USER MANUAL

MCR-ALS GUI v4c

Dr. András Gorzsás

Vibrational Spectroscopy Core Facility, Department of Chemistry, Umeå University, SE-90187 Umeå, SWEDEN

andras.gorzsas@umu.se

2016 September

Table of Contents

Disclaimer and Credits

The GUI itself is open source and free of charge but is subject to

Changelog

2016 September, v4c

1. **Equality constraints** fixed
2. **Unimodality constraints** fixed
3. **Unimodality mode selection** options updated: no longer necessary to have the same mode for concentrations and spectra, they are independent. GUI layout and controls are updated accordingly (checkboxes replaced by popup lists).
4. Added the possibility to **save the normalised Reference Spectra**
5. Added the possibility to **use the normalised Reference Spectra as Initial Estimates**

2016 August, v4b

6. **Input** now accepts **Microsoft Excel .xlsx** files
7. **SOpt, COpt, Wavenumbers** and the **Matrix4MCR** variables are **loaded into the Matlab workspace**, allowing direct copy & paste without saving
8. **Normalisation methods** expanded from “None” and “Total Area Normalisation” to include “Total Min-Max”, “Region Area”, “Region Min-Max”, “Region Max”, “Point Max” and “Offset” types
9. **Reference spectra** do not have to be identical in spectral range or resolution and do not have to match the data in those respects either. They are automatically **made compatible** upon loading. The compatible spectra can also be saved for later use or for use outside the GUI.
10. **Minor layout changes**: The Save MCR Results button relocated, the oversampling and refolding controls adjusted in position

Terms and Conditions and requires MATLAB, which in turn is subject to terms and conditions by Mathworks.

MATLAB version 2013a or later is recommended for all functions to work, although core features are possible to run on older versions.

The code and this manual is based on scripts, techniques and descriptions from the following sources and references therein (sometimes taken almost literally without quotations or references given in the main text):

J. Felten, H. Hall, J. Jaumot, R. Tauler, A. de Juan, A. Gorzsás, *Nature Protocols*, **2015** (GUI element parts)

P. H. C. Eilers, *Analytical Chemistry*, **2004** (Baseline correction)

A. Savitzky, M. J. E. Golay, *Analytical Chemistry*, **1964** (Smoothing)

J. Jaumot, R. Gargallo, A. de Juan, R. Tauler, *Chemometrics and Intelligent Laboratory Systems*, **2005** (MCR-ALS, including constraints and automatic initial estimate calculations)

G. Blanchet, M. Charbit, *Digital Signal and Image Processing Using MATLAB*, ISBN 978-1-905209-13-2, **2006** (Oversampling)

S. Piqueras, L. Duponchel, R. Tauler, A. de Juan, *Analytica Chimica Acta*, **2011** (Segmentation)

MATLAB documentation, http://mathworks.com/help/matlab/ (core function definitions and descriptions)

The Manual is not intended as scientific publication or as an exhaustive resource on any functions of the GUI or the corresponding theories.

The demo examples have been recorded at the Vibrational Spectroscopy Core Facility, Umeå University, Umeå, Sweden and are modified versions of the datasets described in J. Felten, H. Hall, J. Jaumot, R. Tauler, A. de Juan, A. Gorzsás, *Nature Protocols*, **2015** (demo_image.txt and demo_indep_spectra.xlsx) and T. Gillgren, A. Gorzsás, *Wood Science and Technology*, **2016** (demo_series.mat).

Changelog

2016 September, v4c

11. **Equality constraints** fixed
12. **Unimodality constraints** fixed
13. **Unimodality mode selection** options updated: no longer necessary to have the same mode for concentrations and spectra, they are independent. GUI layout and controls are updated accordingly (checkboxes replaced by popup lists).
14. Added the possibility to **save the normalised Reference Spectra**
15. Added the possibility to **use the normalised Reference Spectra as Initial Estimates**

2016 August, v4b

16. **Input** now accepts **Microsoft Excel .xlsx** files
17. **SOpt, COpt, Wavenumbers** and the **Matrix4MCR** variables are **loaded into the Matlab workspace**, allowing direct copy & paste without saving
18. **Normalisation methods** expanded from “None” and “Total Area Normalisation” to include “Total Min-Max”, “Region Area”, “Region Min-Max”, “Region Max”, “Point Max” and “Offset” types
19. **Reference spectra** do not have to be identical in spectral range or resolution and do not have to match the data in those respects either. They are automatically **made compatible** upon loading. The compatible spectra can also be saved for later use or for use outside the GUI.
20. **Minor layout changes**: The Save MCR Results button relocated, the oversampling and refolding controls adjusted in position

Terms and Conditions

1. Introduction

1.1. What This Manual Is

The purpose of this Manual is to provide operational guidelines for the MCR-ALS GUI developed at the Vibrational Spectroscopy Core Facility at Umeå University and it thus lists every user control of the GUI and their actions.

This Manual does NOT aim to provide a comprehensive description of ANY of the techniques that are utilised by the GUI, be it a built-in MATLAB function (such as the Savitzky-Golay filtering) or a custom-built function (such as the MCR-ALS algorithm). For those details, references are given in **Disclaimer and Credits** and in comments in the source code.

This Manual does not explain any of the theoretical backgrounds, although **several demo datasets are provided** to allow testing how the GUI works in action. These datasets are the ones used in the figures of this Manual to illustrate the steps, thus they can be used along the Manual. There is one dataset to show how images are handled ("demo_image.txt"), one for a series of spectra ("demo_series.mat") and another that contains a set of independent spectra ("demo_indep_spectra.xlsx"). One of them is in MATLAB .mat format, another in Microsoft Excel .xlsx format, and the last one as an ASCII .txt file, to demonstrate all kinds of input possibilities and data types. **Auxiliary data is also provided** with the datasets (such as reference spectra to test reference matching, white light image for the image dataset, etc.). **These also serve as templates for determining data input formats for the GUI**.

The code is being maintained, updated and expanded continuously without fixed dates or deadlines. The Manual refers to the current version only, although a **Changelog** is provided to list major alterations. Known bugs and their possible workarounds are described at the relevant sections and marked as "☒ **Known bug:**" and "☑ **Workaround:**", respectively. Actions that require special attention or contain important information are marked as "ⓘ **Warning/Information:**" in the text.

1.2. What The GUI Is

The GUI was originally developed to handle hyperspectral images, in order to facilitate the analysis of vibrational microspectroscopic data of biological tissue without in-depth knowledge of spectroscopy or chemometrics. Thus, the main goal of the GUI is to provide a user-friendly interface, where different data processing parameters can be easily and interactively tested, and their effect on the outcome of the analysis determined (both for educational and scientific purposes). The main objective is to allow the "owner" of the data to process it at least at a basic level, without having to involve a chemometrician or spectroscopist.

The GUI has separate controls for pre-processing the data (baseline correction, smoothing, etc), performing MCR-ALS analysis with a range of possible constraints, and to evaluate the results: visualisation and

classification controls, reference spectra matching and segmentation. Each of them are optional, and data can be saved and exported at any step, for plotting (e.g. in Microsoft Excel) or further processing by other means (such as OPLS-DA analysis in Umetrics' SIMCA-P software).

Special care was taken to write the code in a manner that is easy to follow: most operations are detailed step-by-step in the code, instead of combining them into a single line, and most functions are heavily commented. Thus, the GUI code is optimised for easy understanding and flexibility, and NOT for speed, although care was taken to select the fastest solutions whenever possible.

While the GUI was originally developed for vibrational spectroscopic images, it can handle all spectroscopic datasets, not only images and not only vibrational spectra (although the GUI code, comments and this Manual will frequently use "Wavenumbers" as a synonym for the spectral variables), as long as they meet the **Data Format Requirements**.

1.3. System Requirements

For editing, MATLAB is required. Version 2013a or later is recommended for all functions to work, although core features are possible to run on older versions too. The GUI itself runs on very modest hardware architectures (if MATLAB can run, so can the GUI), but it can be sluggish, especially for large datasets.

Mac and Windows compatibility issues are documented in this Manual wherever they arise.

ⓘ **Warning/Information:** The GUI contains lots of elements and although it rescales to match the screen, it is ideal on **screens with resolutions 1920 x 1200 or higher**. On smaller displays, controls could be partially covered or illegible.

1.4. Data Format Requirements

Data format requirements MUST be met in order for the GUI to work. Changing these requirements requires extensive modification of the source code.

ⓘ **Warning/Information:** There is **no safety mechanism** in the GUI **for detecting and correcting a wrong user input data format**. The GUI will simply crash, with error messages shown in the MATLAB Command Prompt.

ⓘ **Warning/Information:** Mac computers and MATLAB are **case sensitive**. Try to use only lower case extensions for all file types.

ⓘ **Warning/Information:** Use only **standard Latin alphanumeric characters and underscores** in filenames to avoid potential problems. Do not use spaces or any special characters, including accented letters, parentheses, etc.

ⓘ **Warning/Information:** Make sure **regional settings** do not interfere with each other (decimal dots vs decimal commas).

1.4.1. Input data

1.4.1.1. .mat file input

The input .mat file must be of the following structure (default with Bruker's OPUS 7 export into Matlab format):

4000 0.0473 0.0498 0.0953 … 0.0622

3998 0.0423 0.0123 0.1022 … 0.0817

3996 0.0494 0.0331 0.1169 … 0.0724

…. …… …… …… … ……

400 0.0512 0.0678 0.0744 … 0.0688

, where the first column is the wavenumbers, and thereafter each column is one spectrum (see the file "demo_series.mat"). For spectral series, spectra should be in order, there is no possibility to re-order them in the GUI, although refolding for visualisation is possible (see Change Dimensions (Refold)). For image datasets, each spectrum represents one pixel / voxel, with the first row first column spectrum being the first in the list, then first row second column spectrum, and so on. As with spectral series, re-ordering the spectra is not possible. While re-folding is possible (see Change Dimensions (Refold)), it is not recommended for images.

ⓘ **Warning/Information:** Since the .mat files formatted this way do not contain X and Y coordinates, **the .mat filename MUST EITHER HAVE the following format:** AAAxBBB_filename.mat to describe the X and Y dimensions, with AAA denoting the number of pixels in the X dimension and BBB denoting the number of pixels in the Y dimension (e.g. 064x064 –note the leading zeros–, or 128x128), **OR the user MUST ENTER THE CORRECT X and Y** dimensions manually in a pop-up dialog box. The pop-up dialog box by default suggests X = total number of spectra in the dataset, Y = 1.

ⓘ **Warning/Information:** This file format loads the fastest and is recommended for large data files.

1.4.1.2. .xlsx file input

The input .xlsx file must be of the following structure:

4000 3998 3996 … 400

0.0473 0.0498 0.0953 … 0.0622

0.0423 0.0123 0.1022 … 0.0817

0.0494 0.0331 0.1169 … 0.0724

…… …… …… … ……

0.0512 0.0678 0.0744 ... 0.0688

, where the first row is the wavenumbers, thereafter each row is one spectrum (see the file "demo_indep_spectra.xlsx"). For spectral series, spectra should be in order, there is no possibility to re-order them in the GUI, although refolding for visualisation is possible (see Change Dimensions (Refold)). For image datasets, each spectrum represents one pixel / voxel, with the first row first column spectrum being the first in the list, then first row second column spectrum, and so on. As with spectral series, re-ordering the spectra is not possible. While re-folding is possible (see Change Dimensions (Refold)), it is not recommended for images.

ⓘ **Warning/Information: Spectra MUST be in the first worksheet** of the Excel file, starting in cell A1 with the first wavenumber. DO NOT include filenames, pixel numbers or any other descriptors for the spectra in the first worksheet of Excel file, only the wavenumbers and intensities. If such information needs to be included, keep it on another worksheet. The GUI will only read in THE FIRST WORKSHEET of the Excel file, so other information can safely be stored in additional worksheets without causing problems at data load.

ⓘ **Warning/Information: USE DECIMAL DOTS ONLY**. Other regional settings, such as decimal commas, will result in errors at load.

ⓘ **Warning/Information:** Since the .xlsx files formatted this way do not contain X and Y coordinates, **the .xlsx filename MUST EITHER HAVE the following format:** AAAxBBB_filename.xlsx to describe the X and Y dimensions, with AAA denoting the number of pixels in the X dimension and BBB denoting the number of pixels in the Y dimension (e.g. 064x064 –note the leading zeros–, or 128x128), **OR the user MUST ENTER THE CORRECT X and Y** dimensions manually in a pop-up dialog box. The pop-up dialog box by default suggests X = total number of spectra in the dataset, Y = 1.

ⓘ **Warning/Information:** This data format is the easiest to append with additional information (in extra worksheets) and is thus the best choice to process independent spectra.

1.4.1.3. .txt file input (images only, no series)

The input .txt file must be of the following structure (default with Renishaw's WiRE3 export into .txt format):

1 1 1800 4528

1 1 1799 4321

1 1 1798 4413

. . …. ….

1 2 1800 4619

1 2 1799 4721

1 2 1798 4812

. . …. ….

1 3 1800 4311

1 3 1799 4289

1 3 1798 4156

. . …. ….

2 1 1800 4555

2 1 1799 4369

2 1 1798 4611

. . …. ….

, where the first column is the Y(!) coordinate and the second column is the X coordinate of the pixel, the third columm being the wavenumbers and the fourth column is the intensities (see the file "demo_image.txt").

ⓘ **Warning/Information:** This is an extremely inefficient format with lots of unnecessary redundancy, thus **large datasets should preferentially be in .mat format**. The only advantage of this .txt format (apart from being the native export of WiRE 3), is that X and Y dimensions do not have to be supplied (either manually or in the filename, as for .mat files).

1.4.1.3 White light image input

For image datasets, the corresponding visible (white light) map can be automatically loaded, if the file is in the same folder as the data file and has the exact same file name but .jpg extension. Any other image file supported by MATLAB can also be used, but they have to be manually loaded, using the "Load White Light Image" button in the Visualisation section of the GUI (see **White Light Image**).

1.4.2. Reference Spectra

ⓘ **Warning/Information:** Reference spectra **will NOT be pre-processed** by the pre-processing parameters. However, reference spectra **will be made compatible** with the dataset automatically upon loading and will be total area normalised for display. Compatibility means that reference spectra will be either trimmed to the length of the dataset (i.e. excluding all wavenumbers outside the spectral region of the dataset) or/and expanded to match the length of the dataset (adding zeros, i.e. baseline). In addition, spectral resolution will be adjusted to match that of the dataset, using the built-in "spline" interpolation algorithm of MATLAB.

1.4.2.1. .mat or .txt file input

Each file should contain a single spectrum, first column being wavenumbers, second column intensities (see the reference files supplied as .txt files in the "References" folder). Multiple files are allowed to be selected for multiple reference spectra to be loaded, but they all have to be in the same folder. The file names (without path) will be used for labelling the reference spectra.

1.4.2.2. Microsoft Excel .xlsx file input

ⓘ **Warning/Information: This is the preferred option to load reference files.**

Only one file is allowed, which should contain all reference spectra, in the first worksheet.

Each spectrum should take two rows: the first row is the wavenumbers and the second row is spectral intensities. The spectrum name should be given in the first column where the intensities are located. These names will be used for labelling the reference spectra.

Example ("<>" denoting an empty cell in the Excel worksheet, each "<xxx>" denoting cells with content, and "…" denoting repetition):

<> <4000> <3999> <3998> … <400>

<Reference1> <0.2987> <0.2966> <0.2958> … <0.1999>

<> <3900> <3880> <3860> … <380>

<Reference2> <0.1435> <0.1234> <0.1336> … <0.1023>

Note that this example contains 2 reference spectra, with names "Reference1" and "Reference2", with different spectral ranges and spectral resolutions.

See the included "demo_references.xlsx" file for an example.

1.4.3. Initial Estimate Input

All data should be in the first worksheet of a single Microsoft Excel .xlsx file. Non-continuous ranges will be ignored (i.e. do not leave gaps / empty cells).

For spectrum input, the file must have as many rows as the number of components +1 (for the wavenumbers in the first row) and as many columns as there are wavenumbers in the dataset + 1 (for the component names in the first column). If spectra are used for input, each spectrum should be one row, with the first row being the wavenumbers, and first column should be the name / identifier of the spectrum. See "InitEstimates_spectra.xlsx".

For concentration input, the file must have as many rows as the number of spectra in the dataset + 1 (for the component names in the first row), and as many columns as there are components in the MCRALS model. Thus, each component concentration profile should be one column. The first row is the name of the component, thereafter each row should represent one spectrum (i.e. the concentrations of each

component in that spectrum). Component names must contain at least one letter, cannot be only numbers (i.e. “Comp1” is correct, “1” is not). See “InitEstimates_conc.xlsx”.

ⓘ **Warning/Information:** Spectra should have the same length as the spectra in the dataset (same spectral range AND spectral resolution, i.e. the same set of wavenumbers), otherwise the file cannot be loaded. This is intentional and not a bug. If the spectra to be used for initial estimate inputs have different spectral regions / spectral resolution, load them as reference spectra first (see **Microsoft Excel .xlsx file input** for Reference spectra input format requirements). This will make them compatible with the dataset automatically and they can be passed on as Initial Estimates directly using the dedicated button (see **Use References as Initials**).

1.4.4. Equality Constraint Matrices

All data should be in a single .mat file.

If spectra equality is used, the .mat matrix must contain as many rows as the number of components to be resolved, and as many columns as there are wavenumbers in the dataset. For all the components that are to be constrained, the row should contain numbers (i.e. spectral intensities at each wavenumbers), for unconstrained components (or spectral regions), the row (and colums) should contain either NaN or Inf values.

If concentration equality is used, the .mat matrix must contain as many columns as the number of components to be resolved, and as many rows as there are spectra in the dataset. For the components to be constrained, the column should contain numbers describing the concentration of that component in every spectrum in the dataset where the concentration is to be constrained. The column should contain Nan or Inf values for the spectra where the concentration is left unconstrained. For the components that are left unconstrained, their column should contain either NaN or Inf values only.

See the included “csel.mat” and “ssel.mat” auxiliary data files.

1.4.5. Pre-Processing Parameter Matrix

Only use a .mat file that was saved by the GUI or has the exact same list of parameters in the exact same format (see

Pre-Processing Parameter Matrix output format and the “demo_image_PreProcessParam.mat” auxiliary data file).

1.5. Output Data Formats

1.5.1. Pre-Processing Parameter Matrix

All pre-processing parameters are saved in a single .mat file, containing a 2x13 cell matrix called "PreProcParam", with the first row containing the parameter names, and the second row the corresponding parameter values. The parameters that are saved are as follows (in the order they appear in the .mat file)

"**MinSpectralRange**", the lower wavenumber end of the spectral region

"**MaxSpectralRange**", the lower wavenumber end of the spectral region.

"**AsLS Lambda**", the lambda value of asymmetric least squares baseline correction.

"**AsLS p**", the p value of asymmetric least squares baseline correction.

"**Normalisation**", 0 = no normalisation, 1 = total area normalisation, 2 = Min-Max normalisation in the entire spectral range, 3 = area normalisation in a defined region of the spectrum only (the region being defined by the MinIntegralRange and MaxIntegralRange values), 4 = Min-Max normalisation in a defined region of the spectrum only (the region being defined by the MinIntegralRange and MaxIntegralRange values), 5 = Maximum intensity normalisation in a defined region of the spectrum only (the region being defined by the MinIntegralRange and MaxIntegralRange values), 6 = Maximum intensity normalisation at a defined point in the spectrum (the point being defined by the SinglePointIntensity value), 7 = Offset correction at a defined point in the spectrum (the point being defined by the SinglePointIntensity value)

"**S-G Filter**", 1 for Savitzky-Golay filtering (smoothing) to be performed, 0 for no Savitzky-Golay filtering (smoothing) to be performed

"**S-G PolyOrder**", the polynomial order for for Savitzky-Golay filtering (smoothing)

"**S-G FrameNr**", the frame number for for Savitzky-Golay filtering (smoothing)

"**MinIntegralRange**", the lower edge of the integral range

"**MaxIntegralRange**", the higher edge of the integral range

"**ShowIntegral**", 1 for showing the integral area in the Selected Spectrum plot, 0 for not showing it

"**SinglePointIntensity**", the wavenumber at which the intensity should be evaluated

"**Show Single Point Intensity**", 1 for showing the single point intensity marker line in the Selected Spectrum plot, 0 for not showing it

1.5.2. Cut and Pre-Processed Spectra

Irrespective of the original input format, cut spectra are saved in a .mat file in the following format:

4000 0.0473 0.0498 0.0953 ... 0.0622

3998 0.0423 0.0123 0.1022 … 0.0817

3996 0.0494 0.0331 0.1169 … 0.0724

…. …… …… …… … ……

400 0.0512 0.0678 0.0744 … 0.0688

, where the first column is the wavenumbers, and thereafter each column is one spectrum. This .mat file can be read in directly by the GUI.

ⓘ **Warning/Information: Spectra are always saved as .mat files**, irrespective of the original input format. The .mat file is saved in the format that can be directly opened by the GUI.

ⓘ **Warning/Information: The .mat file does NOT contain XY dimension information (for image data sets) or filenames (for spectra series)**. The XY dimensions can be added to the filename upon saving (default suggestion by the GUI) but spectrum names must be saved separately for spectra series

1.5.3. Integral and Single Point Intensity Matrices

Only one integral and one single point intensity matrix can be saved at a time. The result is a .mat file with a single MATLAB matrix variable (double), called “Int” (for integrals) or “SPointInt” for single point intensities, containing a single column of integral / single point intensity values and as many rows as there are spectra in the dataset.

ⓘ **Warning/Information: The .mat file does NOT contain XY dimension information (for image data sets) or filenames (for spectra series) or the integral range / single point intensity position.** The XY dimension and the integral range / single point intensity position can be added to the filename upon saving (default suggestion by the GUI) but spectrum names must be saved separately for spectra series.

1.5.4. MCR-ALS Optimised Spectra

Resolved spectral profiles are saved in a single .mat file, containing a single MATLAB matrix variable (double), called “SOpt”, containing as many rows as there were components +1, and as many columns as there are wavenumbers. The first row contains the wavenumber list, thereafter each row contains one resolved spectral profile.

1.5.5. MCR-ALS Optimised Concentrations

Resolved concentration profiles are saved in a single .mat file, containing a single MATLAB matrix variable (double), called "COpt", containing as many rows as there are spectra in the dataset, and as many columns as the number of resolved components. Each column contains the concentration value of a resolved component in each spectrum.

ⓘ **Warning/Information: The .mat file does NOT contain XY dimension information (for image data sets) or filenames (for spectra series).** The XY dimension can be added to the filename upon saving (default suggestion by the GUI) but spectrum names must be saved separately for spectra series.

1.5.6. MCR-ALS Results

The end results of MCR-ALS optimisations can be saved in a single .mat file, which contains the following MATLAB variables:

"**COpt**", containing the resolved concentrations profiles, consisting of as many rows as there are spectra in the dataset, and as many columns as the number of resolved components. Each column contains the concentration value of a resolved component in each spectrum.

"**Ending**", containing a number to describe whether convergence was reached ("1"), divergence occurred ("2") or the maximum number of iterations were reached ("3").

"**Iteration**", containing a number, denoting the number of iteration after which MCR-ALS was finished.

"**R2Opt**", containing a number, denoting the percentage of variance explained when MCR-ALS was finished (1.00 being 100 percent)

"**SDOpt**", containing two numbers denoting the lack of fit, the first in % PCA, the second in % experimental data.

"**Sigma**", containing a number, denoting the change in sigma value

"**SOpt**", containing the resolved spectral profiles, consisting of as many rows as there were components, and as many columns as there are wavenumbers. As opposed to saving the **MCR-ALS Optimised Spectra**, this variable **does NOT contain the wavenumber list**!

ⓘ **Warning/Information: The .mat file does NOT contain XY dimension information (for image data sets) or filenames (for spectra series).** The XY dimension can be added to the filename upon saving (default suggestion by the GUI) but spectrum names must be saved separately for spectra series.

ⓘ **Warning/Information: The .mat file does NOT contain information about the starting or finishing conditions of the MCR-ALS procedure** (direction, noise level, initial estimates, and convergence criterion)

ⓘ **Warning/Information: The .mat file does NOT contain information about the constraints used for the MCR-ALS procedure** (equality or unimodality).

1.5.7. Marked Spectra

Each marked spectrum is saved as an individual .mat file, containing the MATLAB variable "MarkedSpectrum", with two columns: the first is the wavenumbers, the second is the intensities. These .mat files can be directly loaded as **Reference Spectra**.

ⓘ **Warning/Information: The .mat file does NOT contain the spectrum number (for image data sets) or filename (for spectra series) to denote its origin.** The spectrum number can be added to the filename upon saving as the default suggestion by the GUI, but the spectrum names must be added manually for spectra series.

1.5.8. Compiled Class Matrix

All marked spectra are compiled into the first worksheet of a single Microsoft Excel .xlsx file. The first row contains the wavenumbers, the first column the names (the original dataset name, plus the spectrum number), the second column their assigned class (class IDs), and each row is one spectrum.

This file can be directly imported into SIMCA-P, using the first column as Primary Observation ID and the first row as Primary Variable ID. The second column can be used as Y variable or class ID or secondary observation ID, or left unused.

ⓘ **Warning/Information: This file can NOT be directly used / loaded as Reference Spectra**, since it does not match the input format criteria for Microsoft Excel .xlsx file input without adjustments.

1.5.9. Reference Match Results

The results of reference matching are saved in the first worksheet of a single Microsoft Excel .xlsx file as a table with descriptive headings. Each row is one resolved component, each column is one reference spectrum, and each value of the table describes how well the component and reference matches, based on Euclidean distances.

ⓘ **Warning/Information: The file does NOT contain the resolved components spectral profiles OR the reference spectra, and it does not contain ANY plots to visual validation of the match.**

1.5.10. Segmentation Results

The results of k-means clustering can be saved in a single .mat file, which contains the following MATLAB variables:

"**Centr**", containing the centroid profiles, i.e. the contribution of each resolved component to cluster, and thus consisting of as many rows as there are clusters and as many columns as the number of MCR-ALS resolved components.

"**IDX**", containing the cluster identifiers, i.e. a single number to denote which cluster each spectrum in the dataset belongs to. Thus, it consists of a single column with as many rows as there are spectra in the dataset.

ⓘ **Warning/Information: The .mat file does NOT contain XY dimension information (for image data sets) or filenames (for spectra series).** The XY dimension can be added to the filename upon saving (default suggestion by the GUI) but spectrum names must be saved separately for spectra series.

1.5.11. Plots

Each figure is saved as an individual file.

Plots of the main GUI can be saved by their dedicated buttons in .pdf, .jpg, .tif, .png, .gif or .eps formats.

Plots of pop-up windows opened by the GUI, such as the reference matching results (see **Match References**), showing all pre-processed spectra (see **3.10. Displaying the Pre-Processed** Data), showing PCA results (see **Show PCA**) and showing component maps (see **Show Component Maps**) can be saved by the menus / icons in their own figure windows, in any of the formats that MATLAB supports.

ⓘ **Warning/Information: Mac and Windows compatibility and resolution problems of the saved plots may arise**. Test different formats to find the optimal solution to the current computer architecture.

2. Opening the GUI

2.1 Start

Start MATLAB and navigate to the MCRALS_v4c.m file in the browser pane. Right-click on the file and select "Run" from the opening context sensitive menu.

> ⓘ **Warning/Information: The corresponding MCRALS_v4c.fig file must be in the same folder.**

If the folder in which the MCRALS_v4c.m and .fig files reside is not in the MATLAB path, a dialog box opens. Choose the "Change to Folder" option.

2.2 Layout

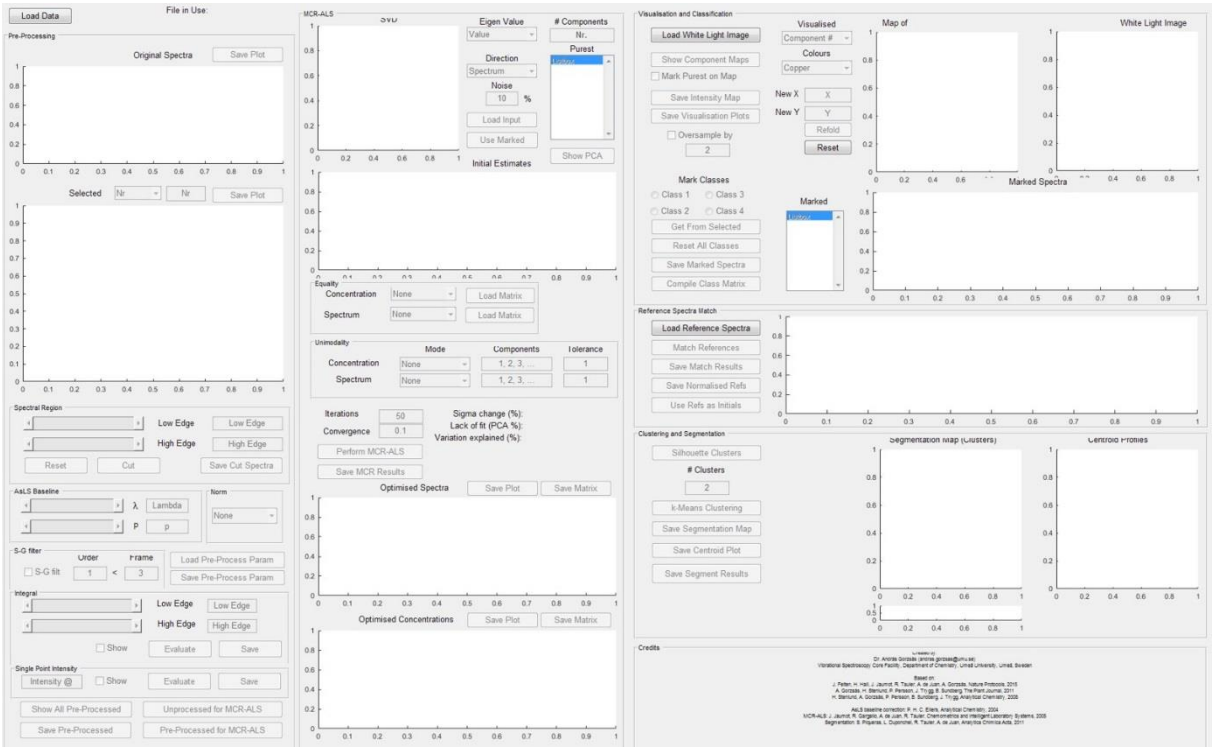

Figure 1: The empty GUI at start

The GUI is designed in the way that controls lacking input value are disabled by default. Thus, at start the only active ones are the "Load Data" and "Load Reference Spectra" buttons (***Figure 1***). As the GUI populates, more options become available.

> ☒ **Known bug:** In certain cases, if the reference spectra are loaded before (or during) loading the spectral data, the "Match References" button remains inactive.
>
> ☑ **Workaround:** Load the reference spectra after the spectral data has finished loading.

☒ **Known bug:** If there has been data in the GUI and a new dataset is loaded, the "Selected" spectrum drop-down list may not get rendered. This is because the last used value is outside the new dataset boundaries (As an example: if the "demo_image.txt" is loaded, and spectrum 102 is shown in the "Selected" spectrum plot, then the "demo_indep_spectra.xlsx" data is loaded, the drop-down menu disappears, as it tries to refer to spectrum 102 in a dataset that only contains 15 spectra.)

☑ **Workaround:** Type "1" in the "Selected" textbox and hit "Enter/Return" on the keyboard. This will set the display to the first spectrum and the "Selected" spectrum drop-down menu appears with correct values.

☒ **Known bug:** In some cases, the new dataset does not load properly (especially if some controls throw error messages during loading, such as the "Selected" spectrum drop-down list) and plots do not update.

☑ **Workaround:** Correct the error (see the "Workaround" tip above for the "Selected" textbox error) and load the dataset again.

ⓘ **Warning/Information:** The GUI contains lots of elements and although it rescales to match the screen, it is ideal on **screen resolutions 1920 x 1200 or higher**. On smaller displays, controls could be partially covered or illegible.

The GUI can be divided into 6 major sections (***Figure 2***), and while each are optional, they may depend on each other for input.

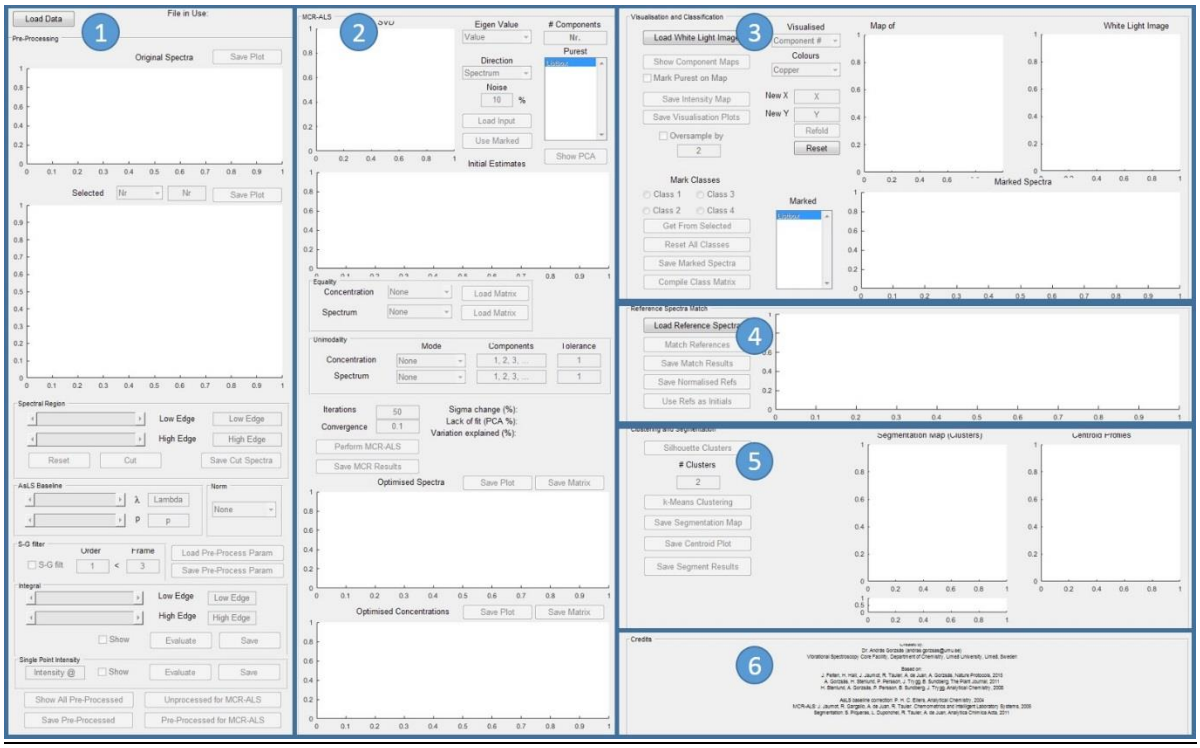

Figure 2: The empty GUI at start, with the sections marked

Section 1: Pre-processing, also including controls for loading the data, saving processed-data and passing on the data for MCR-ALS (Section 2) and Visualisation (Section 3)

Section 2: MCR-ALS, including controls for determining the number of components, determining the initial estimates (with a possibility to use marked spectra from Section 3), setting MCR-ALS constraints and parameters and performing MCR-ALS analysis. The results are automatically passed on for visualisation to Section 3.

