Available online at www.sciencedirect.com

ScienceDirect

Chemometrics and Intelligent Laboratory Systems 85 (2007) 269 – 277

Chemometrics and intelligent laboratory systems

www.elsevier.com/locate/chemolab

Software description

TOMCAT: A MATLAB toolbox for multivariate calibration techniques

Michał Daszykowski [a], Sven Serneels [b], Krzysztof Kaczmarek [a], Piet Van Espen [b], Christophe Croux [c], Beata Walczak [a,*]

[a] *Department of Chemometrics, The University of Silesia, 9 Szkolna Street, 40-006 Katowice, Poland*
[b] *Micro and Trace Analysis Centre, Universiteit Antwerpen, Universiteitsplein 1, B-2610 Wilrijk, Belgium*
[c] *Faculty of Economics and Applied Economics, K.U. Leuven, Naamsestraat 69, B-3000 Leuven, Belgium*

Received 8 February 2006; received in revised form 27 February 2006; accepted 18 March 2006
Available online 9 May 2006

Abstract

We have developed a new user-friendly graphical interface for robust calibration with a collection of m-files, called TOMCAT (*TO*olbox for *M*ultivariate *CA*libration *T*echniques). The graphical interface and its routines are freely available and programmed in MATLAB 6.5, probably one of the most popular programming environments in the chemometrics community. The graphical interface allows a user to apply the implemented methods in an easy way and it gives a straightforward possibility to visualize the obtained results. Several useful features such as interactive numbering of the displayed objects on a plot, viewing the content of the data, easy transfer of the data between the toolbox and the MATLAB workspace and vice versa, are also implemented. Among the implemented methods there are Principal Component Analysis and its robust variant, Partial Least Squares, Continuum Power Regression, Partial Robust M-Regression, Robust Continuum Regression and Radial Basis Functions Partial Least Squares.

Keywords: Partial Robust M-Regression; Robust Continuum Regression; Multivariate calibration; Nonlinear modeling; Radial Basis Functions Partial Least Squares

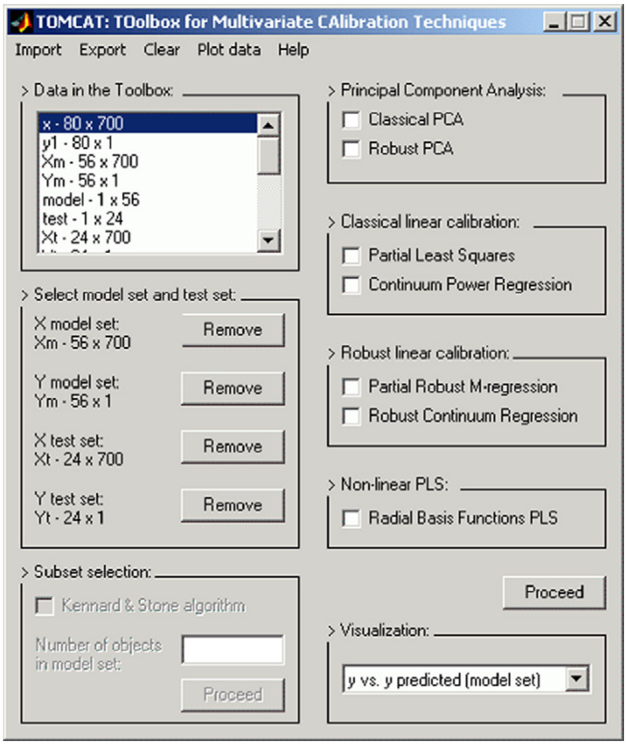
1. Introduction

Chemical data sets are usually multidimensional, complex and often contain more measured parameters than observations. Spectroscopic data are typical examples of such data. This is why in chemometrics latent variables methods are used to explore the information contained in the data. The most popular among them are Principal Component Analysis, PCA, allowing data compression and latent variables modeling techniques such as Principal Component Regression, PCR, and Partial Least Squares, PLS [1–3]. One difficulty that may arise while exploring and modeling the chemical data is a presence of outlying observations. In general, outlying observations are objects that have unique characteristics compared to the data majority. The outlying objects strongly affect all of the least squares methods, including PCA, PCR and PLS [4,5]. Therefore, it is important to detect them, and if necessary, to remove from the data. Another possibility to handle outliers in the data is to use so-called robust approaches that can provide reliable estimates even if outliers are present in the data [4]. Over several years, many robust versions of the classical chemometrical approaches have been proposed such as robust PCA [6–9], robust PCR [10,11], robust PLS [12–14], etc. Although in statistics robust methods have been widely accepted, their use in chemistry is rather limited. Therefore, the goal of our work is to popularize recently proposed robust methods, in particular Partial Robust M-Regression (PRM) [14], by offering to the public a collection of several classical and robust approaches. Another library of robust routines, called LIBRA, has been provided by Verboven et al. [15,16]. To facilitate the use of the implemented methods, we have developed a graphical interface that gives a user possibility to apply the methods in a straightforward way and to visualize the results.

In this article, we focus our attention on presenting the graphical interface and its features. Among others, users will find different methods including classical and robust PCA, linear calibration models such as PLS and its robust variant PRM,

* Corresponding author.
E-mail address: beata@us.edu.pl (B. Walczak).

0169-7439/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.chemolab.2006.03.006


TOMCAT: TOolbox for Multivariate CAlibration Techniques
Import Export Clear Plot data Help
> Data in the Toolbox:
> Principal Component Analysis:
□ Classical PCA
□ Robust PCA
> Select model set and test set:
> Classical linear calibration:
□ Partial Least Squares
□ Continuum Power Regression
> Robust linear calibration:
□ Partial Robust M-regression
□ Robust Continuum Regression
> Non-linear PLS:
□ Radial Basis Functions PLS
> Subset selection:
□ Kennard & Stone algorithm
Number of objects in model set:
Proceed
Remove
> Visualization:

x - 80 x 700
y1 - 80 x 1
Xm - 56 x 700
Ym - 56 x 1
model - 1 x 56
test - 1 x 24
Xt - 24 x 700

X model set: Xm - 56 x 700
Y model set: Ym - 56 x 1
X test set: Xt - 24 x 700
Y test set: Yt - 24 x 1

Visualization: y vs. y predicted (model set)

Fig. 1. The graphical user interface for robust calibration.

