Available online at www.sciencedirect.com

SCIENCE DIRECT®

Chemometrics and Intelligent Laboratory Systems 76 (2005) 101–110

Chemometrics and intelligent laboratory systems

www.elsevier.com/locate/chemolab

Software description

A graphical user-friendly interface for MCR-ALS: a new tool for multivariate curve resolution in MATLAB

Joaquim Jaumot[a], Raimundo Gargallo[a], Anna de Juan[a], Romà Tauler[b,*]

[a]Department of Analytical Chemistry, University of Barcelona, Diagonal 647, Barcelona 08028, Spain
[b]Department of Environmental Chemistry, IIQAB-CSIC, Jordi Girona 18, Barcelona 08034, Spain

Received 22 October 2004; received in revised form 17 December 2004; accepted 21 December 2004

Abstract

A new graphical user-friendly interface for Multivariate Curve Resolution using Alternating Least Squares has been developed as a freely available MATLAB toolbox. Through the use of this new easy-to-use graphical interface, the selection of the type of data analysis (either individual experiments giving a single data matrix or the more powerful simultaneous analysis of several experiments using one or more techniques) and the selection of the appropriate constraints can be performed in an intuitive and easy way, with the help of the options in the graphical interface. Different examples of use of this interface are given.

Keywords: MCR-ALS; Curve resolution; Chemometrics; Graphical user interface; MATLAB

1. Introduction

Multivariate Curve Resolution-Alternating Least Squares (MCR-ALS) has become a popular chemometric method used for the resolution of multiple component responses in unknown unresolved mixtures. On one hand, this recognition is due to the great variety of data sets that can be analyzed by curve resolution methods; essentially, any multicomponent system that gives as a result data tables or data matrices that can be described by a bilinear model. This description includes all kinds of processes and mixtures (e.g., chemical reactions, industrial processes, chromatographic elutions, spectroscopic images or environmental data, to mention a few) monitored by diverse multivariate responses, such as spectroscopic measurements, electrochemical signals, composition profiles or others. On the other hand, other reasons for the acceptance of MCR-ALS are its ability to deal with multiple data matrices simultaneously (reducing factor analysis intrinsic ambiguities [1–3] and/or data rank deficiencies [4,5]) and the diversity and flexible application of constraints to help and improve the resolution results.

Despite all these advantages, MCR-ALS algorithms written in MATLAB have the drawback of how the selected constraints are input in the execution of the MATLAB routine. This process can be troublesome and difficult in complex cases where several data matrices are simultaneously analyzed and/or different constraints are applied to each of them for an optimal resolution. In order to overcome these difficulties and taking advantage of the better MATLAB tools to create graphical user interfaces, an improved MCR-ALS toolbox with a user-friendly graphical interface is presented in this work.

Apart from the software here described, other curve resolution programs have appeared recently. In PLS Toolbox version 3.5 [6], Multivariate Curve Resolution using Alternating Least Squares (ALS) is available via mcr and als m functions and/or via new user graphical interface menu programs. Another curve resolution software, freely available in the Internet, is the GUIPRO software [7,8]. This uses an alternating least squares algorithm based on the use of penalty functions (P-ALS). Differences exist among the different approaches depending on constraints implementation and on how multiway data analysis is performed.

* Corresponding author. Tel.: +34 934006140; fax: +34 932045904.
E-mail addresses: joaquim@apolo.qui.ub.es (J. Jaumot), roma@iiqab.csic.es (R. Tauler).

0169-7439/$ - see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.chemolab.2004.12.007

2. Algorithm

A short description of the MCR-ALS method is given here. For a more detailed description of the method, see Refs. [1–5,9]. Without a loss of generality and to facilitate comprehension, the case of a data set formed by spectroscopic measurements will be considered. However, MCR-ALS should be considered a general purpose factor analysis tool, which may also be applied to other types of instrumental measurements, like electrochemical data [10] or simply to any data table or data matrix decomposable into a bilinear model, such as environmental source apportionment data [11].

Mathematically speaking, MCR methods are based on a bilinear model like the one given in Eq. (1).

$$\mathbf{D} = \mathbf{C}\mathbf{S}^{\mathrm{T}} + \mathbf{E} \tag{1}$$

The goal of MCR-ALS is the bilinear decomposition of the data matrix **D** into the 'true' pure response profiles associated with the variation of each contribution in the row and the column directions, represented by matrices **C** and $\mathbf{S}^{\mathrm{T}}$, respectively, which are responsible for the observed data variance.

In spectroscopic measurements, the rows of matrix **D** are the spectra measured during the experiment, the column profiles of matrix **C** and the row profiles of $\mathbf{S}^{\mathrm{T}}$ are usually associated respectively with the concentration and pure spectra profiles of the resolved components. The superscript T means the transpose of matrix **S**, where pure spectra are column profiles. **E** is the matrix of residuals not explained by the model and ideally should be close to the experimental error. Note that Eq. (1) is the multiwavelength extension of Lambert–Beer's law in matrix form.

MCR-ALS solves iteratively Eq. (1) by an Alternating Least Squares algorithm which calculates concentration **C** and pure spectra $\mathbf{S}^{\mathrm{T}}$ matrices optimally fitting the experimental data matrix **D**. This optimization is carried out for a proposed number of components and using initial estimates of either **C** or $\mathbf{S}^{\mathrm{T}}$. Initial estimates of **C** or $\mathbf{S}^{\mathrm{T}}$ can be obtained either using Evolving Factor Analysis [12] or SIMPLISMA [13] derived methods. During the ALS optimization, several constraints can be applied to model the shapes of the **C** and $\mathbf{S}^{\mathrm{T}}$ profiles, such as non-negativity, unimodality, closure, trilinearity, selectivity or/and other shape or hard-modeling constraints [1,2,9,10]. Convergence is achieved when in two consecutive iterative cycles, relative differences in standard deviations of the residuals between experimental and ALS calculated data values are less than a previously selected value, usually chosen as 0.1%. This value may be modified by the user depending on the stage of the optimization. Usually at the beginning of the study a higher value is used (i.e. 1%) for exploratory purposes. In contrast, once a good model has been found, lower values are attempted to see whether there is any appreciable improvement in **C** and $\mathbf{S}^{\mathrm{T}}$ solutions, both in qualitative and quantitative terms.