Section 3: Visualisation controls, including marking and assigning classes (with a possibility to select the displayed spectrum from Section 1 directly) and exporting the marked classes or using them as initial estimates for MCR-ALS (Section 2), refolding and oversampling and changing the colour scheme of any of the visualisation plots.

Section 4: Reference Spectra Matching, based on Euclidean distances. Requires a completed MCR-ALS analysis (Section 2) AND loaded reference spectra for input.

Section 5: Segmentation, based on k-means clustering. Requires a completed MCR-ALS analysis (Section 2) for input.

Section 6: Static credits, with no user interaction.

Each section is discussed in details in the chapters below.

ⓘ **Warning/Information:** The GUI uses colour coding for the buttons: **black text** means the button is a regular control; **blue text** means the button saves the data in the way that it can be later read in by the GUI directly, without modification; **red text** means the button can be critical in passing information to another part of the GUI.

3. Pre-Processing

3.1 Loading the Data

Click on the “Load Data” button. If there was no data loaded before, a standard dialog box opens, showing the folder in which the MCRALS_v4c.m file resides, and all .mat files in that folder by default. Change folder and file type as needed.

> ☒ **Known bug:** In certain cases, the Finder in OSX systems does not update upon changing the file type in the dialog window, and the newly selected file types remain greyed out.
>
> ☑ **Workaround:** This bug only effects the look of the dialog window, not its function. Thus, the file types can still be opened even if they are greyed out. Alternatively, select “all files” in the file type option or change the folder in the dialog window to force Finder to update the display.

If there was a dataset loaded already, loading a new dataset will completely overwrite all the previous data in the GUI. It will NOT affect the original and saved files, only the data that is not saved. A Warning/Information Dialog window opens to inform about this and allowing for cancellation of the loading of the new data**. Pre-processing and MCR-ALS parameters,** however, **are kept in the GUI,** thus the new dataset can quickly be processed in the exact same way as the previous dataset was, without having to reset (or reload) all parameters.

> ⓘ **Warning/Information:** To make sure that no carryover variable affects a completely new dataset, it may be necessary to close the entire GUI window, type “clear all” in the MATLAB Command Window and restart the GUI (see **Opening the GUI**).
>
> ☒ **Known bug:** When loading a new dataset to overwrite the current one, some controls may be impossible to render, and MATLAB displays a notification message in the command prompt. Example: if integration was performed on the previous dataset, the “Visualised” drop-down list will not be rendered when the new dataset is loaded, because it will be missing the integrals that were visualised in the previous dataset.
>
> ☑ **Workaround:** Either completely close the GUI and restart, or perform a dummy action to generate the data to enable rendering the GUI element. Example: if the “Visualised” drop-down list is not rendered because it misses the integrals from the previous dataset, perform an integration (any integration, it does not have to be the same as the previous dataset had). This will enable the drop-down menu to be rendered correctly.

Select the file to be opened and either double-click it, or click on the “Open” button in the dialog window.

> ⓘ **Warning/Information: .mat and .xlsx files must either contain the X and Y dimensions in the filename (see** .mat file input**) or the correct dimensions must be supplied in the opening dialog**

box. For image files, it is especially important to have the correct X and Y dimensions otherwise the data will be folded wrongly. For independent spectra or a series of spectra it is only important that X*Y equals the total number of spectra in the dataset. The GUI suggests X and Y dimensions by default, with X = total number of spectra in the dataset, and Y = 1 (**Figure 3**). **There is no mechanism to correct for erroneous X and Y input**. **With wrong dimension supplied or the "Cancel" button hit in the X and Y dimension dialog window (Figure 3), the GUI will simply crash with error messages shown in the MATLAB command prompt.** In this case, the GUI window may need to be closed and the script restarted (See **2.1** Start).

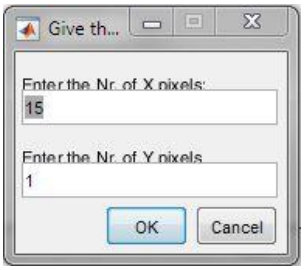

Figure 3: X and Y dimension dialog window, to manually supply the requested dimension for .mat and .xlsx files that do not contain this information in the filename. *By default, the Y dimension is set to 1, and the X dimension to the total number of spectra in the dataset (in this case 15). The example uses the "demo_indep_spectra.xlsx" file.*

The data is loaded, the filename is shown in red under the "File in Use:" title, and spectra are shown in the Original Spectra window of the Pre-Processing section of the GUI. The "Save Plot" button of this plot becomes active (**Figures 4 and 5**).

ⓘ **Warning/Information: If the dataset contains more than 100 spectra, only 100 randomly selected ones are shown in the Original Spectra window, to speed up the GUI (Figure 4).**

ⓘ **Warning/Information: Large files may take a long time to load**, especially on slower computers and in case of .txt files, which have suboptimal structure with lots of redundancy (see

ⓘ **Warning/Information:** This file format loads the fastest and is recommended for large data files.

3.1.1.1. .xlsx file input

The input .xlsx file must be of the following structure:

4000 3998 3996 ... 400

0.0473 0.0498 0.0953 ... 0.0622

0.0423 0.0123 0.1022 ... 0.0817

0.0494 0.0331 0.1169 ... 0.0724

....

0.0512 0.0678 0.0744 ... 0.0688

, where the first row is the wavenumbers, thereafter each row is one spectrum (see the file "demo_indep_spectra.xlsx"). For spectral series, spectra should be in order, there is no possibility to re-order them in the GUI, although refolding for visualisation is possible (see Change Dimensions (Refold)). For image datasets, each spectrum represents one pixel / voxel, with the first row first column spectrum being the first in the list, then first row second column spectrum, and so on. As with spectral series, re-ordering the spectra is not possible. While re-folding is possible (see Change Dimensions (Refold)), it is not recommended for images.

ⓘ **Warning/Information: Spectra MUST be in the first worksheet** of the Excel file, starting in cell A1 with the first wavenumber. DO NOT include filenames, pixel numbers or any other descriptors for the spectra in the first worksheet of Excel file, only the wavenumbers and intensities. If such information needs to be included, keep it on another worksheet. The GUI will only read in THE FIRST WORKSHEET of the Excel file, so other information can safely be stored in additional worksheets without causing problems at data load.

ⓘ **Warning/Information: USE DECIMAL DOTS ONLY**. Other regional settings, such as decimal commas, will result in errors at load.

ⓘ **Warning/Information:** Since the .xlsx files formatted this way do not contain X and Y coordinates, **the .xlsx filename MUST EITHER HAVE the following format:** AAAxBBB_filename.xlsx to describe the X and Y dimensions, with AAA denoting the number of pixels in the X dimension and BBB denoting the number of pixels in the Y dimension (e.g. 064x064 –note the leading zeros–, or 128x128), **OR the user MUST ENTER THE CORRECT X and Y** dimensions manually in a pop-up dialog box. The pop-up dialog box by default suggests X = total number of spectra in the dataset, Y = 1.

ⓘ **Warning/Information:** This data format is the easiest to append with additional information (in extra worksheets) and is thus the best choice to process independent spectra.

.txt file input (images only, no series)). In general, **.mat files are recommended for large data set input**.

The rest of the Pre-Processing section of the GUI is populated, and controls for spectrum selection (see **3.2 Selecting a spectrum for** display), spectral region trimming (see **3.3. Spectral Region Adjustment (**Cutting)), baseline correction (see **3.4. Asymmetrical Least Squares Baseline** Correction), Savitzky-Golay filtering (see **Figure *10: Illustrating the effects of normalisation***. *a: No Normalisation; b: Total Area Normalisation; c: Total Min-Max Normalisation; d: Region Max Normalisation, using the 1550-1640 cm^{-1} region (aromatic -C=C- band); e: Region Area Normalisation, using the 1550-1640 cm^{-1} region (aromatic -C=C- band); f: Region Min-Max Normalisation, using the 1550-1640 cm^{-1} region (aromatic -C=C- band); g: Point Max Normalisation, using the intensity at 1600 cm^{-1} (aromatic -C=C- band); h: Offset Normalisation, using the*

intensity at 1600 cm^{-1} (aromatic -C=C- band); i: Offset Normalisation, using the intensity at 1200 cm^{-1} (assumed baseline point).

*The example uses the "demo_image.txt" file, after optimum baseline correction in **Figure 9** (spectral region high edge cut to "1800", asymmetrical least squares baseline lambda=50 000 and p=0.001 (minimum)).*

3.6. Smoothing), normalisation (see **3.5.** Normalisation), intensity determination (either as band area, see **3.7.** Integration, or as a single value, see **3.8. Single Point Intensity** Evaluation).

The **Visualisation section** of the GUI shows the total intensity map of the dataset (i.e. the total area under all peaks in the spectrum, without pre-processing). The "Visualised" drop-down menu is activated and shows "Tot Int" to reflect this fact. The "Colours" drop-down menu is activated, allowing to change the colour map of the visualised plots to any of MATLAB's built-in sets. The default set is "Copper" to avoid confusions with MCR-ALS component colours. The "Save Intensity Map" and "Save Visualisation Plots" buttons (see **Save Intensity Map** and **Save Visualisation Plots**) become active, allowing to save the current Total Intensity Map as an image file (**Figures 4 and 5**).

> ⓘ **Warning/Information:** Only the image is saved by "Save Intensity Map", the total intensity data is NOT. That can only be saved via the "Save" button in the "Integral" part of the Pre-Processing section (see **3.7.** Integration).

If a hyperspectral image dataset was loaded, a corresponding visible (white light) image is loaded by default (Figure 4), provided that it has the same filename as the data file but with .jpg extension and it resides in the same folder as the loaded data. If that is not the case, a visible image can be loaded manually by the "Load White Light Image" button in the Visualisation section of the GUI (see **White light image input**).

The controls for image refolding also become active, showing the current X and Y coordinates in the "New X" and "New Y" text boxes, with the "Refold" and "Reset" buttons activated underneath (see **Change Dimensions (Refold)**).

The Oversampling option also becomes active, allowing for mathematical refining of the image (virtual resolution increase, see **Oversample**

The lateral resolution of the generated visualisation plots can be improved virtually using oversampling. Select the plot for oversampling by the "Visualised" drop-down list (see **Change Plot**), enter the oversampling factor in the textbox and tick the "Oversample by" textbox. A new popup window appears that contains 2 plots: the original (top) plot as shown in the GUI, and its oversampled version (bottom). The oversampling factor determines the new number of points (pixels) in the oversampled plot, multiplying BOTH the X and Y points with this number and calculating the values of each point based on the original data points. Example: the demo_image.txt dataset contains a 25*5 image. Using an oversampling factor of 2, the new image will consists of 50*10 points (pixels).

> ⓘ **Warning/Information:** The oversampling range is limited to 2 – 10. Numbers outside this range will be reset to the default "2".

ⓘ **Warning/Information:** The oversampling only changes the appearance of the currently active plot, it does NOT change any data matrix, and it cannot be used on more than one plot at a time (i.e. only the currently active visualised plot can be oversampled).

ⓘ **Warning/Information:** Oversampling can generate artefacts, especially close to the right and bottom edges of the images, and in particular when intensities in the original plot are high in those regions (as there is no data beyond these points to be used for value estimations). Be aware of this limitation of the algorithm.

ⓘ **Warning/Information:** New popup windows opened by oversampling do NOT overwrite already existing windows, thus several oversampling options can be compared side by side.

ⓘ **Warning/Information:** To perform a new oversampling, click in the oversampling textbox and either change the oversampling value (if a new oversampling factor is to be tested on the same plot), or simply hit "Enter"/"Return" on the keyboard (to keep the same oversampling factor when generating a new plot, after e.g. changing the "Visualised" drop-down list to a new plot, or changing the "Colours" drop-down list to a new colour scheme), as long as the "Oversample by" checkbox is ticked.

ⓘ **Warning/Information:** Naturally, no Markers (Purest or Class) are included in the oversampling plots, as their position cannot be determined in the virtually enhanced new plots.

ⓘ **Warning/Information:** Unticking the "Oversample by" checkbox does NOT close already opened popup windows. They need to be closed manually.

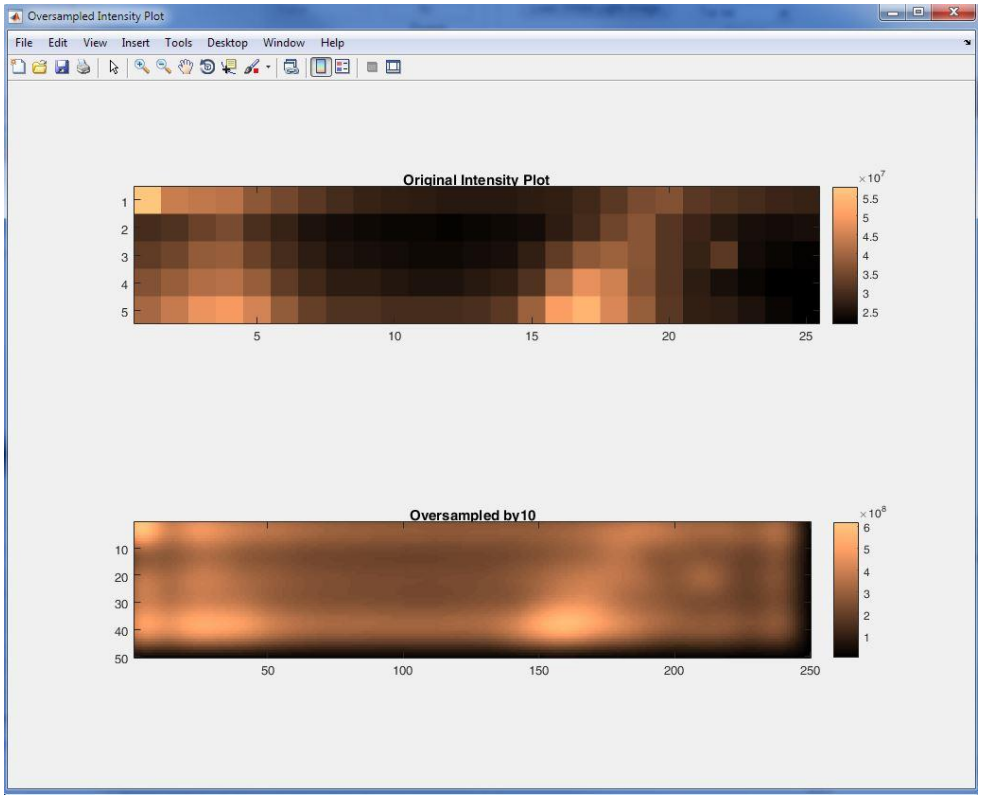

Figure 30. *The Oversampling popup window, using the total intensity plot generated directly after the "demo_image.txt" file is loaded, using an oversampling factor of 10. Top: no oversampling, Bottom: oversampled by a factor of 10. Note the number of pixels in the image (and the resulting smoothness) and the artefacts generated at the bottom and at the right of the oversampled map (drop of intensity). Note also that the window has its own set of menus and icons (allowing for e.g. saving the plots).*

Mark Purest on Map).

The "Mark Classes" radio buttons are activated, together with the "Get From Selected" button and the "Marked" listbox, allowing for manual labelling of selected spectra (assigning them to different classes, see **Mark Classes**).

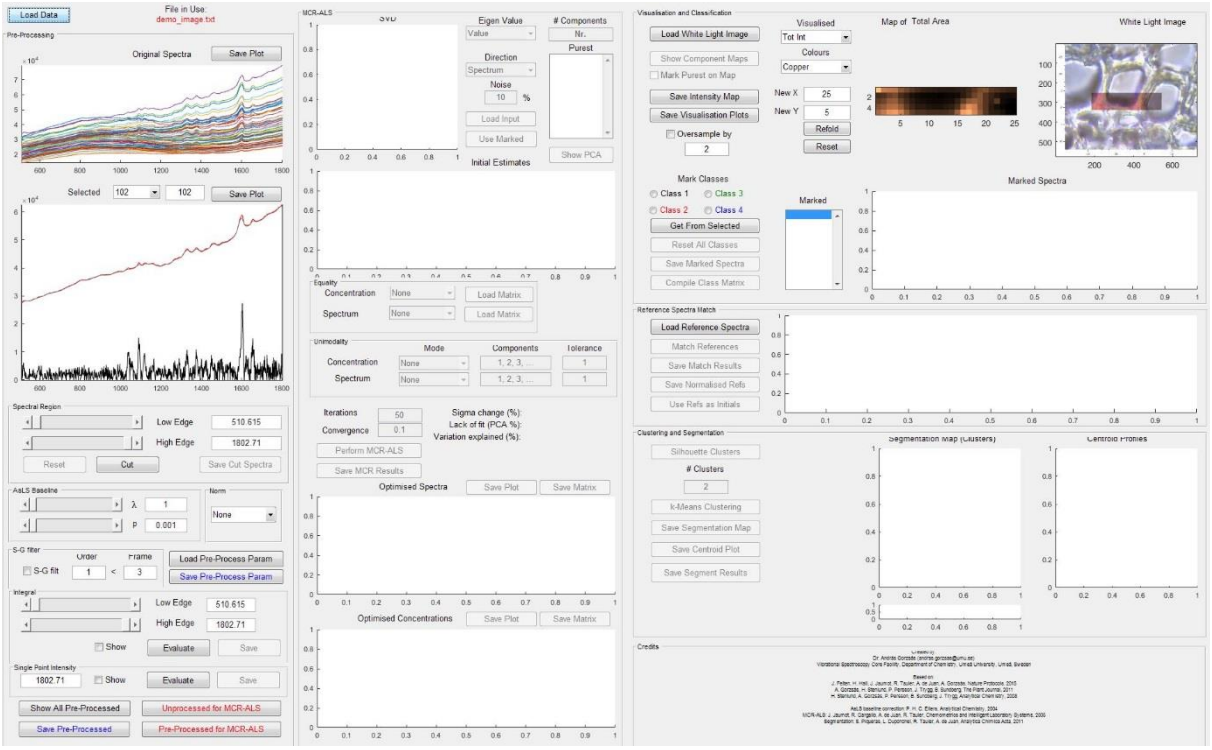

Figure 4: An image dataset is loaded from a .txt file. *This dataset contains more 125 spectra, of which only 100 randomly selected spectra are shown in the "Original Spectra" plot, in rainbow colours. The example uses the "demo_image.txt" file (with "demo_image.jpg" automatically loaded if it resides in the same folder as the "demo_image.txt" file).*

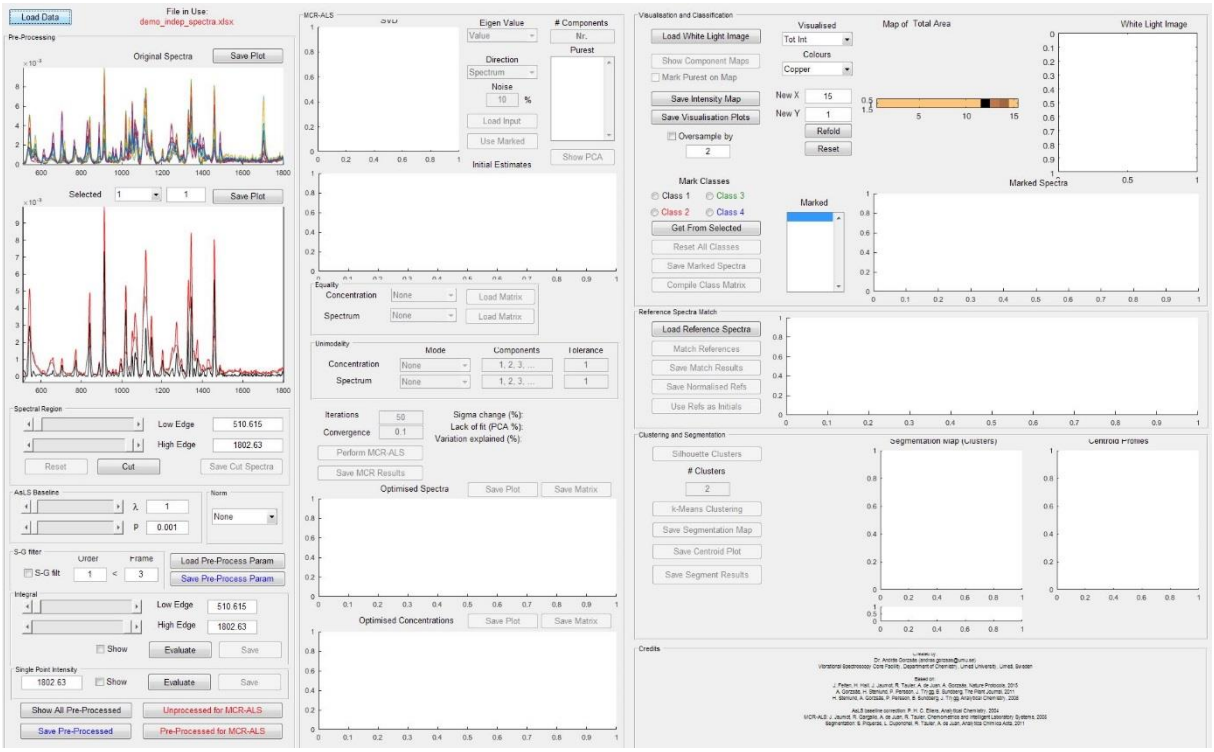

***Figure 5: A spectrum data set is loaded**. This data set contains 15 independent spectra (not in a series). Since the total number is less than 100, all of them are loaded at once, but no corresponding white light image is loaded (upper right). The example uses the "demo_indep_spectra.xlsx" file.*

3.2 Selecting a spectrum for display

Once the data is loaded (see **3.1 Loading the** Data) the "Selected Spectrum" plot is populated with a randomly selected spectrum from the dataset. A drop-down menu and a text box show the number of the spectrum being displayed in the plot, and the corresponding "Save Plot" button is activated. The selected spectrum is shown in red, the baseline (see **3.4. Asymmetrical Least Squares Baseline** Correction) in grey and the resulting baseline corrected spectrum in black (**Figure 6**).

> ⓘ **Warning/Information: If the dataset contains more than 100 spectra, only the 100 randomly selected ones that are shown in the Original Spectra window are listed in the drop-down menu** (and the same spectrum can be randomly selected more than once, meaning that the same spectrum can feature multiple times in the drop-down list).

ANY spectrum can be manually selected by typing the spectrum number in the "Selected" textbox, even it does not feature in the drop-down menu (in case there were more than a 100 spectra in the dataset). This spectrum is then added to the drop-down list and can be assigned to any class by the "Get From Selected" button in the Visualisation section of the GUI (see **Get From Selected**), and from there it can be used as an initial estimate by the "Use Marked" button in the MCR-ALS section of the GUI (see **Use Marked** Spectra

Initial Estimates can be manually specified from the current dataset to complement / replace the automatically determined Initial Estimates. These manually specified Initial Estimates need to be first marked in the dataset either as classless (class 0, gray) or belonging to any of the 4 classes possible to specify (class 1 – 4, black, red, green and blue, see **Mark Classes**).

This method ONLY works in the Spectrum direction!

If some of the Initial Estimates need to be removed after loading (either from among the automatically determined ones or from among the ones imported by the "Use Marked" button), double-click their descriptor (spectrum number) in the "Purest" listbox.

ⓘ **Warning/Information:** Marked spectra are added to the "Purest" listbox and to the "Initial Estimates" plot in the order they appear in the "Marked" textbox (**Figure 21**).

ⓘ **Warning/Information:** If the "Mark Purest on Map" checkbox is ticked, the white diamond markers can completely cover the coloured marker squares. Untick the checkbox of those markers are to be visible.

ⓘ **Warning/Information:** Double-clicking any element of the "Purest" listbox will remove it and adjust the number of components accordingly. This also removes its Marker from the Visualised plot, if it had any. This only works until only 1 element remains in the "Purest" listbox. The last remaining element cannot be removed (i.e. the "Purest" listbox cannot become empty). If this element needs to be removed, a new one must first be added.

ⓘ **Warning/Information:** If the legend obscures the "Initial Estimates" plot, it can be moved to a different position within the plot, by left-clicking and dragging.

ⓘ **Warning/Information: Equality Constraint vs Load Input**: If a certain compound is known / suspected to be a pure component in the dataset, it can be supplied either as an initial estimate or as an equality constraint for the spectral direction. There is a difference in the mechanism how this can practically affect the results. An initial estimate does NOT pose ANY constraints at all during the modelling, it only makes sure that the iteration starts with this value. From that on, the model can take any direction and the spectral profile can change freely, without restrictions (apart from non-negativity). In other words, this way of supplying a spectral profile only ensures that the optimization starts close to the expected final result (i.e. close to the suspected absolute minimum, avoiding falling into a potential local minima, or lowering the number of iterations the model requires to reach convergence). On the other hand, if a spectral profile is supplied as a constraint, it poses a more rigid limitation for how the model can evolve. With an "exact" constraint applied, for instance, the component spectral profile is NOT allowed to change at all. This can be useful when a known compound with a known (and exact) spectral profile needs to be tracked in the dataset. The same is true in the concentration direction, with concentration profiles loaded as initial estimates vs supplied as equality constraint.

☒ **Known bug:** The Markers disappear from the Visualisation plot if a new Integral (3.7. **Integration**) or Single Point Intensity evaluation is performed (**3.8. Single Point Intensity Evaluation**).

☑ **Workaround:** Refresh the plot by either of the following methods: a) untick and tick the "Mark Purest on Map" checkbox; b) Activate the "Visualised" drop-down list (even if clicked on the very same value, it triggers the update of the plot, it does not have to be changed to a new value / new plot selected) (see **Change Plot**); c) Activate the "Colours" drop-down list (even if clicked on the very same value, it triggers the update of the plot, it does not have to be changed to a new value / new plot selected) (see **Change Colour**); d) "Refold" or "Reset" the plot (see **Change Dimensions (Refold)**); d) Change the number of Components to a new value than change back to the desired one (see **Number of Components**).

☒ **Known bug:** If there have been Initial Estimates loaded using the "Load Input" button (see **Load Input**) or by adding them from the loaded References (see **Use Reference Spectra**) before the "Used Marked" button is clicked, those externally loaded Initial Estimates will be listed before the Marked ones, which will create an artefact Marker and an extra line appear in the Visualisation plot when the "Mark Purest on Map" checkbox is checked. This is because the externally supplied Initial Estimates (the loaded inputs or references) are not part of this dataset and thus cannot be plotted.

☑ **Workaround:** Use the "Load Input" and/or "Use Refs as Initials" buttons AFTER using the "Use Marked" button. If there are already externally loaded Initial Estimates, remove them by double-clicking on their names in the "Purest" listbox, uncheck the "Mark Purest on Map" checkbox, click on the "Load Input" button to re-load the Initial Estimates and check the "Mark Purest on Map" box. This will place the externally loaded initial estimates at the end of the list and refresh the Markers to correctly show the position of all Initial Estimates / purest spectra.

☒ **Known bug:** If the "Used Marked" button is clicked repeatedly, spectra will be simply added to the already existing "Purest" list, without checking if they are already listed there or not. In other words, they do not overwrite or replace existing ones, simply add all spectra from the "Marked" plot to the end of the "Purest" list. Unfortunately, this can result in the same spectrum / spectra being added repeatedly.

☑ **Workaround:** Remove multiple instances of the same spectrum by double-clicking on the number in the "Purest" listbox until each spectrum (each number) only features once in the list.

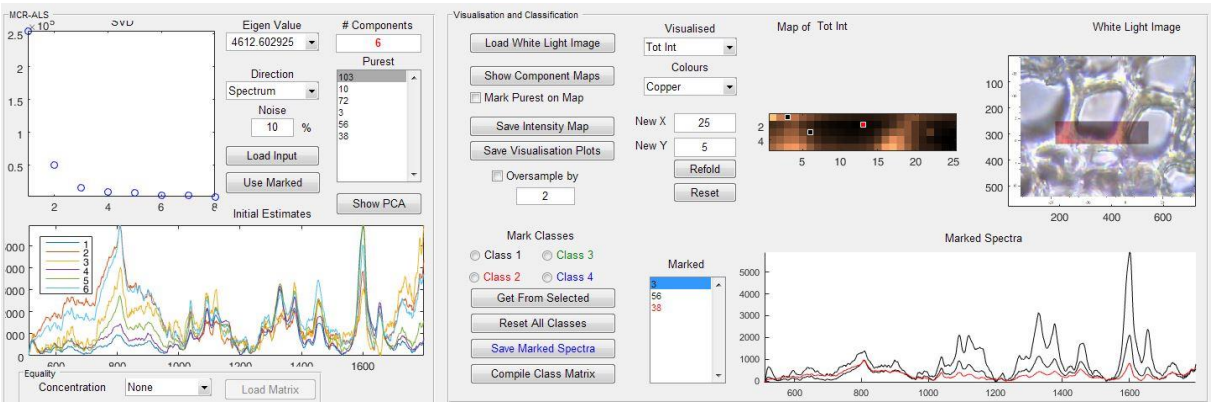

Figure 21. *Use Marked Spectra for Initial Estimates. The example uses the "demo_image.txt" file exactly as in* ***Figure 19****, with manually marking spectra 3 and 56 as Class 1 (black) and spectrum 38 as Class 2 (red), as seen in the "Marked" listbox. The "Marked Spectra" plot shows the spectra in their respective colours as well. Note that the Marked Spectra are added after the automatically determined ones (spectra 103, 10, 72), in the order they appear in the "Marked" listbox. They do not retain their class colours as Components, however (so in the Initial Estimates plot, spectra 3, 56 and 38 appear as violet, green and cyan, respectively, i.e. Components 4-6).*

Use).

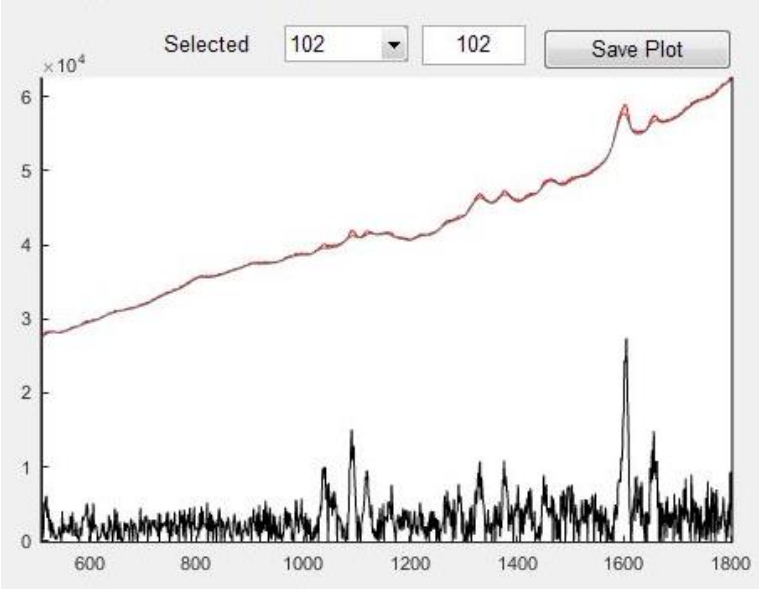

Figure 6: The Selected Spectrum plot*. Red is the original spectrum (nr. 102 in the series, as shown in the drop-down list and the text box), grey is the baseline (clearly overfitted in this case, see* ***Figure 8****), and black is the resulting, baseline-corrected spectrum. The example uses the "demo_image.txt" file.*

3.3. Spectral Region Adjustment (Cutting)

It can be beneficial to trim the spectral region, to eliminate noise or artefacts, or to facilitate baseline correction (see **3.4. Asymmetrical Least Squares Baseline** Correction).

This can be achieved by cutting the lower, higher or both ends of the region, using the controls provided in the "Spectral Region" box of the Pre-Processing section of the GUI.

> ⓘ **Warning/Information:** Only the low and high ends can be trimmed, it is not possible to cut an arbitrary region from within the spectra.

> ⓘ **Warning/Information:** Cutting can only be done in the X variable space (i.e. in the spectral dimension), NOT in the observation space (no image pixel or individual spectrum can be excluded from the dataset)

The region to be cut is marked in semi-transparent yellow colour and can be adjusted by the sliders or directly by typing the desired value in the corresponding text box (**Figure 7**) and hitting "Enter" (PC) or "Return" (Mac) key. **The cutting is only performed after clicking the "Cut" button.** The resulting spectrum will only contain data the falls within the range specified, and the low and high edge textbox will display the new minimum and maximum values (**Figure 7**). The "Selected Spectrum" plot will automatically update, along with the "Marked Spectra" plot in the Visualisation section of the GUI, if there are any marked spectra already (see **Mark Classes**).

> ⓘ **Warning/Information:** The sliders are proportional controls, not absolute. Thus, their precision depends on the range as well. For exact values, use the textboxes.

Once the dataset is cut, it can be saved by the "Save Cut Spectra" button which becomes active. A dialog box opens, allowing for specifying location and filename for the saved data (by default suggesting the original filename with the tag "cut" and the spectral region appended at the end of the filename).

> ⓘ **Warning/Information:** The data is saved in .mat format, irrespective of the original input format (see **Cut and Pre-Processed Spectra**).