Continuum Power Regression (CPR), Robust Continuum Regression (RCR) and nonlinear calibration approach called Radial Basis Functions Partial Least Squares (RBF-PLS) [17,18]. The description of the implemented methods and the obtained results are reported in the related publications.

2. Software specifications and requirements

The routines for robust calibration and the graphical interface have been developed under MATLAB 6.5 (release 13) [19]. Dependent on the user's knowledge about MATLAB, either the m-files or the graphical interface can be used.

The interface routine, called 'TOMCAT', requires for its proper functioning a specific structure of the catalogues where the routines are located. Apart from the main interface, additional interfaces are designed for defining inputs and custom options for the applied methods. It ought to be mentioned that they are a part of the interface only. The following catalogues with m-files are automatically made when the zip file is extracted: 'Calibration', 'Data', 'Interfaces', 'PCA', 'Preprocessing' and 'Subset _ Selection'.

To initialize the graphical interface, the current directory of MATLAB should be changed to the directory where the graphical interface is installed, by using the command 'cd'. Typing the command 'TOMCAT' in the MATLAB command window executes the graphical interface. When the graphical interface starts, the information about the location of the required m-files is added automatically into the MATLAB paths. Executing the graphical interface file results in displaying the interface window on the screen of the computer. After selecting the data and applying any of the implemented methods in the toolbox, the graphical interface looks similar to the one presented in Fig. 1.

The toolbox for robust calibration, is available from the two internet sources [20,21] as a compressed zip-file. Additional information about the implemented methods in the toolbox can be found at [22,23].

3. Collection of the implemented methods

The collection of m-files (implemented in the graphical interface methods) covers aspects of the data analysis such as classical and robust data preprocessing, subset selection, classical, robust as well as nonlinear calibration.

Among the classical data preprocessing methods there are mean column centering, column autoscaling, and Standard Normal Variate [24]. For robust preprocessing purposes, median column centering, L1-median column centering [25] and a standardization based on Sn and Qn scale estimators [26] are included.

Data compression and visualization can be performed with Principal Component Analysis (working always on the smaller data dimension, which speeds up the computations) [27] and its robust version based on the Projection Pursuit algorithm [28].

For classical calibration, a user may choose between PLS (for tall and wide data matrices, WIM-PLS and SIM-PLS) [29] and CPR [30,31]. The WIM-PLS, SIM-PLS [32] and CPR [33] routines are included into the toolbox upon an agreement of their authors. When outliers are present in the data, robust calibration methods such as Partial Robust M-Regression [14] and Robust Continuum Regression [34] can be applied.

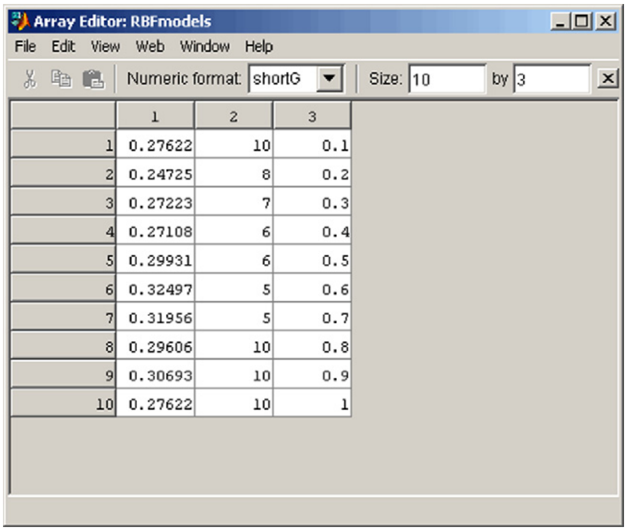
In order to evaluate the complexity of the constructed calibration model, Cross-Validation routine is supplied with two options: a standard leave-n-out Cross-Validation and Monte Carlo Cross-Validation [35,36].

To model a nonlinear relationship between **X** and **y**, Radial Basis Functions Partial Least Squares approach is proposed.

4. Working with the graphical interface

The graphical interface is composed of eight panels and an upper window menu. There are the following panels: 'Data in the Toolbox', 'Select model set and test set', 'Subset selection', 'Principal Component Analysis', 'Classical linear calibration', 'Robust linear calibration', 'Nonlinear PLS' and 'Visualization'. The upper menu of the window contains at most five sub-menus, dependent on the data content in the graphical interface. These folders are 'Import', 'Export', 'Clear', 'Plot data' and 'Help'.

The graphical interface has been designed in such a way that certain options or methods can only be used if appropriate data are available and selected by a user. For instance, the PCA can be performed on the set of independent variables, **X**. Only if both **X** and a dependent variable **y** are present in the toolbox, and if both are selected, then the calibration methods can be used. There are also several checking procedures preserving unwanted action of a user, for instance, selecting twice the same data, setting the inputs of the methods at unacceptable levels, etc.



Numeric format: shortG Size: 10 by 3

	1	2	3
1	0.27622	10	0.1
2	0.24725	8	0.2
3	0.27223	7	0.3
4	0.27108	6	0.4
5	0.29931	6	0.5
6	0.32497	5	0.6
7	0.31956	5	0.7
8	0.29606	10	0.8
9	0.30693	10	0.9
10	0.27622	10	1

Fig. 2. An example of the data content displayed in the MATLAB array editor.

4.1. Loading, exporting and browsing the data

To load the data into the graphical interface, select from the upper window menu a folder ‘Import’ and chose the location of the data: ‘Import data from ∗.mat file’ or ‘Import data from workspace’. The last option is available only if the data are present in the MATLAB workspace. When the file or the workspace contains several variables then the user is asked to select variables that should be imported to the graphical interface.

What should be stressed is that the graphical interface can only handle MATLAB files, i.e., files with ‘mat’ extension, and variables such as vectors and matrices, being double arrays. When the variables are loaded into the graphical interface, the information including their names and sizes (number of objects and variables) is displayed in ‘Data in the Toolbox’ panel.