Figures of merit of the optimization procedure are the percent of lack of fit, the percent of variance explained and the standard deviation of residuals respect experimental data.

Lack of fit is defined as the difference among the input data **D** and the data reproduced from the $\mathbf{C}\mathbf{S}^{\mathrm{T}}$ product obtained by MCR-ALS. This value is calculated according to the expression:

$$\text{lack of fit } (\%) = 100\sqrt{\frac{\sum_{i,j} e_{ij}^2}{\sum_{i,j} d_{ij}^2}} \tag{2}$$

where d_{ij} designs an element of the input data matrix **D** and e_{ij} is the related residual obtained from the difference between the input element and the MCR-ALS reproduction.

Two different lack of fit values are calculated, differing on the definition of the input data matrix **D** (either the raw experimental data matrix or the PCA reproduced data matrix using the same number of components as in the MCR-ALS model). These two lack of fit values are evaluated and shown at the end of each iterative cycle, when all constraints have been already applied and finally given in the results at the end of the optimization. These values are useful to understand whether experimental data were well fitted and also to evaluate whether the ALS optimization fit approached PCA fit.

Percent of variance explained (Eq. (3)) and standard deviation of residuals respect experimental data (Eq. (4)) are calculated according to the following expressions where d_{ij} and e_{ij} are the same as above and n_{rows} and n_{columns} are the number of rows and columns in the **D** matrix.

$$R^2 = \frac{\sum_{i,j} d_{ij}^2 - \sum_{i,j} e_{ij}^2}{\sum_{i,j} d_{ij}^2} \tag{3}$$

$$\sigma = \sqrt{\frac{\sum_{i,j} e_{ij}^2}{n_{\mathrm{rows}}\, n_{\mathrm{columns}}}} \tag{4}$$

Simultaneous MCR-ALS analysis of multiple independent experiments run under different experimental conditions is a useful and powerful strategy in resolution. Eq. (1) can be extended to allow for the MCR-ALS simultaneous analysis of several experiments followed by the same spectroscopic technique. This data arrangement gives rise to a column-wise augmented matrix, where the resolved pure spectra are common to all experiments and the concentration profiles can be different from experiment to experiment. Eq. (5) shows the model for a two-experiment system, giving data matrices $\mathbf{D}^1$ and $\mathbf{D}^2$, a common pure spectra matrix $\mathbf{S}^{\mathrm{T}}$ and submatrices $\mathbf{C}^1$ and $\mathbf{C}^2$ with the concentration profiles related to $\mathbf{D}^1$ and $\mathbf{D}^2$, respectively.

$$\begin{bmatrix} \mathbf{D}^1 \\ \mathbf{D}^2 \end{bmatrix} = \begin{bmatrix} \mathbf{C}^1 \\ \mathbf{C}^2 \end{bmatrix} \mathbf{S}^{\mathrm{T}} + \begin{bmatrix} \mathbf{E}^1 \\ \mathbf{E}^2 \end{bmatrix} \tag{5}$$

Improved conditions for better species resolution may be achieved when the two experiments are analyzed together instead of one by one, particularly in situations of rank-deficiency [4,5].

When the same chemical system is monitored using more than one spectroscopic technique (i.e. fusion of UV-Vis absorption data and fluorescence and/or circular dichroism data), a row-wise augmented data matrix can be built up organizing the individual data matrices corresponding to each spectroscopy one beside each other. For an example of two spectroscopic techniques, the related bilinear model for MCR-ALS analysis is shown in Eq. (3).

$$[\mathbf{D}_A \mathbf{D}_B] = \mathbf{C}\left[\mathbf{S}_A^T \mathbf{S}_B^T\right] + [\mathbf{E}_A \mathbf{E}_B] \quad (6)$$

Now $\mathbf{D}_A$ and $\mathbf{D}_B$ are the measurements for the same experiment obtained with the two techniques. There is a single matrix of concentration profiles $\mathbf{C}$, valid for the two sets of raw measurements and a row-wise augmented matrix of spectra, where $\mathbf{S}_A^T$ and $\mathbf{S}_B^T$ contain the pure spectra for the techniques used in $\mathbf{D}_A$ and $\mathbf{D}_B$, respectively. Solving Eq. (6) for $\mathbf{C}$ and $[\mathbf{S}_A^T\ \mathbf{S}_B^T]$ helps to resolve all the species of the system, particularly if their spectra are not very different or lack signal in one of the two spectroscopies simultaneously analyzed.

Finally, a still more evolved data analysis approach can be proposed for the simultaneous analysis of data acquired with different spectroscopies on multiple experiments. Data set augmentation is now carried out both column-wise (as in Eq. (5)) and row-wise (as in Eq. (6)). The bilinear model for n experiments studied by 2 different spectroscopies is described by Eq. (7):

$$\begin{bmatrix} \mathbf{D}_A^1 & \mathbf{D}_B^1 \\ \vdots & \vdots \\ \mathbf{D}_A^n & \mathbf{D}_B^n \end{bmatrix} = \begin{bmatrix} \mathbf{C}^1 \\ \vdots \\ \mathbf{C}^n \end{bmatrix} \left[\mathbf{S}_A^T \mathbf{S}_B^T\right] + \begin{bmatrix} \mathbf{E}_A^1 & \mathbf{E}_B^1 \\ \vdots & \vdots \\ \mathbf{E}_A^n & \mathbf{E}_B^n \end{bmatrix} \quad (7)$$

where $\mathbf{D}_i^j$ accounts for the data matrix related to experiment i monitored by the technique j and the resolved concentration profiles and pure spectra are in a column-wise matrix formed by n $\mathbf{C}^j$ submatrices and a row-wise augmented matrix formed by two $\mathbf{S}_i^T$ submatrices, respectively.

This kind of simultaneous data analysis is still even more powerful than those described by Eqs. (1), (5) or (6) and allows for the improvement of resolution of very complex data structures, as those found in Refs. [10,11,14,15]. The approach assembles the profits of both augmentations previously described and gives more reliable solutions, eventually removing rotational ambiguities and rank-deficiency problems.