> ⓘ **Warning/Information: When a new dataset is loaded**, which contains a broader spectral range than the previously cut one, the cutting limits are kept in the GUI. Thus, the new dataset can be quickly trimmed to the same zone as the previous dataset, by clicking the "Cut" button, without having to set the low and high edges again. However, if the new dataset contains a spectral region with any or both ends WITHIN the cut limits, the cut limits will be overwritten with the new minimum/maximum/both spectral region limits. That is, the spectral region can only be cut, NOT expanded (spectra cannot be padded to expand the spectral region on either ends).

The cutting can be undone by clicking the "Reset" button, which also de-activates the "Save Cut Spectra" button (and itself) after restoring the original spectral region.

> ⓘ **Warning/Information:** There is no stepwise undo. After multiple cuts, the dataset is restored to its original spectral region, not to the result of the previous cut.

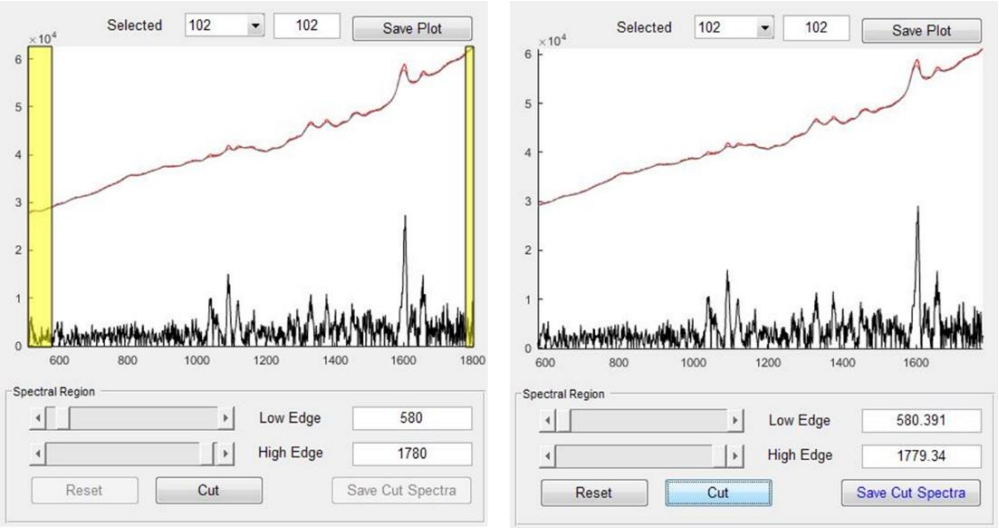

***Figure 7: Cutting the spectral region: before (left) and after (right)**. The region was set to 580 – 1780 cm^{-1} (left). Thus the new dataset contains only points within this region, the first recorded data being 580.391 cm^{-1}, and the last 1779.34 cm^{-1} (right) in the present example, which uses the "demo_image.txt" file.*

3.4. Asymmetrical Least Squares Baseline Correction

Baseline correction is performed via **asymmetrical least squares fitting**, a method developed by Eilers (*Analytical Chemistry*, 2004). It uses only two parameters to perform a fit to each spectrum in the dataset: **lambda** and **p**. **This baseline correction method does not operate with a pre-defined baseline shape, but fits a baseline to each spectrum individually.** The lambda value determines how close the baseline needs to follow (fit) the original data. At minimum value, it follows very closely (**Figure 8 left**), often resulting in overfitting. At maximum value, it produces a linear baseline, connecting the first and last data point in the spectrum, which can result in underfitted baselines (**Figure 8 middle**). The p value determines the weight of negative contributions to the sum of fit. In practical terms it means that the minimum p value does not allow negative data during baseline fit. In other words, the baseline cannot "cut through" the spectrum, so all points in the spectrum must be positive, and no point can be below the baseline (**Figure 8 middle**). This should be the default setting for normal spectral data (no derivatives). Conversely, maximum p values allow maximum negative contributions (**Figure 8 right**).

The baseline can be perfectly reproduced for any spectrum using only these two parameter values.

Adjust the baseline primarily by varying the lambda value (either via the slider or by directly typing the desired number in the corresponding textbox) until optimum is reached, ideally as close to linear baselines as possible for vibrational spectroscopic (in particular FTIR spectroscopic) data. Results can be seen in black in the "Selected Spectrum" plot (see **3.2 Selecting a spectrum for** display). If there are marked spectra, the "Marked Spectra" plot of the Visualisation Section updates as well, showing the effects of the baseline correction (**Mark Classes**).

ⓘ **Warning/Information:** The sliders are proportional controls, not absolute. For exact values, use the textboxes. It is especially **recommended to use the textbox for lambda values**, since it is on a broad logarithmic scale from 1 to 10^9. Thus, moving the slider can a) result in big jumps in lambda values) and b) result in long, hard to report, hard to remember and hard to reproduce numbers. On the other hand, providing an easier number in the textbox that is close enough in value will produce a virtually identical baseline fit. For example: moving the slider can result in a lambda value of 486425125.9253. In this case, manually providing 500000000 in the textbox can be the preferred alternative, as the baseline will be virtually identical but the number "$5x10^8$" is more practical.

It can happen that no optimum can be found because of the presence of negative bands (due to e.g. automatic background subtractions resulting in negative CO_2 bands in FTIR spectra) or because of other artefacts (e.g. in PCA recombined noise filtered datasets, as in the high wavenumber end in the "demo_image.txt" example file). In this case, try changing the p value or trim the spectral region to eliminate the problematic zone (in case of the "demo_image.txt" file, it is sufficient to set the high edge of the spectral region to "1800" instead of the original "1802" (see **3.3. Spectral Region Adjustment (**Cutting)) to eliminate the artefact and produce a good quality baseline).

☒ **Known bug:** In some cases, certain lambda values do not allow setting the p value to the maximum 1, resulting in an impossible baseline calculation followed by an error message in the MATLAB command prompt (division by 0) and the "Selected Spectrum" plot becomes blank.

☑ **Workaround:** Set the p value to 0.99 or 0.9999, instead of 1.

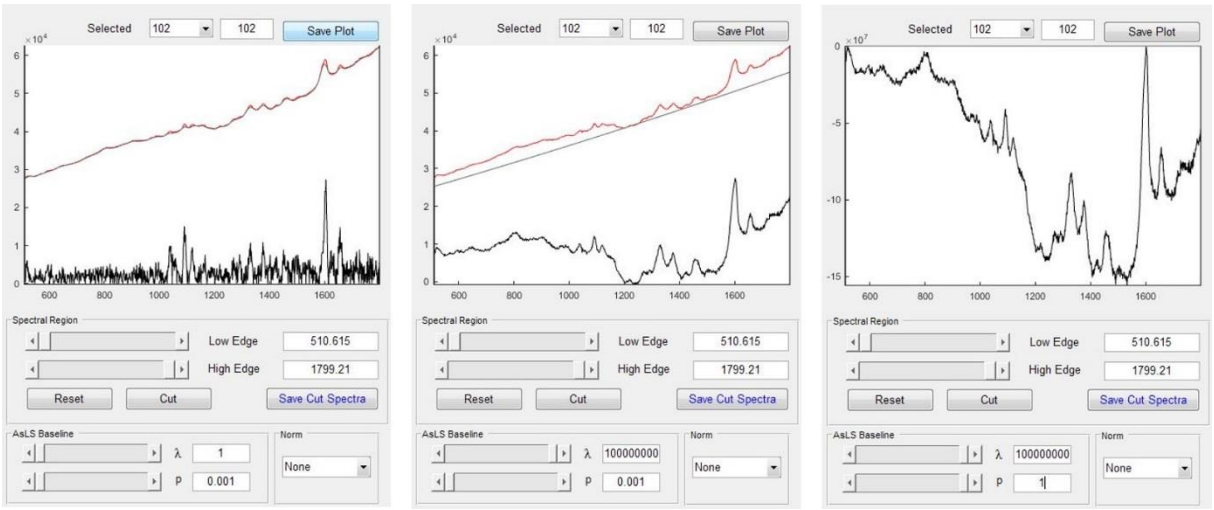

Figure 8: Asymmetrical Least Squares baseline fitting. *Left: overfitted baseline (lambda = 1 (minimum), p = 0.001 (minimum)), middle: underfitted baseline (lambda=10^9 (maximum), p=0.001 (minimum)), right: demonstrating the effect of high p values (resulting in a baseline that is higher or cuts through the spectrum, producing negative spectral intensity values as a result, and a display scaling that cannot show the original spectrum in red. Lambda=10^9 (maximum), p=1 (maximum)). The example uses the "demo_image.txt" file after cutting the high edge to "1800".*

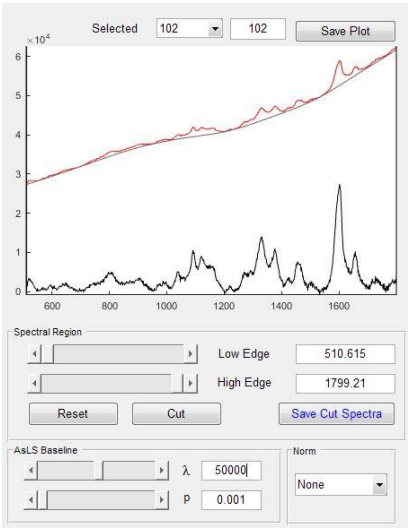

Figure 9: Optimum baseline *(lambda=50 000, p=0.001 (minimum)). The example uses the "demo_image.txt" file after cutting the high edge to "1800".*

3.5. Normalisation

Normalisation is an optional spectrum standardisation step. This is not required for MCR-ALS, which has a built-in spectral equal length constraint in the GUI. However, it can still be beneficial if the spectra needs to be standardised and exported for further processing by other methods, or simply for visualisation or educational purposes.

To perform normalisation, select the type of normalisation from the "Norm" drop-down menu. The default setting is "None".

ⓘ **Warning/Information:** Normalisation is only performed when the pre-processing parameters need to be updated (such as when the "Save Pre-Processed" or "Pre-Processed for MCR-ALS" buttons are clicked in the GUI, or intensity maps need to be generated (see **3.7.** Integration and **3.8. Single Point Intensity** Evaluation). However, if there are marked spectra (see **Mark Classes**), the "Marked Spectra" plot in the Visualisation section of the GUI immediately updates to reflect the changes!

ⓘ **Warning/Information: The Total Area map in the Visualisation section ALWAYS shows the intensity without normalisation, i.e. the raw input!** If, for some reason, a normalised total area map should is required (for instance to detect erroneous spectra in the dataset that contains no information, etc.) it can be generated by performing integration over the entire spectral region (see **3.7.** Integration). In this case, the Visualisation section will label this as an Integral plot, which will be added to the list of Visualised plots in the drop-down list (see **Change Plot**), without overwriting the non-normalised Total Area map.

ⓘ **Warning/Information:** Bear in mind that other pre-processing parameters (mainly spectral region and baseline correction but also Savitzky-Golay filtering, see 3.3. Spectral **Region Adjustment (Cutting)**, **3.4. Asymmetrical Least Squares Baseline** Correction and **Figure *10: Illustrating the effects of normalisation***. *a: No Normalisation; b: Total Area Normalisation; c: Total Min-Max Normalisation; d: Region Max Normalisation, using the 1550-1640 cm^{-1} region (aromatic -C=C- band); e: Region Area Normalisation, using the 1550-1640 cm^{-1} region (aromatic -C=C- band); f: Region Min-Max Normalisation, using the 1550-1640 cm^{-1} region (aromatic -C=C- band); g: Point Max Normalisation, using the intensity at 1600 cm^{-1} (aromatic -C=C- band); h: Offset Normalisation, using the intensity at 1600 cm^{-1} (aromatic -C=C- band); i: Offset Normalisation, using the intensity at 1200 cm^{-1} (assumed baseline point).*

*The example uses the "demo_image.txt" file, after optimum baseline correction in **Figure 9** (spectral region high edge cut to "1800", asymmetrical least squares baseline lambda=50 000 and p=0.001 (minimum)).*

3.6. Smoothing) naturally will have a great impact on the results of the normalisation.

The types of normalisations that can be performed are as follows:

3.5.1. Total Area

The area under all bands in the entire spectral region is set to a constant value. In practical terms, each intensity (i.e. the value at each wavenumber) is divided by the total sum of all intensities, i.e. the sum of all intensities is constant (**Figure 10b**).

3.5.2. Total Min-Max

The lowest intensity in the entire spectral range is set to 0, and the highest is set to 100. The spectrum is scaled accordingly and each spectrum is treated this way, independent of one another. That is, the minimum and maximum intensities do not have to be at the same wavenumbers in every spectrum, but every spectrum will have an intensity range 0 – 100 (**Figure 10c**).

3.5.3. Region Area

Similar to the Total Area normalisation, but instead of using the entire spectral range, each spectrum is normalised for the area in a user specified range only. The range is specified by the "Integral" controls in the GUI (see **3.7.** Integration). This option can be used to normalise for the intensity of a certain band intensity, using it as an internal standard (provided that its intensity can be accurately determined) (**Figure 10e**).

3.5.4. Region Min-Max

Similar to the Total Min-Max normalisation, but instead of using the entire spectral range, each spectrum is normalised using a user defined range only. The range is specified by the "Integral" controls in the GUI (see **3.7.** Integration). The minimum intensity in this region is set 0, and the maximum to 100 as in the Total Min-Max normalisation option. This option is similar to the Region Area normalisation above, but scales along the intensity axis, and not for the band area (**Figure 10f**).

3.5.5 Region Max

Similar to the Region Min-Max normalisation, but it only sets the maximum intensity in the region to 100 and sets the minimum intensity to 0, wherever that is in the spectrum (i.e. it does not have to be in the region), and scales the spectra accordingly. The region in which the maximum intensity is set to 100 is specified by the "Integral" controls in the GUI (see **3.7.** Integration).This normalisation option can be preferred over Region Area and Region Min-Max normalisation if the spectral intensity does not decrease to baseline (0) in the desired region due to e.g. overlapping bands (**Figure 10d**).

3.5.6. Point Max

Similar to the Region Min-Max normalisation, but it sets the highest intensity to 100 at a defined point, and not in a range. The point is defined by the "Single Point Intensity" input in the GUI (see **3.8. Single Point Intensity** Evaluation). Since it uses only one point to set the maximum, not a range, it is very sensitive to band shifts. The minimum intensity in the entire spectral range is set 0 (**Figure 10g**).

3.5.7. Offset

Similar to Point Max normalisation, in a sense that it uses a point defined by the Single Point Intensity input in the GUI to perform the normalisation (see **3.8. Single Point Intensity** Evaluation), but instead of setting the intensity to 100 at this point, it sets the intensity to 0 (i.e. baseline). Most importantly**, it does NOT scale the spectrum, only shifts (offsets) all intensities** (i.e. no multiplication, only addition). Consequently, it is best used on a baseline point (**Figures 10h and i**).

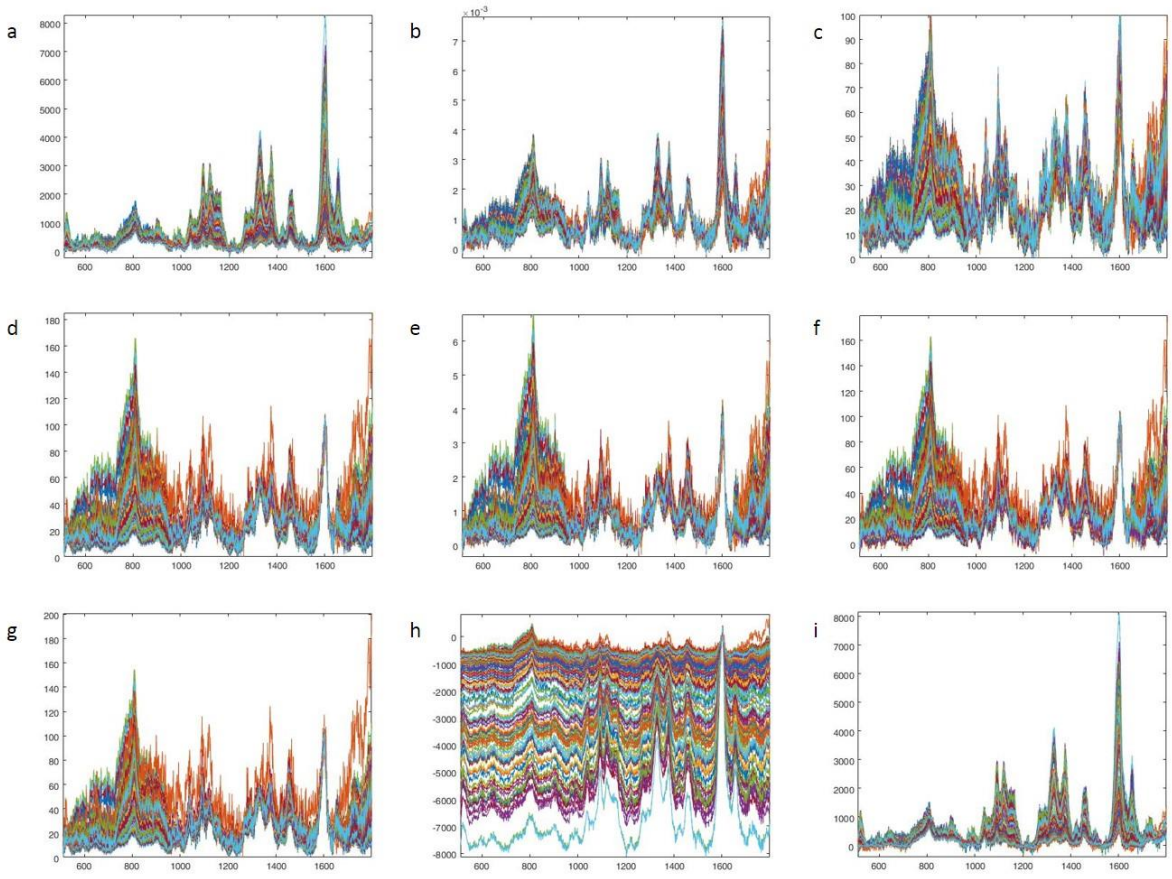

Figure 10: Illustrating the effects of normalisation*. a: No Normalisation; b: Total Area Normalisation; c: Total Min-Max Normalisation; d: Region Max Normalisation, using the 1550-1640 cm^{-1} region (aromatic -*

C=C- band); e: Region Area Normalisation, using the 1550-1640 cm^{-1} region (aromatic -C=C- band); f: Region Min-Max Normalisation, using the 1550-1640 cm^{-1} region (aromatic -C=C- band); g: Point Max Normalisation, using the intensity at 1600 cm^{-1} (aromatic -C=C- band); h: Offset Normalisation, using the intensity at 1600 cm^{-1} (aromatic -C=C- band); i: Offset Normalisation, using the intensity at 1200 cm^{-1} (assumed baseline point).

*The example uses the "demo_image.txt" file, after optimum baseline correction in **Figure 9** (spectral region high edge cut to "1800", asymmetrical least squares baseline lambda=50 000 and p=0.001 (minimum)).*

3.6. Smoothing

Savitzky-Golay filtering is an optional step that can be applied to smooth the spectra, potentially improving signal-to-noise ratio. The GUI utilises MATLAB's built-in "sgolayfilt" function. The smoothing uses two parameters: "Order", referring to the polynomial order of the fit, and "Frame". Consult MATLAB's documentation for details.

> ⓘ **Warning/Information:** The Frame can only be an odd number, and must ALWAYS be higher than the Order, which must be a positive integer (even or odd). The GUI will revert to default values if wrong input is attempted.

The smoothing is only performed if the "S-G filt" checkbox is ticked. The "Selected Spectrum" plot updates to show the smoothed spectrum in blue. If there are marked spectra, the "Marked Spectrum" plot in the Visualisation section of the GUI also updates automatically.

Generally, the larger the difference between Order and Frame, the more intensive the smoothing gets. Experiment with various settings to find the ideal parameters, but avoid overfitting (**Figure 11**, right). Often low Order and Frame rates (1 and 5, respectively, for instance) are sufficient.

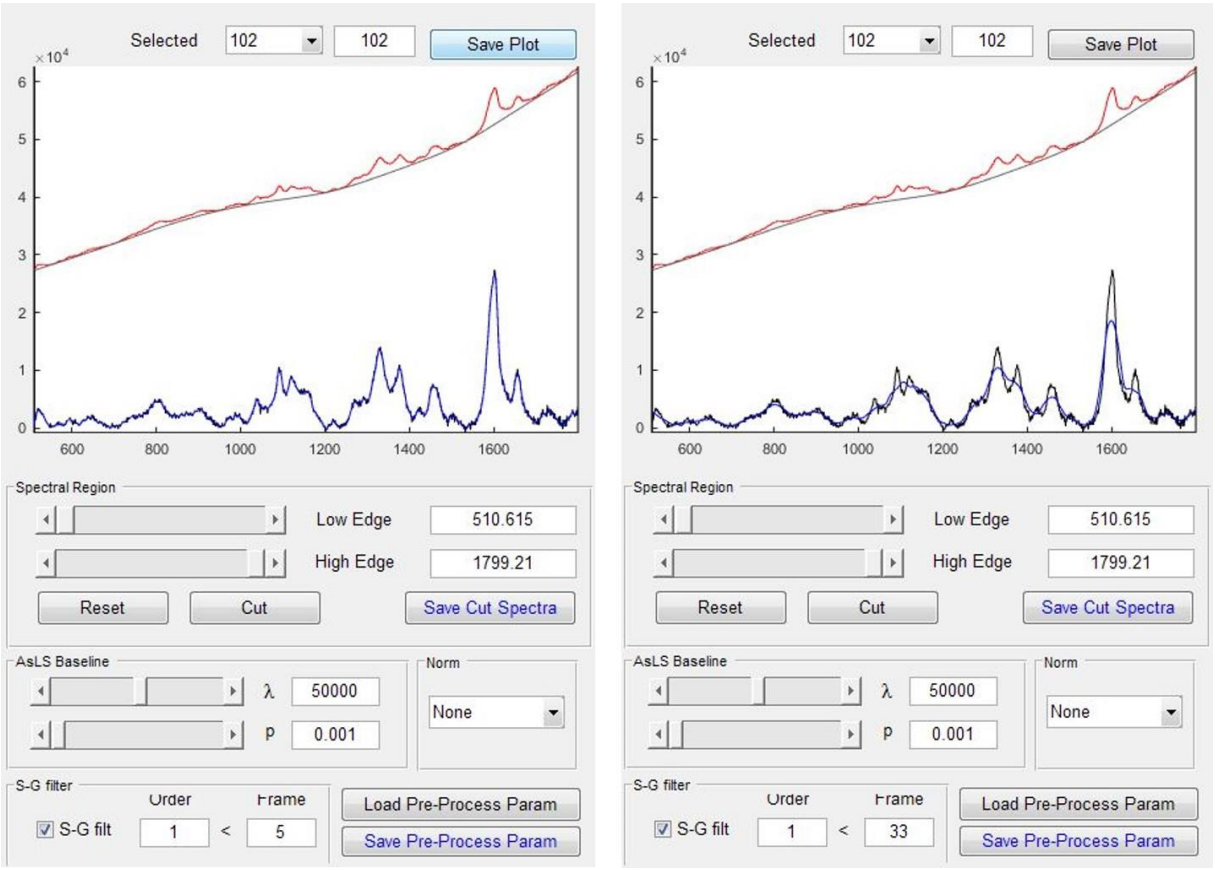

Figure 11: The effects of smoothing*. Left: correct (mild) smoothing resulting in reduced signal to noise ratio (Order=1, Frame=5). Right: oversmoothing, resulting in loss of information and artefact generation (Order=1, Frame=33). The example uses the "demo_image.txt" file, after optimum baseline correction in* ***Figure 9*** *(spectral region high edge cut to "1800", asymmetrical least squares baseline lambda=50 000 and p=0.001 (minimum)).*

3.7. Integration

The area under the spectral bands can be determined using the controls in the "Integral" box of the Pre-Processing section of the GUI.

> ⓘ **Warning/Information: This is a crude integration, with no deconvolution**, only the sum of intensities between the low and the high edge of the specified region (see **Figure 12**).

To perform the integration, simply move the sliders or manually type in the low and high edge of the integral region (preferred) and click "Evaluate".

> ⓘ **Warning/Information:** For large datasets, integration can take longer on slow computers if there has been no pre-processing done yet, or pre-processing parameters have been changed since the last time integrals were evaluated (i.e. new baseline, smoothing, normalisation or cut has been

performed). The reason for this is that all spectra must be pre-processed before the integral intensities are evaluated. A status bar shows the progress. Please wait until it finishes.

Naturally, baseline correction, smoothing and normalisation all affect the Integral values (**Figures 12 and 13**). Thus, if any of them are changed after the integration has been performed, the "Evaluate" button needs to be clicked again to refresh the Integral values. In this case, the entire dataset will be pre-processed again (see the **Warning/Information** above).

ⓘ **Warning/Information:** There is NO information showing whether the integral currently displayed in the Visualisation plot has been calculated with the current pre-processing parameters or not! In other words, the visualisation plot will show the last evaluated integral, which may not be the current one if pre-processing values have been changed but the "Evaluate" button has NOT yet been clicked.

The integral can only be shown in the "Selected Spectrum" plot (as a light blue area) by ticking the corresponding "Show" checkbox (**Figure 12**). However, the Visualisation plots are automatically updated whenever the "Evaluate" button is clicked, even if the "Show" checkbox is unticked. The "Visualised" drop-down list is also updated to include the integral plot, using a label starting with "Int @", followed by the integral region and finished by an abbreviation to denote the kind of normalisation performed (**Figure 13**). "NN" stands for "No Normalisation", "TA" for "Total Area", "TMM" for "Total Min-Max", "RA" for "Region Area", "RMM" for "Region Min-Max", "RM" for "Region Max", "PM" for "Point Max" and "OFN" for "Offset Normalisation" (see **3.5.** Normalisation). The colour map of the plot can be changed by the "Colours" drop-down list in the Visualisation section of the GUI (**Change Colour**).

ⓘ **Warning/Information:** There can only be one integral evaluated and visualised at the time. Any new integral will therefore overwrite the previous ones. If the integral results are to be kept, they need to be saved before a new integral is evaluated. However, integrals and single point intensities do NOT overwrite each other, i.e. there can be one integral and one single point intensity visualised. In addition, the non-normalised total intensity plot is always kept, irrespective of any integration.

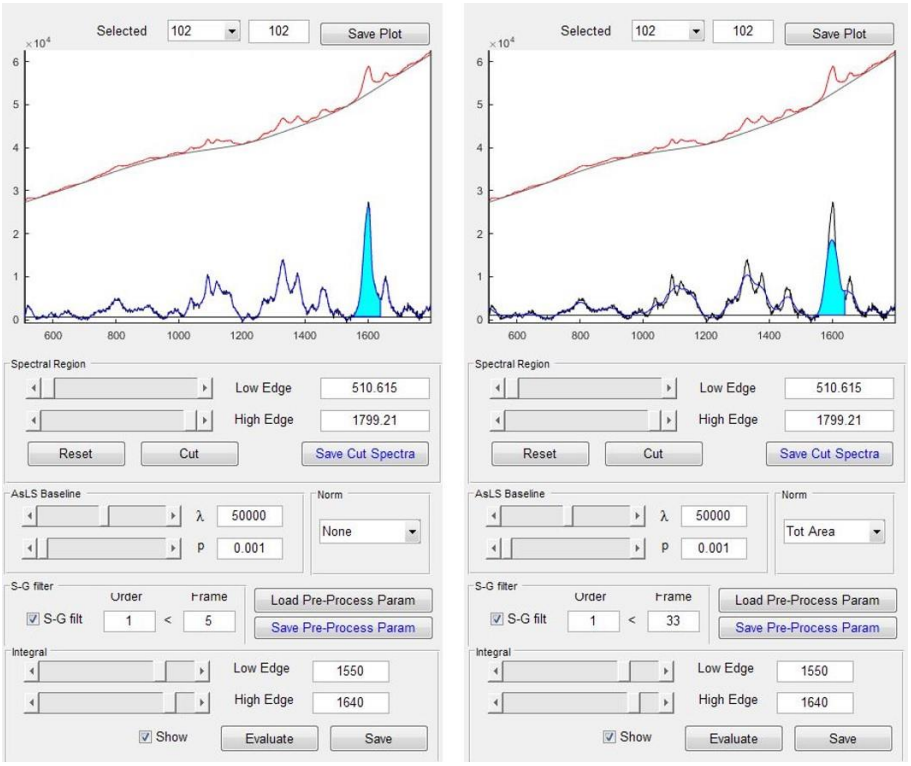

Figure 12: Integration. *Left: showing how crudely the integral values are calculated (see the abrupt end of the area at 1640 cm^{-1}). Right: smoothing, baseline correction and normalisation affects the integral value. The example uses the "demo_image.txt" file.*

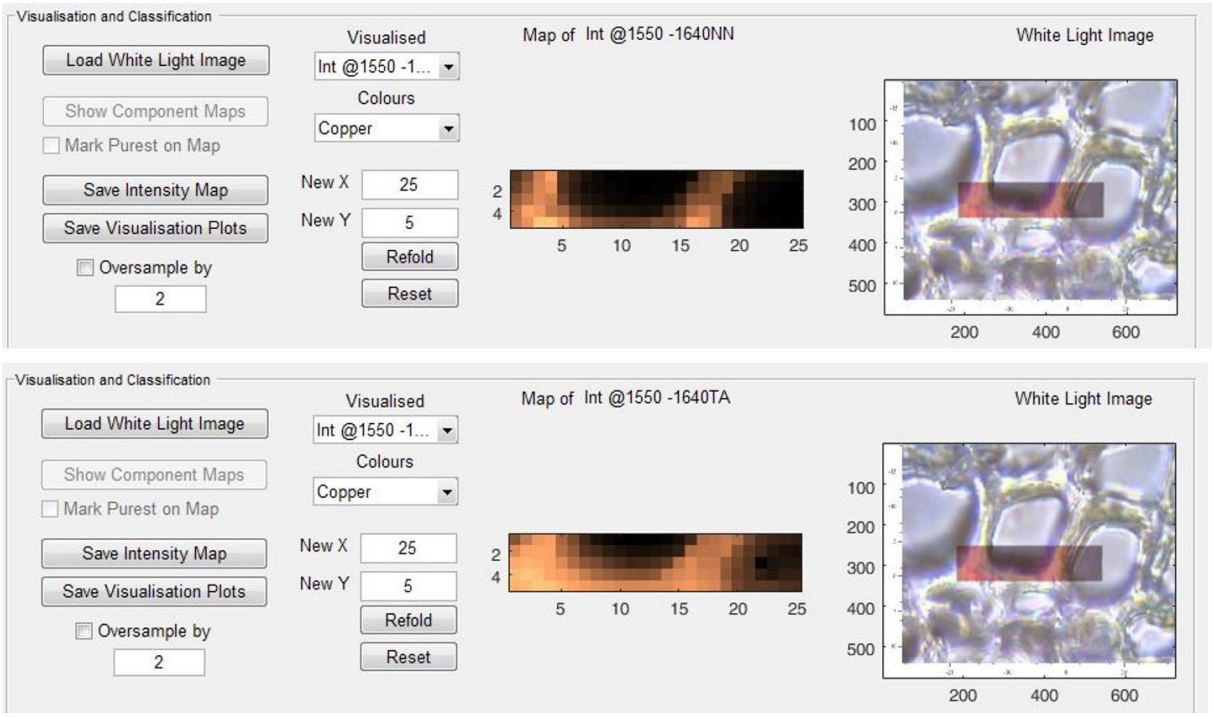

Figure 13: Visualising the integral result. *The "Visualised" drop-down list is updated, together with the plot and its title, to show the evaluated integrals. The example uses the "demo_image.txt" file with the same settings and integrals as in* ***Figure 12*** *above, the top image corresponding to* ***Figure 12 left****, and the bottom one corresponding to* ***Figure 12 right****.*

If the entire spectral range is specified for the Integral, it generates a total area map. However, this as an Integral plot, which will be added to the list of Visualised plots in the drop-down list, and will NOT overwrite the original, non-normalised Total Area map (see Warning/Information above).

If the spectral region is cut after the integral region was set, there are two possibilities. 1) The cut region still contains the entire integral region. In this case, nothing happens to the integral. 2) The cut region does not contain the entire integral region. In this case, the integral region will be adjusted in the controls to reflect the change, but a new integral is NOT calculated until the "Evaluate" button is clicked.

> ⓘ **Warning/Information:** Following the resetting of the spectral range to its original values after it has been cut, the Integral controls do not change. This is not a bug, it is intentional. Keep this in mind if the total area intensity needs to be determined after reset! In that case, set the low and high edge to the minimum and maximum values, respectively.

After evaluating the integral, the corresponding "Save" button will become active, allowing for saving the result in .mat format (see **Integral and Single Point Intensity Matrices**).

> ⓘ **Warning/Information:** The "Save" button saves the last evaluated integrals. If there have been changes (in pre-processing or in integration) click the "Evaluate" button before attempting to save the results.

> ⓘ **Warning/Information:** The saved integral results **cannot** be loaded and visualised in GUI afterwards (note that the button has a black text, not blue). If the generated visualisation plot is to be kept as well, it needs to be saved separately using the controls in the Visualisation section of the GUI (see **Save Visualisation Plots**).

3.8. Single Point Intensity Evaluation

The intensity at any given wavenumber in the spectrum can be evaluated for entire dataset using the "Single Point Intensity" textbox of the Pre-Processing section of the GUI. This is a very simple evaluation, reading out the intensity value at that point in the spectrum (see **Figure 14**).

Simply enter the wavenumber at which the intensity should be determined in the textbox and click the corresponding "Evaluate" button.

> ⓘ **Warning/Information:** For large datasets, integration can take longer on slow computers if there has been no pre-processing done yet, or pre-processing parameters have been changed since the last time single point intensities were evaluated (i.e. new baseline, smoothing, normalisation or cut has been performed). The reason for this is that all spectra must be pre-processed before the integral intensities are evaluated. A status bar shows the progress. Please wait until it finishes.

Naturally, baseline correction, smoothing and normalisation all affect the Single Point Intensity values (see **Figures 14 and 15**). Thus, if any of them are changed after the single point intensity evaluation has been performed, the "Evaluate" button needs to be clicked again to refresh the values. In this case, the entire dataset will be pre-processed again (see the **Warning/Information** above).