At any time a user can easily export the obtained results either to a ‘mat’ file or to the MATLAB workspace, by selecting from the upper window menu an option ‘Export’ and choosing the destination: ‘Export data to ∗.mat file’ or ‘Export data to workspace’. This gives a possibility to apply other methods that are not available in the graphical interface on the saved or exported data, or to create custom plots.

Additionally, we have included in the graphical interface an option for browsing the content of the data. By double clicking with the left mouse button on the highlighted variable in the ‘Data in the Toolbox’ panel, its content is displayed as a data sheet in the MATLAB Array Editor (see Fig. 2). Simultaneously, the selected variable is exported to the workspace. However, due to the limitation of the Array Editor, only the variables containing at most 65,536 elements can be viewed.

4.2. Clearing the graphical interface and the MATLAB workspace

The variables in the graphical interface and/or the workspace can be erased by selecting from the upper window menu a folder ‘Clear’ and then ‘Clear data from Toolbox’ or ‘Clear data from workspace’. Erasing the variables from the graphical interface will result in disabling previously active options, and at this stage the user is allowed to import the data only.

4.3. Selecting the data

Once the data are loaded into the graphical interface, it is necessary to select a set of independent variables, **X**, and optionally independent variables, **y** (both variables should have the same number of objects, and **X** should be a matrix, otherwise an error dialog box appears on the screen). This can be done using the buttons from ‘Select model set and test set’ panel. There are four buttons for selecting model set data—independent variables (denoted as Xm), a dependent model set variable (denoted as Ym) and, if necessary, test set independent variables (denoted as Xt) and a dependent test set variable (denoted as Yt). As soon as the model set independent variables are selected, a new variable Xm, appears in the ‘Data in the Toolbox’ panel and the ‘Principal Component Analysis’ panel becomes active allowing to apply PCA and robust PCA on this data. Anytime, the user may remove a selected variable by clicking on the corresponding button ‘Remove’ and to re-select a new one. In the same way, the dependent model set variable can be chosen, and then, four additional panels become active (‘Subset selection’, ‘Classical linear calibration’, ‘Robust linear calibration’ and ‘Nonlinear PLS’).

Often, for the purpose of calibration, the constructed model is validated with the independent test set. The test set can be selected in the graphical interface in two possible ways: either by importing an external test set to the graphical interface or by splitting the selected model set with the Kennard and Stone algorithm [37]. The aim of this algorithm is to design a model set in such way that the objects are scattered uniformly around the calibration domain. In this way, all sources of the data variance are included into the calibration model. By selecting the Kennard and Stone algorithm from the ‘Subset selection’ panel, the selected model set is by default split into two new sets: a model set containing approximately 70% of the total number of objects in the data, and the test set containing the remaining samples. The user is given also the possibility to define the cardinality of the model set by imputing a custom number. After clicking the button ‘Proceed’ the algorithm is executed and the ‘waitbar’ showing the progress of the algorithm is displayed on the screen. The ‘waitbar’ feature was implemented for robust PCA, for the Cross-Validation routine and for RBF-PLS. It gives the user an idea about the remaining computation time, since depending on the data size, the computational cost can vary to a high extent.

4.4. A short presentation of methods inputs, outputs and visualization options

Aside from the main graphical interface, the implemented methods have separate interfaces allowing to specify custom parameters and options. For calibration methods, dependent on the applied method, the user may select a preprocessing method, the number of components in the model, the percentage of the data contamination, the type of Cross-Validation, etc. The interfaces of the calibration methods are presented in Fig. 3.

4.4.1. Principal Component Analysis

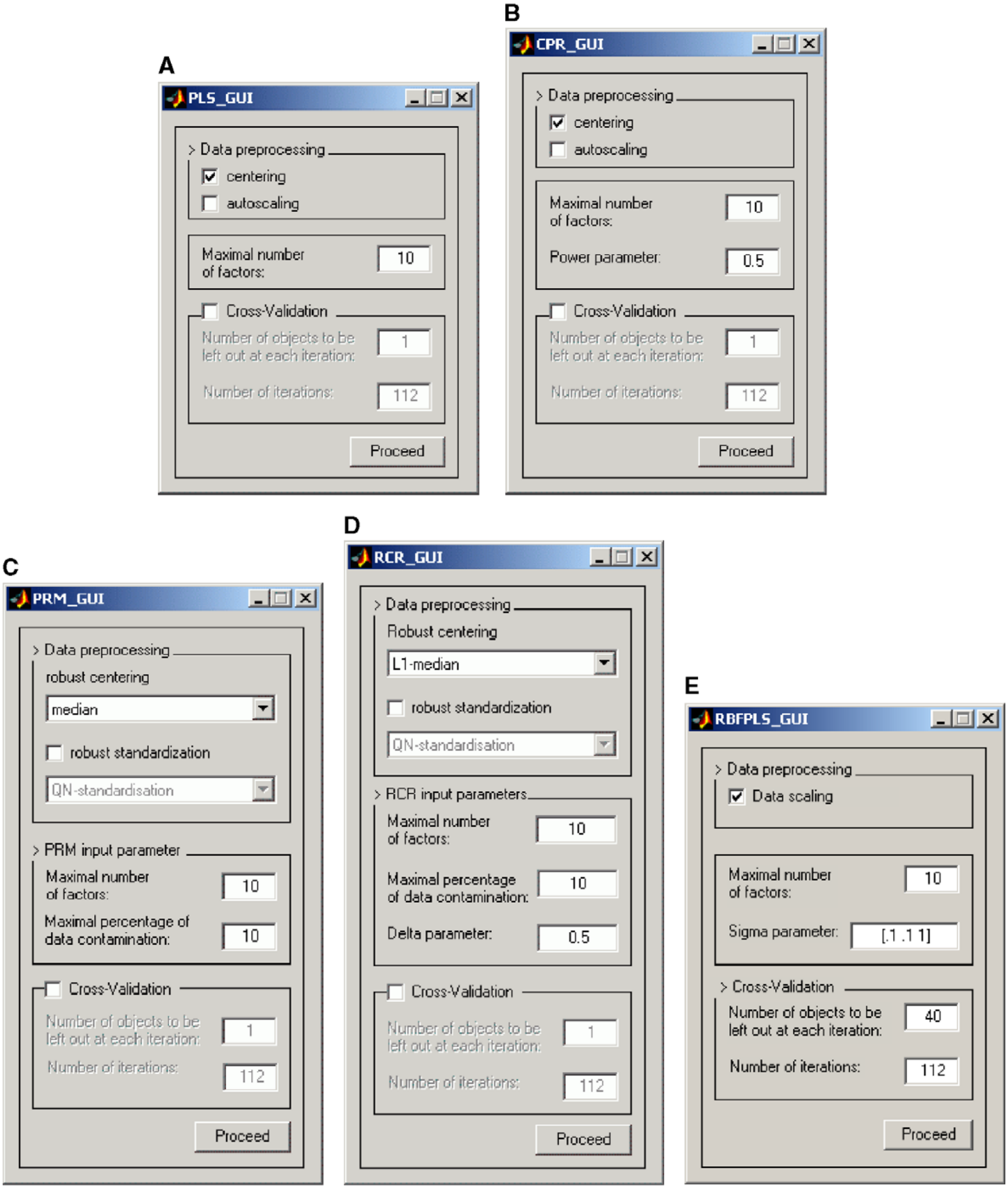
Before performing PCA the data set can be preprocessed with several preprocessing methods including column mean centering, autoscaling and standard normal variate transformation.

