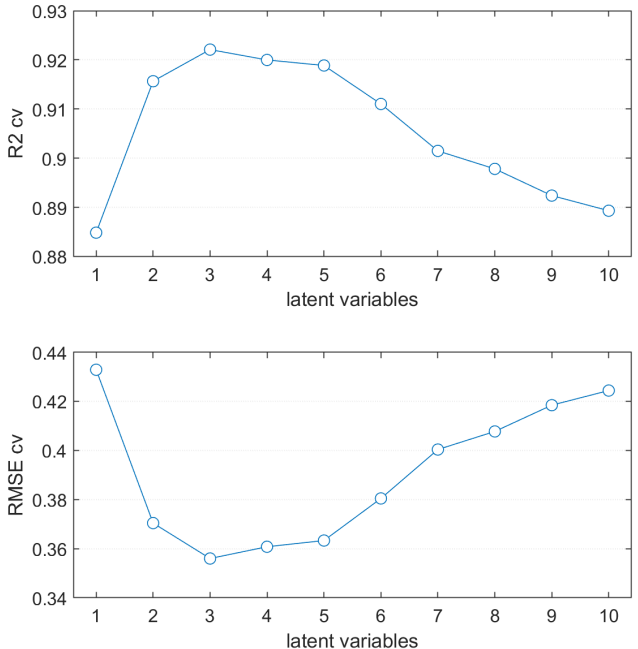
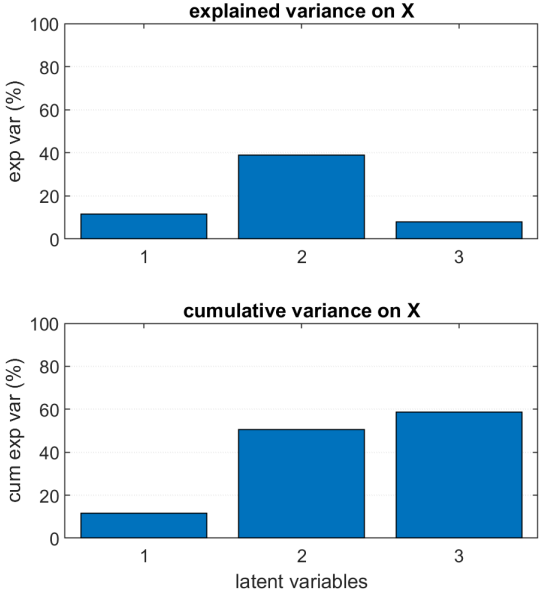


Figure 03

a)



b)