> ⓘ **Warning/Information:** There is NO information showing whether the currently displayed single point intensities in the Visualisation plot have been calculated with the current pre-processing parameters or not! In other words, the visualisation plot will show the last evaluated values, which may not be the current one if the pre-processing parameters have since changed but the "Evaluate" button has NOT yet been clicked.

The position in the spectrum at which the single point intensity is evaluated can only be shown in the "Selected Spectrum" plot (as a vertical magenta line) by ticking the corresponding "Show" checkbox (**Figure 14**).

> ☒ **Known bug:** Sometimes, the magenta line can be hidden behind the light blue integral area if both are to be shown in the "Selected Spectrum" plot.
>
> ☑ **Workaround:** Untick and tick the "Show" checkbox for the Single Point Intensity. This will trigger a new render of the "Selected Spectrum" plot, with the magenta line correctly plotted on the top layer.

However, the Visualisation plots are automatically updated whenever the "Evaluate" button is clicked, even if the "Show" checkbox is unticked. The "Visualised" drop-down list is also updated to include the single intensity plot, using a label starting with "SPoint @", followed by the wavenumber at which the intensity was evaluated and finished by an abbreviation to denote the kind of normalisation performed (**Figure 15**). "NN" stands for "No Normalisation", "TA" for "Total Area", "TMM" for "Total Min-Max", "RA" for "Region Area", "RMM" for "Region Min-Max", "RM" for "Region Max", "PM" for "Point Max" and "OFN" for "Offset Normalisation" (see **3.5.** Normalisation). The colour map of the plot can be changed by the "Colours" drop-down list in the Visualisation section of the GUI (**Change Colour**).

> ⓘ **Warning/Information:** There can only be one single point intensity evaluated and visualised at the time. Any new single point intensity evaluation will therefore overwrite the previous one. If the evaluated results are to be kept, they need to be saved before a new evaluation is performed. However, integrals and single point intensities do NOT overwrite each other, i.e. there can be one integral and one single point intensity visualised.

If the spectral region is cut after the single point intensity position was set, there are two possibilities. 1) The cut region still contains this position. In this case, nothing happens to the single point intensity. 2) The cut region does not contain the position set for the single point intensity evaluation. In this case, the single point intensity position will be automatically set to the highest wavenumber of the new (cut) spectral region, but a new evaluation at this position is NOT performed until the "Evaluate" button is clicked.

ⓘ **Warning/Information:** Following the resetting of the spectral range to its original values after it has been cut, the single point intensity position do not change. This is not a bug, it is intentional. Keep this in mind when cutting and resetting.

After evaluating the single point intensity, the corresponding "Save" button will become active, allowing for saving the result in .mat format (see **Integral and Single Point Intensity Matrices**).

ⓘ **Warning/Information:** The "Save" button saves the last evaluated intensities. If there have been changes since then (in pre-processing or in the position of the single point intensity evaluation), click the "Evaluate" button before attempting to save the results.

ⓘ **Warning/Information:** The saved intensity results **cannot** be loaded and visualised in the GUI afterwards (note that the button has a black text, not blue). If the generated visualisation plot is to be kept as well, it needs to be saved separately using the controls in the Visualisation section of the GUI (see **Save Visualisation Plots**).

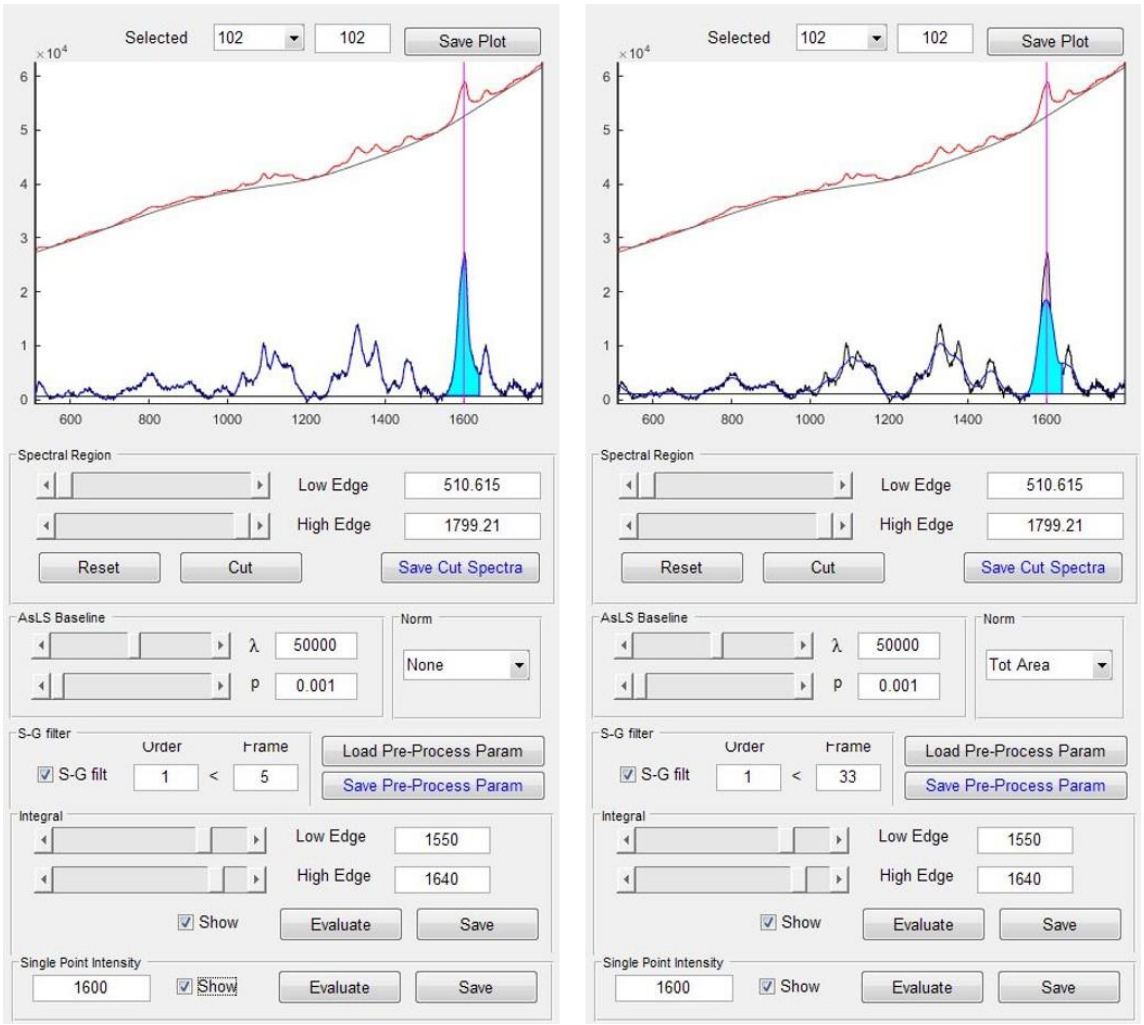

Figure 14: Single Point Intensity Evaluation. *Left: showing where the intensity values are calculated (see vertical magenta line at 1600 cm^{-1}). Right: smoothing, baseline correction and normalisation affects the intensity value. The example uses the "demo_image.txt" file.*

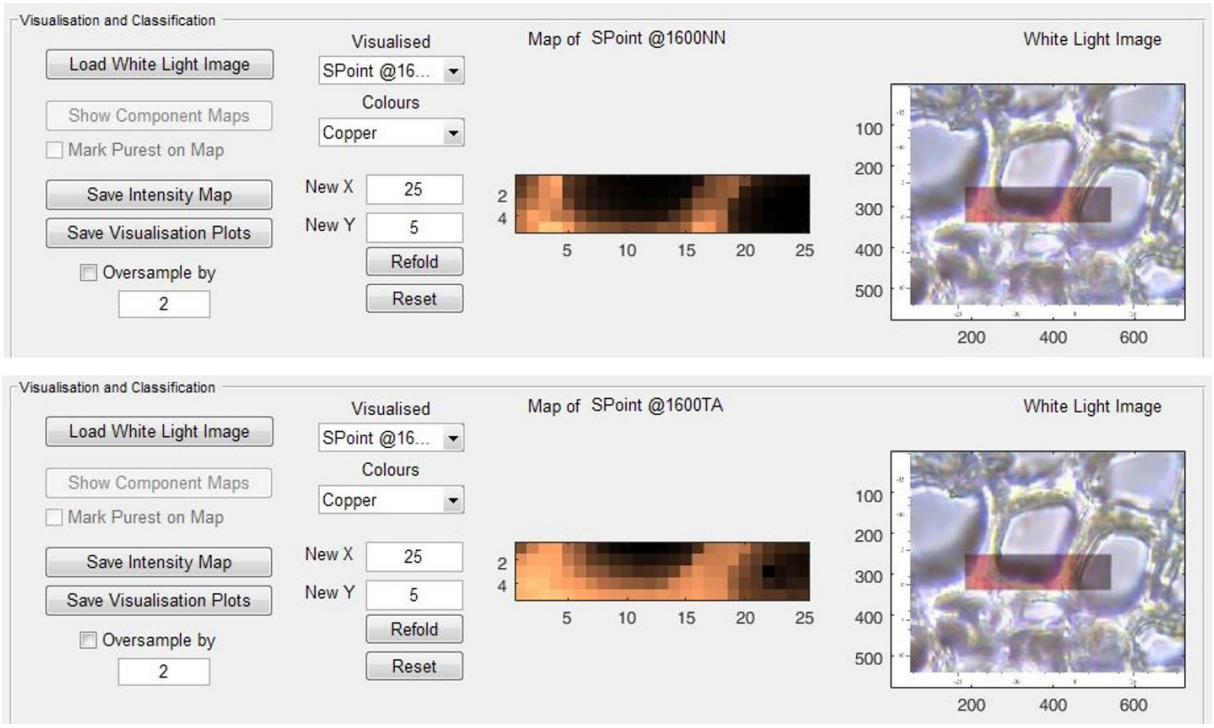

Figure 15: Visualising the Single Point Intensity Evaluation result. *The "Visualised" drop-down list is updated, together with the plot and its title. The example uses the "demo_image.txt" file with the same settings and integrals as in* ***Figure 14*** *above, the top image corresponding to* ***Figure 14 left****, and the bottom one corresponding to* ***Figure 14 right****.*

3.9. Saving and Loading Pre-Processing Parameters

To make processing different datasets in identical ways easier, the pre-processsing parameters can be saved and loaded, by using the respective buttons in the GUI. The parameters that are included in the save / load are listed in the section: **Pre-Processing Parameter Matrix**.

> ⓘ **Warning/Information: Only those parameters will be loaded that are within the range of the dataset, with warning messages given about which parameters have been ignored!** Consider the following scenario: The spectral region and integral region are both set to 400 – 450 cm^{-1} in the old dataset, with the "Show" checkbox ticked, and these pre-processing parameters are saved. Later, a new dataset is loaded, but it only contains spectral data in the 510-1800 cm^{-1} region. When the previously saved parameters are loaded, the spectral region cannot be cut and the integrals cannot be calculated, since the new dataset has no values in the 400-450 cm^{-1} region. Thus, the spectral region and the integral region will keep their current values in the GUI, as they cannot be changed to values stored in the loaded parameters. However, the "Show" checkbox will be ticked, even if it was unticked before, as that is a valid setting even for the new dataset. Keep in mind that there will be no indication that the parameters in the spectral and integral region have not been changed / updated, i.e. no indication that they do not match the saved parameters.

3.10. Displaying the Pre-Processed Data

The results of the pre-processing are only shown "live" in the GUI on the spectrum that is displayed in the "Selected Spectrum" plot and on the spectra that are marked in the Visualisation section (if there are any, **Mark Classes**). To visualise the effect of the pre-processing on all spectra, click on the "Show All Pre-Processed" button. A separate figure window will pop up, displaying ALL spectra in the dataset in a single plot. The new pop-up window has a full menu and icon bar, allowing the change of display parameters (zoom, add legend, etc.) and saving the plot (in all image formats MATLAB supports (see **Plots**). Note that all spectra are displayed, even if there is more than a 100 (as opposed to the "Original Spectra" plot, which only shows 100 randomly selected ones if there are more than 100 in the dataset).

> ⓘ **Warning/Information:** Since ALL spectra will be displayed, this can be a slow process on less powerful computers, if the pre-processing needs to be performed. A status bar will indicate the progress. Please wait until it completes. If, however, pre-processing has already been performed and no pre-processing parameters have been changed since then, displaying the results is fast.

3.11. Saving the Pre-Processed Data

The first exit point of the GUI is the saving of the pre-processed data. It marks the end of the analysis if the GUI is only used for pre-processing purposes. It is performed by simply clicking the "Save Pre-Processed" button. A dialog window opens where the filename and the folder can be specified. The GUI automatically suggests a filename, based on the original, adding the X and Y dimensions at the start, lambda and p values from the baseline correction (which is generally the most important part of the pre-processing, and sometimes the only required step) and appending the word "Processed" in the end. The saved data is in a format that can be directly loaded by the GUI without modification later (hence the blue text on the button).

> ⓘ **Warning/Information:** Irrespective of the initial input format, data will be saved as a .mat file (for the format of the resulting .mat file, see **Cut and Pre-Processed Spectra**). Thus, image .txt inputs that contained the X and Y coordinates will lose this information. This is why the GUI automatically suggests to append the dimensions to the filename in the XXX_YYY number format.

3.12. Passing the Data for MCR-ALS

The data can be passed for MCR-ALS in two ways: **unprocessed or pre-processed**, using the respective buttons in the GUI. Passing the data without processing can be preferred if the data has already been processed before (for instance by the GUI, saved (3.11. **Saving the Pre-Processed Data**) and now re-loaded) or if the unprocessed data is used for comparison / reference, etc. Otherwise pre-processing (especially **3.4.**

Asymmetrical Least Squares Baseline Correction) is generally recommended, even if only using a linear baseline.

> ⓘ **Warning/Information:** If pre-processing needs to be performed, it can be slow for large datasets and slow computers. A status bar will indicate the progress. Please wait until it completes. If, however, pre-processing has already been performed (e.g. for **3.7.** Integration) and no pre-processing parameters have been changed since then, passing the data for MCR-ALS is fast.

Once the data is passed for **MCR-ALS**, the MCR-ALS section of the GUI starts to populate, including the "Singular Value Decomposition" plot, the "Eigen Value" drop-down list, the "# Components" text box, the MCR-ALS "Direction" drop-down list, the "Noise" text box, the "Purest" spectrum list box and the "Initial Estimates" plot. The "Load Input" button is also activated (**Figure 16**).

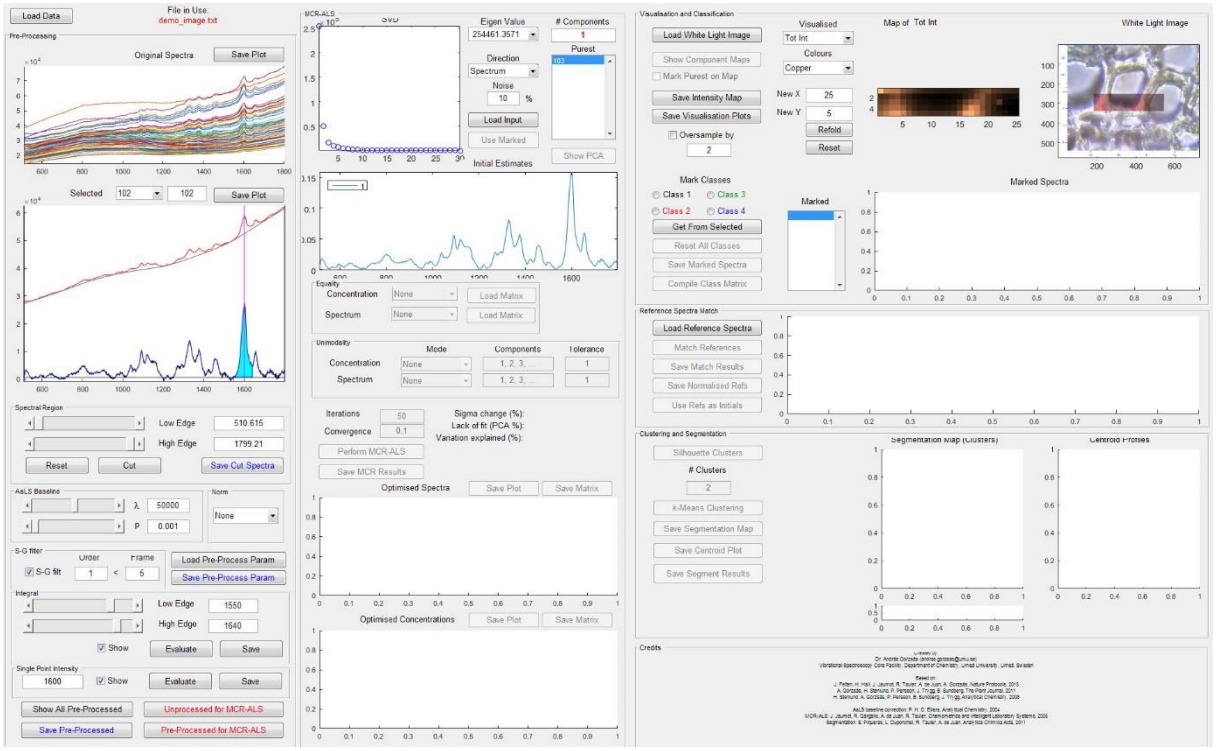

Figure 16: Passing the data for MCR-ALS. *Additional elements of the GUI are activated and plots populated. The example uses the "demo_image.txt" file.*

> ⓘ **Warning/Information:** If there have already been MCR-ALS results in the MCR-ALS part of the GUI, these will be permanently erased from the GUI without a warning! Variables in the MATLAB Workspace will remain, however (see **Perform MCR-ALS**).

4. MCR-ALS

Multivariate Curve Resolution – Alternating Least Squares (MCR-ALS) is a resolution technique based on the assumption that any spectrum can be described as the linear combination of a set of "pure" component contributions (plus some noise). "Pure" components ideally mean pure chemical compounds, but such resolution is not always possible. Thus, in practice "pure" means as clearly resolved as possible. As an iterative technique, the level of resolution depends on the iteration parameters (see **Number of Iterations and Convergence Criterion**), and to a large extent on the initial estimates from where the iterations start (to avoid local minima and find absolute minimum) (see **Initial Estimates**). In addition, resolution can be aided (the number of potentially correct solutions reduced) by applying various constraints (see **Constraints**). Naturally, the input data is of vital importance, thus pre-processing significantly affects the results.

> ⓘ **Warning/Information:** There is NO INDICATOR telling whether the current Pre-Processing values have been implemented or not yet. For instance, if spectra have been pre-processed with a set of parameters, and MCR-ALS analysis has been performed, the optimised spectra and concentration plots will be populated. If the user now changes any of the pre-processing parameters, the MCR-ALS results remain in the GUI display but they are still referring to the results obtained by the previous pre-processing until a new MCR-ALS is done. This is not a bug, it is a deliberate design.

The GUI is an improved version of the one published in Nature Protocols in 2015 (http://www.nature.com/nprot/journal/v10/n2/abs/nprot.2015.008.html), but it is does NOT include several extremely powerful and useful MCR-ALS tools. For instance, it does not utilise image information to aid the resolution (such as local rank constraints), and it cannot simultaneously handle multimethod and/or multiset data. For a more versatile MCR-ALS package and more information about MCR-ALS, visit www.mcrals.info.

4.1. Number of Components

The first step in performing MCR-ALS is to determine the number of pure components that are required to describe the dataset. This can be done based on a priori knowledge of the system or by analysing the dataset using singular value decomposition.

If the number of components is known, type the number directly in the "# Components" box in the MCR-ALS section of the GUI. Alternatively, use the "Eigen Value" drop-down list to see the numerical results of the singular value decomposition, and select the one after which a large drop in Eigen Values is observed or when the Eigen Values approximate 0 (this can be observed in the "SVD" plot of the MCR-ALS section of the GUI) (**Figure 17**).

> ⓘ **Warning/Information:** Double-clicking any element of the "Purest" listbox will remove it and adjust the number of components accordingly. This only works until only 1 element remains in the "Purest" listbox. The last remaining element cannot be removed (i.e. the "Purest" listbox cannot become empty). If this element needs to be removed, a new one must first be added.

After providing the number of components (either through the “Eigen Values” drop-down list or via the “# Components” textbox, the SVD plot zooms in to show only the number of components + 5 values (to have a better overview), the Initial Estimates plot and the “Purest” spectra listbox updates (see **Automatic Estimation and Mark Purest on Map**) and the “Show PCA” button becomes active (**Figure 17**).

ⓘ **Warning/Information:** Technically it is possible to select (close to) as many pure components as there are spectra in the dataset, but this is practically meaningless and can cause errors / warnings in the MCR-ALS algorithm (overfitting). Try to keep the number of components as low as possible. For similar reasons, the maximum number of singular values is limited to 30. This is a completely arbitrary number, and can easily be changed in the source code of the GUI without affecting any other parts of the script.

ⓘ **Warning/Information:** Determining the number of components by singular values can be less than straightforward at times, since the values may not drop sharply at a certain point but gradually decrease (**Figure 17**). In this case, try the following: a) test different numbers of components, perform MCR-ALS and observe the results to decide which is the optimum; b) use any a priori knowledge / reasoning to motivate the optimal number of components (try to keep to as low numbers as possible); c) adjust pre-processing parameters and see if this helps determining the number of components.

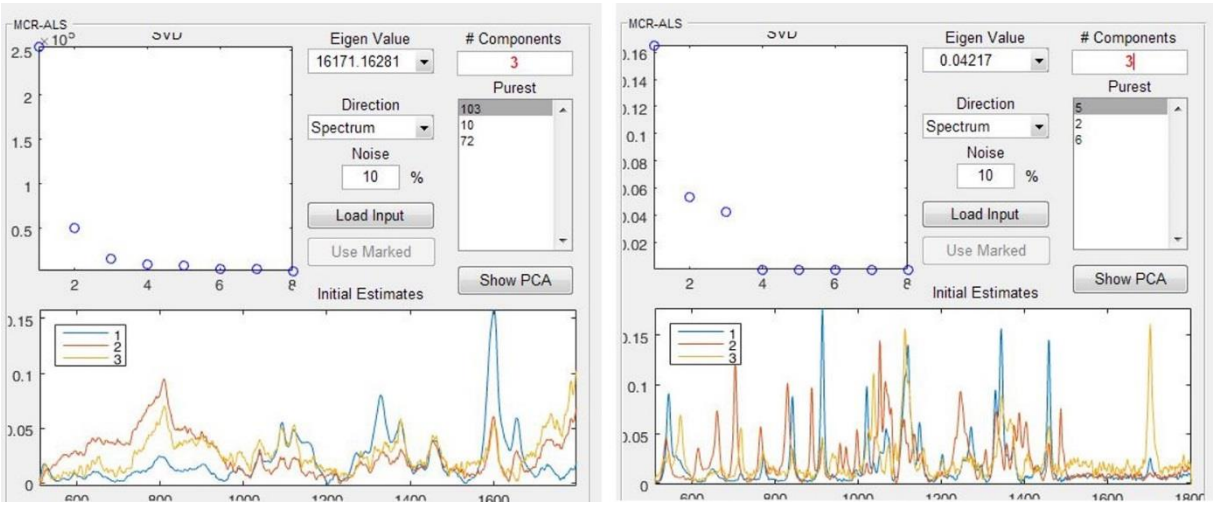

Figure 17. *Determining the number of components by singular value decomposition. Left: the “demo_image.txt” data (pre-processed with the parameters shown in* ***Figure 16****) displays a gradual decrease with no abrupt drop in Eigen Values. The number of components can reasonably be assumed to be 3-5 (6 and 7 are already close to 0 on the SVD plot) and it is recommended to test MCR-ALS with all those possibilities. Right: the unprocessed “demo_indep_spectra.xlsx” dataset has a clear drop in Eigen Values after the 3rd component. Thus, this data is most likely to have 3 components.*

4.2. Show PCA

After the number of Components have been set (), principal component analysis (PCA) of the data is possible, using as many principal components (PCs) for the reconstruction of the dataset as the number of Components. Click on the "Show PCA" button to perform the PCA analysis and to visualise the results in a separate, pop-up window. This window contains 4 plots: the Initial Pure Spectra Estimates and Unconstrained concentration profiles (using a single iteration least squares fit), compared to the PCA Loadings and Scores, respectively, prior to MCR-ALS (**Figure 18**). This could be especially interesting when initial estimates are not determined automatically but are supplied in another way (manually loaded, selected from marked pixels or from reference spectra (see **Initial Estimates**)).

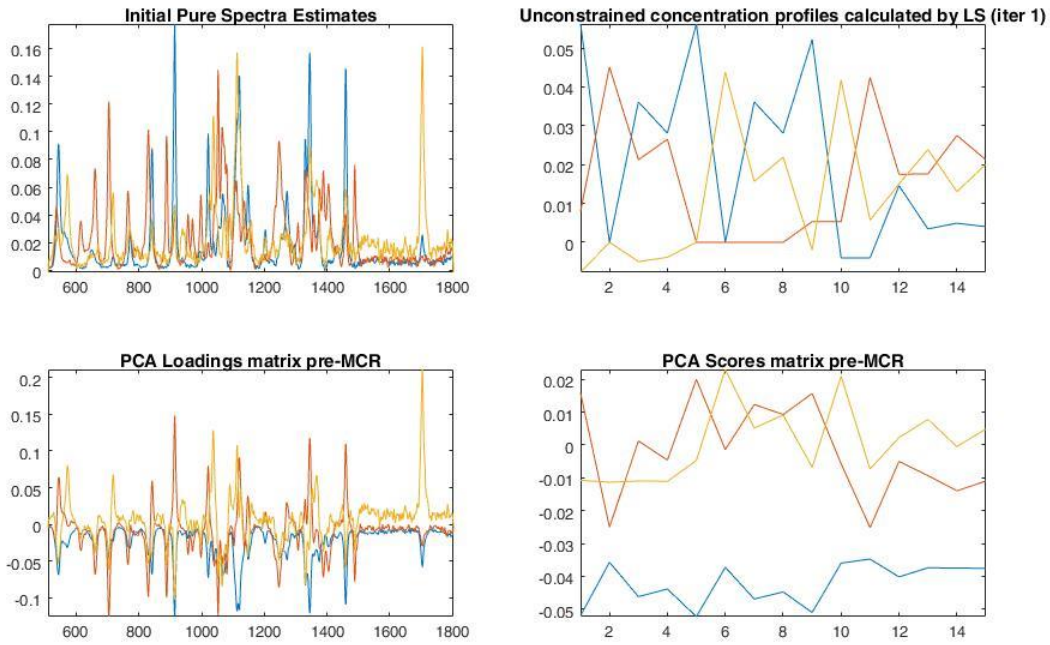

Figure 18. *PCA reconstruction results visualised in a separate pop-up window after clicking the "Show PCA" button in the GUI. The example uses un-processed "demo_indep_spectra.xlsx" dataset with 3 Components*

4.3. Initial Estimates

The initial estimates are very important to the success of the MCR-ALS resolution. Therefore the GUI allows for different methods of providing them.

The direction of the MCR-ALS procedure (see **Direction**) determines the kind of input required: "Spectrum" direction (default in the GUI) requires spectra as initial estimate input, while "Concentration" direction naturally requires concentration values as input. The examples below only use the "Spectrum" direction for illustration (i.e. in Figures) because these are easier to follow (less abstract) and some options are only viable for those (such as **Use Marked Spectra** and **Use Reference Spectra**), but notes are given for "Concentration" direction as well.

Potential overlaps with constraints are listed as well.

4.3.1. Automatic Estimation and Mark Purest on Map

The **default option** in the GUI is the automatic estimation of the initial values, using a SIMPLISMA-based algorithm (see www.mcrals.info for more information). **This method works both in the Spectrum and the Concentration directions** is completely **automatic.** It is performed as soon as the data is passed for MCR-ALS from the Pre-Processing section of the GUI (3.12. **Passing the Data for MCR-ALS**), and Initial Estimates are updated automatically whenever the number of Components is changed (see **Number of Components**), including updating the Initial Estimates plot and the estimation of the purest spectrum (or wavenumber if Concentration direction is used) (see **Figure 19**).

The only control the user have is the "Noise" textbox, where the level of allowed noise can be set (in percentages) for the Initial Estimation procedure. The default value of 10% Noise is usually sufficient, and small changes in the Noise percentage usually have no or minimal effect on the initial estimates on normal vibrational spectral datasets. This can be easily tested by setting different noise levels and observing the changes in the Initial Estimates plot and selecting the best result.

If the Spectrum direction is selected, the "Mark Purest on Map" checkbox is activated, allowing for displaying which spectrum in the dataset is estimated to be the purest for each Component. When this checkbox is ticked, white diamonds with cyan outline appear in the active plot in the Visualisation section of the GUI (**Figure 19**). This option is naturally NOT available if Concentration direction is selected (see **Direction**).

ⓘ **Warning/Information:** The Initial Estimates are in order in the "Purest" listbox and are coloured in the "Initial Estimates" plot according to the legend there. That is, the first Component is listed as the first in the "Purest" listbox and is plotted in blue in the "Initial Estimates" plot in the example in Figure 19. Please note that the colours used for the Components and their order is determined by the version of MATLAB. Nevertheless, the markers on the visualisation plot are **all white (not colour coded) and have no labels** to indicate which Component they belong to (see **Figure 19** for an example).

ⓘ **Warning/Information:** The Markers remain even if the Visualised plot is changed using the "Visualised" drop-down list (see **Change Plot**) or a different colour coding is selected via the "Colours" drop-down list (see **Change Colour**), as long as the checkbox is ticked.

ⓘ **Warning/Information:** The position of the Markers automatically update when the image is refolded / reset (see **Change Dimensions (Refold)**).

ⓘ **Warning/Information:** The Markers automatically update when the number of Components is altered (see **Number of Components**).

☒ **Known bug:** The Markers disappear from the Visualisation plot if a new Integral (3.7. **Integration**) or Single Point Intensity evaluation is performed (**3.8. Single Point Intensity Evaluation**).

☑ **Workaround:** Refresh the plot by either of the following methods: a) untick and tick the "Mark Purest on Map" checkbox; b) Activate the "Visualised" drop-down list (even if clicked on the very same value, it triggers the update of the plot, it does not have to be changed to a new value / new plot selected) (see **Change Plot**); c) Activate the "Colours" drop-down list (even if clicked on the very same value, it triggers the update of the plot, it does not have to be changed to a new value / new plot selected) (see **Change Colour**); d) "Refold" or "Reset" the plot (see **Change Dimensions (Refold)**); d) Change the number of Components to a new value than change back to the desired one (see **Number of Components**).

ⓘ **Warning/Information:** Markers do NOT appear in the plots of pop-up windows (i.e. the plots displayed by the "Show Component Maps" button, see **Show Component Maps**)

ⓘ **Warning/Information:** If the Markers are visible in the Visualisation section of the GUI, they will be visible in the Saved plots (see **Save Intensity Map** and **Save Visualisation Plots**).

ⓘ **Warning/Information:** If the legend obscures the "Initial Estimates" plot, it can be moved to a different position within the plot, by left-clicking and dragging.

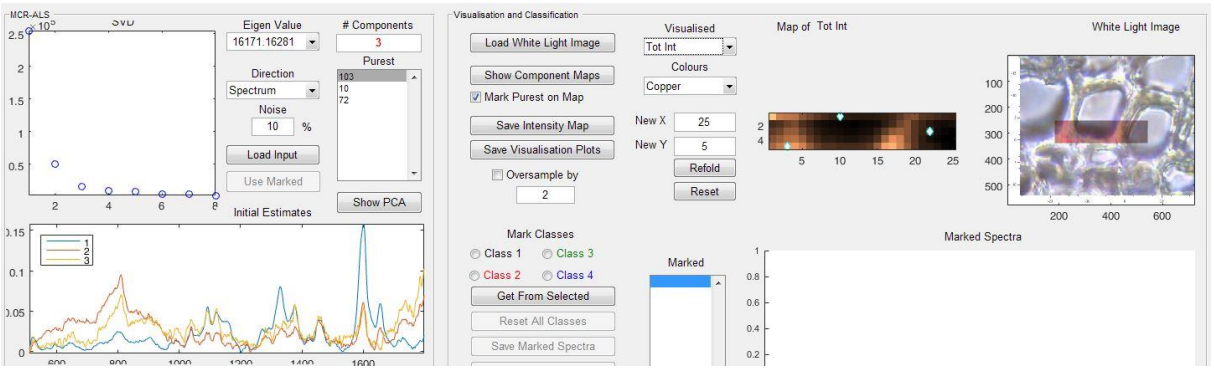

Figure 19. *Automatic determination of Initial Estimates and marking the purest spectra on the Visualisation plot. The example uses the "demo_image.txt" file, the exact same way as in* ***Figure 17 left****. The 1st Component is shown in blue in the "Initial Estimates" plot and the purest spectrum belonging to this Component is estimated to be nr. 103 (1st value the "Purest" listbox). The 2nd Component is sown in red in the "Initial Estimates" plot and the purest spectrum belonging to this Component is estimated to be nr. 10 (2nd value the "Purest" listbox). The 3rd Component is shown in yellow in the "Initial Estimates" plot and the purest spectrum belonging to this Component is estimated to be nr. 72 (3rd value the "Purest" listbox). However, there is no indicator on the Markers in the Visualisation plot, they are all white diamonds with cyan outline. Based on the spectrum number, it can be deduced, however, that the bottom left marker must be spectrum 103 (the highest number), and thus Component 1, the topmost marker (in the first row of the image) must be spectrum 10 (the lowest number) and thus Component 2, while the marker most to the right must be spectrum 72 and thus Component 3.*

4.3.2. Load Input

Input data can be manually loaded, using the "Load Input" button in the MCR-ALS section of the GUI, to complement / replace the automatically determined Initial Estimates. The Inputs to be loaded must be in a .mat file that meets the formatting requirements as described in the section "**Initial Estimate Input**" and must match the direction of the Initial Estimates (spectrum or concentration). The Loaded Initial Estimates are added to the already listed Initial Estimates and displayed in the "Initial Estimates" plot as well as in the "Purest" listbox, using the names supplied in the input .mat file (**Figure 20**).