After executing PCA the following outputs are obtained and stored in the toolbox:

□ *pc*, a matrix containing scores in columns,
□ *s*, a matrix containing normalized scores in columns,
□ *v*, a vector of singular values,
□ *d*, a matrix containing in columns loadings,
□ *pr*, a vector with percent of variance explained by each principal component.

Based on the obtained outputs, score plots, loading plots, an eigenvalue scree-plot and a bar plot of the explained data variance can be constructed. An example of such plots is given in Fig. 4.

In all types of scatter plots, we implemented a feature allowing to add a number to every object by clicking on the



A
> Data preprocessing
☑ centering
□ autoscaling
Maximal number of factors: 10
□ Cross-Validation
Number of objects to be left out at each iteration: 1
Number of iterations: 112

B
> Data preprocessing
☑ centering
□ autoscaling
Maximal number of factors: 10
Power parameter: 0.5
□ Cross-Validation
Number of objects to be left out at each iteration: 1
Number of iterations: 112

C
> Data preprocessing
robust centering: median
□ robust standardization
QN-standardisation
> PRM input parameter
Maximal number of factors: 10
Maximal percentage of data contamination: 10
□ Cross-Validation
Number of objects to be left out at each iteration: 1
Number of iterations: 112

D
> Data preprocessing
Robust centering: L1-median
□ robust standardization
QN-standardisation
> RCR input parameters
Maximal number of factors: 10
Maximal percentage of data contamination: 10
Delta parameter: 0.5
□ Cross-Validation
Number of objects to be left out at each iteration: 1
Number of iterations: 112

E
> Data preprocessing
☑ Data scaling
Maximal number of factors: 10
Sigma parameter: [.1 .1 1]
> Cross-Validation
Number of objects to be left out at each iteration: 40
Number of iterations: 112

Fig. 3. Different interfaces for: (A) Partial Least Squares, (B) Continuum Power Regression, (C) Partial Robust M-Regression, (D) Robust Continuum Regression and (E) Nonlinear PLS (Radial Basis Functions Partial Least Squares).