Click here to access/download;Figure;fig3.pdf

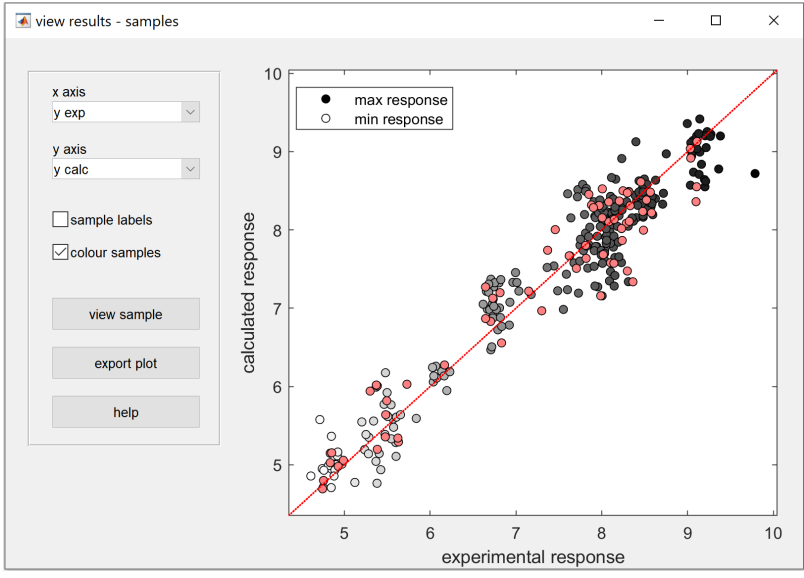
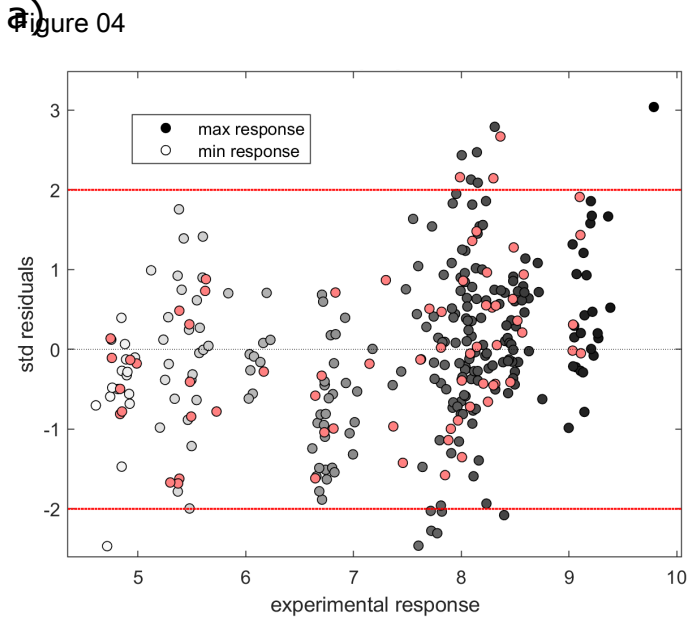
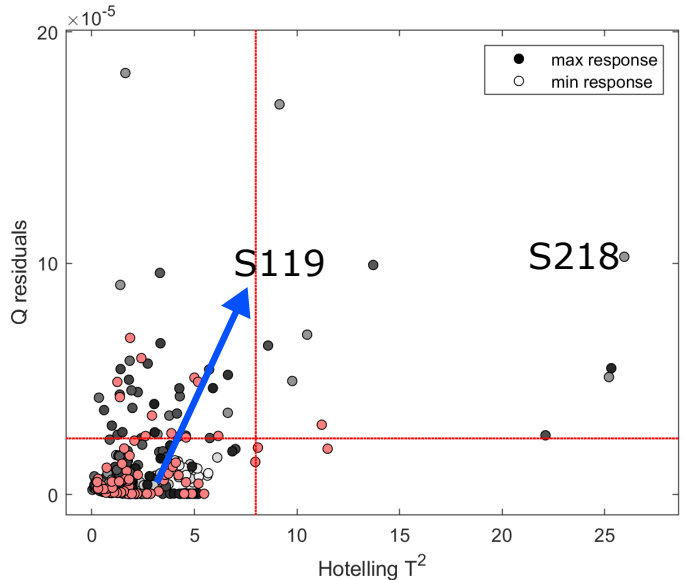