This method works both in the Spectrum and the Concentration directions.

If some of the Initial Estimates need to be removed after loading (either from among the automatically determined ones or from the newly loaded ones), double-click its descriptor in the "Purest" listbox.

ⓘ **Warning/Information:** Double-clicking any element of the "Purest" listbox will remove it and adjust the number of components accordingly. This also removes its Marker from the Visualised plot, if it had any. This only works until only 1 element remains in the "Purest" listbox. The last remaining element cannot be removed (i.e. the "Purest" listbox cannot become empty). If this element needs to be removed, a new one must first be added.

ⓘ **Warning/Information:** Naturally, the loaded Initial Estimates cannot have Markers in the Visualised plot, as they do not feature in the original dataset plotted there (**Figure 20**).

ⓘ **Warning/Information:** If the legend obscures the "Initial Estimates" plot, it can be moved to a different position within the plot, by left-clicking and dragging.

ⓘ **Warning/Information: Equality Constraint vs Load Input**: If a certain compound is known / suspected to be a pure component in the dataset, it can be supplied either as an initial estimate or as an equality constraint for the spectral direction. There is a difference in the mechanism how this can practically affect the results. An initial estimate does NOT pose ANY constraints at all during the modelling, it only makes sure that the iteration starts with this value. From that on, the model can take any direction and the spectral profile can change freely, without restrictions (apart from non-negativity). In other words, this way of supplying a spectral profile only ensures that the optimization starts close to the expected final result (i.e. close to the suspected absolute minimum, avoiding falling into a potential local minima, or lowering the number of iterations the model requires to reach convergence). On the other hand, if a spectral profile is supplied as a constraint, it poses a more rigid limitation for how the model can evolve. With an "exact" constraint applied, for instance, the component spectral profile is NOT allowed to change at all. This can be useful when a known compound with a known (and exact) spectral profile needs to be tracked in the dataset. The same is true in the concentration direction, with concentration profiles loaded as initial estimates vs supplied as equality constraint.

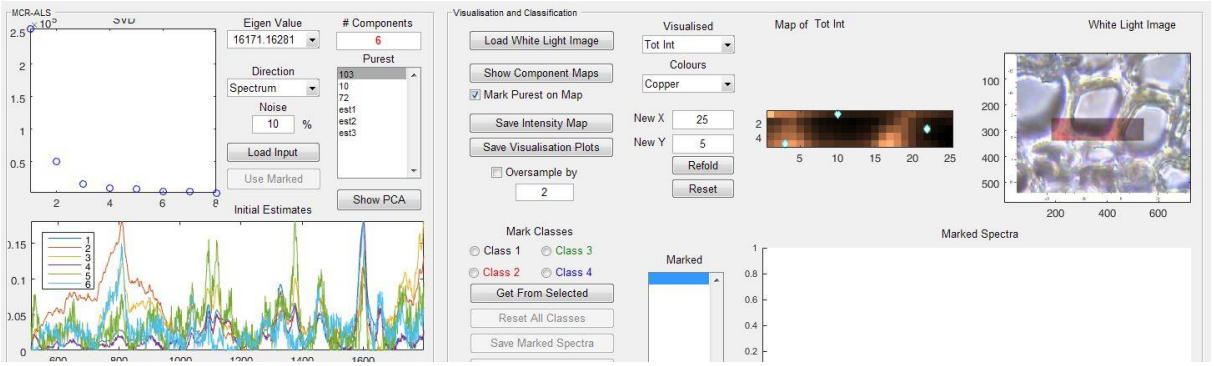

Figure 20. *Load Input manually for Initial Estimates. The example uses the "demo_image.txt" file exactly as in* ***Figure 19****, with the manually loaded auxiliary data "InitEstimates_spectra.xlsx". This auxiliary data file contains the spectral profiles "est1", "est2" and "est3" (as seen in the "Purest" listbox). Note that these names are NOT listed in the Initial Estimates plot legend, which only contains the colours and the list numbers. It can be deduced, however, that nr. 6 is "est3", since that is listed as the 6th in the "Purest" listbox. There are no Markers in the Visualisation plot for the newly loaded Initial Estimates.*

4.3.3. Use Marked Spectra

Initial Estimates can be manually specified from the current dataset to complement / replace the automatically determined Initial Estimates. These manually specified Initial Estimates need to be first marked in the dataset either as classless (class 0, gray) or belonging to any of the 4 classes possible to specify (class 1 – 4, black, red, green and blue, see **Mark Classes**).

This method ONLY works in the Spectrum direction!

If some of the Initial Estimates need to be removed after loading (either from among the automatically determined ones or from among the ones imported by the "Use Marked" button), double-click their descriptor (spectrum number) in the "Purest" listbox.

ⓘ **Warning/Information:** Marked spectra are added to the "Purest" listbox and to the "Initial Estimates" plot in the order they appear in the "Marked" textbox (**Figure 21**).

ⓘ **Warning/Information:** If the "Mark Purest on Map" checkbox is ticked, the white diamond markers can completely cover the coloured marker squares. Untick the checkbox of those markers are to be visible.

ⓘ **Warning/Information:** Double-clicking any element of the "Purest" listbox will remove it and adjust the number of components accordingly. This also removes its Marker from the Visualised plot, if it had any. This only works until only 1 element remains in the "Purest" listbox. The last remaining element cannot be removed (i.e. the "Purest" listbox cannot become empty). If this element needs to be removed, a new one must first be added.

ⓘ **Warning/Information:** If the legend obscures the "Initial Estimates" plot, it can be moved to a different position within the plot, by left-clicking and dragging.

ⓘ **Warning/Information: Equality Constraint vs Load Input**: If a certain compound is known / suspected to be a pure component in the dataset, it can be supplied either as an initial estimate or as an equality constraint for the spectral direction. There is a difference in the mechanism how this can practically affect the results. An initial estimate does NOT pose ANY constraints at all during the modelling, it only makes sure that the iteration starts with this value. From that on, the model can take any direction and the spectral profile can change freely, without restrictions (apart from non-negativity). In other words, this way of supplying a spectral profile only ensures that the optimization starts close to the expected final result (i.e. close to the suspected absolute minimum, avoiding falling into a potential local minima, or lowering the number of iterations the model requires to reach convergence). On the other hand, if a spectral profile is supplied as a constraint, it poses a more rigid limitation for how the model can evolve. With an "exact" constraint applied, for instance, the component spectral profile is NOT allowed to change at all. This can be useful when a known compound with a known (and exact) spectral profile needs to be tracked in the dataset. The same is true in the concentration direction, with concentration profiles loaded as initial estimates vs supplied as equality constraint.

☒ **Known bug:** The Markers disappear from the Visualisation plot if a new Integral (3.7. **Integration**) or Single Point Intensity evaluation is performed (**3.8. Single Point Intensity Evaluation**).

☑ **Workaround:** Refresh the plot by either of the following methods: a) untick and tick the "Mark Purest on Map" checkbox; b) Activate the "Visualised" drop-down list (even if clicked on the very same value, it triggers the update of the plot, it does not have to be changed to a new value / new plot selected) (see **Change Plot**); c) Activate the "Colours" drop-down list (even if clicked on the very same value, it triggers the update of the plot, it does not have to be changed to a new value / new plot selected) (see **Change Colour**); d) "Refold" or "Reset" the plot (see **Change Dimensions (Refold)**); d) Change the number of Components to a new value than change back to the desired one (see **Number of Components**).

☒ **Known bug:** If there have been Initial Estimates loaded using the "Load Input" button (see **Load Input**) or by adding them from the loaded References (see **Use Reference Spectra**) before the "Used Marked" button is clicked, those externally loaded Initial Estimates will be listed before the Marked ones, which will create an artefact Marker and an extra line appear in the Visualisation plot when the "Mark Purest on Map" checkbox is checked. This is because the externally supplied Initial Estimates (the loaded inputs or references) are not part of this dataset and thus cannot be plotted.

☑ **Workaround:** Use the "Load Input" and/or "Use Refs as Initials" buttons AFTER using the "Use Marked" button. If there are already externally loaded Initial Estimates, remove them by double-clicking on their names in the "Purest" listbox, uncheck the "Mark Purest on Map" checkbox, click on the "Load Input" button to re-load the Initial Estimates and check the "Mark Purest on Map" box. This will place the externally loaded initial estimates at the end of the list and refresh the Markers to correctly show the position of all Initial Estimates / purest spectra.

☒ **Known bug:** If the “Used Marked” button is clicked repeatedly, spectra will be simply added to the already existing “Purest” list, without checking if they are already listed there or not. In other words, they do not overwrite or replace existing ones, simply add all spectra from the “Marked” plot to the end of the “Purest” list. Unfortunately, this can result in the same spectrum / spectra being added repeatedly.

☑ **Workaround:** Remove multiple instances of the same spectrum by double-clicking on the number in the “Purest” listbox until each spectrum (each number) only features once in the list.

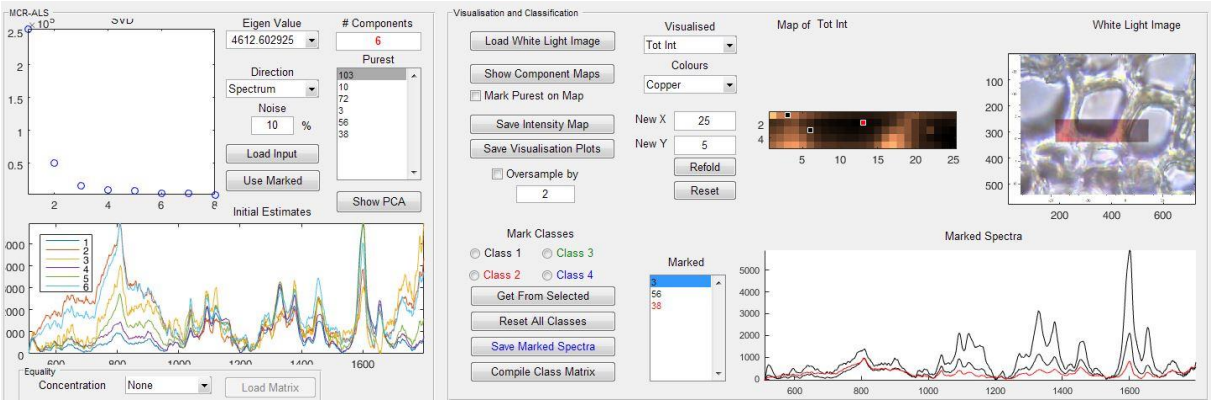

Figure 21. *Use Marked Spectra for Initial Estimates. The example uses the “demo_image.txt” file exactly as in* ***Figure 19****, with manually marking spectra 3 and 56 as Class 1 (black) and spectrum 38 as Class 2 (red), as seen in the “Marked” listbox. The “Marked Spectra” plot shows the spectra in their respective colours as well. Note that the Marked Spectra are added after the automatically determined ones (spectra 103, 10, 72), in the order they appear in the “Marked” listbox. They do not retain their class colours as Components, however (so in the Initial Estimates plot, spectra 3, 56 and 38 appear as violet, green and cyan, respectively, i.e. Components 4-6).*

4.3.4. Use Reference Spectra

The loaded Reference Spectra can be passed to the MCR-ALS part of the GUI as Initial Estimates to complement / replace the automatically determined Initial Estimates. First, load the Reference Spectra (see **Load Reference Spectra**), then click on the “Use Refs as Initials”. **Note that his button is located in the “Reference Spectra Match” section of the GUI** (**Figure 22**), not in the MCR-ALS section! All Reference Spectra will be added to the list of already existing Initial Estimates, and also displayed in the “Initial Estimates” plot as well as in the “Purest” listbox, using the reference names as they appear in the legend of the Reference Spectra plot (**Figure 22**).

This method ONLY works in the Spectrum direction.

If some of the Initial Estimates need to be removed (either from among the automatically determined ones or from the newly added ones from the references), double-click its descriptor in the “Purest” listbox.

ⓘ **Warning/Information:** Double-clicking any element of the “Purest” listbox will remove it and adjust the number of components accordingly. This also removes its Marker from the Visualised plot, if it had any. This only works until only 1 element remains in the “Purest” listbox. The last

remaining element cannot be removed (i.e. the "Purest" listbox cannot become empty). If this element needs to be removed, a new one must first be added.

ⓘ **Warning/Information:** Naturally, the Initial Estimates from Reference Spectra cannot have Markers in the Visualised plot, as they do not feature in the original dataset plotted there.

ⓘ **Warning/Information:** If the legend obscures the "Initial Estimates" plot, it can be moved to a different position within the plot, by left-clicking and dragging.

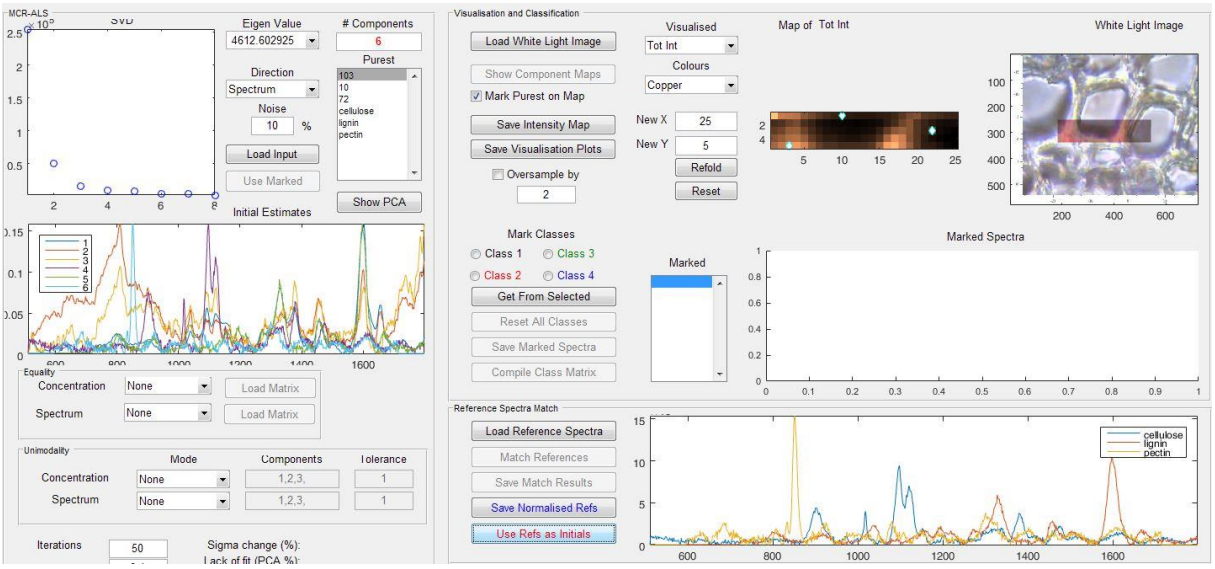

Figure 22. *Use Reference Spectra for Initial Estimates. The example uses the "demo_image.txt" file exactly as in* ***Figure 19****, with the manually loaded auxiliary data "demo_references.xlsx". This auxiliary data file contains the reference spectra "cellulose", "lignin" and "pectin" (as seen in the "Purest" listbox and in the legend of the Reference Spectra plot). Note that these names are NOT listed in the Initial Estimates plot legend, which only contains the colours and the list numbers. It can be deduced, however, that nr. 6 is "pectin", since that is listed as the 6th in the "Purest" listbox. There are no Markers in the Visualisation plot for the newly loaded Initial Estimates. The Reference Spectra plot shows the spectra in the default MATLAB colour order, which is conflicting the colouring of the "Initial Estimates" plot. Thus, the Reference Spectra used for Initial Estimates will NOT keep their colours, but will be assigned a new colour depending on their position in the "Purest" listbox (so in the Initial Estimates plot, cellulose is NOT blue as in the Reference Spectra plot, but violet, as it has position 4 in the "Purest" listbox, i.e. Component 4).*

ⓘ **Warning/Information: Equality Constraint vs Load Input**: If a certain compound is known / suspected to be a pure component in the dataset, it can be supplied either as an initial estimate or as an equality constraint for the spectral direction. There is a difference in the mechanism how this can practically affect the results. An initial estimate does NOT pose ANY constraints at all during the modelling, it only makes sure that the iteration starts with this value. From that on, the model can take any direction and the spectral profile can change freely, without restrictions (apart from non-negativity). In other words, this way of supplying a spectral profile only ensures that the optimization starts close to the expected final result (i.e. close to the suspected absolute minimum, avoiding falling into a potential local minima, or lowering the number of iterations the model

requires to reach convergence). On the other hand, if a spectral profile is supplied as a constraint, it poses a more rigid limitation for how the model can evolve. With an "exact" constraint applied, for instance, the component spectral profile is NOT allowed to change at all. This can be useful when a known compound with a known (and exact) spectral profile needs to be tracked in the dataset. The same is true in the concentration direction, with concentration profiles loaded as initial estimates vs supplied as equality constraint.

☒ **Known bug:** If the "Use Refs as Initials" button is clicked repeatedly, spectra will be simply added to the already existing "Purest" list, without checking if they are already listed there or not. In other words, they do not overwrite or replace existing ones, simply add all spectra from the References to the end of the "Purest" list. Unfortunately, this can result in the same spectrum / spectra being added repeatedly.

☑ **Workaround:** Remove multiple instances of the same spectrum by double-clicking on its name in the "Purest" listbox until each spectrum only features once in the list.

4.4. Direction

Use the "Direction" drop-down list in the MCR-ALS part of the GUI to specify the direction in which the Initial Estimates are determined and the alternating least squares algorithm starts.

With high enough number of iterations and correct convergence limits (see **Number of Iterations and Convergence Criterion**), the direction of the iterations should have minimal impact on the MCR-ALS results. However, setting the direction to either "Spectrum" (default) or "Concentration" can be helpful in evaluating the quality of the data already before MCR-ALS modelling takes place, simply by inspecting the Initial Estimates. For example, if "Spectrum" direction is selected, the initial spectral profiles can be informative about the compounds the model is estimating to be present in the dataset, and changing the number of components can reveal further information about minor components / noise levels in the data. Similarly, if certain concentration profiles are known / expected (e.g. from monitoring a reaction and knowing at which spectrum a reactant was added, etc), "Concentration" direction can provide a very quick quality assessment already at the Initial Estimates plot.

ⓘ **Warning/Information:** Some Initial Estimate inputs are only possible with "Spectrum" direction selected (see **Initial Estimates**).

4.5. Constraints

Constraints can dramatically alter the result of the MCR-ALS process and are a valuable tool in reaching optimum resolution. Thus, using the correct constraints constitutes the "art" of MCR-ALS and can make or break a model. However, while correct constraints can salvage something even from vastly suboptimal datasets, ultimately it is the input data quality that determines the success of MCR-ALS. (Conversely, incorrect constraints can easily distort MCR-ALS results even from perfectly good datasets.)

Constraints can be mild (allowing a certain degree of flexibility) or hard (when the rules they set must be strictly followed). Generally, hard constraints should only be used when the data is certain to fulfil their prerequisites, in order to minimise the chance of artefacts / bias in the results.

The present GUI only contains constraints that have chemical / spectroscopic foundations (e.g. non-negativity), not only mathematical (local rank constraints, etc).

> ⓘ **Warning/Information:** Constraints need to be applied with care as they have profound effects on the outcome of the model and can constitute to serious bias. Improper constraints can be easy to discover if the MCR-ALS modelling gives any kind of erroneous result or MATLAB shows error or warning messages in the Command Window (such as rank deficiencies, NaN values instead of numbers in the results, etc), but they can be hard to spot if the model appears to work correctly computationally. As a general rule, MCR-ALS models with the minimum necessary constraints are preferred, and constraints should preferably be based on established features of the dataset.

> ⓘ **Warning/Information:** Constraints can be set independently from each other and multiple constraints can be applied at the same time (for the same or for different components). The more constraints applied, the harder the limitations become.

4.5.1. Non-Negativity

This is a default constraint in **both the Spectrum and the Concentration** profiles, hard-coded in the GUI with **no user control**. It is implemented as a mild constraint, using a fast non-negative least squares algorithm, as opposed to having a hard non-negativity constraint, where negative values would simply be force-replaced by zeros, without fitting.

The present non-negativity constraints are built-in and **cannot be removed or modified** in the GUI without extensive modification of the source code. Thus, the GUI is not well suited for datasets that inherently contain negativity (e.g. EPR spectra), although this restriction is not a general MCR-ALS feature otherwise.

4.5.2. Equality

Equality constraints can be set for one, several or all components, independently for concentrations and spectra or both. The default setting is “None”, i.e. no equality constraint applied for either concentration or spectra.

Equality constraints allow the user to fix values in the concentration or spectral profiles, such as the spectrum of a known compound present in the dataset, and/or its concentration in certain spectra / pixels of the dataset (including even 0 concentrations, i.e. the absence of a compound. For instance the starting material can be fixed at 100% concentration (i.e. being pure) in the first few spectrum of a reaction dataset before the reactant is added, while the concentration of both the reactant and the product(s) can be fixed at 0% in the same spectra).

The method for equality constraints can be set using the drop-down list in the “Equality” box of the MCR-ALS section of the GUI. The default option is “None” (i.e. no equality constraint), but it can be changed to “Exact” (i.e. the constraint values must be met exactly) or “Maximum” (i.e. the constraint values cannot be exceeded, but they are allowed to be lowered). For example, if Component 1 concentration is set to be 80%

in a certain spectrum of the dataset, it will remain exactly at 80% in that spectrum during MCR-ALS modelling with the "Exact" method, while it can obtain any value between 0 – 80% during MCR-ALS modelling with the "Maximum" method. (0 is the minimum value it can obtain, due to the fixed non-negativity constraints. Thus, if a concentration is set to 0 in a spectrum, "Exact" and "Maximum" type of equality constraints will have the same effect.)

ⓘ **Warning/Information:** Setting "Exact" for **spectral equality constraint** is a very hard constraint, since it will not consider noise or small shifts in the spectra. While "maximum" can be somewhat milder, it is still much more rigid in the spectral direction than it is in the concentration direction, since it still does not cope well with shifting peaks, or with varying total intensities. Be wary of using spectral equality constraints and make sure the spectral profiles used for constraining are of good quality and matching the dataset.

Once the method of the equality constraint is selected, the respective "Load Matrix" button will become active. Click and select the files to apply the constraints.

ⓘ **Warning/Information:** There is no way of adjusting the equality constraint interactively in the GUI. Constraints need to be loaded from an already existing .mat file, which must meet the formatting criteria required by the GUI (see **Equality Constraint Matrices** and the included "csel.mat" and "ssel.mat" auxiliary data files).

ⓘ **Warning/Information:** The GUI will show an error message if the dimensions do not match (e.g. the number of components is wrong, the number of spectral variables is wrong (spectra cut to different length), etc.), but it will not give further details regarding the origin of the dimension mismatch. MCR-ALS cannot be started until a correct Matrix is loaded, OR the Equality constraint is set to "None".

☒ **Known bug:** The GUI can be fooled by changing the number of components after the Equality Constraint Matrix has been loaded (i.e. selecting a different number of components afterwards, which does not match the dimension of the loaded .mat file of the equality constraint). If the new number of components is lower than the one used in the Equality Constraint Matrix, an error message will be shown in the MATLAB Command prompt: "Index exceeds matrix dimensions". If, however, the new number of components is higher than the one used in the Equality Constraint Matrix, no error message appear, but MCR-ALS is NOT performed correctly.

☑ **Workaround:** Make sure dimension match: A) do not change the number of Components after the Equality Constraint Matrix has been loaded; B) If the number of Components must be changed, either load a new matrix with the correct dimensions, or select "None" in the Equality Constraint drop-down list to remove the wrong matrix / constraint.

ⓘ **Warning/Information:** Once an Equality Constraint Matrix is loaded, it is kept until "None" is selected in the drop-down list. This allows for changing between "Exact" and "Maximum" methods quickly without the need to load the matrix again. However, once "None" is selected, the matrix is deleted.

Two equality constraint matrices are supplied (ssel.mat and csel.mat) as auxiliary data, matching the demo_image.txt file dimension for 3 components, provided that the spectral range of the dataset is cut to 1800 cm^{-1} on the high edge during pre-processing (see **Figure 19**). Both of these matrices are extreme (and completely fabricated, i.e. they have nothing to do with the actual dataset apart from matching its dimensions) for pedagogical reasons: the impact of the equality constraint can be easily visualised with such drastic (and unfitting) constraints magnifying their effect.

The csel.mat matrix constrains Component1 to have a concentration of 1 in the first 3 spectra (pixels) of the dataset, and 0 in all other spectra (pixels), i.e. it should only be present in the first 3 spectra (pixels), for both "Exact" and "Maximum" type of concentration constraining (the difference between the "Exact" and "Maximum" types can be visualised easily by changing the type of normalisation during Pre-Processing, between "None" and "Tot Area". This will change the intensity values to be outside or within the range of the csel matrix, respectively). Component 2 is unconstrained in the csel matrix except for the last spectrum (pixel) of the dataset, where its concentration is set to 1 (**Figure 23**).

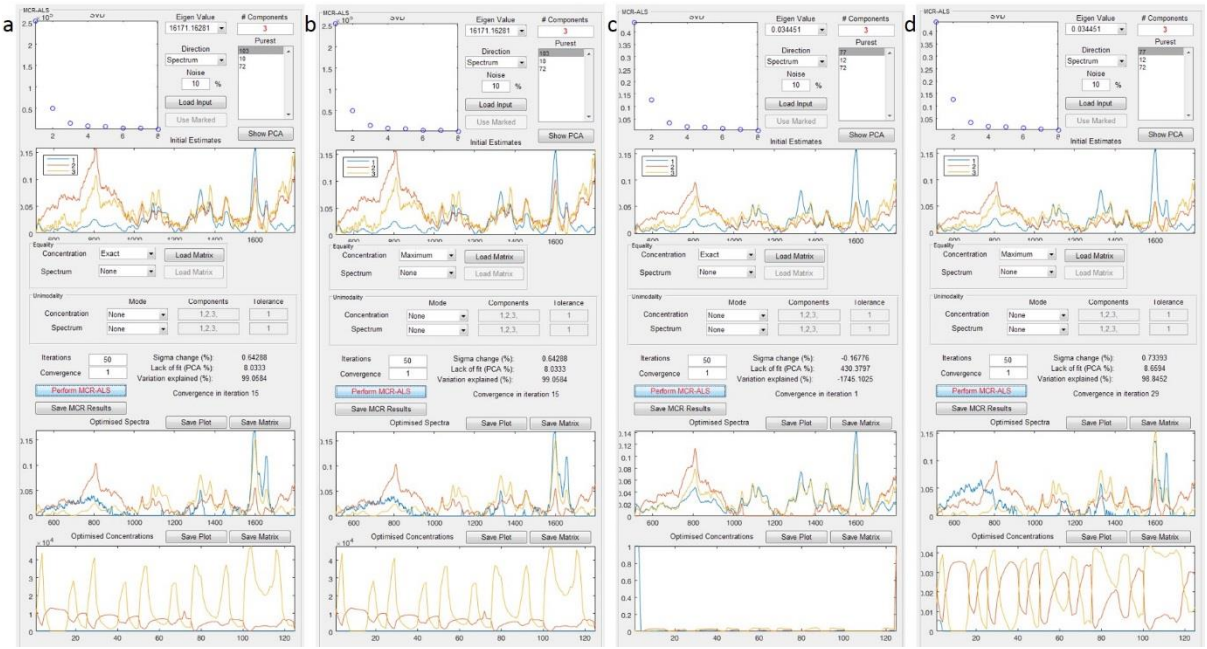

Figure 23. *The effect of Concentration Equality Constraints. The example uses the "demo_image.txt" file exactly as in* ***Figure 19*** *for* ***a*** *and* ***b****, and with Total Area Normalisation for* ***c*** *and* ***d****. 3 components are selected for MCR-ALS, using automatic initial estimate determination, spectral direction and 10% noise. Convergence limit is set to 1 for speed, and maximum number of iterations is kept at the default 50. Results are identical without normalisation (c.f.* **a** *and* ***b****) since the constraints are numerically outside the range of the dataset (see the scale on the Y-axis of the Optimal Concentration Plot). For visualisation purposes, either the csel matrix values need to be adjusted, or the spectra scaled, by e.g. total area normalisation. This highlights the effect of the concentration equality constraint and the differences between "equal" (a hard constraint) and "maximum" (a milder constraint), c.f.* ***c*** *and* ***d****.*

The ssel.mat matrix constrains Component1 to only contain two bands around 1600 and 1650 cm^{-1}, and 0 values (baseline) everywhere else. The other 2 components are unconstrained and thus can have any spectral profiles during MCR-ALS modelling. Since both "Exact" and "Maximum" are hard constrains for

spectral profiles, visualising the differences between these two methods is difficult in the spectrum direction. Slight differences can be noticed when "Point Max" normalisation is used at 1600 cm^{-1} and MCR-ALS is performed with a Convergence Criterion of 1 (Convergence is reached in Iteration 15 using the "Maximum" method, with 98.3294% variation explained and 10.663% lack of fit. Using the "Exact" method, convergence is reached in 18 iterations only, with slightly worse results: 98.1688% variation explained and 11.3927% lack of fit). The resolved spectral and concentration profiles are very similar, however, due to the severity of the constraint applied (**Figure 24**).

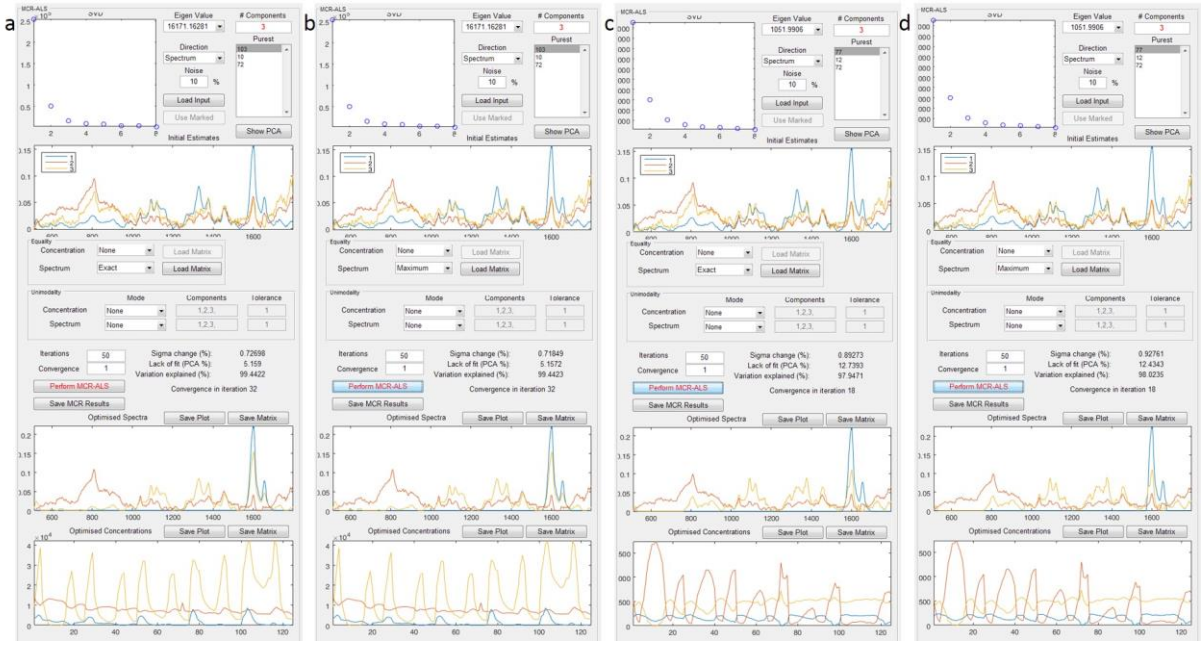

Figure 24. *The effect of Spectrum Equality Constraints. The example uses the "demo_image.txt" file exactly as in* ***Figure 19*** *for* ***a*** *and* ***b****, and with Point Max Normalisation at 1600 cm^{-1} for* ***c*** *and* ***d****. 3 components are selected for MCR-ALS, using automatic initial estimate determination, spectral direction and 10% noise. Convergence limit is set to 1 for speed, and maximum number of iterations is kept at the default 50. Results are almost identical without normalisation (c.f.* **a** *and* ***b****) and somewhat more differing with normalisation (c.f.* ***c*** *and* ***d****). The effect of normalisation is drastic in the concentration profiles (understandably) but milder in the spectral profiles (c.f. band ratios in Component 3 spectra in the range 1300-1400 cm^{-1} and the intensity of the 1600 cm^{-1} band (directly affected by the normalisation) in* ***a/b*** *vs* ***c/d****).*

> ⓘ **Warning/Information:** It is possible to set BOTH spectral and concentration equality constraint, independently from each other. They do not need to concern the same component(s) either, although that is also possible. The more constraints applied, the harder the limitations become.

4.5.3. Unimodality

Unimodality constraints can be set independently for concentration and spectral profiles, by first selecting the mode unimodality should be applied for each, in their respective drop-down list. "None" means no unimodality constraint is applied and in this case the respective "Components" and "Tolerance" textboxes are disabled (grayed out). "Vertical" and "Horizontal" modes mean that secondary maxima are cut either

vertically or horizontally, respectively, during modelling. “Average” mode means that secondary maxima are adjusted by averages (similarly to least squares fitting algorithms). While this is the smoothest application of the constraint, it is also by far the slowest, due to nested loops inside conditional statements.

> ⓘ **Warning/Information:** Choosing “Average” with large number of variables (i.e. many spectra if concentration unimodality is set to “Average”, or many spectral points (high spectral resolution and/or wide spectral region) if spectra unimodality is set to “Average”) can be **extremely slow**. Once the MCR-ALS algorithm starts (see Perform MCR-ALS) there is no “Cancel” option, and the user cannot exit the loops until the modelling is done.

Once the unimodality mode is selected, the Components and Tolerance textboxes become active (**Figure 23**). By default, all components are listed to be constrained and the tolerance is set to “1”.

If only a certain number of components need to be constrained, their number needs to be entered in the “Components” textbox, separated by commas. For instance, if only components 1 and 3 need to constrained, enter “1, 3” or “1,3” (without spaces) in the textbox. Once the numbers entered, hit “Return” / “Enter” to manually approve the list.

> ⓘ **Warning/Information:** The component list for concentration and spectrum unimodality are independent. Thus, any component may be constrained by only in concentration or spectral profiles, or both (or neither).