3. Software specifications and requirements

The new proposed MCR-ALS algorithm with a user-friendly interface consists of a few MATLAB (m- and their related fig-) files developed under MATLAB 6.5 (Release 13). The main routine is called *als2004* and calls auxiliary routines (*als2004m, clas3way, als2004multi, closure_no and als_res*) related to the characterization of the data set and to the selection of the constraints used during the optimization. Launching of the main routine can be performed either by writing 'als2004' in the MATLAB command line or by clicking the MCR-ALS user-friendly interface icon in the MATLAB Start menu or in the Shortcuts bar with the 'xml' file provided with the toolbox. The software is compatible with MATLAB release versions R13 and R14 and does not need any toolbox apart from the MATLAB standard core program. The software has been tested in computers under different Microsoft Windows operating systems (from 98/2000/XP) with no need of any particular additional resources and, as expected, the time of analysis is dependent on the experimental data nature. The MCR-ALS code, the related tutorials and the data sets for practice are available at the Multivariate Curve Resolution web page: http://www.ub.es/gesq/mcr/mcr.htm.

4. Data set

The data set used for this software demonstration is formed by four HPLC-DAD runs, sized (51×96, as example *m1*), each one of them containing four compounds. The augmented data set forming a column-wise matrix with the four runs appended one on top of each other is in a MATLAB variable called *MATRIX*, sized (204×96). The matrix of initial estimates for spectra profiles is in variable *spure*. These variables, needed to start the MCR-ALS optimization, are in a MATLAB file called *mtril.mat*, downloadable with the MATLAB code. In this file are also available other variables used in the following examples: *csel_matrix* (containing the local rank information in the concentration direction) and *isp_matrix* (containing identification of species between different matrices).

5. Operating procedure

MCR-ALS optimization with a user-friendly interface works following the classical scheme of the alternating least squares procedures, i.e., the iterative optimization of the resolved concentration profiles and spectra subject to selected constraints The dialog boxes that appear during the MCR-ALS execution are mainly related to: a) input of initial information, b) selection of constraints and selection of optimization parameters, and c) display of resolution results.

5.1. Input of initial information

The initial graphic input window is launched by the *als2004* function (Fig. 1). In this window, the user has to select: a) the matrix **D** to be analyzed (the *matrix* variable in this case), b) the initial estimates of either concentration or

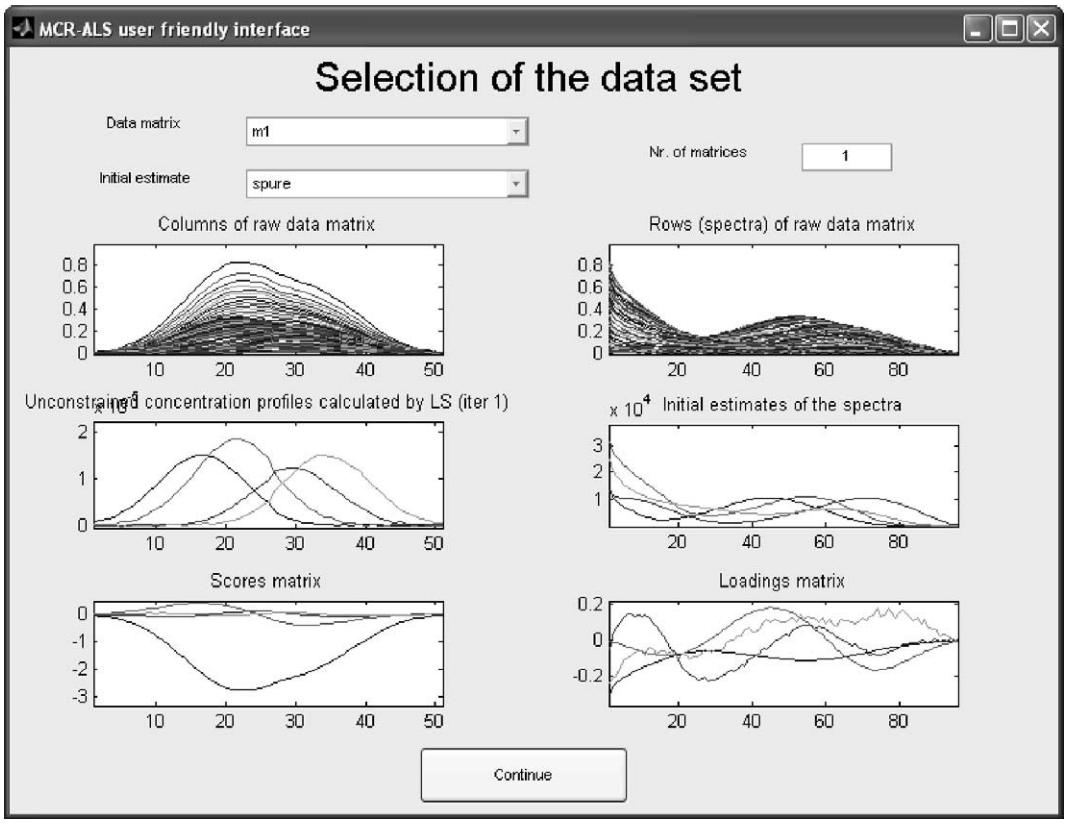

Fig. 1. Selection of data set window. The user has to select data set, initial estimation and number of matrices. The plots show information about the system as described in the text.

spectra profiles (**C** or $\mathbf{S}^T$) (e.g. *spure* variable in Fig. 1), and c) the number of submatrices simultaneously analyzed in the augmented data matrix **D** (if necessary). This value is defined by the number of experiments performed at different conditions and/or monitored by different spectroscopic techniques (four in our system of four HPLC-DAD runs). The only requirement is that all these matrices (**D** and the initial estimates of **C** or $\mathbf{S}^T$) have to be already stored in the MATLAB workspace before the *als2004* program function is launched.