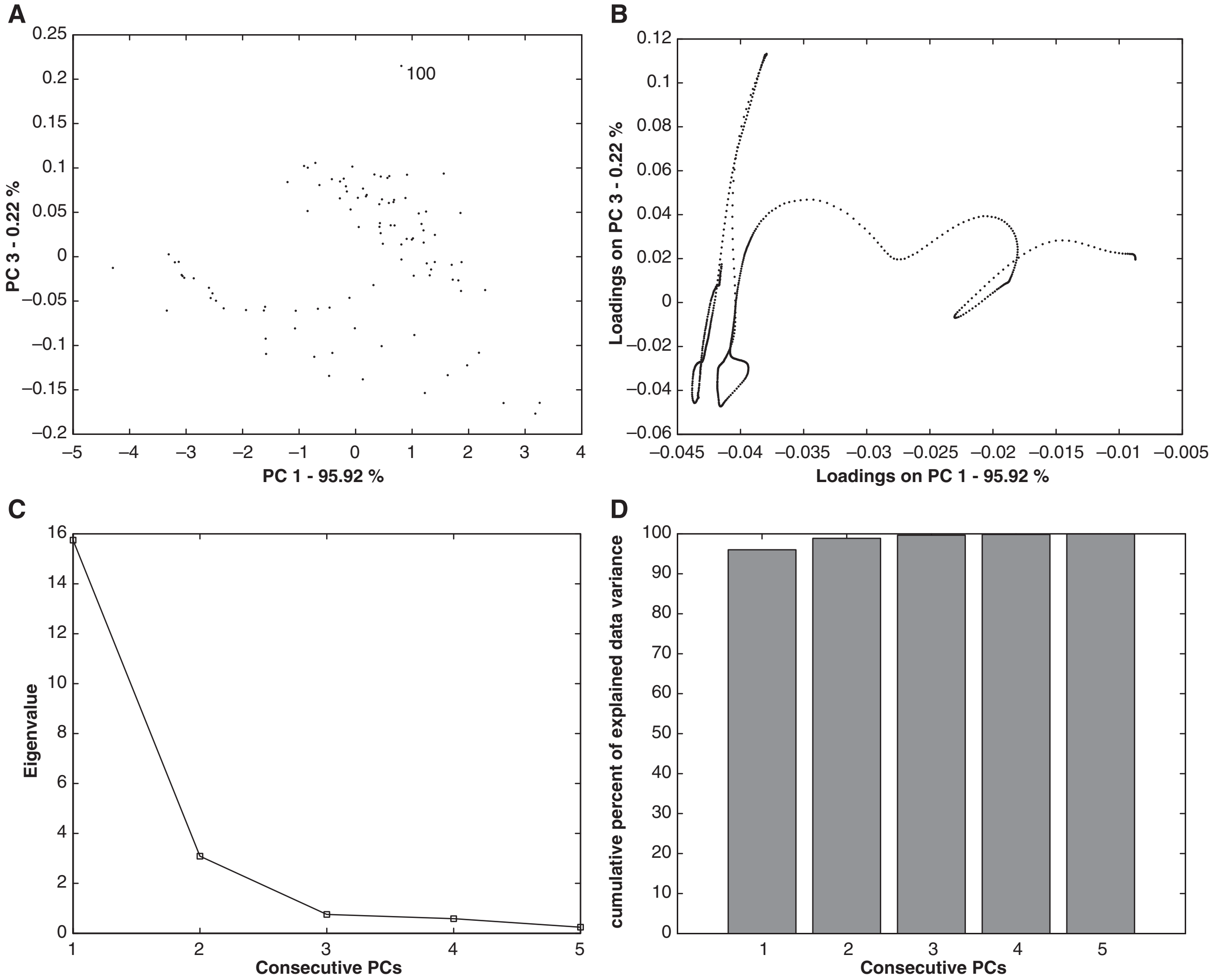


Fig. 4. Plots obtained after PCA: (A) projection of objects on the plane defined by PC 1 and PC 3, (B) projection of loadings on the plane defined by PC 1 and PC 3, (C) eigenvalue scree plot and (D) cumulative percentage of explained data variance.

object with the left mouse button. In order to remove object number, the right mouse button should be used.

4.4.2. Robust Principal Component Analysis

Before carrying out the robust PCA (rPCA), the user can apply to the data several types of preprocessing: L1-median column centering [25], Sn and Qn autoscaling [26] and standard normal variate transformation.

Applying robust PCA leads to the following outputs:

□ *rpc*, a matrix containing in its columns the robust principal components,

□ *rv*, a vector of robust singular values,

□ *rd*, a matrix of robust loadings,

□ *ROD*, a matrix (*n*,*f*) of *z*-transformed (standardized) Robust Orthogonal Distances calculated for different rPCA models with an increasing number of components. All the distances obtained with the rPCA model of certain complexity are centered around the median and the absolute values of the centered distances are divided by their corresponding robust Qn-scale,

□ *RD*, a matrix (*n*,*f*) of *z*-transformed Robust Distances, in the same manner as RODs, calculated for different rPCA models with an increasing number of components,

□ *exRD*, a vector with indices of objects with *z*-transformed Robust Distances (obtained for the selected number of robust PCs in the rPCA model) exceeding a cutoff value, i.e., 3,

□ *exROD*, a vector with indices of objects with *z*-transformed Robust Orthogonal Distances (obtained for the selected number of robust PCs in the rPCA model) exceeding cutoff value, i.e., 3.

The obtained outputs from the robust PCA allow construction of similar plots as presented ones in panels of Fig. 3. Additionally, using Robust Distances and Robust Orthogonal Distances, a distance–distance plot can be made, facilitating identification of outlying samples (see Fig. 5). For the *i*-th sample, the Robust Distance, RD_i, is defined as follows:

$$\mathrm{RD}_i = \sqrt{\sum_{j=1}^{n} \left(\frac{t_{ij}^{\mathrm{R}}}{s_j^{\mathrm{R}}} \right)^2} \qquad (1)$$

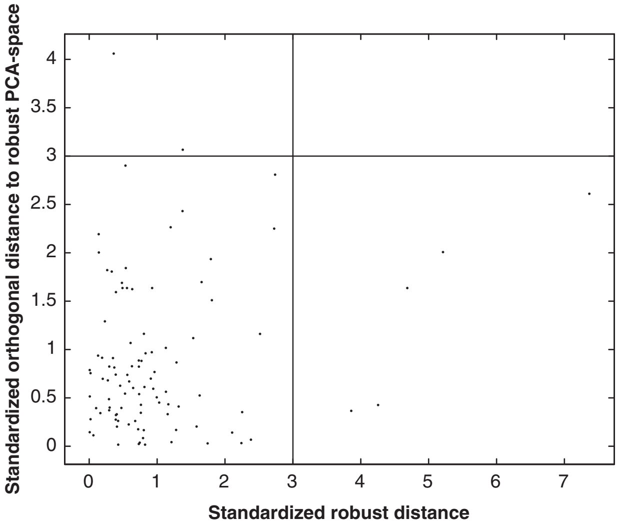


Fig. 5. Distance–distance plot obtained with robust PCA.

where t_{ij}^{R} are the elements of the robust score matrix and s_j^{R} is the squared root of the j-th robust eigenvalue.

The Orthogonal Distances are obtained as:

$$\mathrm{OD}_i = ||\mathbf{x}_i - \mathbf{P}_f \mathbf{t}_{i,f}^{\mathrm{T}}|| \qquad (2)$$

where $\mathbf{t}_{i,f}$ is the i-th score vector with f elements and $\mathbf{P}_f$ is a matrix (n,f) of f robust loadings.

To detect outlying observation one can use z-transformed distances, i.e., to center every vector with distances around median and to divide all elements by corresponding Qn-scale of the distances. For z-transformed distances, a general cutoff value is 3, it means that all objects with a z-transformed distance above 3 can be considered as outliers.

4.4.3. Classical modeling with Partial Least Squares and Continuum Power Regression

Before applying PLS or PCR, the user may specify the type of the data preprocessing (centering or standardization), the number of components in the model and the type of Cross-Validation: classical leave-n-out, by setting in the field 'Number of iterations' value 0, or Monte Carlo. By default the number of iterations is set to be twice the number of objects in the data, see Fig. 3a and b. In CPR the continuum parameter 'power' can be specified, and multiple values of the 'power' parameter are allowed if the Cross-Validation is carried out. The 'power' parameter can take values from the interval [0 1]. By default it is set to 0.5. Multiple values of the 'power' parameter, taking as input a vector like [0 0.1 0.3 0.6], are only allowed if Cross-Validation is performed. The obtained values of RMSECV, are trimmed according to the assumed fraction of data contamination and organized into a matrix, where in rows and columns are the values of RMSECVs obtained based on a model with h factors and the i-th 'power' parameter, respectively.