> ☒ **Known bug:** The GUI can be fooled by changing the number of components after the Unimodality Constraint Matrix has been set. In this case, the “Component” textbox automatically updates to include all components but this information is NOT passed on automatically to the MCR-ALS algorithm UNTIL the “Component” textbox is manually edited. This is intentional, to make sure the user actively decides on the components to be constrained (even if all of them need to be constrained) before each MCR-ALS modelling run. However, if the user forgets to manually edit the “Components” textbox, MATLAB will show an error message in the Command Window, informing about a dimension mismatch.

> ☑ **Workaround:** Make sure to manually approve the components to be constrained by clicking in the “Components” textbox and hitting “Enter” / “Return” on the keyboard (if all components are to be constrained) or edit the list to include the correct components.

> ⓘ **Warning/Information:** There is a safety mechanism to stop the user entering a wrong input in the “Components” textbox. In case of improper input (such as letter instead of numbers), a warning dialog box appears and the list of components reverts to the default (i.e. including all components). This also deactivates the “Iterations” and “Convergence” textboxes and the “Perform MCR-ALS” button, so that MCR-ALS modelling cannot be started UNTIL the user manually approves of the content of the “Components” textbox (to make sure the correct constraints are applied to the correct components). Simply click in the “Components” textbox and hit “Enter” / “Return” on the keyboard, or edit the contents to contain the correct list of components.

A tolerance level for the unimodality constraint can be set in the "Tolerance" textbox. The tolerance level softens the unimodality constraint by allowing for a certain level of local departures (i.e. "noise"). For example, 1.3 means a maximum of 30% local departure is allowed. Thus, on the decreasing slopes of the main peak a particular point can have an increased value that corresponds to maximum 30% of the previous value before the unimodality constraint is applied (i.e. non-monotonous decrease or noise). Values between 1.0 (hardest constraint with no local departures allowed, only monotonous slopes) and 1.1 (10% departure allowed) are usual in systems with low to medium noise levels.

ⓘ **Warning/Information:** It is NOT possible to set different (individual) tolarence levels to each component, but it is possible to set different tolerance levels for the spectrum and concentration unimodality constraints.

ⓘ **Warning/Information:** There is no safety mechanism to stop the user entering a wrong input in the "Tolerance" textbox. In case of improper input (such as letter instead of numbers), the unimodality constraint will simply be ignored (not applied).

ⓘ **Warning/Information:** Unimodality constraints should NOT be applied for images and independent spectra as a general rule. Similarly, unimodality constraints are seldom useful for spectral profiles, as they more or less assume a single maximum (i.e. a single peak) in the spectrum. They are most useful in the concentration profiles of a reaction series (**Figure 25**).

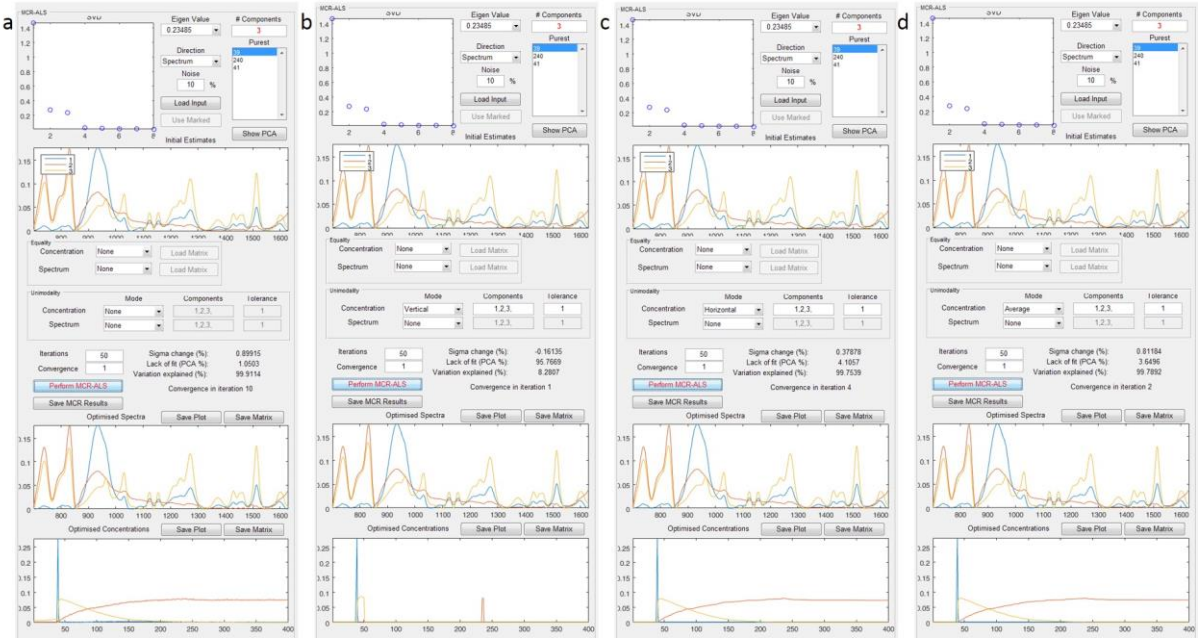

Figure 25. *The effect of Concentration Unimodality Constraints. The example uses the "demo_series.mat" file, with the following pre-processing parameters: spectral region cut to 700 – 1630 cm^{-1}; lambda=10,000 and p=0.001; No Normalisation and Savitzky-Golay smoothing with 1st order polynomial and Frame=5. 3 components are selected for MCR-ALS, using automatic initial estimate determination, spectral direction and 10% noise. Convergence limit is set to 1 for speed, and maximum number of iterations is kept at the default 50. Results are shown with no unimodality constraints (**a**), and concentration unimodality constraints (vertical, **b**, horizontal, **c**, and average, **d**) for all 3 Components. Note that apart from the vertical constraint (**b**), all results are very similar, achieved in much fewer iterations with the constraints.*

4.5.4. Closure

A default **spectrum equal length closure constraint** is hard-coded in the GUI with **no user control**. This means an equal spectral area normalisation for the MCR-ALS algorithm to avoid possible scale indeterminacies. This is the most common closure constraint for general spectroscopic data.

> ⓘ **Warning/Information:** This default spectrum equal length closure constraint is the reason why normalisation is generally not recommended for the MCR-ALS, in order to avoid double normalisations potentially skewing the data.

The spectrum equal length closure constraint **cannot be removed or modified** in the GUI without extensive modification of the source code. Thus, the GUI is not well suited for datasets that would require no closure or a different kind of closure constraints (such as accounting for mass balance equations or total concentration changes in reaction systems).

> ⓘ **Warning/Information:** The plan is to allow for different closure constraints in a future version of the GUI, including spectral equal height and no closure (relying on normalisations during pre-processing).

4.6. Number of Iterations and Convergence Criterion

The maximum number of iterations and the convergence criterion need to be supplied in the "Iterations" and "Convergence" textboxes of the MCR-ALS section of the GUI. The default values are "50" and "0.1", respectively. These parameters provide the user with controls to end the MCR-ALS modelling.

The number entered in the "Iterations" box sets the maximum number of iterations that will be performed if convergence or divergence does not happen before (see **Perform MCR-ALS**).

The convergence criterion should be provided as the percentage of change in the standard deviation of residuals between two consecutive iterations ("0.1 meaning that convergence is achieved when the change in the standard deviation of the residuals is lower or equal to 0.1% between two consecutive iterations). In other words, it sets a threshold for what the user considers as an acceptable fit for the model.

> ⓘ **Warning/Information:** Low convergence limits are generally harder to meet, especially in low number of iterations. For fine tuning the model, consider low convergence and high iterations. However, it can prove to be unnecessarily time consuming if the model does not change / improve significantly after a certain number of iterations. For a quick overview, either a high convergence limit or a low number of iterations can be used.

4.7. Perform MCR-ALS

After selecting the number of components (see **Number of Components**), the direction (**Direction**), the initial estimates (starting values, see **Initial Estimates**), setting the constraints (optional, see **Constraints**) and the end conditions for the modelling **(Number of Iterations and Convergence Criterion**), the MCR-ALS algorithm can be started by clicking on the "Perform MCR-ALS" button. The button has red text, to mark

that this step provides essential input for other parts of the GUI, even though they are optional (see **Reference Matching** and **Clustering and Segmentation**).

After clicking the button, the algorithm starts and the GUI updates at each iteration, until the convergence condition is met, the maximum number of iterations is reached (see **Number of Iterations and Convergence Criterion**), or divergence is encountered (see divergence stop **Warning/Information** below).

> ⓘ **Warning/Information:** Once the MCR-ALS algorithm starts (see Perform MCR-ALS) there is **no "Cancel" option**. Thus, the modelling cannot be stopped until the loops finish, i.e. convergence achieved, maximum number of iterations reached (see **Number of Iterations and Convergence Criterion**) or no improvement of fit is achieved in a certain number of consecutive iterations (see **divergence stop** above).

The "Sigma change (%):" text updates showing how the **sigma value** changes after each iteration (see **Number of Iterations and Convergence Criterion** for explanation).

The "Lack of fit (PCA %):" text updates, showing how the **lack of fit** (as expressed in percentage of the PCA reproduced matrix)

The "Variation explained (%):" text updates, showing the **percent of variation explained by the model (r^2 value of the model)**.

Below these texts, a **status display** continuously updates, informing about the current iteration number and whether the fit is improving or not.

At the bottom part of the MCR-ALS section of the GUI, two plots visualise the results of each iteration, showing the **Optimised Spectra** and the **Optimised Concentrations**, respectively. The colouring is kept for each component throughout the MCR-ALS section of the GUI (i.e. if Component 1 was blue in the Initial Estimates plot, it is blue in the Optimised Spectra and in the Concentration Plots as well)

> ⓘ **Warning/Information:** Pay attention to values shown in the text displays as well as in the Optimised Spectra and Optimised Concentration plots to see if the model can / needs to be improved (different constraints and starting conditions have more profound effects, while adjusting the number of iterations and convergence limit can help fine tuning the results or save time).

> ⓘ **Warning/Information:** Erroneous MCR-ALS models can be easy to discover if MATLAB shows error or warning messages in the Command Window (such as rank deficiencies, NaN values instead of numbers in the results, etc), but they can be hard to spot if the modelling appears to work correctly computationally. Be critical of the results and compare them with known features of the dataset (for instance resolved component maps matching the features in the white light image or not, etc.). The reason for suboptimal models (such as **divergence stop**) is usually worth investigating, as it can provide valuable information about the dataset.

When the iteration stops, this status display informs about the **end condition** that was met: a) **convergence** achieved (and in which iteration); b) **divergence** encountered; or c) the **maximum number of iterations** reached.

> ⓘ **Warning/Information: Divergence stop**. If the fit does not improve in a number of consecutive iterations, it can be considered divergence, instead of convergence and it can signal the end MCR-ALS modelling. The number of consecutive iterations that are tolerated without improving the fit before it is considered divergence and the MCR-ALS modelling is stopped is **hard coded in the GUI**. However, it is easy to modify, simply by changing the value of the "**idevmax**" variable in the code. The default value of "idevmax" is 10, and it is generally not recommended to have lower numbers, in order to allow the model to escape local minima and find the absolute minimum.

The MATLAB default workspace is updated with 4 variables at the end of the MCR-ALS modelling:

"COpt" contains the optimised concentration profiles, in Number of Spectra * Number of Components format.

"MCRMatrix" contains the input data for the MCR-ALS algorithm (i.e. the dataset as it was passed from the Pre-Processing part of the GUI, either processed or unprocessed), in Number of Spectra * Number of Spectral Variables (e.g. wavenumbers) format. It does NOT contain the Spectral Variables (Wavenumbers).

"SOpt" contains the optimised spectral profiles, in Number of Components * Number of Spectral Variables (e.g. wavenumbers) format. It does NOT contain the Spectral Variables (Wavenumbers).

"Wavenumbers" contains the Spectral Variables (e.g. wavenumbers, hence the name) in a single row.

> ⓘ **Warning/Information:** There is no information showing whether the results shown are obtained with the current values or not. For example, if a model was generated using a Vertical mode Unimodality Constraint for the Concentrations for all components, then the user removes this constraint, the results will NOT be deleted until the "Perform MCR-ALS" button is clicked again, which will overwrite the previous results. In short, **changing any MCR-ALS parameters does NOT delete the results of an already existing MCR-ALS model, until a new model is created**.

> ⓘ **Warning/Information:** In contrast to changing any MCR-ALS parameters, **changing the input data will erase ALL previous MCR-ALS models** from the GUI (although the controls for constraints and end criteria remain, making it easy to process the new input with the same parameters.) This is true both for loading a new dataset (3.1 **Loading the Data**) and for clicking on either the "Unprocessed for MCR-ALS" or the "Pre-Processed for MCR-ALS" buttons of the Pre-Processing section of the GUI (see **3.12. Passing the Data for** MCR-ALS). Variables in the MATLAB Workspace will remain, however, even though they are outdated (i.e. refer to a previous MCR-ALS model, not the currently loaded dataset)

> ⓘ **Warning/Information:** During MCR-ALS optimisation, not all parameters fit in the GUI display. Thus, the **MATLAB Command Window also shows the most important parameters** during each iteration and at the end of the modelling process.

Example MATLAB Command Window display during iteration:

"ITERATION 3

Sum of squares respect PCA reprod. = 24743903512.8037

Old sigma = 453.795 -----> New sigma = 443.148

Sigma respect experimental data = 444.1634

FIT IMPROVING

Sigma change(%) = 2.4026

Lack of fit in % (PCA) = 69.9875

Lack of fit in % (exp) = 70.0692

Percent of variance explained (r2) is 50.9031"

Example MATLAB Command Window display at the end of modelling:

"CONVERGENCE ACHIEVED

Lack of fit (%) at optimum = 69.9875(PCA) 70.0692(exp)

Percent of variance explained (r2) at optimum 50.9031

Plots are at optimum in iteration 3"

Finally, the Visualisation section of the GUI updates as well, including the folded concentration profiles (concentration maps) of the Components (see **Change Plot**).

An example is shown in **Figure 26**.

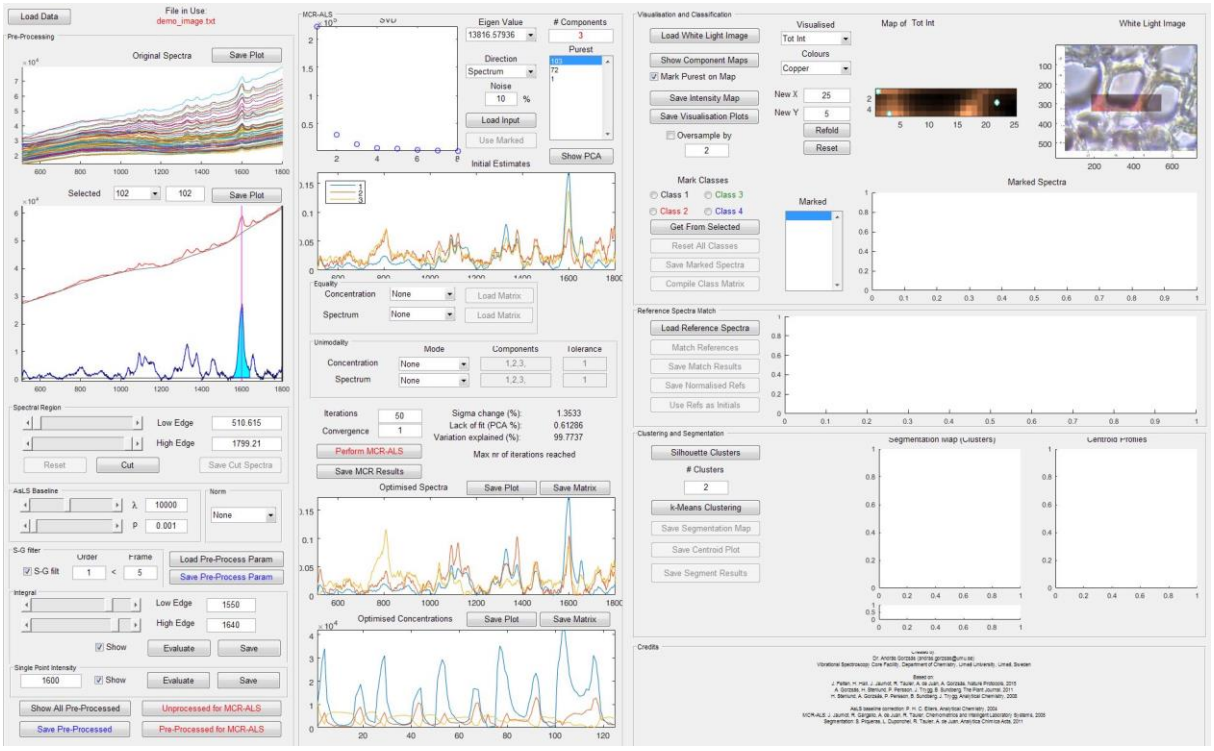

Figure 26. *The updated GUI after MCR-ALS modelling has finished. The example uses the "demo_image.txt" file, with the pre-processing parameters as in* ***Figure 16****, and in the same stage as if the analysis was continued from* ***Figure 19****. 3 Components with automatic initial estimate determination, spectral direction and 10% noise. No constraints are applied, the number of iterations is left at the default 50 but the convergence limit is changed to 1 for speed. The purest spectra estimates are marked as well.*

4.8. Optimised Spectra

The Optimised Spectra plot continuously updates with each iteration during MCR-ALS modelling. Once the modelling is finished, the plot remains static and the "Save Plot" and "Save Matrix" buttons become active. The colouring is kept for each component throughout the MCR-ALS section of the GUI (i.e. if Component 1 was blue in the Initial Estimates plot, it is blue in the Optimised Spectra and in the Concentration Plots as well)

4.8.1. Save the Optimised Spectra Plot

Click on the "Save Plot" button above the Optimised Spectra Plot to save it as an image file. A standard dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded, and suggests a name consisting of the dataset name, appended with the text "_MCRALS_SpectralPlot" at the end.

The default format is .pdf, but it can be changed to all major formats MATLAB allows for export (.jpg, .tif, .png, .gif, .eps).

ⓘ **Warning/Information: Mac and Windows compatibility and resolution problems of the saved plots may arise**. Test different formats to find the optimal solution to the current computer architecture.

ⓘ **Warning/Information:** As the text suggests, this button only saves the plot, NOT the data matrix with the resolved spectral profile values.

ⓘ **Warning/Information:** There is no information showing whether the results shown are obtained with the current values or not. For example, if a model was generated using a Vertical mode Unimodality Constraint for the Concentrations for all components, then the user removes this constraint, the results will NOT be deleted until the "Perform MCR-ALS" button is clicked again, which will overwrite the previous results. In short, **changing any MCR-ALS parameters does NOT delete the results of an already existing MCR-ALS model, until a new model is created**.

4.8.2. Save the Optimised Spectra Matrix

Click on the "Save Matrix" button above the Optimised Spectra Plot to save the MCR-ALS resolved (optimised) spectral profiles as a single file. A standard dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded, and suggests a name consisting of the dataset name, appended with the text "_MCRALS_SOptX" at the end, where X stands for the number of components the model used (so for a 3 component model, it is SOpt3).

The default format is .mat, but .xlsx can also be selected. Both formats will save the results in Number of Components + 1 * Number of Spectral Variables (e.g. wavenumbers) format. The first row contains the Spectral Variables (Wavenumbers), thereafter each row is one spectral profile, in the order of components (i.e. first component first, etc.)

ⓘ **Warning/Information:** This is different from the "SOpt" variable in the MATLAB Workspace, which does NOT contain the Spectral Variables (as they are stored in the separate "Wavenumber" variable).

ⓘ **Warning/Information:** As the text suggests, this button only saves the data matrix with the resolved spectral profile values, NOT the plot.

ⓘ **Warning/Information:** There is no information showing whether the results shown are obtained with the current values or not. For example, if a model was generated using a Vertical mode Unimodality Constraint for the Concentrations for all components, then the user removes this constraint, the results will NOT be deleted until the "Perform MCR-ALS" button is clicked again, which will overwrite the previous results. In short, **changing any MCR-ALS parameters does NOT delete the results of an already existing MCR-ALS model, until a new model is created**.

4.9. Optimised Concentrations

The Optimised Concentration plot continuously updates with each iteration during MCR-ALS modelling. Once the modelling is finished, the plot remains static and the "Save Plot" and "Save Matrix" buttons become active. The colouring is kept for each component throughout the MCR-ALS section of the GUI (i.e. if Component 1 was blue in the Initial Estimates plot, it is blue in the Optimised Spectra and in the Concentration Plots as well)

4.9.1. Save the Optimised Concentration Plot

Click on the "Save Plot" button above the Optimised Concentration Plot to save it as an image file. A standard dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded, and suggests a name consisting of the dataset name, appended with the text "_MCRALS_ConcPlot" at the end.

The default format is .pdf, but it can be changed to all major formats MATLAB allows for export (.jpg, .tif, .png, .gif, .eps).

ⓘ **Warning/Information: Mac and Windows compatibility and resolution problems of the saved plots may arise**. Test different formats to find the optimal solution to the current computer architecture.

ⓘ **Warning/Information:** As the text suggests, this button only saves the plot, NOT the data matrix with the resolved concentration profile values.

ⓘ **Warning/Information:** There is no information showing whether the results shown are obtained with the current values or not. For example, if a model was generated using a Vertical mode Unimodality Constraint for the Concentrations for all components, then the user removes this constraint, the results will NOT be deleted until the "Perform MCR-ALS" button is clicked again, which will overwrite the previous results. In short, **changing any MCR-ALS parameters does NOT delete the results of an already existing MCR-ALS model, until a new model is created**.

4.9.2. Save the Optimised Concentration Matrix

Click on the "Save Matrix" button above the Optimised Concentration Plot to save the MCR-ALS resolved (optimised) concentration profiles as a single file. A standard dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded, and suggests a name consisting of the dataset name, appended with the text "_MCRALS_COptX" at the end, where X stands for the number of components the model used (so for a 3 component model, it is COpt3).

The default format is .mat, but .xlsx can also be selected. Both formats will save the results in Number of Spectra * Number of Components format.

ⓘ **Warning/Information:** This is the same as the "COpt" variable in the MATLAB Workspace.

ⓘ **Warning/Information:** As the text suggests, this button only saves the data matrix with the resolved concentration profile values, NOT the plot.

ⓘ **Warning/Information:** There is no information showing whether the results shown are obtained with the current values or not. For example, if a model was generated using a Vertical mode Unimodality Constraint for the Concentrations for all components, then the user removes this constraint, the results will NOT be deleted until the "Perform MCR-ALS" button is clicked again, which will overwrite the previous results. In short, **changing any MCR-ALS parameters does NOT delete the results of an already existing MCR-ALS model, until a new model is created**.

4.9.3. Show the Component Maps

The MCR-ALS optimised concentration profiles can be refolded into a map, in addition to being shown in the Optimised Concentrations Plot. This is most valuable for image datasets, but is not technically limited to those (results from independent spectra or even spectral series datasets can benefit from this kind of visualisation).

Simply click on the "Show Component Maps" button.

ⓘ **Warning/Information:** The "Show Component Maps" button is located in the Visualisation section of the GUI, NOT in the MCR-ALS section, as strictly speaking this is NOT a core part of the MCR-ALS algorithm, but is a visualisation option

The results are shown in a new popup window, that contains two columns and as many rows as the number of Components. The first column shows the refolded Concentration profiles, each with a title "ComponentX Map", where X stands for the Component number. The second column shows the resolved spectral profiles of the Components, each with a title "ComponentX Spectrum", where X stands for the Component number.

The refolded concentration profiles are shown in the same colours as the Intensity maps of the Visualisation section of the GUI (see **Change Colour**), while the Spectral profiles have the same colours as in the MCR-ALS section of the GUI (i.e. if Component 1 was blue in the Initial Estimates plot, it is blue in the Optimised Spectra and Concentration Plots as well as in the Spectral profile of this popup window).

ⓘ **Warning/Information:** There is no information showing whether the results shown are obtained with the current values or not. For example, if a model was generated using a Vertical mode Unimodality Constraint for the Concentrations for all components, then the user removes this constraint, the results will NOT be deleted until the "Perform MCR-ALS" button is clicked again, which will overwrite the previous results. In short, **changing any MCR-ALS parameters does NOT delete the results of an already existing MCR-ALS model, until a new model is created**.

ⓘ **Warning/Information:** Changing the colouring in the Visualisation section of the GUI (see **Change Colour**), does **NOT** change the colouring in an already existing popup window. This is NOT a bug, but an intentional feature, since clicking the "Show Component Maps" multiple times opens up a new popup window instead of overwriting the existing one. This allows for testing and directly comparing different colour schemes for visualisation side by side.

ⓘ **Warning/Information:** The “Mark Purest on Map” checkbox (see **Mark Purest on Map**) does **NOT** add Markers to the Component Maps in the new popup window, only in the Visualisation section of the main GUI window. Thus, if plots need to be saved with the Markers, they should be saved using the “Save Visualisation Plots” button (**Save Visualisation Plots**).

☒ **Known bug:** When the “Show Component Maps” button is clicked, all markers for the purest spectra are removed from the main GUI window visualisation plot as well (Class Markers are not affected)

☑ **Workaround:** Untick the “Mark Purest on Map” checkbox and tick it again to bring back the markers.

ⓘ **Warning/Information:** Refolding (see **Change Dimensions (Refold)**) does **NOT** change the dimensions of to the Component Maps in the new popup window. Thus, if plots need to be saved with the changed dimensions, they should be saved one by one, using the “Save Intensity Map” button (**Save Intensity Map**).

ⓘ **Warning/Information:** Oversampling (see **Oversample**) does **NOT** change the look of the Component Maps in the new popup window. Thus, if plots need to be saved with the mathematically increased spatial resolution, they should be saved one by one, using the controls of the popup window opened up when oversampling is performed (see **Oversample**).

To **save** the plots in the popup window that appears after the “Show Component Maps” button is clicked, use the menu or the icons of the popup window itself (see **Plots**).

ⓘ **Warning/Information:** Each visualised Component concentration map can be visualised individually by selecting it from the “Visualised” drop-down list (see **Change Plot**), and **saved** on its own using the “Save Intensity Map” button (see **Save Intensity Map**). In this case, **all Markers** (Class and Purest) will be saved as well.

ⓘ **Warning/Information:** All visualised Component concentration map can be saved in a single click, together with all other generated maps (including Total Intensity, Integral and Single Point intensity maps, if any) using the “Save Visualisation Plots” button (see **Save Visualisation Plots**).

☒ **Known bug:** When the “Save Visualisation Plots” button is clicked, the GUI cycles through all plots. This loses all Class Markers (but not the Purest Markers, and not the Marked Spectra).

☑ **Workaround:** Use the “Save Intensity Map” button instead of the “Save Visualisation Plots” button, and save each plot individually (one by one) to keep all Markers in the plots. Change the “Visualised” drop-down list to another plot to reset the maps and show the Class Markers again.

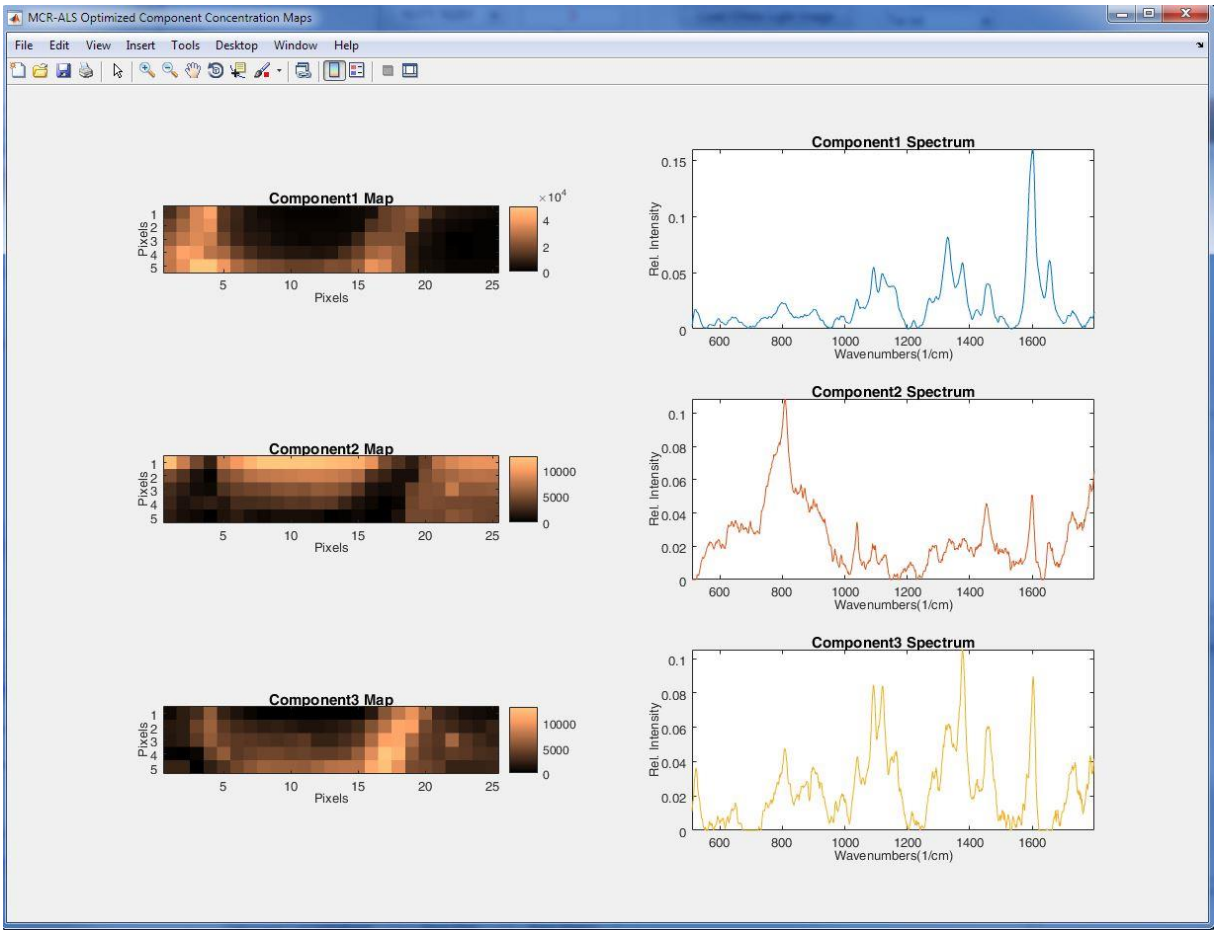

Figure 27. *The popup window showing the MCR-ALS resolved Components, after MCR-ALS modelling finished in* ***Figure 26****. Note that the window has its own set of menus and icons (allowing for e.g. saving the plots, zooming in, etc.). The Component spectra colours are matching the colours in the MCR-ALS section of the GUI (i.e. Component 1 is blue, Component 2 is red and Component 3 is yellow, see* ***Figure 26****) and the Component maps are using the Colour scheme set by the "Colours" drop-down list in the Visualisation section of the GUI (see* **Change Colour***).*

4.10. Save the MCR-ALS Results

There are several ways of saving the MCR-ALS results.

For the generated plots, the following options are available.

a. Optimised Concentration and Spectral Plots can be saved by their dedicated buttons in the MCR-ALS section of the GUI (see **Save the Optimised Spectra Plot** and **Save the Optimised Concentration Plot**).
b. Individual Concentration Maps can be saved using the **Save Intensity Map** and **Save Visualisation Plots** options.
c. All Component Maps and their Spectral Profiles can be saved using the menu / icon of the popup window that appears after clicking on the "Show Component Maps" button (see **Show the Component Maps**).

To save the generated data (i.e. the matrices, not the plots), the following options are available:

d. Optimised Concentration and Spectral profile matrices can be saved by their dedicated buttons in the MCR-ALS section of the GUI (see **Save the Optimised Spectra Matrix** and **Save the Optimised Concentration Matrix**).
e. Variables in the MATLAB Workspace (see **Perform MCR-ALS**) can be saved individually by right-clicking on their name in the Workspace pane of the main MATLAB window and selecting "Save As…" in the opening context-sensitive menu.
f. Variables in the MATLAB Workspace (see **Perform MCR-ALS**) can be opened by double-clicking on their name in the Workspace pane of the main MATLAB window, and their content copied and pasted into third party software (such as Microsoft Excel).

However, the above options do NOT save the end conditions of the MCR-ALS modelling.

In order to do that, click on the "Save MCR Results" button. A standard save dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded, and suggests a name consisting of the dataset name, appended with the text "_MCRResultsX" at the end, where X stands for the number of components the model used (so for a 3 component model, it is MCRResult3).

The only available format is .mat, and a single file is generated that can be opened/loaded in MATLAB and includes the following 7 variables:

"COpt" contains the optimised concentration profiles, in Number of Spectra * Number of Components format. This is the same as the variable loaded in the MATLAB Workspace at the end of the MCR-ALS modelling (see **Perform MCR-ALS**).

"Ending" contains a single number referring to how the MCR-ALS optimisation ended. "1" means convergence was achieved, "2" means divergence was encountered, "3" means the maximum number of iterations was reached (to see at which number of iteration the ending happened, see "Iteration").

"Iteration" contains a single number, referring to the number of iteration when the MCR-ALS stopped (for the reason of stopping, see "Ending").

"R2Opt" contains a single number, referring to the percent of variation explained, in fraction format (i.e. 0.9975 means 99.75%)

"SDOpt" contains a vector with two values, both describing the lack of fit in relative standard deviation units. The first number is with respect to the PCA reconstructed dataset (i.e. the same number as shown in the GUI as "Lack of fit (PCA%):"), and the second is with respect to the original data matrix.

"Sigma" contains a single number, describing the last sigma value change during iterations with respect to the experimental data, normalised (It is calculated as sqrt(sum(sum(resn.*resn))/(nrow*ncol)), where "sqrt" and "sum" are the square root and sum functions of MATLAB, "resn" are the normalised residuals of the MCR-ALS optimisation, "nrow" is the number of rows (number of spectra) in the dataset, and "ncol" is the number of columns (number of spectral variables, e.g. wavenumbers) in the dataset).