Figure 04



b) Click here to access/download;Figure;fig4.pdf



a) Figure 05

b) Click here to access/download;Figure;fig5.pdf

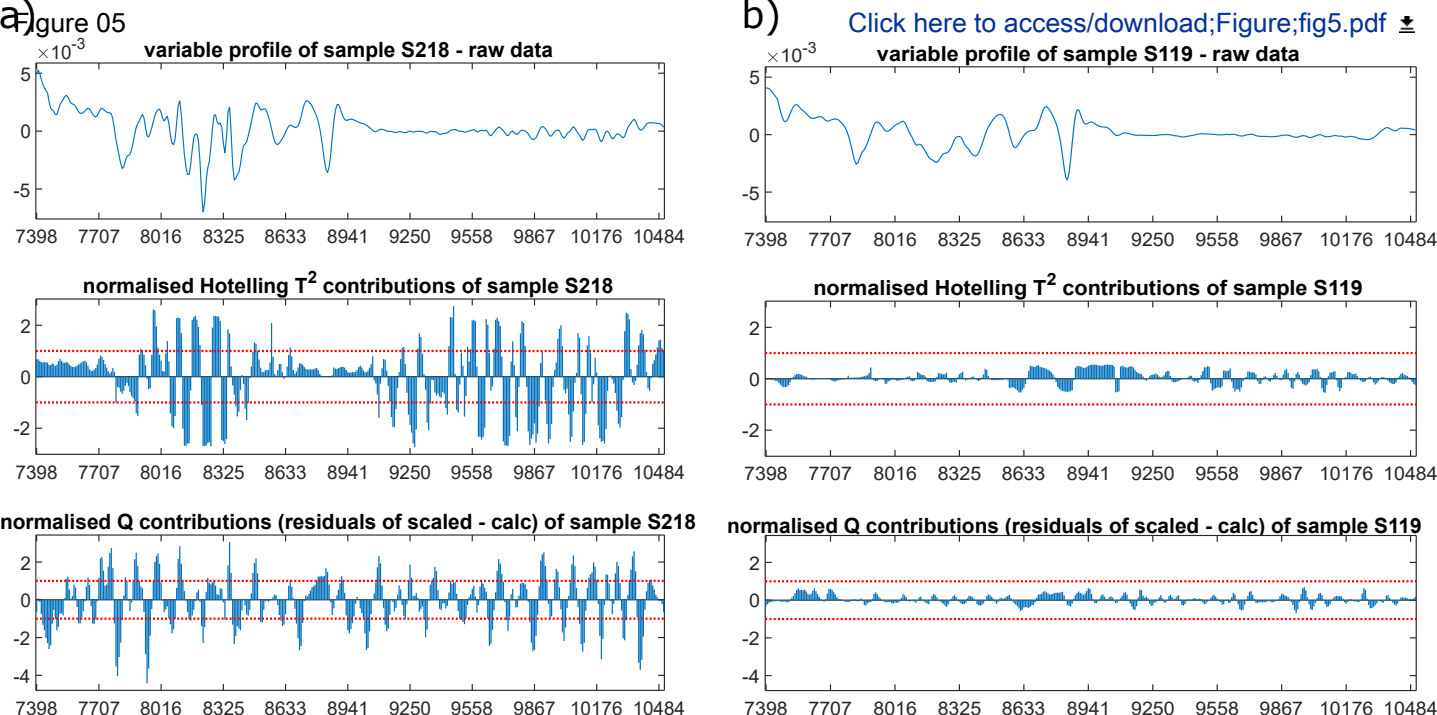
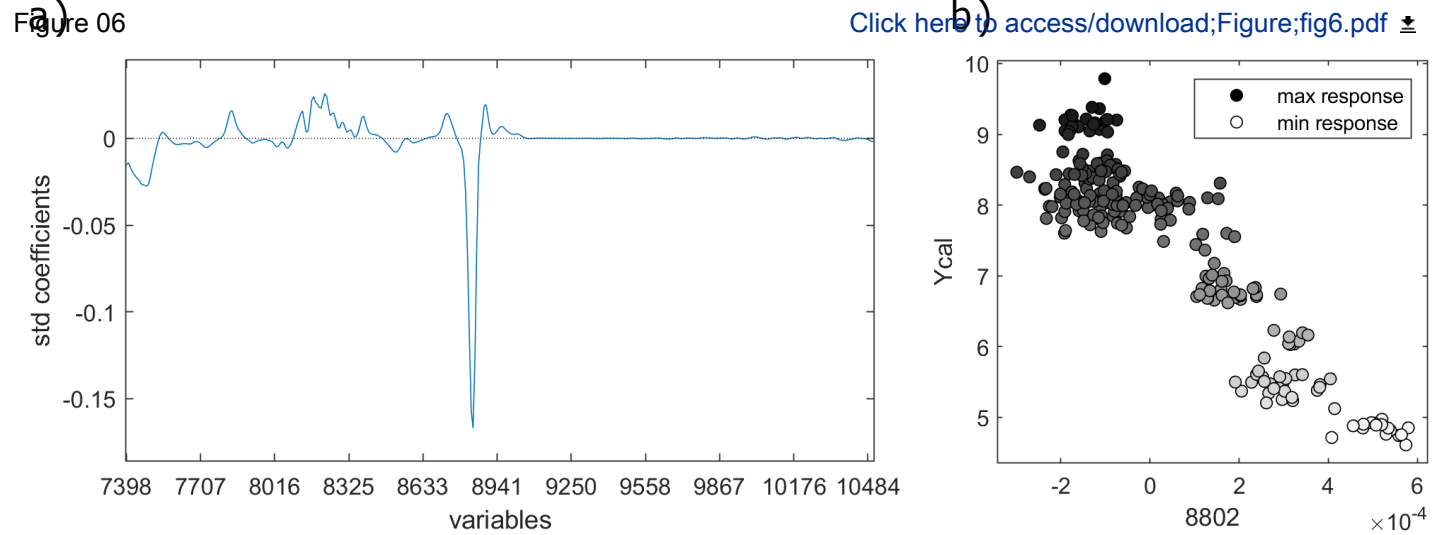