In order to do the selection of the data matrix **D** and of the initial estimates of **C** or $\mathbf{S}^T$, appropriate boxes of the initial input window (Fig. 1, *Selection of the data set* window, m1 in data matrix box and spure in initial estimates box) should be filled in. Once these matrices have been selected, six different plots are obtained completing the graphical representation of the initial known information about the system under study. The top plots of Fig. 1 display the columns of the raw data set showing the process evolution at each wavelength (left plot) and the rows of the raw data set, i.e., the experimental spectra acquired along the evolution of the process (right plot). Left and right in the middle of Fig. 1, the plots display the initial estimates of the species spectra $\mathbf{S}^T$ matrix and the paired concentration profiles **C** matrix estimated by linear least squares from these spectra initial estimates $\mathbf{S}^T$ and data matrix **D**. Finally, at the bottom of Fig. 1, Principal Component Analysis [16] results of matrix **D** are obtained i.e. plot of scores and loadings matrices, for the preselected number of components in the system. When clicking the *Continue* button, two different situations can occur. If a single experiment is analyzed (Number of experiments in the selection data set window is equal to 1), the software will go directly to the *Selection of ALS constraints* window (Fig. 2).

On the other hand, if the user wants to analyze a row- and/or column- wise augmented matrix (several experiments or techniques), an intermediate window (Fig. 3) will appear prompting the user to set the number of **C** submatrices (different experiments), and/or $\mathbf{S}^T$ submatrices (different spectroscopic techniques) needed to describe the data set and the dimensions (number of rows or columns) of each one of them.

In Fig. 3, the implementation of the four HPLC-DAD runs is shown. First, the definition of the data set is required and the user selects one of the three options of matrix augmentation (column-wise, row-wise or column- and row-wise augmented data matrix). In our example, the column-wise augmented data matrix option is selected. Next, the user has to define the number of **C** and/or **S** submatrices needed to model the **D** matrix. If the matrix **D** is a column- or a row-wise augmented matrix, only the pop-up menu related to **C** or to **S**, respectively, will get activated. In these cases, the number of **C** or **S** submatrices should equal the total number of submatrices in **D**. If the matrix **D** is a

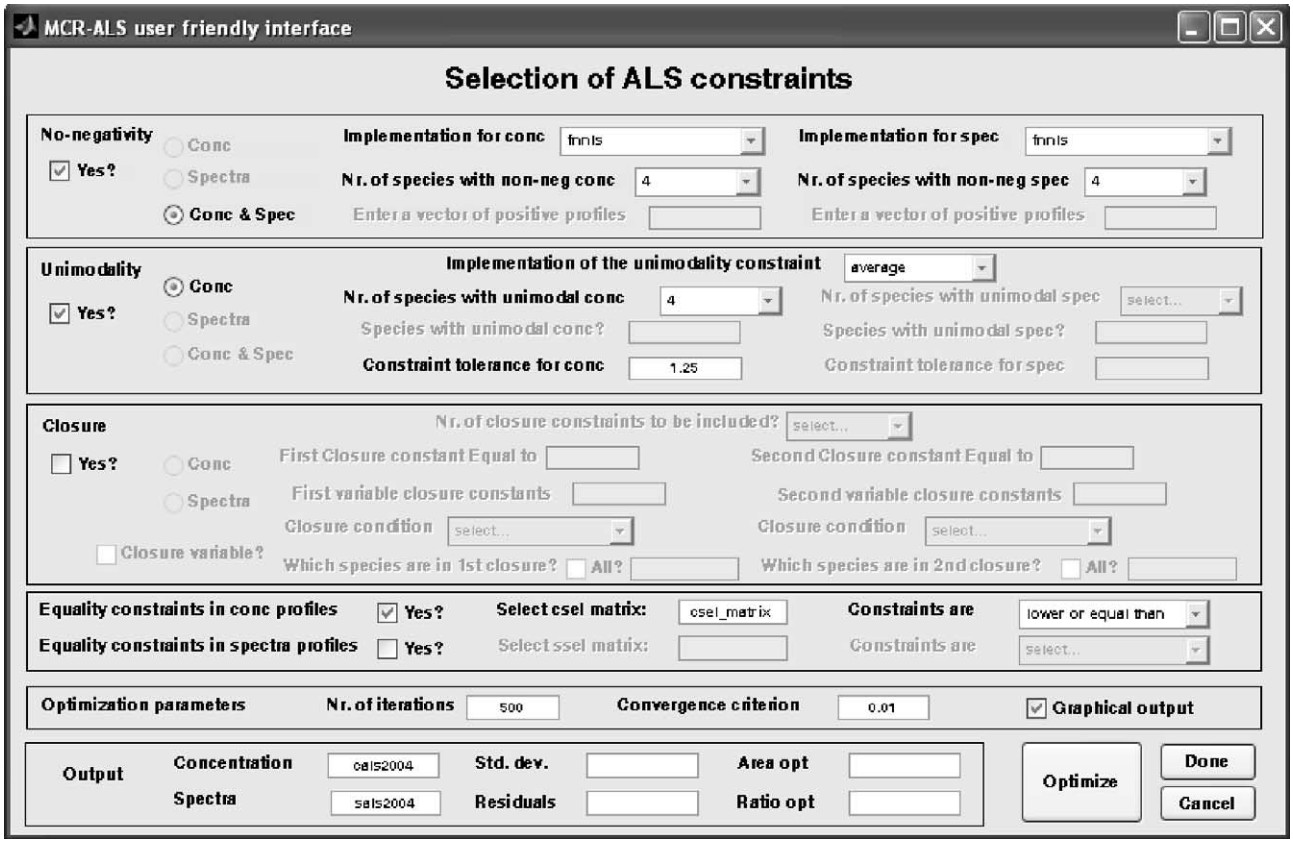

Fig. 2. Selection of the ALS constraints window in the case of a single experiment analysis. See text for a detailed explanation about the constraints selection and implementation.

column- and row-wise augmented matrix, then both **C** and **S** pop-up menus will be active and the user must select the number of submatrices of each kind. For the double augmentation, the product of the number of **C** submatrices by the number of **S** submatrices must be equal to the total number of submatrices in **D**. Once this general data structure is defined, the user has to input the number of rows or columns of each **C** and **S** submatrix according to the size of the submatrices in the augmented matrix **D**. Finally, when clicking the *OK* button the selection of constraints window for the three-way case will appear (Fig. 4).

