Supplementary material

The PLS-DA models shown in this tutorial were calculated by means of the Classification toolbox for MATLAB (version 2.0). A GUI graphical interface is provided with the toolbox. The graphical interface enables the user to perform all the steps of the analysis. The toolbox was released by Milano Chemometrics and QSAR Research Group. Updates of the toolbox can be found in the following webpage: http://michem.disat.unimib.it/chm/.
In order to calculate PLS-DA models, MATLAB should be installed, while no other toolboxes are needed. In order to install the toolbox, simply copy the files to a folder. Then, in order to run it, select the same folder as MATLAB current directory.

Table I. Steps of analysis of Sediment dataset by means of the Classification toolbox for MATLAB.
The following table collects all commands needed to perform the steps of analysis on the Sediment data, as explained in the tutorial. In italics the command to be used directly in the MATLAB command window, in normal character the commands to be used in the graphical interface (GUI) of the Classification toolbox for MATLAB.

step	**MATLAB command window**	
1	*load sediment*	Load the dataset in the MATLAB workspace (see Table II).
2	*figure; boxplot(exp(Xtrain_log))* *figure; boxplot(Xtrain_log)*	Display box plots of variables of training samples (raw data and log transformed).
3	*class_gui*	Open the graphical interface of the Classification toolbox for MATLAB.
	Classification toolbox GUI command	
4	File -> load data, load class, load labels	Load training data (Xtrain_log), training class (class_train), training sample labels (samples_train) and variable labels (variables) in the GUI.
5	View -> Wilk's lambda	Plot the Wilk's Lambda values for all variables.
6	Calculate -> optimal components for PLSDA	Calculate the optimal number of Latent Variables (LVs) in cross-validation; select the following PLS-DA settings: data scaling: none; assignation criterion: bayes; cross validation: venetian blinds; number of cv groups: 5.
7	Calculate -> PLSDA	Calibrate the PLSDA model and (optional) make cross-validation; select the following PLS-DA settings: number of components: 2; data scaling: none; assignation criterion: bayes; cross validation: venetian blinds; number of cv groups: 5.

8	Results -> classification results	See the classification performance (error rate, non error rate, specificity, sensitivity) and the confusion matrices in fitting and cross-validation.
9	Results -> PLSDA scores and loadings	Open the graphical interface to plot model details for samples (scores, calculated responses in fitting and cross validation, leverages, Hotelling T^2 and Q residuals) and variables (loadings and regression coefficients).
10	Results -> ROC curves	Display the ROC curves.
11	File -> save model	Save the calculated model (e.g. mymodel) as MATLAB structure in the MATLAB workspace.
12	File -> load data, load class, load labels	Load test data (Xtest_log), test class (class_test), test sample labels (samples_test) in the GUI.
13	predict -> predict samples	Predict the test samples with the model calibrated on the training samples.
14	predict -> prediction results	See the classification performances (error rate, non error rate, specificity, sensitivity) and confusion matrix for the test samples.
15	results -> PLSDA scores and loadings	Open the graphical interface to plot model details for samples and variables. Test samples are plotted with different marks.
	MATLAB command window	
16	*Xc=mymodel.T*mymodel.P';* *E = Xtrain_log(1344,:) - Xc(1344,:);* *Qres=E*E'* *bar(E)*	Make the bar plot of the Q contributions of sample 1344.
17	*figure* *plot(Xtrain_log','k')* *hold on* *plot(Xtrain_log(1344,:)','r')*	Variable profile of all samples (in black) and of sample 1344 (in red).
18	*in1=find(mymodel.T(find(class_train==1),2)>0);* *in2=find(class_train==2);* *in3=find(mymodel.T(find(class_train==1),2)<0);* *hold on* *plot(mean(Xtrain_log(in1,:)),'-k')* *plot(mean(Xtrain_log(in1,:)),'ok')* *plot(mean(Xtrain_log(in2,:)),'-r')* *plot(mean(Xtrain_log(in2,:)),'or')* *plot(mean(Xtrain_log(in3,:)),'-b')* *plot(mean(Xtrain_log(in3,:)),'ob')*	Find a) samples of class 2 (toxic); b) samples of class 1 with scores on the second LV lower than 0; c) samples of class 1 with scores on the second LV higher than 0; than, plot average of all variables for the three groups of samples. Results are shown in Figure 10.
19	*Xtrain_log=Xtrain_log([1:1343 1345:end],:);*	Remove sample 1344 from the

	class_train=class_train([1:1343 1345:end]); *samples_train=samples_train([1:1343 1345:end]);*	dataset.

Table II. MATLAB Workspace content of the Sediment dataset.

Name	**Size**	**Content**
Xtest_log	471x9	test set data (log transformed)
Xtrain_log	1413x9	training set data (log transformed)
class_test	471x1	test class vector
class_train	1413x1	training class vector
info	5x1	data information and reference
samples_test	471x1	labels of training samples
samples_train	1413x1	labels of test samples
variables	1x9	labels of variables