“SOpt” contains the optimised spectral profiles, in Number of Components * Number of Spectral Variables (e.g. wavenumbers) format. It does NOT contain the Spectral Variables (Wavenumbers). This is the same as the variable loaded in the MATLAB Workspace at the end of the MCR-ALS modelling (see **Perform MCR-ALS**).

ⓘ **Warning/Information:** Importantly, the variables “**Wavenumbers**” and “**MCRMatrix**” are **NOT included** in the saved .mat file. This is because these variables are not the results of the MCR-ALS modelling, but its input. They can be saved using the “Save Pre-Processed” button of the Pre-Processing section of the GUI (**3.11. Saving the Pre-Processed** Data), or by saving them directly from the MATLAB Workspace (see options **e** and **f** above).

5. Visualisation and Classification

The Visualisation section of the GUI provides tools to visualise the results and customise the visualisation. This can be helpful for reporting and / or verifying the results, but it is entirely optional. Some visualisation controls are more suited for images (such as the possibility to load a corresponding White Light Image) and some are more suited for a spectral (e.g. reaction) series or a set of independent spectra (such as the possibility to change the dimensions of the visualisation plots), but most of them are generic (such as changing the colour scheme of the plots)

Classification controls are also located in this part of the GUI, allowing manually marking spectra in the dataset and assigning them to a class. These Class Marked spectra can be used as Initial Estimates, saved independently (i.e. extracting spectra from a dataset), or compiled into a single Microsoft Excel .xlsx file that can be directly processed by other software (such as Umetrics SIMCA-P), for e.g. discriminant analysis (since the class variable follows the data).

> ⓘ **Warning/Information:** The number of classes to be assigned is limited to 5 at present. If no class is selected via the four "Mark Classes" radio buttons, it is considered class 0.

5.1. White Light Image

A White Light Image is automatically loaded with the dataset if it is located in the same folder as the dataset itself, and has the exact same name as the dataset, with .jpg extension.

When it is not loaded automatically, a White Light Image can be manually loaded by clicking the "Load White Light Image" button in the Visualisation section of the GUI. This opens a standard dialog window, where the file to be loaded can be selected. The default option lists .jpg files, but by choosing the "All Files" option, all image formats supported by MATLAB can be loaded.

5.2. Visualisation Controls

The Visualisation section of the GUI contains at least one concentration map: the total intensity plot, that displays the total spectral intensity (without any normalisation) for every spectrum in the dataset. As soon as new concentration maps are generated (integral3.7. Integration, or single point intensity, see **3.8. Single Point Intensity** Evaluation, MCR-ALS resolved concentration profiles of the components, see **Perform MCR-ALS** and **Show the Component Maps**, segmentation map, see **K-Means Clustering**), they are added to the list of plots, and the latest added / selected is displayed in the GUI with a title above the plot indicating which one is currently being displayed.

5.2.1. Change Plot

To change the currently displayed map, use the "Visualised" drop-down list in the Visualisation section of the GUI.

This list always starts with "Tot Int", which refers to the non-normalised total intensity plot of the dataset.

The second item in the list is the single point intensity, if there is any (even if single point intensity evaluation was performed later than any of the other plots were generated), labelled as "SPoint @XXXXNN", where XXXX is the spectral variable (wavenumber) at which the single point intensity was evaluated and NN is the kind of normalisation performed (see **3.8. Single Point Intensity** Evaluation).

The next item in the list is the Integral plot, if there is any (irrespective in which order the integral plot was generated, it always precedes the Component maps and follows the single point intensity map, if there are any of those present), labelled as "Int @XXXX-YYYYNN", where XXXX-YYYY denotes the spectral variables (wavenumbers) between which the intensity was evaluated (integral region) and NN is the kind of normalisation performed (see **3.7.** Integration).

The next items in the list are the folded concentration profiles (maps) of the MCR-ALS resolved Components, if any (irrespective when these were generated). There are as many of these as there are components, with labels "Comp X", where X denotes the component number (from 1 to the total number of components).

The last item in the list is the segmentation map, if there is any, labelled as "KMeansXClusters", where X denotes the number of clusters used for k-means clustering.

ⓘ **Warning/Information:** The plots and the "Visualised" drop-down list updates as soon as there is new input for it, but NOT otherwise. Old values are kept as long as they are meaningful (single point intensity and integral evaluations), but if they get obsolete (component and segmentation maps), they are removed. Examples:

Step 1: Single Point Intensity evaluated at 1600 cm^{-1}, with no normalisation, MCR-ALS performed with 3 components, k-means clustering performed with 3 clusters. The "Visualised" drop-down list will contain the following items: "Tot Int", "SPoint @1600NN", "Comp1", "Comp2", "Comp3", "KMeans3Clusters".

Step2: Total Area normalisation is selected in the Pre-Processing section of the GUI, and the data is passed on to MCR-ALS, using the "Pre-Processed for MCR-ALS" button. The "Visualised" drop-down list will contain only the following items: "Tot Int" (always present) and "SPoint @1600NN" (as the single point intensity has NOT been re-evaluated with the total area normalisation, i.e. the "Evaluate" button in the Single Point Intensity box of the Pre-Processing section of the GUI was not clicked after total area normalisation was selected in the "Norm" drop-down list in the Pre-Processing section of the GUI). The maps "Comp1", "Comp2" and "Comp3" are deleted and they are removed from the "Visualised" list as the MCR-ALS results are cleared and a new MCR-ALS model must be generated. Similarly, the "KMeans3Clusters" plot is also removed, since it is based on the MCR-ALS results that are now deleted. (However, the segmentation map is NOT deleted from the Clustering and Segmentation section of the GUI, only from the Visualisation section).

Step3: A new MCR-ALS model is generated using 4 components. The "Visualised" drop-down list will contain the following items now: "Tot Int" (always present), "SPoint @1600NN" (still no new single point evaluation performed to reflect the total area normalisation), "Comp1", "Comp2", "Comp3", and "Comp4". It will not contain a segmentation map, as there is still no new input for it (i.e. no new segmentation map is generated. The Clustering and Segmentation section of the GUI still displays the previous map, but it was generated with 3 clusters based on a 3-component

MCR-ALS model of the non-normalised dataset, as opposed to the current 4-component model of the total area normalised dataset).

Step 4: Integration is performed in the region 1550-1640 cm^{-1}. The "Visualised" drop-down list will now contain the following items: "Tot Int" (always present), "SPoint @1600NN" (still no new single point evaluation performed to reflect the total area normalisation), "Int @1550-1640TA" (i.e. the newly generated integrals, in the 1550-1640 region, using total area normalisation ("TA")), "Comp1", "Comp2", "Comp3", and "Comp4".

ⓘ **Warning/Information:** There can only be maximum one of any kind of plot at any given time in the "Visualisation" list:

One total intensity plot (non-normalised). It is always present.

One single point intensity plot. If a new one is generated, the old one is overwritten.

One integral plot. If a new one is generated, the old one is overwritten.

One set of Component maps (with as many individual maps as there are number of components). If the MCR-ALS model is deleted, they are deleted. If a new MCR-ALS modelling is performed, a new set is generated and the old set is overwritten.

One segmentation map. If it gets obsolete (the MCR-ALS model is deleted or a new MCR-ALS modelling is performed), it is deleted. If a new segmentation map is generated, the old one is overwritten.

If the Purest Markers are shown (the "Mark Purest on Map" checkbox is ticked), they are retained when the map is changed in the "Visualised" drop-down list.

☒ **Known bug:** When the "Show Component Maps" button is clicked, all markers for the purest spectra are removed from all visualised plots (Class Markers are not affected)

☑ **Workaround:** Untick the "Mark Purest on Map" checkbox and tick it again to bring back the markers.

If there are Class Markers shown in the plot, they are retained when the map is changed in the "Visualised" drop-down list.

☒ **Known bug:** When the "Save Visualisation Plots" button is clicked, the GUI cycles through all plots. This loses all Class Markers (but not the Purest Markers, and not the Marked Spectra).

☑ **Workaround:** Use the "Save Intensity Map" button instead of the "Save Visualisation Plots" button, and save each plot individually (one by one) to keep all Markers in the plots. Change the "Visualised" drop-down list to another plot to reset the maps and show the Class Markers again.

5.2.2. Change Colour

The colour of the visualised plots in the main GUI window and subsequently opened popup windows (see **Show the Component Maps**, **Change Dimensions (Refold)** and Oversample) can be changed by the "Colours" drop-down list of the Visualisation section of the GUI.

ⓘ **Warning/Information:** Popup windows opened BEFORE the colour change are NOT affected.

ⓘ **Warning/Information:** Changing the colour also affects the segmentation map shown in the "Clustering and Segmentation" section of the GUI, since this plot is also part of the visualised plots (can be included the "Visualised" drop-down list)

The default colouring scheme is "Copper" (one of the built-in options of the "colormap" function of MATLAB), as this provides a good range from dark to light and minimum interference with other potential colours in the maps (including Class Markers and Purest Markers) and no confusion with the Component colours (which use the default MATLAB colour sequence) or centroid profile colours (which are matching the Component colours).

Other colour schemes are all built-in options of the "colormap" function of MATLAB (see MATLAB documentation and **Figure 28**) and are as follows: Jet, HSV, Hot, Cool, Spring, Summer, Autumn, Winter, Gray, Bone and Pink.

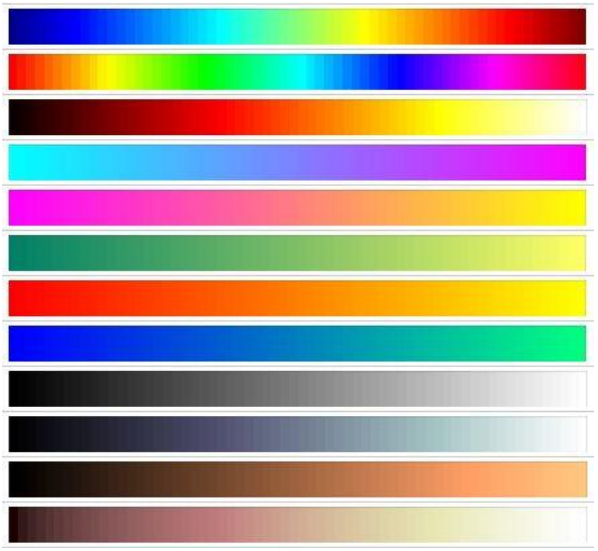

Figure 28. *The values of the "colormap" function of MATLAB used by the "Colours" drop-down list. From top to bottom: Jet, HSV, Hot, Cool, Spring, Summer, Autumn, Winter, Gray, Bone, Copper (**default**) and Pink. See the MATLAB documentation for more details.*

5.2.3. Change Dimensions (Refold)

The dimension for the dataset needs to be specified upon loading (see **3.1 Loading the** Data). For image data, the X and Y dimensions are set by the image dimensions and should not be changed. Thus, this option is NOT recommended for images.

For spectral series and sets of independent spectra, however, changing the dimensions can help the visualisation and interpretation. The default option upon loading (if no other dimensions are specified) is

Y=1 and X=total number of spectra, generating a single row in the visualisation plots. This can be suboptimal when a large number of spectra are included in the dataset. Alternatively, if there is a pattern in the data, this can be retained during visualisation.

As an example, consider a batch experiment, where each batch contains 5 replicates (spectra). Batch 1-2 experiments are performed at pH=3 (one experimental variable), Batches 3-4 are at pH=5, batches 5-6 at pH=7 and batches 7-8 at pH=9. This means 8 batches in total, with 5 spectra in each = 40 spectra. If the data is visualised as 40*1 spectra, it can be hard to see (and mark) individual spectra in the set. However, if the dataset is refolded as 8*5, each batch will have its own row in the visualisation plots. Thus, patters are easier to notice (also beneficial for segmentation) and correlate to features of the dataset, and spectra are easier to mark. Alternatively, a 10*4 refolding can be helpful as well, in which case every spectra of a certain pH will be shown in one row, with the first half (5 spectra) from the first batch at that pH and the second half (5 spectra) from the second batch at that pH.

The dimension change is entirely optional and the dimensions are left to the user to decide.

Type in the new X and Y dimensions in the respective textboxes in the Visualisation section of the GUI and click the “Refold” button.

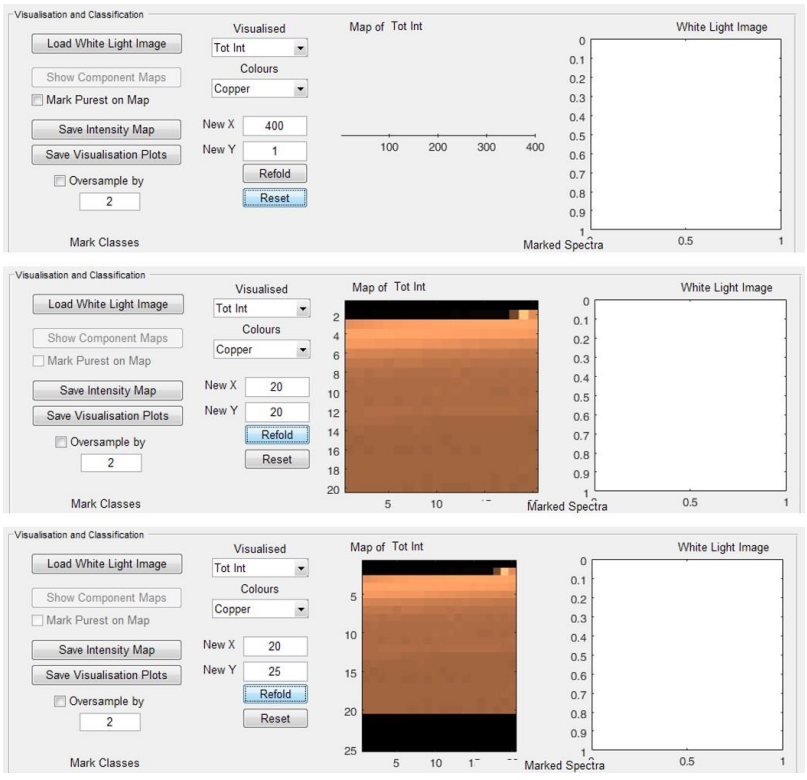

Figure 29. *Changing the dimensions of the dataset for better visualisation. The example uses the “demo_series.mat” file, and its total intensity plot directly after loading. On top: Accepting the default suggestion of the GUI upon loading (X=400, Y=1, i.e. all spectra in a single row) creates a plot that is impossible to see. Middle: changing the dimensions to 20 x 20 makes visualisation easier and reveals that the total intensity of the first 17 spectra are considerably lower than the rest. Bottom: the dimensions can be set to include more data points than the original dataset, but not less. In this example the dimension 20x25 is specified, which equals to 500 data points, although the dataset only contains 400 in total. Thus, the matrix for visualisation is padded with 100 NaNs (not-a-number values) at the end.*

To revert to the original dimensions (supplied during loading), click the “Reset” button.

ⓘ **Warning/Information:** The new dimensions **CANNOT be lower** than the total number of spectra (i.e. a dataset with 40 spectra cannot be refolded into a 4*4 dimension, as that only contains 16 elements). In this case, the GUI will simply ignore the “Refold” button. However, the “Reset” button works, so accidentally wrong inputs can be easily reset.

ⓘ **Warning/Information:** The new dimensions are **allowed to be higher** than the total number of spectra (i.e. a dataset with 40 spectra is allowed to be refolded into a 7*6 dimension, which should contain 42 spectra). In this case, the GUI will add the extra points at the end of the plot with NaN values. (So in the above example, the plot will contain 6 rows and 7 columns, with the last two points of the last row being blank, and all the rest containing values). This is ONLY done for the currently active visualisation plot, i.e. strictly a display/visualisation tool: NO extra data (spectra) will be added to the dataset (i.e. no new rows in the data matrix).

ⓘ **Warning/Information:** Changing the “Visualised” drop-down list will reset the dimension to the original ones, but it does NOT reset the values of the “New X” and “New Y” textboxes, so a new refolding can be easily done for another visualisation plot by simply clicking the “Refold” button.

ⓘ **Warning/Information:** Changing the “Colours” drop-down list does NOT reset the dimension, so a new colour scheme can be tested without having to refold again.

ⓘ **Warning/Information:** If there are Purest Markers in the plot, their position will be recalculated to match the new dimensions, i.e. they are automatically updated.

ⓘ **Warning/Information:** Classes CANNOT be marked in refolded plots.

☒ **Known bug:** If there are marked spectra, refolding loses all Class Markers (but not the Marked Spectra).

☑ **Workaround:** Change the “Visualised” drop-down list to another plot. This will reset the maps and show the Class Markers. There is currently no way in the GUI to save the Class Markers in refolded maps. If that is required, supplied the desired dimensions already when loading the dataset (if possible), instead of using the refolding option of the GUI.

ⓘ **Warning/Information:** The “Save Intensity Map” button saves the current plot with the current dimensions (i.e. the plot with the refolded dimensions can be saved).

☒ **Known bug:** The “Save Visualisation Plots” button resets the dimensions of the plots to the original one while the GUI cycles through all plots. This loses all Class Markers (but not the Purest Markers, and not the Marked Spectra) if there were any and plots will be saved with their original dimensions only (no refolded plot is saved).

☑ **Workaround:** Use the “Save Intensity Map” button instead of the “Save Visualisation Plots” button, and save each plot individually (one by one) to keep the refolded dimensions. Once the plot is saved, change the “Visualised” drop-down list to another plot to reset the maps and show the Class Markers again.

5.2.4. Oversample

The lateral resolution of the generated visualisation plots can be improved virtually using oversampling. Select the plot for oversampling by the “Visualised” drop-down list (see **Change Plot**), enter the oversampling factor in the textbox and tick the “Oversample by” textbox. A new popup window appears that contains 2 plots: the original (top) plot as shown in the GUI, and its oversampled version (bottom). The oversampling factor determines the new number of points (pixels) in the oversampled plot, multiplying BOTH the X and Y points with this number and calculating the values of each point based on the original data points. Example: the demo_image.txt dataset contains a 25*5 image. Using an oversampling factor of 2, the new image will consists of 50*10 points (pixels).

ⓘ **Warning/Information:** The oversampling range is limited to 2 – 10. Numbers outside this range will be reset to the default “2”.

ⓘ **Warning/Information:** The oversampling only changes the appearance of the currently active plot, it does NOT change any data matrix, and it cannot be used on more than one plot at a time (i.e. only the currently active visualised plot can be oversampled).

ⓘ **Warning/Information:** Oversampling can generate artefacts, especially close to the right and bottom edges of the images, and in particular when intensities in the original plot are high in those regions (as there is no data beyond these points to be used for value estimations). Be aware of this limitation of the algorithm.

ⓘ **Warning/Information:** New popup windows opened by oversampling do NOT overwrite already existing windows, thus several oversampling options can be compared side by side.

ⓘ **Warning/Information:** To perform a new oversampling, click in the oversampling textbox and either change the oversampling value (if a new oversampling factor is to be tested on the same plot), or simply hit “Enter”/”Return” on the keyboard (to keep the same oversampling factor when generating a new plot, after e.g. changing the “Visualised” drop-down list to a new plot, or changing the “Colours” drop-down list to a new colour scheme), as long as the “Oversample by” checkbox is ticked.

ⓘ **Warning/Information:** Naturally, no Markers (Purest or Class) are included in the oversampling plots, as their position cannot be determined in the virtually enhanced new plots.

ⓘ **Warning/Information:** Unticking the “Oversample by” checkbox does NOT close already opened popup windows. They need to be closed manually.

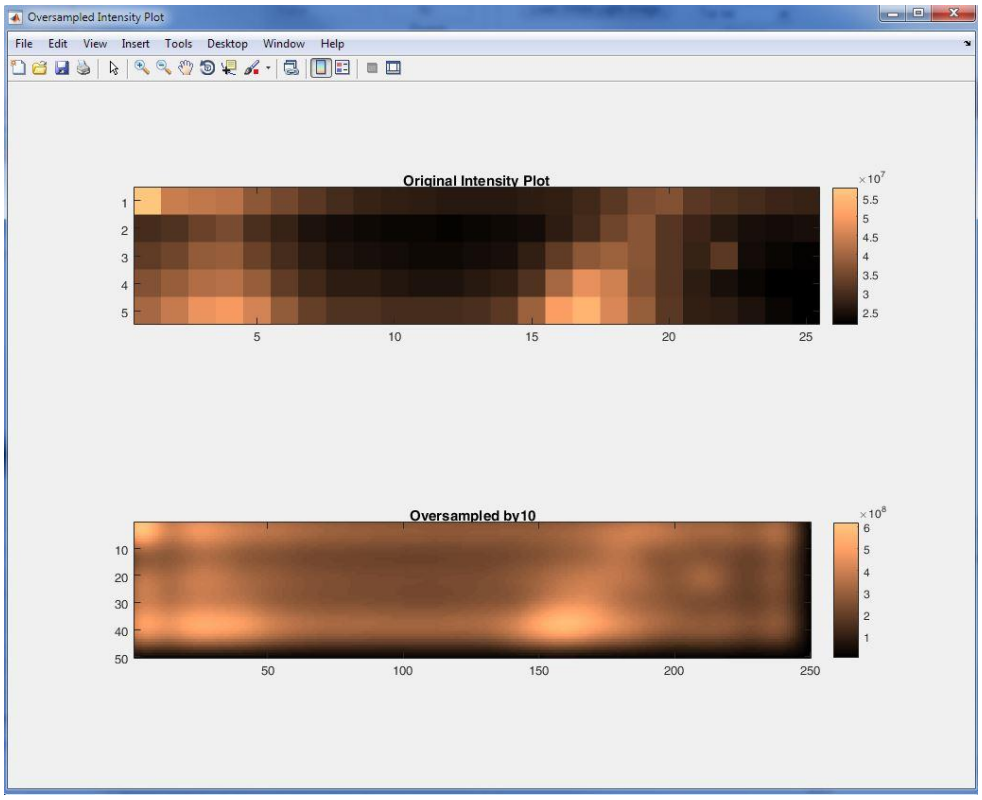

Figure 30. *The Oversampling popup window, using the total intensity plot generated directly after the "demo_image.txt" file is loaded, using an oversampling factor of 10. Top: no oversampling, Bottom: oversampled by a factor of 10. Note the number of pixels in the image (and the resulting smoothness) and the artefacts generated at the bottom and at the right of the oversampled map (drop of intensity). Note also that the window has its own set of menus and icons (allowing for e.g. saving the plots).*

5.2.5. Mark Purest on Map

The location of the spectra shown in the "Purest" listbox in the MCR-ALS section of the GUI can be visualised on the plots by ticking the "Mark Purest on Map" checkbox. This will add white diamond markers with cyan outlines to the plots in the Visualisation section of the GUI (see e.g. **Figures 26** and **31**).

ⓘ **Warning/Information:** Naturally, only those elements of the "Purest" listbox can be shown in the plot that are members of the dataset (see **Automatic Estimation and Mark Purest on Map** and **Use Marked Spectra**). Externally supplied Initial Estimates, either manually loaded (see **Load Input**) ones or those imported from Reference Spectra (see **Use Reference Spectra**) cannot be visualised and can result in bugs.

ⓘ **Warning/Information:** The Purest Markers on the visualisation plot are **all white (not colour coded) and have no labels** to indicate which Component they belong to (see **Figure 19** for an example). This information has to be deduced from their position.

ⓘ **Warning/Information:** The Markers remain even if the Visualised plot is changed using the "Visualised" drop-down list (see **Change Plot**) or a different colour coding is selected via the "Colours" drop-down list (see **Change Colour**), as long as the checkbox is ticked.

ⓘ **Warning/Information:** The position of the Markers automatically update when the image is refolded / reset (see **Change Dimensions (Refold)**).

ⓘ **Warning/Information:** The Markers automatically update when the number of Components is altered (see **Number of Components**).

☒ **Known bug:** The Markers disappear from the Visualisation plot if a new Integral (3.7. **Integration**) or Single Point Intensity evaluation is performed (**3.8. Single Point Intensity Evaluation**).

☑ **Workaround:** Refresh the plot by either of the following methods: a) untick and tick the "Mark Purest on Map" checkbox; b) Activate the "Visualised" drop-down list (even if clicked on the very same value, it triggers the update of the plot, it does not have to be changed to a new value / new plot selected) (see **Change Plot**); c) Activate the "Colours" drop-down list (even if clicked on the very same value, it triggers the update of the plot, it does not have to be changed to a new value / new plot selected) (see **Change Colour**); d) "Refold" or "Reset" the plot (see **Change Dimensions (Refold)**); d) Change the number of Components to a new value than change back to the desired one (see **Number of Components**).

ⓘ **Warning/Information:** Markers do NOT appear in the plots of pop-up windows (such as the plots displayed by the "Show Component Maps" button, see **Show Component Maps**)

☒ **Known bug:** When the "Show Component Maps" button is clicked, all markers for the purest spectra are removed from all visualised plots (Class Markers are not affected)

☑ **Workaround:** Untick the "Mark Purest on Map" checkbox and tick it again to bring back the markers.

ⓘ **Warning/Information:** If the Markers are visible in the Visualisation section of the GUI, they will be visible in the Saved plots (see **Save Intensity Map** and **Save Visualisation Plots**).

☒ **Known bug:** In certain cases the "Mark Purest on Map" checkbox can get deactivated unintentionally.

☑ **Workaround:** To activate the checkbox again, click in the "# Components" textbox of the MCR-ALS section of the GUI and hit "Enter"/"Return" with the "Spectrum" option chosen in the "Direction" drop-down list of the MCR-ALS section of the GUI. This will re-activate the checkbox.

5.3. Show Component Maps

See **Show the Component Maps** and **Change Plot**.

5.4. Save Intensity Map

Click on the “Save Intensity Map” button in the Visualisation section of the GUI to save the currently active visualised plot exactly as it is shown in the GUI. A standard save dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded, and suggests a name consisting of the dataset name, appended with the text “_Visualised_Title” at the end, where Title stands for the title shown above the plot and in the “Visualised” drop-down list.

> ⓘ **Warning/Information:** If there are Markers (Purest or Class, see **Mark Purest on Map** and **Mark Classes**) are visible in the Visualisation section of the GUI, they will also be visible in the saved plot.

The default save format is .pdf, but .jpg, .tif, .png, .gif or .eps formats can also be selected.

> ⓘ **Warning/Information: Mac and Windows compatibility and resolution problems of the saved plots may arise**. Test different formats to find the optimal solution to the current computer architecture.

5.5. Save Visualisation Plots

Click on the “Save Visualisation Plots” button in the Visualisation section of the GUI to save all plots in the Visualisation section (i.e. all plots listed in the “Visualised” drop-down list). A standard save dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded, and suggests the same name as the original dataset with .pdf extension, although jpg, .tif, .png, .gif or .eps formats can also be selected. EACH PLOT will be saved as an individual file with the selected file format/extension, and the name supplied in the dialog box is used only as a prefix: each file name will be appended by the Title of the plot as shown in the GUI and in the “Visualised” drop-down list (e.g. if “demo_image.jpg” is selected in the dialog box, the non-normalised total intensity plot will be saved as “demo_image_TotInt.jpg”, since “Tot Int” is its title).

> ⓘ **Warning/Information: Mac and Windows compatibility and resolution problems of the saved plots may arise**. Test different formats to find the optimal solution to the current computer architecture.
>
> ☒ **Known bug:** Class Markers are NOT saved for the plots, ONLY Purest Markers, if they are present.
>
> ☑ **Workaround:** If Class Markers are to be included, each plot must be saved individually by the “Save Intensity Map” button of the Visualisation section of the GUI (see **Save Intensity Map**).

5.6. Mark Classes

Spectra in the dataset can be assigned to any one of 5 possible classes: class 0 (grey, when no class is specified by the "Mark Classes" radio buttons), class 1 (black), class 2 (red), class 3 (green) and class 4 (blue). Any particular spectrum can only belong to one class.

Marked spectra are shown in the Marked Spectra Plot of the Visualisation section of the GUI, using the colour of their classes. The Marked Spectra Plot updates automatically as soon as any Pre-Processing parameter is changed, or a spectrum is added or removed from the Marked Spectra list.

The positions of the Marked Spectra are also indicated on the Visualised plots, using square markers with the respective class colours and white outlines. These Class Markers remain even if the "Visualised" drop-down menu is changed to show a different map, or the "Colours" drop-down menu is changed to use a different colour scheme for visualisations.

Marked Spectra are also listed in the "Marked" listbox by their spectrum number shown in their class colours. Spectra are listed in the "Marked" listbox by increasing class numbers first and then increasing spectrum number within each class.

> ☒ **Known bug:** Class Markers can be lost from the plots even though the Marked Spectra Plot still shows them and they are listed in the "Marked" listbox as well. (For example when the "Save Visualisation Plots" button is clicked and the GUI cycles through all plots, see **Save Visualisation Plots**; or during refolding, see **Change Dimensions (Refold)**).
>
> ☑ **Workaround:** Change the "Visualised" drop-down list to another plot to reset the maps and show the Class Markers again.
>
> ☒ **Known bug:** Class Markers are NOT saved for the plots, ONLY Purest Markers, if they are present.
>
> ☑ **Workaround:** If Class Markers are to be included, each plot must be saved individually by the "Save Intensity Map" button of the Visualisation section of the GUI (see **Save Intensity Map**).
>
> ⓘ **Warning/Information:** There is no way in the GUI to save the Marked Spectra Plot. However, the Marked Spectra can be saved (see **Save Marked Spectra** and **Compile Class Matrix**) and re-plotted outside the GUI.
>
> ⓘ **Warning/Information:** There is no option in the GUI to mark more than one spectrum at a time.

Marking spectra activates the "Reset All Classes", "Save Marked Spectra" and "Compile Class Matrix" buttons in the Visualisation section of the GUI. If there are no marked spectra, these buttons are deactivated.

5.6.1. Manual Marking / Unmarking

To manually mark a spectrum first select a class the spectrum should belong to by ticking any of the four "Mark Classes" radio buttons. If no radio button is ticked, the class is set to 0 (grey). Then click on the spectrum in the active Visualised Plot to mark it.

> ⓘ **Warning/Information:** It can be very difficult to accurately mark a spectrum this way if the plot contains many spectra or folded in the wrong dimensions. In this case, use the **Get From Selected** option.

> ⓘ **Warning/Information:** If a spectrum is already assigned to a class, it cannot be selected for a new one directly this way (i.e. class membership cannot be overwritten). It first needs to be cleared (unmarked) and then marked to belong to the new class.

To unmark a spectrum of any class, either click on its marker in the Visualised plot, or double click its identifier (the spectrum number shown in the class colour) in the "Marked" listbox (preferred).

> ⓘ **Warning/Information:** It is very difficult to unmark a spectrum by clicking on its marker in the Visualised plot, as it requires hitting the invisible boundary around the marker, which is usually to the lower right of the visible marker and can be over another spectrum to the lower right (which may already be marked as well). It is considerably easier to use the "Marked" listbox for unmarking.

5.6.2. Get From Selected

It can be difficult to manually mark a particular spectrum in the dataset. In this case, the desired spectrum number can be typed in the "Selected" textbox of the Pre-Processing section of the GUI. This will show that particular spectrum in the Selected Spectrum Plot (in the Pre-Processing section of the GUI) even if it was not originally included in the "Selected" list (in case the dataset contains more than a 100 spectra, see **3.2 Selecting a spectrum for** display). Once the spectrum is selected, choose the class the spectrum should belong to by ticking any of the four "Mark Classes" radio buttons. If no radio button is ticked, the spectrum will belong to class 0 (grey). Finally, click on the "Get From Selected" button in the Visualisation section of the GUI. This will add the selected spectrum to the "Marked Spectra" Plot, its spectrum number will appear in the "Marked" listbox in its class colour and its position will be marked in the Visualised Plot with class colour as well.

5.7. Reset All Classes

Instead of manually removing all marked spectrum one by one (see **Manual Marking / Unmarking**), all classifications can be cleared at once. Do to this, click on the "Reset All Classes" button in the Visualisation section of the GUI. This will clear the Marked Spectra Plot, remove all Class Markers from the Visualised plots and clear the "Marked" listbox. It will not reset the "Mark Classes" radiobuttons, but it will deactivate the "Reset All Classes", "Save Marked Spectra" and "Compile Class Matrix" buttons in the Visualisation section of the GUI.

> ⓘ **Warning/Information:** Once deleted, the Markers cannot be recovered and the GUI gives no warning prior to deleting them!

5.8. Save Marked Spectra

All marked spectra can be saved as individual .mat files by clicking the “Save Marked Spectra” button in the Visualisation section of the GUI. A standard save dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded. It also suggest a name that contains the original dataset name and AsLS baseline correction parameters (lambda and p) in the following format: “demo_image_lvalue_92dot5514_pvalue_0dot001_Extracted.mat” where “demo_image” is the name of the original dataset name, “lvalue_92dot5514” refers to lambda=92.5514 and “pvalue_0dot001” stands for p=0.001.

Since each file will be saved as an individual .mat file, the final filenames will also contain the text “SpectrumXX”, where XX stands for the spectrum number. This information is automatically appended to the filename, there is no need to specify it. As a result, there will be as many .mat files saved as there were Marked Spectra.

> ⓘ **Warning/Information:** If files with the same name already exist in the target location (folder), they will be overwritten without warning!

Each saved .mat file contains a single MATLAB variable called “MarkedSpectrum”, which contains 2 rows: the first is the spectral variables (wavenumbers), the second is the corresponding intensities.

> ⓘ **Warning/Information:** Saving the spectra this way saves each spectrum as an individual file, and class information is lost. To retain class information and to save all marked spectra in a single file, use the “Compile Class Matrix” button (see **Compile Class Matrix**)

5.9. Compile Class Matrix

Instead of saving each Marked Spectrum individually (see **Save Marked Spectra**), all Marked Spectra can be saved together in a single Microsoft Excel .xlsx file. This way, class information is retained as well.

Click on the “Compile Class Matrix” button in the Visualisation section of the GUI. A standard save dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded. It also suggest a name that contains the original dataset name and AsLS baseline correction parameters (lambda and p) in the following format: “demo_image_lvalue_92dot5514_pvalue_0dot001_ClassMatrix.xlsx” where “demo_image” is the name of the original dataset name, “lvalue_92dot5514” refers to lambda=92.5514 and “pvalue_0dot001” stands for p=0.001.