By applying Partial Least Squares or Continuum Power Regression to the selected data the following outputs are obtained:

□ *RMS*, a scalar, trimmed Root Mean Squared Error according to the assumed fraction of data contamination,

□ *RMSEP*, a scalar, Root Mean Squared Error of Prediction (if the test set is selected),

□ *RMSECV*, a scalar, trimmed Root Mean Squared Error of Cross-Validation (if the Cross-Validation is performed) according to the assumed fraction of data contamination,

□ *h*, a scalar, the number of factors in the calibration model,

□ *power*, a scalar or vector with values of the '*power*' parameter in CPR,

□ *Ymp*, a vector, predicted dependent variable for the model set,

□ *Ytp*, a vector, predicted dependent variable for the test set (if the test set is selected),

□ *b*, a vector, regression coefficients,

□ *T*, a matrix of PLS or CPR factors,

□ *E*, a matrix containing in each column model set residuals from a model with 1, 2, 3 …, f factors,

□ *Et*, a matrix containing in each column test set residuals of a model with 1, 2, 3 …, f factors.

The following plots are available if the model set and test set are selected: a plot of the Root Mean Squared Error of Cross-Validation (if Cross-Validation is performed), plots of the selected pairs of scores, a plot of observed y vs. y predicted for model and test sets, a plot of regression coefficients, a color map of absolute values of residuals for model and test sets for models of different complexity, a plot of absolute values of residuals of the model and test sets and a distance–distance plot. When the distance–distance plot is created, the following additional variables are computed and introduced into the data content in the graphical interface:

□ *rd*, a vector with z-transformed absolute residuals for each object,

□ *sd*, a vector with z-transformed distances in the space of factors PLS or CPR for each object,

□ *crd*, a scalar, a cut-off value for z-transformed residuals,

□ *csd*, a scalar, a cut-off value for z-transformed distances in the space of PLS or CPR factors.

Some of the figures that are typical for the calibration techniques are presented in Fig. 6.

4.4.4. Modeling with Partial Robust M-Regression

In PRM, the user ought to preprocess the data in a robust way using the classical median or the L1-median, and possibly Qn or Sn standardization. Additionally, it is possible to specify the percent of the data contamination. The fraction of contamination is taken into account during Cross-Validation procedure. The assumed fraction of objects with the highest values for the residuals is rejected and trimmed RMSECV is computed.

Applying Partial Robust M-Regression on the selected data will lead to additional output such as:

□ *wy*, a vector, containing y-weights of objects computed based on the y-residuals,

□ *wx*, a vector, containing x-weights of objects based on objects residuals computed in the scores space.