5.2. Selection of constraints and selection of optimization parameters

The input of the constraints is carried out using the *Selection of ALS constraints* graphical window. Fig. 2 shows the simplest dialog box, when only a single experi-

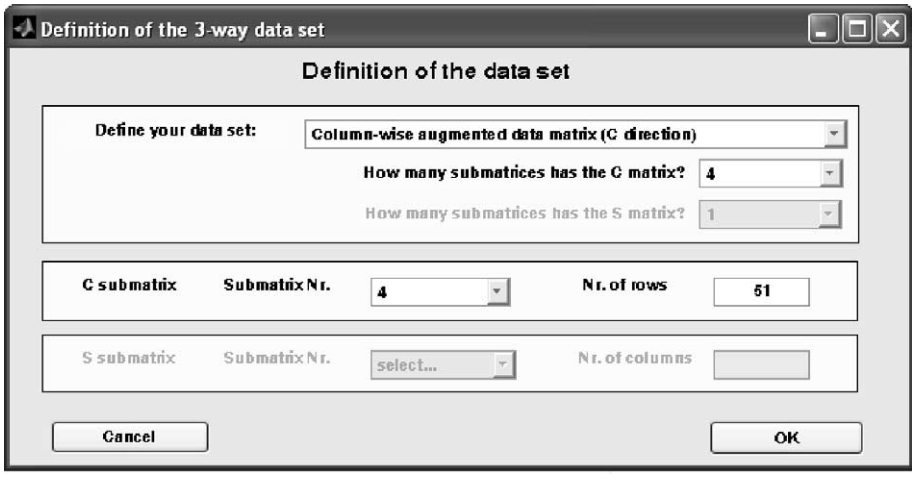

Fig. 3. Definition of the 3-way data set window. Definition of the kind of matrix augmentation and characterization of the size of each submatrix is performed.

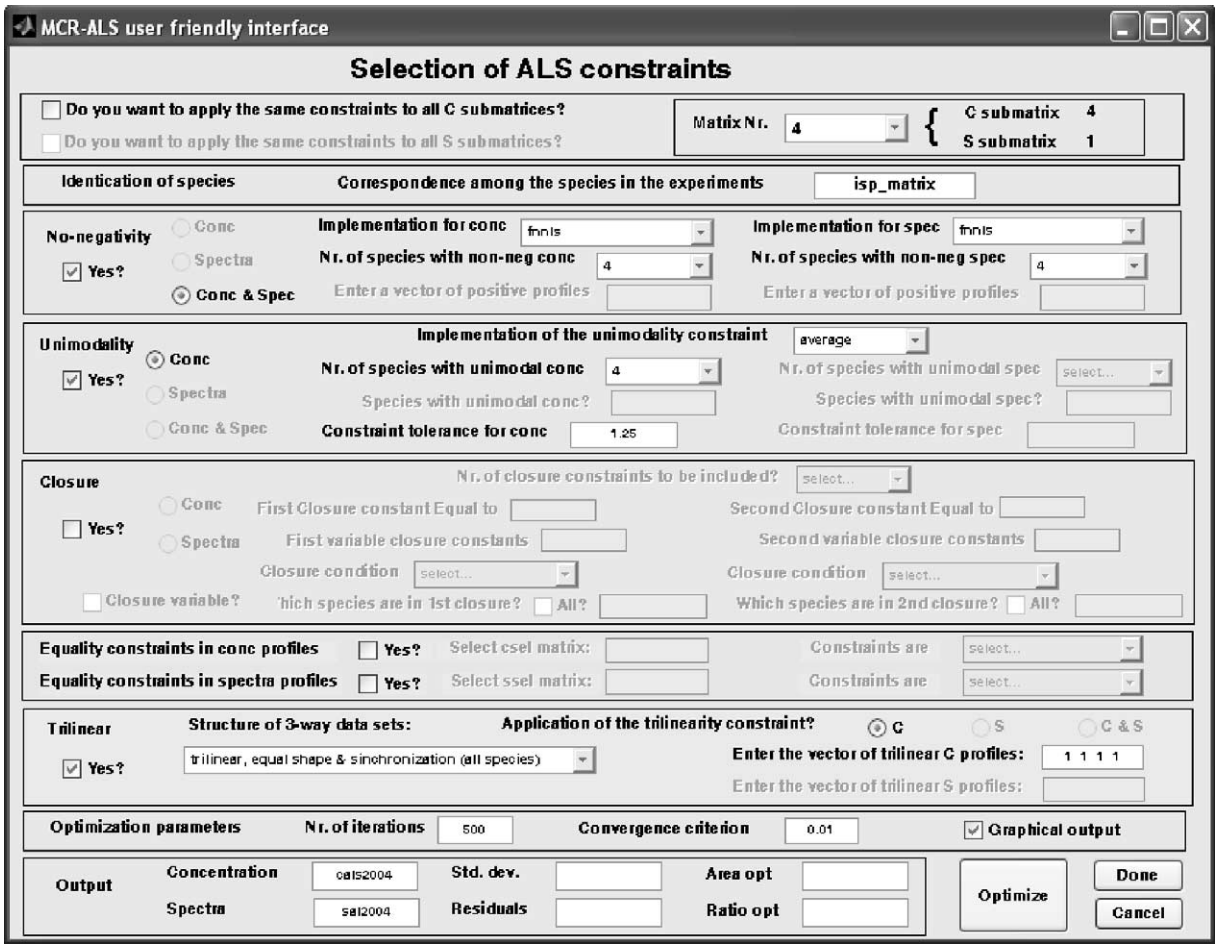

Fig. 4. Selection of the ALS constraints window in the analysis of several experiments. The selected constraints in this figure are applied to the HPLC-DAD chromatographic runs shown in Fig. 1. See text for a detailed explanation about the constraints selection and implementation.

ment is analyzed. Compared with previous versions of the algorithm, the step of selection of constraints has been greatly simplified. Fig. 2 adapts to the case of a single four-component HPLC-DAD run, where constraints of non-negativity in the concentration and spectral direction, unimodality for the concentration profiles and equality or local rank constraints could be applied and are, therefore, selected (see the related *Yes?* box ticked).