Figure 06



variable selection

Figure 07

Click here to access/download;Figure;fig

all subsets selection

forward selection

Use variables with frequency of selection higher than:

0.25

cancel

help

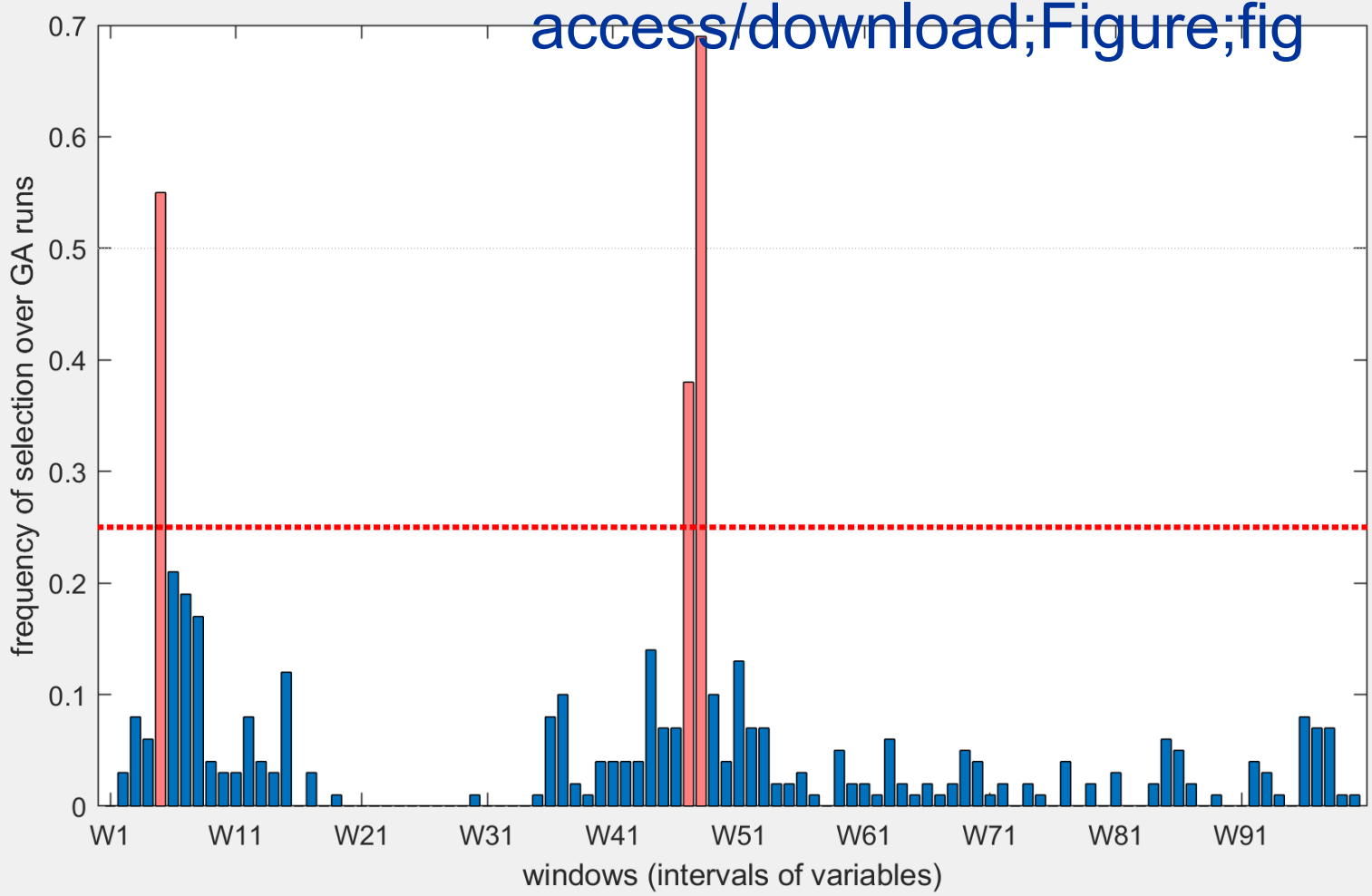


Table 1. MATLAB routines for the calculation and validation of regression models and their output. For each routine, the output is collected as a field of a unique MATLAB structure. I is the number of training samples, I_T is the number of test samples, J the number of variables, M is the number of retained components/latent variables for PLS and PCR, K is the number of neighbours for KNN.

MATLAB routine	Description	Output	Description
olsfit, ridgefit, plsfit, pcrfit, knnfit, bnnfit	fitting of OLS, Ridge, PLS, PCR, and local models based on KNN and BNN	yc	calculated response vector ($I \times 1$)
		reg_param	regression quality metrics (R^2, *RMSE*)
		b, b_std	coefficients, standardised coefficients ($J \times 1$)
		r, r_std	residuals, standardised residuals ($I \times 1$)
		H	leverages ($I \times 1$)
		T*	scores ($I \times M$)
		L*	loadings ($J \times M$)
		exp_var*	explained variance % ($M \times 1$)