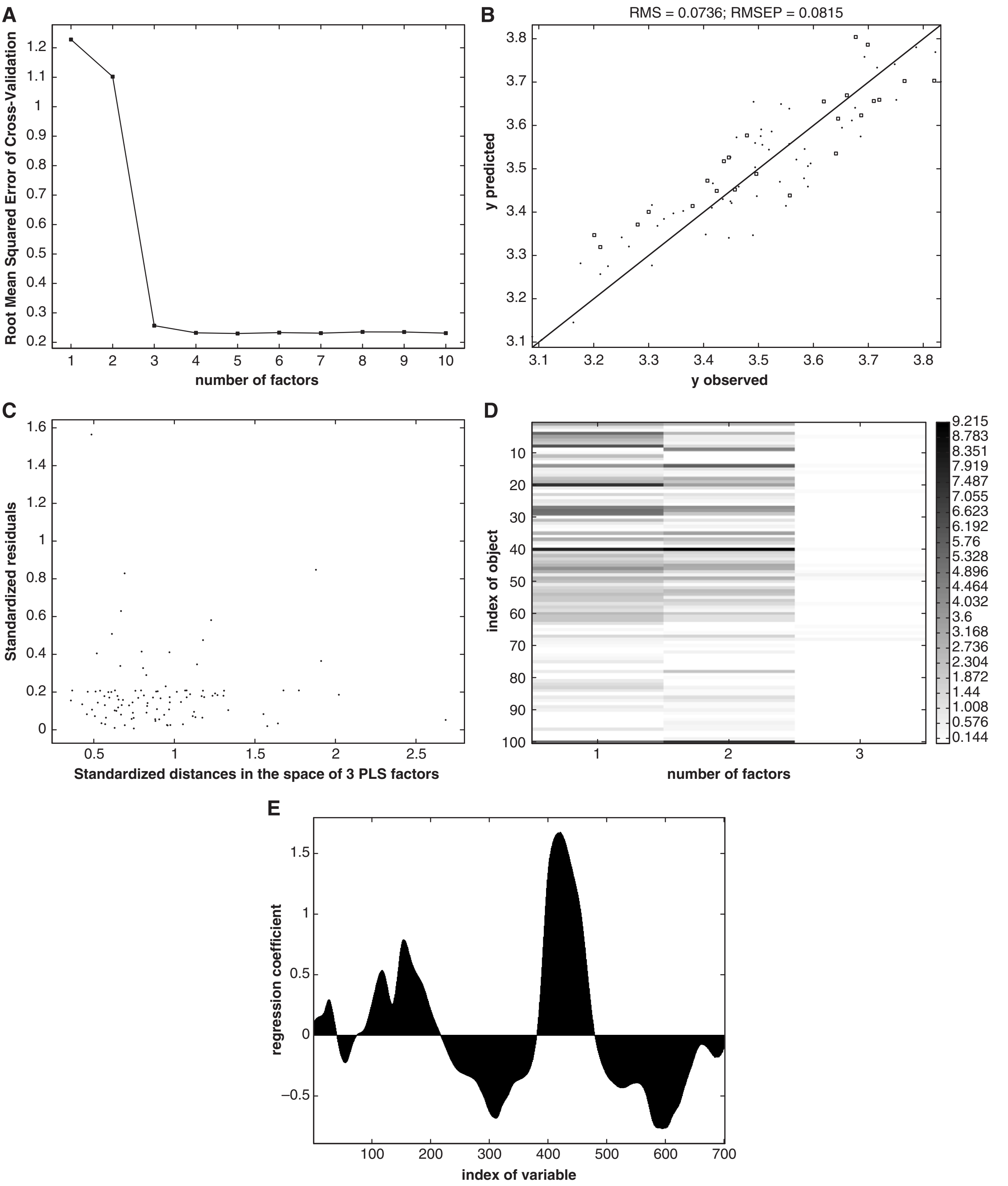


Fig. 6. Typical figures for calibration methods: (A) plot of Root Mean Squared Error of Cross-Validation, (B) *y* observed vs. *y* predicted for model set '·' and the test set '□', (C) distance–distance plot, (D) color map of residuals for model set for models with different number of factors and (E) plot of regression coefficients.

Additionally to the plots that can be constructed for PLS and CPR, the users have a possibility to inspect the *x*-weights and *y*-weights.

4.4.5. Modeling with robust continuum regression

In RCR, the data can be preprocessed with the same methods as in PRM. The delta parameter can be specified as a number from the interval [0 1] (by default it is set to 0.5). While Cross-Validation, several delta parameters can be considered, and then they should be introduced as a vector e.g., [0.1 0.2 0.3 0.5 0.7]. If several delta parameters are specified, the obtained values of RMSECV are organized into a matrix where in rows and in columns are the values of RMSECVs obtained with a model with *h* factors and the *i*-th delta parameter, respectively.

The remaining outputs are as the ones obtained in PLS and CPR. Also the figures that can be constructed are similar, except the RMSECV plot. If many delta parameters are tested then a color map can be plotted presenting the obtained values of RMSECV for models with different number of factors and different values of the delta parameter.

4.4.6. Nonlinear PLS for nonlinear modeling and classification

For nonlinear PLS, which can be used either for calibration or classification purposes, the Radial Basis Function-Partial Least Squares method is proposed [17,18]. For a continuous dependent variable Ym calibration is carried out, otherwise classification. In order to code classes the Ym vector can be a binary vector (in case of discrimination between two classes).

There are several inputs that should be specified by a user. Among them are: the option of scaling the data into the [0 1] interval, the maximal number of factors considered in the model, the different widths of Gaussians used for modeling (sigma parameter), Cross-Validation parameters including the number of objects to be left out in each step of the Cross-Validation procedure and the number of Cross-Validation iterations. To construct models with Gaussians of different widths the values of sigma should be inserted as follows [0.1 0.1 1]. This means that in the first model *m* Gaussians of width 0.1 are considered. For constructing the next nonlinear models, Gaussians with width of 0.1 wider than in the previous model up to Gaussian widths equal to 1 are used.

By applying nonlinear PLS for calibration to the selected data the following outputs are obtained and introduced to data window:

- □ *RMS*, a scalar, Root Mean Squared Error,
- □ *RMSEP*, a scalar, Root Mean Squared Error of Prediction (if the test set is selected),
- □ *RMSECV*, a scalar, Root Mean Squared Error of Cross-Validation (if the Cross-Validation is performed),
- □ *h*, a scalar, number of factors in the calibration model,
- □ *Ymp*, a vector, predicted dependent variable for the model set,
- □ *Ytp*, a vector, predicted dependent variable for the test set (if the test set is selected),
- □ *E*, a matrix containing in each column model set residuals from a model with an optimal number of factors,
- □ *Et*, a matrix containing in each column test set residuals from a model with an optimal number of factors,
- □ *final*, a vector giving an overview of the best nonlinear PLS model by specifying the sigma parameter, the number of factors in the final model, RMS, RMSECV and, if the test set is used, also RMSEP. In a classification setting, the final variable contains also information on the percent of correctly classified objects for the model set and test set and within Cross-Validation course,
- □ *RBFmodels*, a matrix containing the results for all RBF-PLS models where in the first column there are RMSECV values, in the second column the number of factors in the model and in the last column consecutive values of the sigma parameter,
- □ *activation*, a matrix (*m*,*m*) giving activation values of every Gaussian.

Besides the typical plots for calibration methods, the following figures can be made as well:

- – plot of the RMSECV displaying its values for a certain sigma and number of factors in the model,
- – color map of Gaussians activations.

5. Independent testing

The Toolbox aims to facilitate the data analysis and calibration, while its evaluation has proven to be useful in this context. The Toolbox was installed on my computer and I found the procedures performing properly. The routines of the Toolbox and Graphical User Interface (GUI) work as the authors described in the publication. The GUI is simple to use and no knowledge of the Matlab software is required to construct multivariate models. Many versatile methods are implemented in the Toolbox. There are classical PCA and PLS, which are the most applied in chemometrics, and the other interesting ones, but known to a lesser degree such as Continuum Power Regression (CPR), robust PCA, Partial Robust M-Regression, Robust Continuum Regression and Radial Basis Function PLS (RBF-PLS). I do appreciate various visualization options, for instance, possibility to plot scores, loadings, regression coefficients, color maps of the residuals, distance plots, etc. To give a help to a user, the authors equipped the GUI with the so-called 'tool tips' giving a short information when the mouse cursor is drag over some GUI buttons or fields.

Nevertheless, I regret that some more specific preprocessing methods (e.g. Multiplicative Scatter Correction or derivatives) are not available at the moment. I wish only the Kennard and Stone algorithm can be used to design model and test sets. On the other hand, when necessary, the GUI enables exporting and importing the data from the Matlab workspace, and hence additional data pre-processing can be done outside the GUI. This gives a lot of flexibility to the program. A list of comments was sent to the authors. All of the remarks were taken into account and the Toolbox was improved as it was required.

My overall opinion is that TOMCAT is a very efficient tool for multivariate analysis of the data and worthy to try by the others.