Initially, when the *Selection of ALS Constraints* window in Fig. 2 is loaded, the only active buttons are those to select *which* constraints should be applied. When one particular constraint and the matching checkbox button are selected, new options are gradually activated in the graphical interface to give details on *where* and *how* the constraint should come into play in the resolution process. First, some general comments for the application of any constraint, expressed in a common form in the dialog box, are described. Particular features of the different constraints will be treated afterwards when necessary.

The first thing to be decided is the direction of application of the constraint (concentration and/or spectral). To do so, the suitable radio buttons can be clicked at the left hand side of the constraints that allow for this option. Once the direction(s) of application are selected, the user must answer how many and which components (species) in the **C** and/or $\mathbf{S}^T$ matrix should be constrained. If the number of species to be constrained is lower than the total, an additional vector using a binary code (1 for component constrained and 0 for component unconstrained) should be introduced to indicate which components must obey the constraint (e.g., a vector 1 1 0 0, would indicate that only the first two profiles from a four-component system should be constrained). Note that the identity of the components is defined by the sequence that they follow in the matrix of initial estimates; therefore, all the vectors defined in the selection of constraints should match this initial sequence. In the application of several constraints, different implemented algorithms can also be selected.

Thus, the application of the non-negativity constraint can be carried out according to different least squares approaches, the classical non-negative least squares, *nnls* [17] and the more recent fast non-negative least squares, *fnnls* [18]. An additional option, designed 'forced', which replaces negative values by zeroes, is also available. This option is useful when only some of the profiles in **C** or $\mathbf{S}^T$ must be constrained or when statistically sounder non-

negative least squares (*nnls* [17] and *fnnls* [18]) algorithms fail for some reason or take too long.

For the unimodality constraint, the options 'vertical' and 'horizontal' mean that secondary maxima are cut vertically or horizontally [19]. In the 'average' implementation (the smoothest one), the secondary maxima are corrected taking averages similarly as in unimodal least squares algorithms [20]. A constraint tolerance can be selected to allow for some local departures of the unimodality condition. For instance, 1.5 means that 50% of local departure of the unimodal condition is allowed, i.e. that in the decreasing slopes of the main peak a particular point can increase a maximum of 50% of the previous value before the unimodality constraint is applied. Values between 1.0 (no departures from the unimodal condition allowed) and 1.1 are usual in systems with low to medium noise levels.

Although the constraint of closure does not apply to the HPLC-DAD example, it should be commented because it is of general use in many reaction systems. Closure in the concentration direction is related to mass balance equations in closed reaction systems. The total concentration of the system (closure constant) can be fixed to a single value or to a variable (changing) value. If the variation of the closure constant along the experiment is known (e.g., titration experiments with known dilutions), the name of a vector variable that contains the total concentration values at each point of the process should be introduced in the suitable box. The program allows also for the introduction of two closure conditions (two mass balance equations); however, the application of this option is not recommended when common species are shared by both mass balances. Finally, closure can be implemented as an equality constraint (the closure constant is exactly equal to some preselected value) or as a smoother inequality constraint. In the latter case, the mass balance should 'equal or be lower than' the preselected value for the closure constant.

When no closure constraint is selected, as is the case in our HPLC-DAD example and of many other chemical systems, a new window will open suggesting the use of an alternative normalization to avoid scale indeterminacies during ALS optimization (see Fig. 5). Very commonly, for unclosed systems, equal spectra area or intensity normalizations are selected. If no closure or normalization is applied, the scale of the profiles during the ALS optimization can become erratic and troublesome.

The boxes designed *Equality constraints in concentration and/or spectra* refer to the possibility to fix known values in the concentration profiles or in the spectra during the optimization, e.g., pure spectra of known compounds or selectivity/local rank information. Selectivity/local rank information can be defined as an equality constraint in the sense that we know that all absent species in a selective or in a low rank region have a concentration (or response) value equal to zero. To be applied, equality constraints need a 'filter' concentration and/or spectra matrix, sized as $\mathbf{C}$ and/or $\mathbf{S}^T$, formed by real numbers equal to the known values in the positions to be constrained and 'NaN' (Not a Number) or 'Inf' (infinite) MATLAB notation values in the positions left unconstrained. This matrix should be present in the MATLAB workspace and the name written down in the appropriate input box in the *Selection of ALS constraints* window. For instance, in Fig. 2, *csel_matrix* is given containing the selectivity/local rank information in the concentration direction. In this example, selectivity information in the concentration profiles is applied by setting to 0 values of concentration at the beginning and at the end of the chromatographic run when only one and two components are expected. Despite the name of this constraint, the user may choose between the implementation as 'equal' or 'equal or lower than' constraint, as in the closure example.

The comments below refer to the additional options available for the selection of constraints when several data matrices are simultaneously analyzed, either from several different experiments and/or from several different spectroscopic techniques, as shown in the dialog box in Fig. 4.

On top of Fig. 4, a first check-box option is available to decide whether the same constraints should apply to all submatrices in augmented matrices $\mathbf{C}$ and/or $\mathbf{S}^T$. When the related checkbox is not ticked, the user will be able to select different constraints for each individual submatrix in the augmented data matrix.

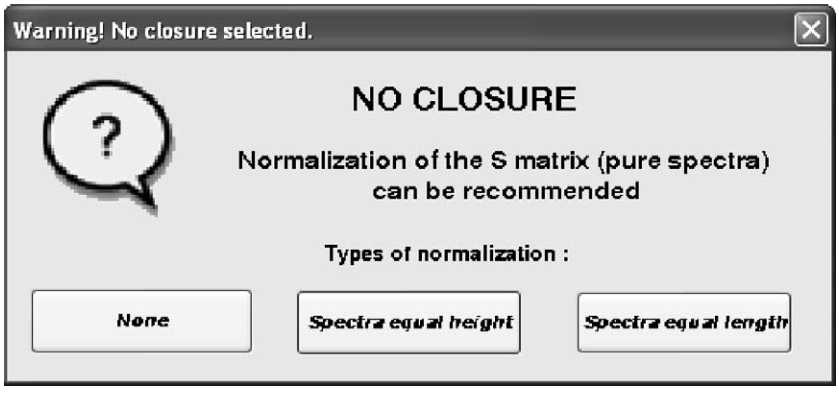

Fig. 5. No closure selected window. Different possibilities of spectra normalization are available when no closure is selected.