		cum_var*	cumulative variance % ($M \times 1$)
		Thot*	Hotelling's T^2 ($I \times 1$)
		Tcont*	Hotelling's T^2 contributions ($I \times J$)
		Qres*	Q residuals ($I \times 1$)
		Qcont*	Q residual contributions ($I \times J$)
		neighbours**	neighbours ID ($I \times K$)
		D**	distance matrix ($I \times I$)
		setting	structure with settings used to calculate model (scaling parameters, Hotelling's T^2 and Q confidence limits)
olscv, ridgecv, plscv, pcrcv, knncv, bnncv	cross-validation, bootstrap and Montecarlo procedures	yc	predicted responses by cross-validation ($I \times 1$)
		reg_param	regression metrics in validation (R^2_{cv}, *RMSECV*)
		settings	settings used for cross-validation
olspred, ridgepred, plspred, pcrpred, knnpred, bnnpred	prediction for new samples	yc	predicted responses for test samples ($I_T \times 1$)
		H	leverages ($I_T \times 1$)
		T*	scores ($I_T \times M$)
		Thot*	Hotelling's T^2 ($I_T \times 1$)
		Tcont*	Hotelling's T^2 contributions ($I_T \times J$)
		Qres*	Q residuals ($I_T \times 1$)
		Qcont*	Q residuals contributions ($I_T \times J$)
		neighbours**	neighbours ID ($I_T \times K$)
		D**	distance matrix ($I_T \times I$)

* only for PLS and PCR
** only for KNN and BNN