MCR-ALS Graphical User-Frienly Interface user-guide

MCR-ALS with a user-friendly interface optimization 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, c) display of resolution results.

Input of initial information

The initial graphic input window is launched by the *als2004* function (Figure 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 spectra profiles (**C** or S^T) (i.e. *cpure* variable), 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 S^T) have to be already stored in the MATLAB workspace before the *als2004* program function is launched.
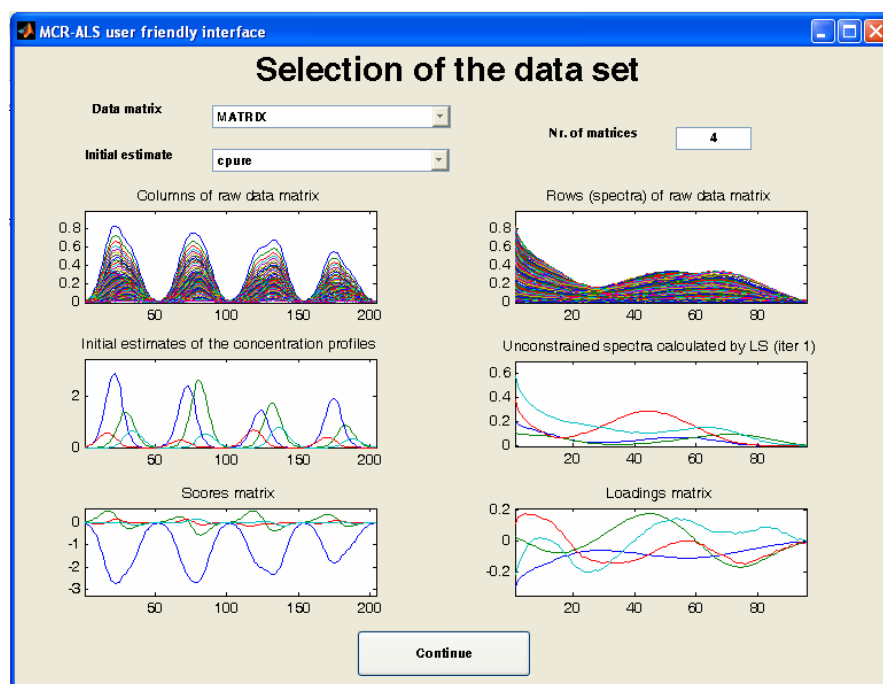

Figure 1.

In order to do the selection of the data matrix **D** and of the initial estimates of **C** or **S^T**, appropriate boxes of the initial input window (Figure 1, *Selection of the data set* window) 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 Figure 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 Figure 1, the plots display the initial estimates in the input data (matrix **C** of elution profiles, *cpure,* in Figure 1) and the paired matrix (**S^T** in the example), estimated by linear least-squares from the initial estimates and matrix **D**. Finally, at the bottom of Figure 1, Principal Component Analysis [13] 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 one), the software will go directly to the *Selection of ALS constraints* window (Figure 2).

Figure 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 (Figure 3) will appear prompting the user to set the number of **C** submatrices (different experiments),

and/or S^T submatrices (different spectroscopic techniques) needed to reproduce the data set and the dimensions (number of rows or columns) of each one of them.

Figure 3.

In Figure 3, the characterization 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 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 the three-way general 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 (Figure 4).

3

Figure 4.

Selection of constraints and selection of optimization parameters

The input of the constraints is carried out using the *Selection of ALS constraints* graphical window. Figure 2 shows the simplest dialog box, when only a single experiment is analyzed. Compared with previous versions of the algorithm, the step of selection of constraints has been enhanced and simplified. Figure 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 Figure 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 **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* and the more recent fast non-negative least-squares, *fnnls*. 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 **S**^{T} must be constrained or when statistically sounder non-negative least squares (*nnls* and *fnnls*) 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. In the 'average' implementation (the smoothest one), the secondary maxima are corrected taking averages similarly as in unimodal least squares algorithms. 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 it is the case of 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 Figure 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.
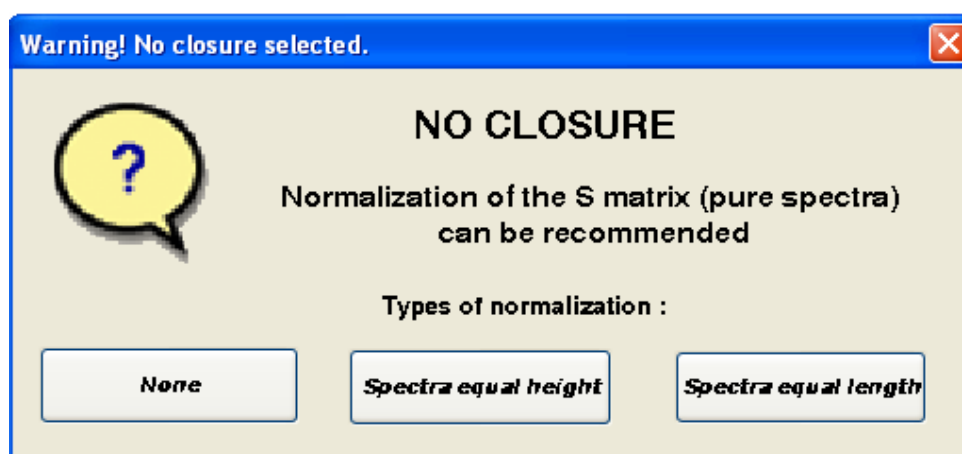