The identification and correspondence of species between different matrices is another option available in the analysis of data sets with an augmented concentration matrix. The user can select which species are present and absent in each **C** submatrix. This information is provided by a binary coded matrix (*isp_matrix* in Fig. 4). This matrix has a number of rows equal to the number of submatrices in **C** and a number of columns equal to the total number of components or species present in the system (counting all **C** submatrices). The presence or absence of a particular species in a submatrix is coded by 1 or 0, respectively. In a three-experiment data set formed by a mixture matrix of three components (A, B and C) and two standard matrices containing only component A and B, respectively, the matrix would be like:

$$\text{isp\_matrix} = \begin{array}{c} \begin{matrix} A & B & C \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

In our example of the four HPLC-DAD runs with the same four compounds in each, the *isp_matrix* would be sized (4×4) and all elements would be equal to one because all components are present in all HPLC-DAD runs. The speciation of a system can be known beforehand or can be elucidated in a first resolution analysis and be used in subsequent runs of the program. When nothing is known about the speciation or when all species are present in all matrices, the user can leave empty the *Correspondence among species in the experiments* box and a 'ones' matrix with appropriate dimensions will be generated automatically.

When the geometrical shape of an augmented data set can be displayed as a cube or a parallelepiped, i.e., data matrices having the same variables and dimensions in the row and column direction, the data set can be forced to obey characteristic models of three-way data sets. Different models have been proposed to analyze these more complex data structures, such as the trilinear or Parafac/Candecomp model [22] and the Tucker models [23]. Although MCR-ALS was initially designed for the analysis of individual or augmented data matrices under the assumption of a bilinear model, it can be easily adapted to the fulfillment of a trilinear model. This is achieved in MCR-ALS algorithmically as a constraint during the ALS optimization and it has been described elsewhere [24]. From the point of view of MCR-ALS, trilinearity means that the resolved profiles of the same component in the different data matrices for a particular direction (**C** or $\mathbf{S}^T$) have the same shape. In contrast to traditional three-way resolution methods, the trilinear condition can be implemented for each species separately. The main advantage of using MCR-ALS in this

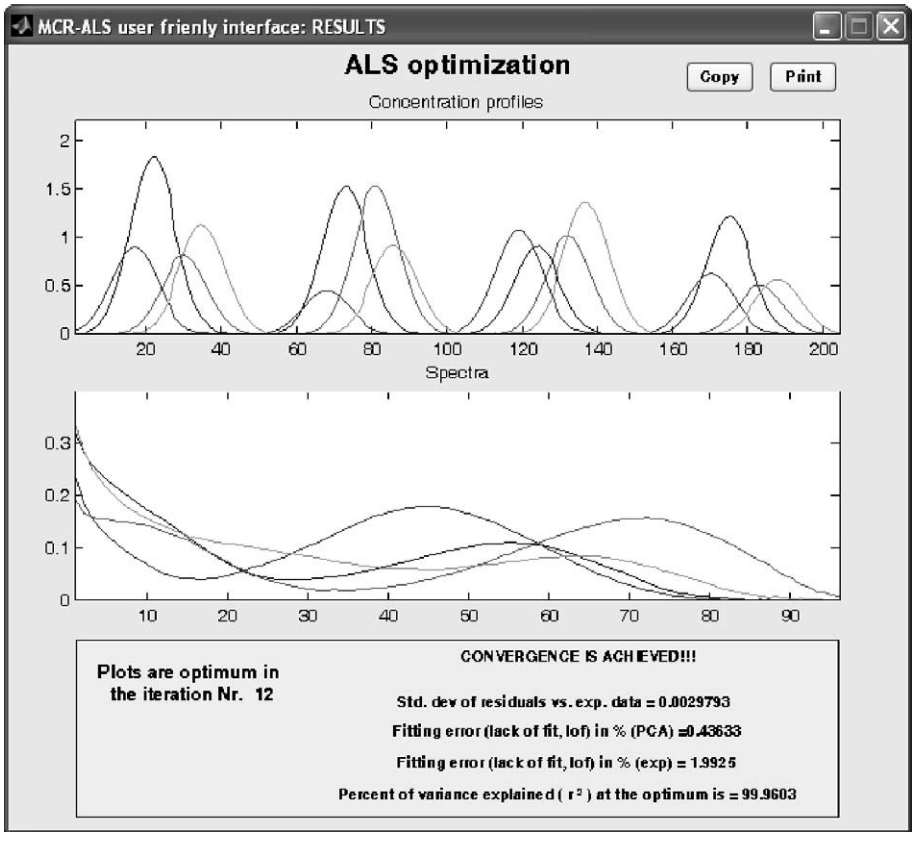

Fig. 6. ALS optimization results window. In this window, a representation of the resolved concentration and spectra is shown with other optimization (percentage of variance explained, lack of fit). These results correspond to the analysis of the HPLC runs with constraints selected in Fig. 4.

way is the flexibility to cover intermediate situations between a pure bilinear model and a completely trilinear model [21]. In the application of trilinearity, several options exist that account for shift correction among profiles of different matrices when needed (with or without synchronization options).

Once the constraints are selected, the choice of the optimization parameters and the information needed to present the output of the resolution method are carried out in the same way for single and augmented matrices.

Thus, in both dialog boxes in Figs. 2 and 4, the bottom part concerns the parameters used to control the end of the optimization process like the maximum number of iterations allowed and the convergence criterion (in percentage of change of standard deviation of residuals between two consecutive iterations, i.e. 0.01 means that convergence is achieved when the change in the standard deviation of the residuals is lower or equal to 0.01% between two consecutive iterations,). Ticking the *Graphical output* box, a plot of the resolved concentration profiles and spectra is shown after each iteration.