The Toolbox was independently tested by Xavier Capron, Department of Analytical Chemistry and Pharmaceutical Technology, Vrije Universiteit Brussel, Laarbeeklaan 103, B-1090 Brussels, Belgium.

Acknowledgements

M. Daszykowski would like to express his sincere gratitude to Foundation for Polish Science for financial support.

S. Serneels: Research financed by a PhD Grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

The authors would like to thank S. de Jong for providing WIM-PLS and SIM-PLS routines.

References

[1] E.R. Malinowski, Factor Analysis in Chemistry, John Wiley and Sons, INC., New York, 1991.

[2] H. Martens, T. Næs, Mutivariate Calibration, John Wiley and Sons, Chichester, UK, 1989.

[3] T. Næs, T. Isaksson, T. Fearn, T. Davis, Multivariate Calibration and Classification, NIR Publications, Chichester, UK, 2002.

[4] P.J. Rousseeuw, A.M. Leroy, Robust Regression and Outlier Detection, John Wiley and Sons, New York, 1987.

[5] P.J. Huber, Robust Statistics, Wiley, New York, 1981.

[6] G. Li, Z.L. Chen, Projection pursuit approach to robust dispersion matrices and principal components—primary theory and Monte-Carlo, Journal of the American Statistical Association 381 (1985) 759–766.

[7] M. Hubert, P.J. Rousseeuw, S. Verboven, A fast method for robust principal components with applications to chemometrics, Chemometrics and Intelligent Laboratory Systems 60 (2002) 101–111.

[8] C. Croux, G. Haesbroeck, Principal component analysis based on robust estimators of the covariance or correlation matrix: influence function and efficiencies, Biometrika 87 (2000) 603–618.

[9] H. Hove, Y.-Z. Liang, O.M. Kvalheim, Trimmed object projection: a nonparametric latent-structure decomposition method, Chemometrics and Intelligent Laboratory Systems 27 (1995) 33–40.

[10] B. Walczak, D.L. Massart, Robust principal components regression as a detection tool for outliers, Chemometrics and Intelligent Laboratory Systems 27 (1995) 41–54.

[11] M. Hubert, S. Verboven, A robust PCR method for high-dimensional regressors, Journal of Chemometrics 17 (2003) 438–452.

[12] I.N. Wakeling, H.J.H. Macfie, A robust PLS procedure, Journal of Chemometrics 4 (1992) 189–198.

[13] M. Hubert, K. Vanden Branden, Robust methods for partial least squares regression, Journal of Chemometrics 17 (2003) 537–549.

[14] S. Serneels, C. Croux, P. Filzmoser, P.J. Van Espen, Partial Robust M-Regression, Chemometrics and Intelligent Laboratory Systems 79 (2005) 55–64.

[15] S. Verboven, M. Hubert, LIBRA: a MATLAB library for robust analysis, Chemometrics and Intelligent Laboratory Systems 75 (2005) 127–136.

[16] Can be downloaded from: http://www.wis.kuleuven.ac.be/stat/robust.html.

[17] B. Walczak, D.L. Massart, The Radial Basis Functions-Partial Least Squares approach as a flexible non-linear regression technique, Analytica Chimica Acta 331 (1996) 177–185.

[18] B. Walczak, D.L. Massart, Application of Radial Basis Functions-Partial Least Squares to non-linear pattern recognition problems: diagnosis of process faults, Analytica Chimica Acta 331 (1996) 187–193.

[19] MatLab, The MathWorks, Inc. Natwick, MA (USA), http://www.mathworks.com.

[20] http://www.chemometria.us.edu.pl/RobustToolbox/TOMCAT.zip.

[21] http://www.chemometrix.ua.ac.be/dl/Tomcat.zip.

[22] http://www.chemometria.us.edu.pl/RobustToolbox/.

[23] http://www.chemometrix.ua.ac.be/tomcat/.

[24] R.J. Barnes, M.S. Dhanoa, S.J. Lister, Standard normal variate transformation and de-trending of near-infrared diffuse reflectance spectra, Applied Spectroscopy 43 (1989) 772–777.

[25] O. Hössjer, C. Croux, Generalizing univariate signed rank statistics for testing and estimating a multivariate location parameter, Non-parametric Statistics 4 (1995) 293–308.

[26] P.J. Rousseeuw, C. Croux, Alternatives to the median absolute deviation, Journal of the American Statistical Association 88 (1993) 1273–1283.

[27] W. Wu, D.L. Massart, S. de Jong, The kernel PCA algorithms for wide data: Part I. Theory and algorithms, Chemometrics and Intelligent Laboratory Systems 36 (1997) 165–172.

[28] C. Croux, A. Ruiz-Gazen, High breakdown estimators for principal components: the projection-pursuit approach revisited, Journal of Multivariate Analysis 95 (2005) 206–226.

[29] S. de Jong, SIMPLS: an alternative approach to partial least squares regression, Chemometrics and Intelligent Laboratory Systems 18 (1993) 251–263.

[30] M. Stone, R.J. Brooks, Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal component regression, Journal of the Royal Statistical Society B 5 (1990) 237–269.

[31] S. de Jong, R.W. Farebrother, Extending the relationship between ridge regression and continuum regression, Chemometrics and Intelligent Laboratory Systems 25 (1994) 179–181.

[32] Courtesy of S. de Jong.

[33] Courtesy of B.M.W. Wise, Eigenvector Research.

[34] S. Serneels, P. Filzmoser, C. Croux, P.J. Van Espen, Robust continuum regression, Chemometrics and Intelligent Laboratory Systems 76 (2005) 197–204.

[35] Q.-S. Xu, Y.-Z. Liang, Monte Carlo cross validation, Chemometrics and Intelligent Laboratory Systems 56 (2001) 1–11.

[36] K. Baumann, H. Albert, M. von Korff, A systematic evaluation of the benefits and hazards of variable selection in latent variable regression: Part I. Search algorithm, theory and simulations, Journal of Chemometrics 16 (2002) 339–350.

[37] R.W. Kennard, L.A. Stone, Computer aided design of experiments, Technometrics 11 (1969) 137–148.