Figure 5

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 **C** and/or **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 Figure 2, *csel_matrix* is given containing the selectivity/local rank information in the concentration direction. 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 Figure 4.

On top of Figure 4, a first check-box option is available to decide whether the same constraints should apply to all submatrices in augmented matrices **C** and/or **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.

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 Figure 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:

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

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 or Tucker models. 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. 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 ($\mathbf{C}$ 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 way is the flexibility to cover intermediate situations between a pure bilinear model and a completely trilinear model. 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 Figures 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 % of change of standard of 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 Figures 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 (equation 5). *Residuals* represent the **E** matrix of residuals between the original data and the pure resolved concentration and spectra profiles (see equation 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.

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 (Figure 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.
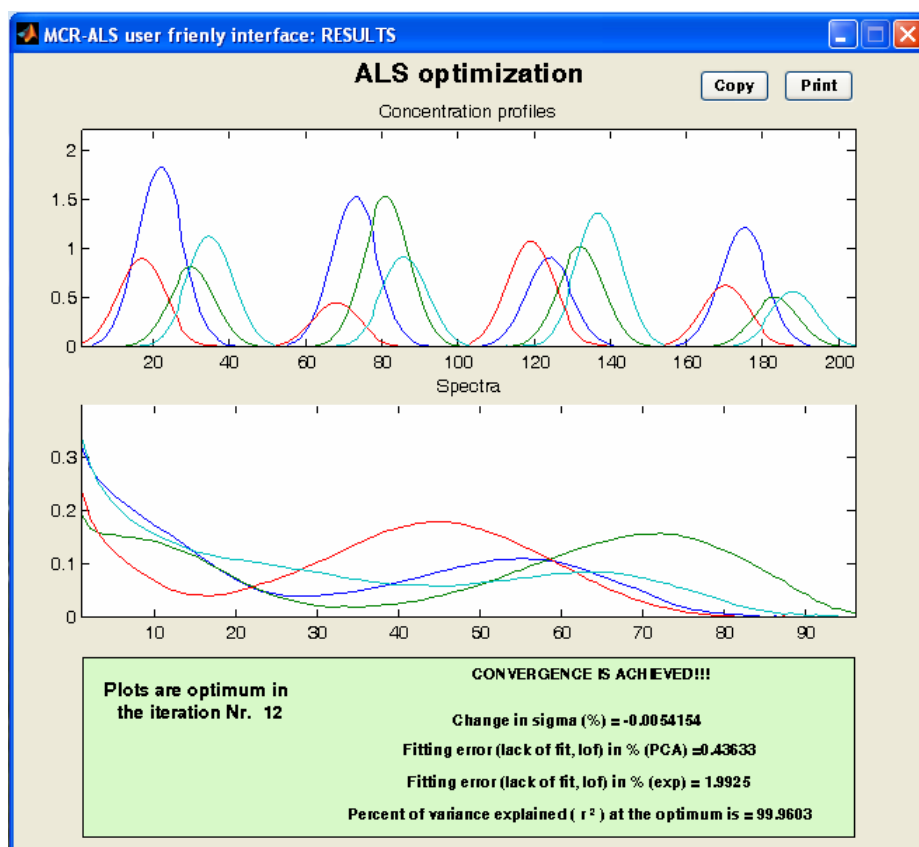
Figure 6.

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

If you find some mistakes in this user guide or in the software, please contact one of the authors:

- Romà Tauler: rtaqam@iiqab.csic.es
- Anna de Juan: annaj@apolo.qui.ub.es
- Joaquim Jaumot: joaquim@apolo.qui.ub.es

Last update: October 2004