The final frame in dialog boxes in Figs. 2 and 4 asks for variable names to store the output of the resolution process. This output information is structured as six different variables that consist of two matrices related to the resolved pure concentration and spectra profiles (*Concentration and Spectra*), two matrices related to figures of merit of the optimization procedure (*Std. Dev. and Residuals*) and two parameters used for quantitative purposes in data sets where augmented **C** matrices are obtained (*Area opt* and *Ratio opt*). *Standard deviation* represents a vector that contains the optimal percent of lack of fit in relative standard deviation units: the first element is the lack of fit value of the resolution results with respect to the PCA reproduced matrix, while the second is the lack of fit between the resolution results and the original matrix (Eq. (5)). *Residuals* represent the **E** matrix of residuals between the original data and the pure resolved concentration and spectra profiles (see Eq. (1)). *Area opt* contains the area under the concentration profile of each species in each **C** submatrix. *Ratio opt* provides relative quantitative information as the ratio between the area of a certain species in the different **C** submatrices and the area of that species in the first **C** submatrix, taken as reference.

Clicking the "*Optimize*" button, the optimization procedure starts showing the partial results obtained in the different iterations. When graphical output has been selected, MCR-ALS resolved profiles are graphically shown after each iteration.

5.3. Display of resolution results

Once convergence is achieved or after the maximum number of iterations is exceeded or in case of divergence, the optimal resolution results are shown (Fig. 6). In this window, a plot of the resolved concentration and spectra profiles is given, as well as figures of merit related to the optimization results.

Results shown in Fig. 6 correspond to results obtained in the analysis of the previous example of four simulated HPLC runs subject to the constraints selected in Fig. 4.

6. Conclusions

The multivariate curve resolution of a data set is a dynamic process, where the data analysis is performed several times under different conditions (initial estimates, constraints selected,...) to learn more about the system behavior and end up with an optimal result from both a mathematical and a chemical point of view. The graphical interface incorporated to the MCR-ALS algorithm offers a user-friendly tool that can allow for an easy way to change the parameters of analysis keeping the power and freedom of application of the original MATLAB routine.

7. Validations

The Barcelona chemometrics group has now developed a graphical user interface for MCR-ALS. The rather cumbersome task of organising the data and the different configurations of constraints is now arranged in a well-structured series of graphical user interfaces. As the authors correctly state in the conclusions, ALS is an iterative process. Often, different starting guesses for **C** or $\mathbf{S}^T$ as well as different regimes of constraints have to be tested. This task is now tremendously easier, in particular as the selected values are 'remembered' from one attempt to the next.

There are a few outstanding issues which I would like to see addressed in the future: The first is the construction of a similar gui-based algorithm for the development initial guesses of either **C** or $\mathbf{S}^T$ based on the different methods available. The selection of optimal starting values is crucial in many instances. The other is a wider collection of worked examples which demonstrate the strengths and weaknesses of the different regimes of constraints and concatenation. The website related to this ALS toolbox is the perfect channel for such a collection.

Marcel Maeder
Department of Chemistry
University of Newcastle
Callaghan NSW 2308
Australia
chmm@cc.newcastle.edu.au

Acknowledgements

J. Jaumot acknowledges a PhD scholarship from the Ministerio de Educación y Ciencia (MEC). This research

was supported by the Spanish MEC (BQU2003-00191) and the Generalitat de Catalunya (2001SGR0056).

References

[1] R. Tauler, Chemometr. Intell. Lab. Syst. 30 (1995) 133–146.
[2] R. Tauler, A.K. Smilde, B.J. Kowalski, J. Chemom. 9 (1995) 31–58.
[3] R. Tauler, J. Chemom. 15 (2001) 627–646.
[4] M. Amrhein, B. Srinivasan, D. Bonvin, M.M. Schumacher, Chemometr. Intell. Lab. Syst. 33 (1996) 17–33.
[5] J. Saurina, S. Hernández-Cassou, R. Tauler, A. Izquierdo-Ridorsa, J. Chemom. 12 (1998) 183–203.
[6] PLS_Toolbox 3.5 for use with MATLAB, Eigenvector research Inc., 2004, Manson, WA, USA.
[7] P.J. Gemperline, E. Cash, Anal. Chem. 75 (2003) 4236–4243.
[8] http://personal.ecu.edu/gemperlinep/.
[9] A. de Juan, R. Tauler, Anal. Chim. Acta 500 (2003) 195–210.
[10] F. Berbel, E. Kapoya, J.M. Díaz-Cruz, C. Ariño, M. Esteban, R. Tauler, Electroanalysis 15 (2003) 499–508.
[11] R. Tauler, D. Barceló, E.M. Thurman, Environ. Sci. Technol. 34 (2000) 3307–3314.
[12] M. Maeder, Anal. Chem. 59 (1987) 527–530.
[13] W. Windig, J. Guilment, Anal. Chem. 63 (1991) 1425–1432.
[14] J. Jaumot, N. Escaja, R. Gargallo, C. González, E. Pedroso, R. Tauler, Nucleic Acids Res. 30 (2002) e92/1–e92/10.
[15] S. Navea, A. de Juan, R. Tauler, Anal. Chem. 75 (2003) 5592–5601.
[16] E.R. Malinowski, Factor Analysis of Chemistry, 3rd ed., Wiley-VCH, New York, USA, 2002.
[17] C.L. Lawson, R.J. Hanson, Solving Least-squares Problems, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[18] R. Bro, S. De Jong, J. Chemom. 11 (1997) 393–401.
[19] A. de Juan, Y. Vander Heyden, R. Tauler, D.L. Massart, Anal. Chim. Acta 346 (1997) 307–318.
[20] R. Bro, N.D. Sidiropoulos, J. Chemom. 12 (1998) 223–247.
[21] A. de Juan, R. Tauler, J. Chemom. 15 (2001) 749–772.
[22] R. Bro, Chemometr. Intell. Lab. Syst. 38 (1997) 149–171.
[23] L.R. Tucker, Psychometrika 31 (1966) 279–311.
[24] R. Tauler, I. Marqués, E. Casassas, J. Chemom. 12 (1998) 55–75.