> ⓘ **Warning/Information:** If a file with the same name already exist in the target location (folder), a warning message is shown. Click “Cancel” and choose a different name if the file should not be overwritten.

The saved .xlsx file contains the Marked Spectra in the first worksheet: The row contains the spectral variables (wavenumbers), thereafter each row is one spectrum. The first column contains the file names (which include the name of the .xlsx file appended with "SpectrumXX" at the end, where XX stands for the spectrum number), and the second column contains the Class number for each spectrum, thereafter each column contains the intensities at a certain spectral variable (wavenumber).

ⓘ **Warning/Information:** The resulting .xlsx file can be directly imported into Umetrics SIMCA-P, although it is recommended that an identifier is added to the column containing the Class number (by default it is blank). The spectrum names may automatically be marked as OBS IDs in Umetrics SIMCA-P. Variable IDs, X and Y variables likely required to be manually specified during importing.

ⓘ **Warning/Information:** The resulting .xlsx file CANNOT be directly used as an input for the GUI, as it does not match the input format requirements (see **.xlsx file input**)

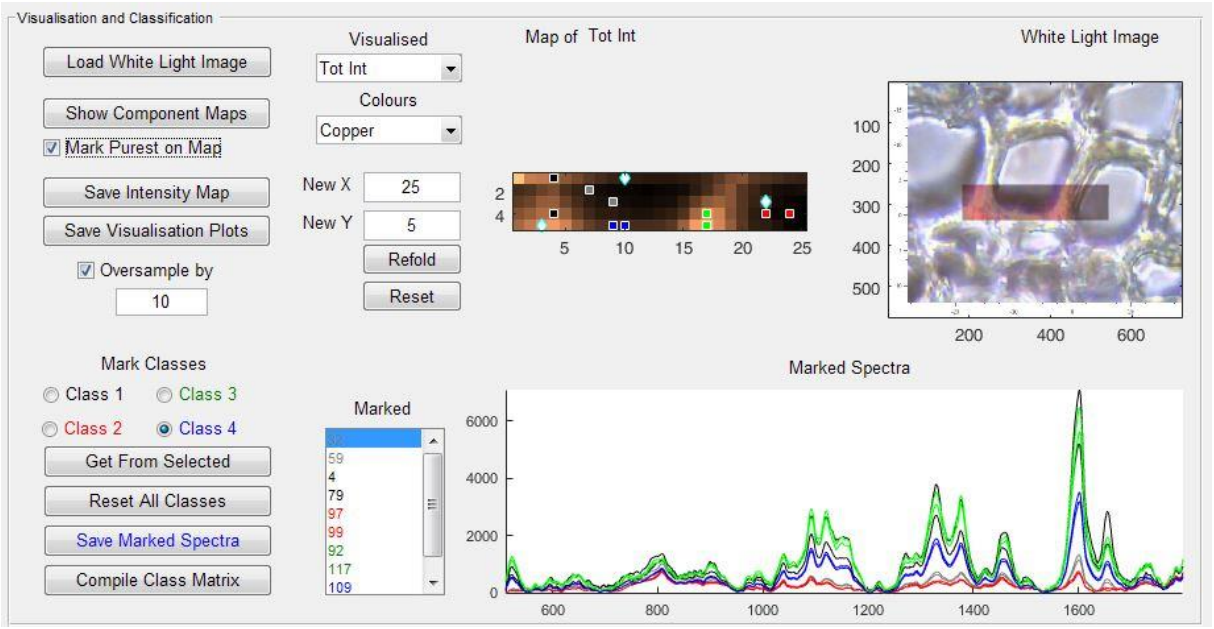

Figure 31. *The Visualisation section of the GUI, after spectra are marked for Classes. The example uses the "demo_image.txt" file, after MCR-ALS modelling finished in* ***Figure 26****. The "Mark Purest on Map" checkbox is ticked so the purest spectra are marked by white diamonds with cyan outline. Classes are marked by gray (Class 0), black (Class 1), red (Class 2), green (Class 3) and blue (Class 4) boxes with white outlines in the visualised map, and these colours are used in the Marked Spectra plot as well as in the "Marked" listbox, which contains the spectra in class order first, and thereafter spectrum number order. Since there are more spectra marked than space in the listbox, a scroll bar automatically appears on the side of the listbox.*

6. Reference Matching

The resolved spectral profiles of the Components can be matched to a set of spectra (References) for identification / validation. The reference matching is built solely on simple **Euclidean distances** (see **Match References**). The user **must always verify the results** of the reference matching by visual inspection, the numbers provided in the table are only indicators.

6.1. Load Reference Spectra

Click on the "Load Reference Spectra" button in the Reference Matching section of the GUI. A standard dialog window opens to navigate to the folder where the Reference Spectra are located and open the file(s).

There are three ways of providing Reference Spectra input for the GUI:

1. A set of .mat files. Each Reference Spectrum must be in one .mat file, which must contain a single data matrix: the first column is the spectral variables (wavenumbers), the second is the corresponding intensities. The filenames will be used as the names (identifiers) for the reference spectra in the GUI. Hold "Shift" or "Ctrl" for selecting and loading multiple files simultaneously.

   ⓘ **Warning/Information:** All files must be in the same folder!
2. A set of .txt files. Each Reference Spectrum must be in one .txt file, which must contain two columns (space or tab separated): the first column is the spectral variables (wavenumbers), the second is the corresponding intensities. The filenames will be used as the names (identifiers) for the reference spectra in the GUI. Hold "Shift" or "Ctrl" for selecting and loading multiple files simultaneously.

   ⓘ **Warning/Information:** All files must be in the same folder!
3. A single Microsoft Excel .xlsx file. All spectra must be in the first worksheet of the file, first column containing the names (identifiers) for the reference spectra (these names will be used in the GUI) in every even number row, every odd number row containing the spectra variables (wavenumbers) and every even number row containing the corresponding intensities.

   ⓘ **Warning/Information:** Only one file can be loaded if .xlsx is chosen for input format.

For exact formatting requirements of the Reference Spectra files, see **Reference Spectra**).

ⓘ **Warning/Information:** Only one type of spectra input can be used at a time, i.e. all reference spectra must be either in a set of .mat OR .txt files OR a single .xlsx file, they cannot be combined!

ⓘ **Warning/Information:** Reference spectra **will be made compatible** with the dataset automatically upon loading and will be total area normalised for display. Compatibility means that reference spectra will be either trimmed to the length of the dataset (i.e. excluding all wavenumbers outside the spectral region of the dataset) or/and expanded to match the length of the dataset (adding zeros, i.e. baseline). In addition, spectral resolution will be adjusted to match that of the dataset, using the built-in "spline" interpolation algorithm of MATLAB.

ⓘ **Warning/Information:** Upon loading any new Reference Spectra, they will NOT be added to the already existing ones, but overwrite them, without warning. This will NOT affect the Initial Estimates, i.e. if Reference Spectra were passed to the Initial Estimates (see **Use Reference Spectra**), they will remain there even if a new set of References are loaded. This is intentional, not a bug, so a set of spectra can be provided as Initial Estimates and another set loaded as References.

Once the Reference Spectra are loaded, the Reference Spectra plot updates to show the loaded spectra, the "Save Normalised Refs" and the "Use Refs as Initials" button become active, as well as the "Match References" button, if MCR-ALS modelling has been completed (i.e. there are resolved spectral profiles to be matched) (**Figure 32**).

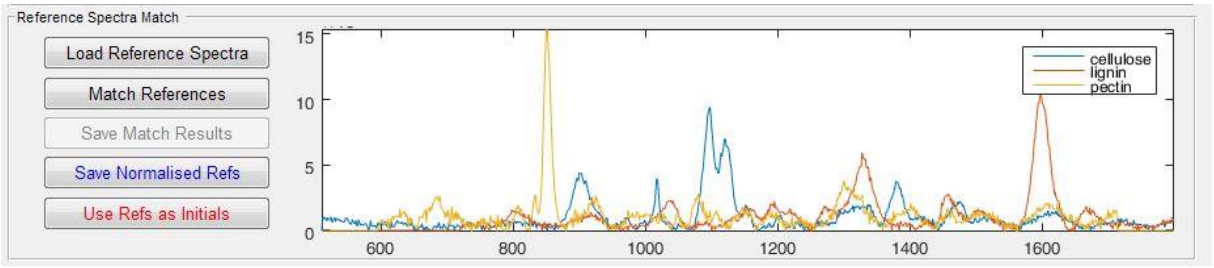

Figure 32. *The Reference Matching section of the GUI after loading Reference Spectra. The example uses the "demo_image.txt" file, after MCR-ALS modelling finished in* ***Figure 26*** *and the "demo_references.xlsx" auxiliary data file. Note the conflicting colour scheme of the reference spectra and Components, in e.g. the MCR-ALS section of the GUI (****Figure 26****).*

ⓘ **Warning/Information:** Reference Spectra are automatically area normalised upon loading (to match the MCR-ALS resolved spectral profiles, which are also normalised by the built-in spectral equal length constraint (see **Closure**).

ⓘ **Warning/Information:** Reference Spectra are also min-max scaled and offset corrected in the Reference Spectra plot. This scaling is for display purposes only and is NOT used during reference matching.

ⓘ **Warning/Information:** All Pre-Processing parameters are ignored by the Reference Spectra, i.e. **Reference Spectra are NOT pre-processed**! (If they need to be pre-processed by the same parameters as the dataset spectra, save the pre-processing parameters, load the Reference Spectra as if they were a dataset, pre-process them with the saved parameters and save them. This process involves repeated format changes to match different the input requirements of the dataset and the reference spectra, see **Input data** and **Reference Spectra**).

ⓘ **Warning/Information:** The Reference Spectra plot uses the same colour scheme as the Components in the MCR-ALS section of the GUI. Do not confuse Reference Spectra colours with Component colours!

☒ **Known bug:** In certain cases, if the reference spectra are loaded before (or during) loading the spectral data, the "Match References" button remains inactive.

☑ **Workaround:** Load the reference spectra after the spectral data has finished loading.

☒ **Known bug:** The GUI will try to make reference spectra compatible with resolved MCR-ALS profiles. If the data has not been passed for MCR-ALS (see **3.12. Passing the Data for** MCR-ALS), this may result in errors.

☑ **Workaround:** Load the reference spectra after the dataset has been passed for MCR-ALS.

6.2. Match References

Click on the "Match References" button to match the MCR-ALS resolved spectral profiles to the loaded Reference Spectra.

Reference matching is based on simple **Euclidean distances** of two row matrices (the resolved spectral profile of a Component and a Reference Spectrum). The Euclidean distance in mathematics is the straight line distance between two points in the Euclidean space, and MATLAB defines it as follows:

d = sum((x-y).^2).^0.5, where d is the Euclidean distance, x and y are the two row vectors (the resolved spectral profiles and the reference spectrum, respectively).

The smaller the distance, the closer the points are to each other (better matching).

The direction of the vectors can be determined by the scalar dot product of two real vectors of length *n*:

$u \cdot v = u_1 v_1 + u_2 v_2 + \ldots + u_n v_n$.

If the dot product is equal to zero, then *u* and *v* are perpendicular. This relation is commutative for real vectors, such that dot(u,v) equals dot(v,u).

Once the Reference Match is performed, the results are shown in two popup windows; one contains a Table with the numerical results (**Figure 33**), the other contains as many plots as there are Components in the MCR-ALS model: each plot showing one resolved Component (in thick black lines) and all Reference Spectra in the same colours as in the Reference Spectra plot (**Figure 34**).

ⓘ **Warning/Information:** The Reference Spectra plot uses the same colour scheme as the Components in the MCR-ALS section of the GUI. Do not confuse Reference Spectra colours with Component colours (see **Figures 26**, **32** and **34**)!

ⓘ **Warning/Information:** The matches (Euclidean distances) will be converted automatically to percentages for easier interpretation. Values below 0.01 (less than 1% match) will be replaced by 0. Otherwise values will be rounded to whole percentages.

Warning/Information: The matches shown in the Table of the popup window are ONLY indicators and **visual inspection of the spectral profiles is ALWAYS necessary** and should be trusted over the numeric results (**Figures 33** and **34**). This is especially true for complex spectra with diagnostic bands. Consider the following two hypothetical scenarios:

1. A resolved Component contains a single band between 515-525 cm^{-1} and the Reference Spectrum a single band between 1550-1560 cm^{-1}. If the spectral range is from 400-2000 cm^{-1} and the spectral resolution is 1 cm-1, there are 1600 cm^{-1} points in each spectrum. Of these, only 20 points are not agreeing (10 in the 515-525 cm-1 region, and 10 in the 1550-1560 cm-1 region), and 1580 points are matching completely. Thus the Euclidean match will be excellent, whereas spectroscopically it is clear that the match should be 0%, since both spectra feature only one band (diagnostic), and they do not match. This false positive match is provided by matching the absence of bands (baseline), which is indeed a valuable information, but in this case numerically outweighs the information of the diagnostic bands.
2. The resolved Component and the Reference Spectra contain the EXACTS same spectra, but due to experimental errors (such as improper/lacking calibration), one of the two spectra is shifted by 10 cm-1. Since all bands are shifted, this can result in a relatively poor Euclidean match (not even the baselines are matching), whereas a visual inspection of the spectral profiles reveals the presence of all diagnostic bands in both spectra, the only difference being the shifted position, indicating a potential calibration error.

In real life examples these extremes do not occur, but the presence/absence of diagnostic bands is hard to quantify numerically (case 1), and the positions of diagnostic bands can easily shift (e.g. due to different chemical environments, pH, etc) even though the same compounds are recorded (case 2).

Warning/Information: Reference Spectra are automatically area normalised upon loading (to match the MCR-ALS resolved spectral profiles, which are also normalised by the built-in spectral equal length constraint (see **Closure**).

Warning/Information: All Pre-Processing parameters are ignored by the Reference Spectra, i.e. **Reference Spectra are NOT pre-processed**! (If they need to be pre-processed by the same parameters as the dataset spectra, save the pre-processing parameters, load the Reference Spectra as if they were a dataset, pre-process them with the saved parameters and save them. This process involves repeated format changes to match different the input requirements of the dataset and the reference spectra, see **Input data** and **Reference Spectra**). Consider this carefully if any kind of normalisation has been performed to the dataset, as this can skew the match results. Due to the built-in spectral equal length constraint (see **Closure**), it is recommended NOT to use any normalisation for the dataset, and this is especially important if Reference Matching is to be performed.

Known bug: In certain cases, if the reference spectra are loaded before (or during) loading the spectral data, the “Match References” button remains inactive.

☑ **Workaround:** Load the reference spectra after the spectral data has finished loading.

To **save** the Reference Matching plots in the popup window, use the menu or the icons of the popup window itself (see **Plots**). There is currently no option in the GUI to save the Reference Spectra Plot in the Reference Matching section.

ⓘ **Warning/Information: Mac and Windows compatibility and resolution problems of the saved plots may arise**. Test different formats to find the optimal solution to the current computer architecture.

To save the Table containing the numerical results of the Reference Matching, use the "Save Match Results" button (see **Save Match Results**).

ⓘ **Warning/Information:** The Table showing the numerical results of the Reference Matching may not be fully visible in the popup window, depending on the number of Components and Reference Spectra and their names. However, slider / scroll bars should be visible to allow for scrolling to different parts of the Table. In addition, MATLAB allows for resizing the popup window, which should resize the Table as well.

☒ **Known bug:** Resizing the popup window may not resize the Table.

☑ **Workaround:** Manually resize the Table after resizing the popup window: click on the Table until its boundary is highlighted, select one of the corner markers by clicking on it, then hold and drag to the required size.

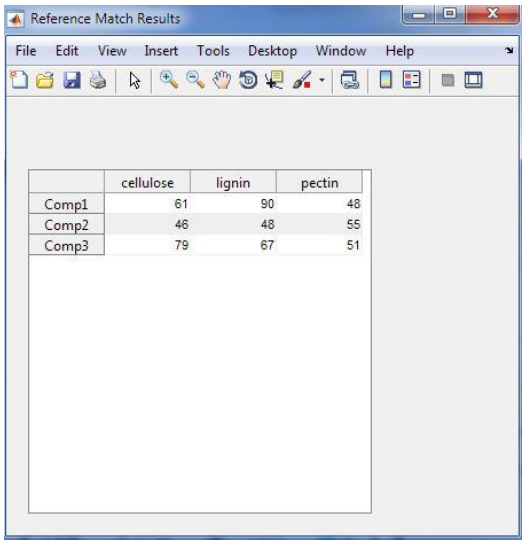

Figure 33. *The numerical results of Reference Matching, as shown in the popup window. The example uses the "demo_image.txt" file, after MCR-ALS modelling finished in* ***Figure 26*** *and the "demo_references.xlsx" auxiliary data file. Note that the window has its own set of menus and icons. Resizing can change the window, but changing Table dimensions can be difficult. A scroll bar appears if the table does not fit in the frame. Note that Component 1 matches the "lignin" reference spectrum well, but Component 3 is almost equally matching the "cellulose" and "lignin" reference spectra. See also* ***Figure 34****.*

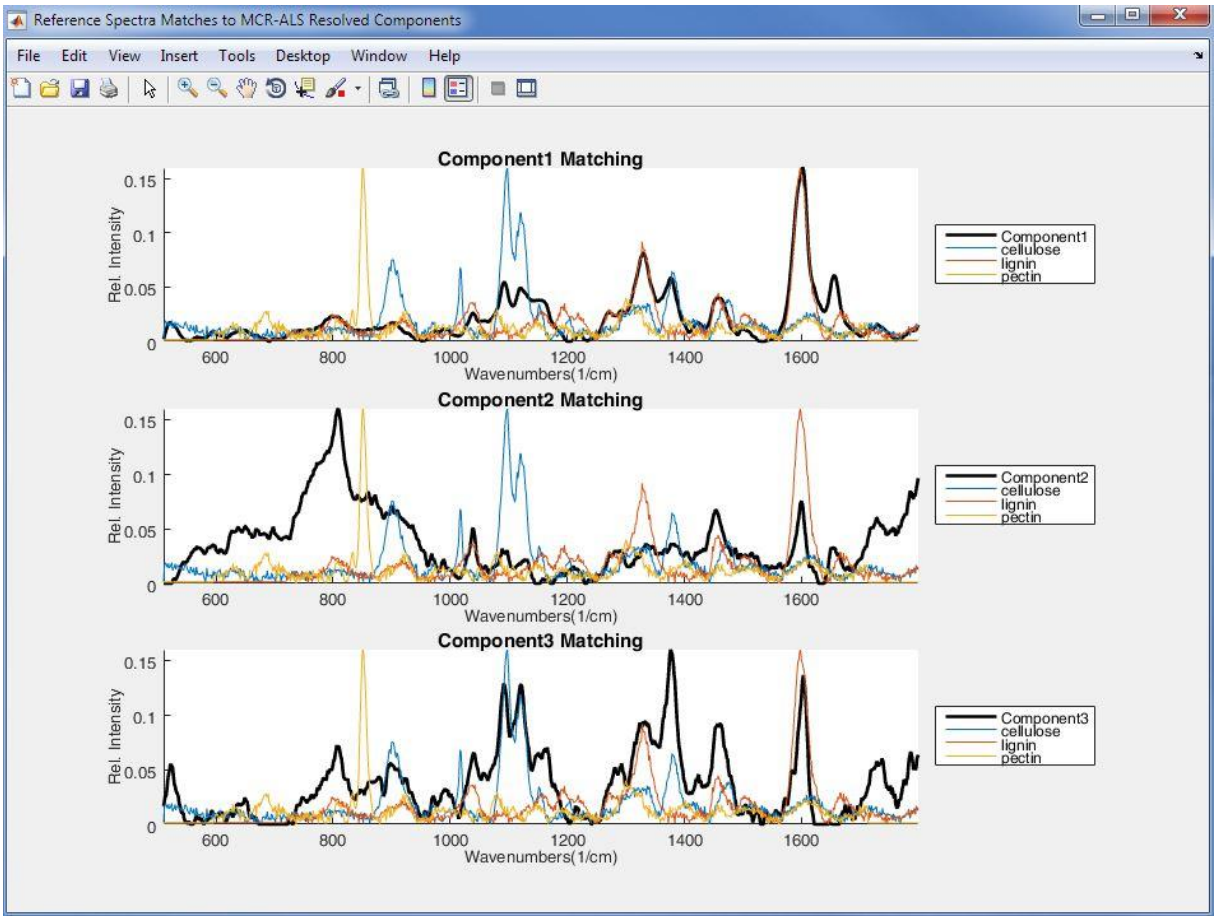

Figure 34. *The graphical results of Reference Matching, as shown in the popup window. The example uses the "demo_image.txt" file, after MCR-ALS modelling finished in* ***Figure 26*** *and the "demo_references.xlsx" auxiliary data file. Note that the window has its own set of menus and icons, allowing for e.g. saving the plots. Note that Component 1 matches the "lignin" reference spectrum well, as noted from the tabular results as well (see* ***Figure 33****) but it is not entirely "clean" (bands between 1050 – 1150 cm^{-1} are more similar to cellulose, for instance). Component 3 has significant contributions from both "cellulose" and "lignin". This indicates that the current MCR-ALS model was unable to resolve pure lignin and cellulose spectral profiles in this dataset with these processing parameters.*

6.3. Save Match Results

The numerical results of the Reference Matching can be saved either using the controls (menu / icon) of the popup widiw, or by clicking on the "Save Match Results" button in the Reference Matching section of the GUI (recommended). A standard save dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded. It also suggests a name that contains the original dataset name and the number of components used in the MCRALS model, appended by the text "_RefMatching" at the end (e.g. "demo_image_Components3_RefMatching.xlsx").

ⓘ **Warning/Information:** If a file with the same name already exist in the target location (folder), a warning message is shown. Click "Cancel" and choose a different name if the file should not be overwritten.

The saved .xlsx file contains the numerical Reference Matching results in the first worksheet, as in the table in the popup window (see **Figure 33)**. The first row contains the Reference Spectra names and the first column the Component names, the cells containing the numerical matching values as shown in the Table in the popup window generated by the "Match References" button (see **Match References**).

ⓘ **Warning/Information:** The matches are shown in whole percentages (rounded). Values below 0.01 (less than 1% match) will be replaced by 0.

6.4. Save Normalised References

Since the Reference Spectra are automatically made compatible and are area normalised upon loading, they are not identical to the original spectra that were loaded. Thus, they may need to be saved along with the resolved spectral profiles (**Save the Optimised Spectra Matrix** and **Save the MCR-ALS Results**) and the results of the Reference Matching (see **Match References** and **Save Match Results**), for complete documentation or use by third party software.

Click on the "Save Normalised References" button in the Reference Matching section of the GUI. A standard save dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original Reference files were loaded. Irrespective of the input format (see **Reference Spectra** and **Load Reference Spectra**), the normalised Reference Spectra will be saved as a single Microsoft Excel .xlsx file. The GUI suggests a name that contains the original dataset name appended by the text "_NormRefs" at the end..

ⓘ **Warning/Information:** If a file with the same name already exist in the target location (folder), a warning message is shown. Click "Cancel" and choose a different name if the file should not be overwritten.

The saved .xlsx file contains all normalised Reference Spectra in the first worksheet, in the same format as used for Reference Spectra input (see **Microsoft Excel .xlsx file input** and **Load Reference Spectra**). That is, the first column containing the names (identifiers) of the reference spectra in every even number row, every odd number row containing the spectra variables (wavenumbers), even though they are identical after they have been made compatible, and every even number row containing the corresponding intensities.

6.5. Use References as Initials

See **Use Reference Spectra**.

7. Clustering and Segmentation

The maps generated by the intensity evaluations (see **3.7.** Integration and **3.8. Single Point Intensity** Evaluation) or by mapping the concentration profiles of MCR-ALS resolved Components (see **MCR-ALS**, **Visualisation and Classification** and **Show the Component Maps**) often show gradients within the sample but no sharp boundaries. Different segmentation methods are available to generate maps with sharp boundaries and the GUI uses k-means clustering (see **K-Means Clustering**). The clustering is based on the concentration profiles of the MCR-ALS resolved Components only (no clustering can be done in the GUI using integrals or single point intensities) and it is generally useful for image datasets, i.e. identify chemically distinct zones in the image. However, clustering can be useful for spectral series or even for a set of independent spectra, especially when they are ordered in chemically / experimentally meaningful sets instead of randomised sequences.

7.1. Number of Clusters

The number of clusters must be supplied in the "# Clusters" textbox in the Clustering and Segmentation section of the GUI before k-means clustering can take place. There are two ways of adding the number of clusters:

7.1.1. Manual Input

Simply click in the "# Clusters" textbox in the Clustering and Segmentation section of the GUI and type the number of clusters and hit "Enter/Return" on the keyboard.

> ⓘ **Warning/Information:** The minimum number of clusters is 2 (default in the GUI upon Loading) and the maximum is half the number of total spectra in the dataset. If the user supplies any number outside this range or non-numerical input, the textbox reverts to the default value of 2. Decimals inputs are rounded.

> ⓘ **Warning/Information:** Manual input is ideal for experimenting with different numbers of clusters.

7.1.2. Silhouette Clusters

If there is no a priori information available regarding the possible number of clusters, silhouette clustering can be used to explore the dataset. In MATLAB, the silhouette value for each point is defined as a measure of how similar that point is to points in its own cluster, when compared to points in other clusters. The silhouette value for the i^{th} point, S_i, is thus calculated as

$S_i = (b_i-a_i)/ \max(a_i,b_i)$, where a_i is the average distance from the i^{th} point to the other points in the same cluster as i, and b_i is the minimum average distance from the i^{th} point to points in a different cluster, minimized over clusters. This distance in the GUI is based on Euclidean distances (for a definition, see), which is the default setting of MATLAB (although MATLAB allows for any supported distance metric to be used for the silhouette evaluation).

The resulting silhouette values range from -1 to +1. A high silhouette value (i.e. close to +1) indicates that the point (spectrum in the dataset) matches to other points (spectra) in its own cluster very well, and is rather dissimilar (matches poorly) to points (spectra) in neighbouring clusters. If most points have a high

silhouette value, then the clustering solution is considered appropriate, otherwise the clustering solution may have either too many or too few clusters.

When the user clicks on the "Silhouette Clusters" button in the Clustering and Segmentation section of the GUI, the script starts k-means clustering with 2 clusters, and increasing the number of clusters by 1 in each loop until the silhouette value improves. During this time, a warning dialog appears, telling the user to wait until the process finishes. The number of clusters that gives the best silhouette value is considered the appropriate clustering solution, and this number is shown in the "# Clusters" textbox when the silhouette clustering process is finished (and the warning dialog automatically closes).

> ⓘ **Warning/Information:** Silhouette clustering is not fail proof and is not an iterative optimisation procedure, but a simple loop. Feel free to override the silhouette cluster number manually, if needed (a priori knowledge of the dataset) or for exploratory purposes (testing different cluster numbers)

> ⓘ **Warning/Information:** Silhouette clustering is normally fast, but in case of large datasets (many Components and many spectra) on slow computers it may take several seconds. Be patient.

> ⓘ **Warning/Information:** The dialog box can be closed manually before silhouette clustering is finished. In this case, there is no indication in the GUI whether the process is finished or not, unless the "# Clusters" textbox updates to show a different number of clusters than before.

7.2. K-Means Clustering

Click on the "k-Mean Clustering" button in the in the Clustering and Segmentation section of the GUI to perform segmentation by k-means clustering, based on squared Euclidean distances and MATLAB's built-in k-means++ algorithm for cluster centre initialisation. The number of times to repeat the clustering using new initial values is determined by the 'Replicates' options. It is set to 50 in the GUI by default, to make sure the initial values do not influence the results.

> ⓘ **Warning/Information:** Although the number of Replicates is hard-coded in the GUI, it is easy to change in the source code. It is set to 50 as it is relatively fast, yet allows for clusters to be stable. (If the Replicates value is low, cluster numbering could jump, i.e. one spectrum could belong to cluster 1 in during one clustering, and to cluster 2 in the next. This does not mean that the segmentation boundaries would change, i.e. the same spectra would belong to the same cluster, only the number of that cluster would change. In practical terms, this would mean that the colouring of the segmentation map would change every time clustering is done, although the pattern would remain constant).

When clustering is finished, the Segmentation Map and the corresponding Centroid Profiles are automatically updated in the GUI (see **Figure 35**). The Segmentation Map shows the cluster distributions in the dataset, folded to the same dimensions as the initial dataset and in the same colour scheme as set by the "Colours" drop-down list in the Visualisation section of the GUI (see **Change Colour**). The Segmentation Map is also added to the end of the list of plots in the "Visualised" drop-down list of the GUI (see **Change**

Plot). The Centroid Profiles show a stacked bar plot, showing the contribution of each Component to that cluster, in percentage. The colouring in the Centroid Profiles matches the colour coding of the Components throughout the GUI (see the Legend of the Initial Estimates Plot in the MCR-ALS section of the GUI, **Initial Estimates**).

ⓘ **Warning/Information:** There can only be maximum one of any kind of plot at any given time in the "Visualisation" list, which means that if a new Segmentation Map is generated, the old one gets overwritten. If the Segmentation Plot in the Visualisation section of the GUI gets obsolete (the MCR-ALS model is deleted or a new MCR-ALS modelling is performed), it is **deleted**. However, the Segmentation Map and the corresponding Centroid Profiles are **NOT deleted** from the Clustering and Segmentation section of the GUI until the MCR-ALS model is deleted (by loading a new dataset, see **3.1 Loading the** Data or by passing the same dataset with new pre-processing values to the MCR-ALS section of the GUI (see **3.12. Passing the Data for** MCR-ALS).

ⓘ **Warning/Information:** Only the Segmentation Map is added to the list of "Visualisation" plots, the Centroid Profiles are NOT. Thus, Centroid Plots cannot be saved by the buttons in the Visualisation section of the GUI (see **Save Intensity Map** and **Save Visualisation Plots**). Use the "Save Centroid Plot" button to save the plot (see **Save Centroid Plot**), or the "Save Segmentation Results" button to save the data (see **Save Segmentation Results**) from which the plots can be re-generated in MATLAB or in third party software.

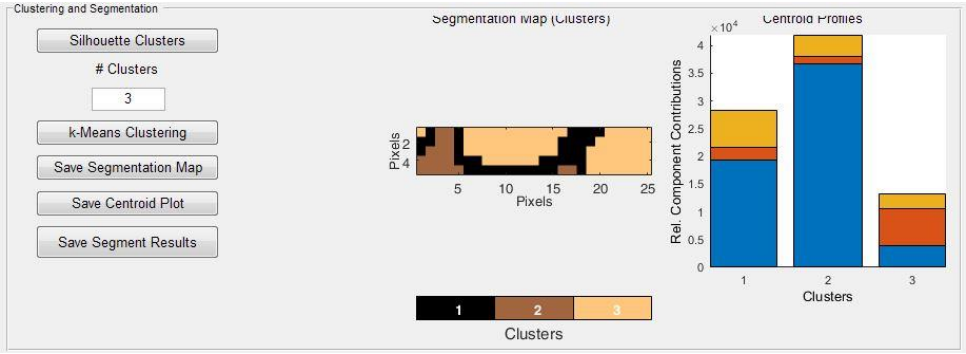

Figure 35. *The Clustering and Segmentation section of the GUI. The example uses the "demo_image.txt" file, after MCR-ALS modelling finished in* ***Figure 26*** *and Silhouette Clusters, resulting in 3 clusters. The Segmentation Map uses the default "Copper" colour scheme and the Centroid Profiles use the Component colours matching the MCR-ALS section of the GUI (i.e. Component 1 is blue, Component 2 is red, Component 3 is yellow, see e.g.* ***Figures 19*** *and* ***26****). The Segmentation map identifies the lumen areas of the image as cluster 3 (beige colour), and the Centroid profiles indicate that spectra in these regions are low in intensity and dominated mostly by Component 3. Cluster 2 (brown) in the Segmentation map matches the location of the middle lamella, and accordingly the centroid profiles show a high contribution of Component 1 to this cluster. Component 1 is mostly lignin, as indicated by Reference Matching (see* ***Figures 33*** *and* ***34****). Finally, Cluster 1 (black) is the cell wall, with lignin and cellulose as main constituents (Components 1 and 3, blue and yellow, see Centroid Profiles and* ***Figures 33*** *and* ***34****). Note, that while the resolution provided by this MCR-ALS model is suboptimal (see* ***Figure 34*** *caption), the segmentation map is still valuable for identifying chemically different regions of the image.*

7.3. Save Segmentation Results

Click on the “Save Segment Results” button in the Clustering and Segmentation section of the GUI to save the results of the segmentation in a single .mat file. A standard save dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded. The GUI will also suggest a name that contains the original dataset name, appended at the end by the text “_ClusteringResultsXComp_YClusters”, where X is the number of Components the MCR-ALS model had and Y is the number of Clusters used for k-means clustering.

> ⓘ **Warning/Information:** If a file with the same name already exist in the target location (folder), a warning message is shown. Click “Cancel” and choose a different name if the file should not be overwritten.

The saved single .mat file which will contain two MATLAB variables:

“**Centr**” and “**IDX**”. “Centr” stands for the centroid profiles, i.e. the contribution of each resolved component to a certain cluster, and thus consisting of as many rows as there are clusters and as many columns as the number of MCR-ALS resolved components. “IDX”, contains the cluster identifiers, i.e. a single number to denote which cluster each spectrum in the dataset belongs to. Thus, it consists of a single column with as many rows as there are spectra in the dataset.

> ⓘ **Warning/Information: The .mat file does NOT contain XY dimension information (for image data sets) or filenames (for spectra series).** The XY dimension can be added to the filename upon saving (default suggestion by the GUI) but spectrum names must be saved separately for spectra series.

> ⓘ **Warning/Information:** Only the data matrices will be saved, the plots will not be included!

7.4. Save Segmentation Map

Click on the “Save Segmentation Map” button in the Clustering and Segmentation section of the GUI to save the Segmentation Map plot as shown in the GUI. A standard dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded, and suggests a name consisting of the dataset name, appended with the text “_SegmentationMapXComp_YClusters” at the end, where X is the number of Components the MCR-ALS model had and Y is the number of Clusters used for k-means clustering.

The default format is .pdf, but it can be changed to all major formats MATLAB allows for export (.jpg, .tif, .png, .gif, .eps).

ⓘ **Warning/Information: Mac and Windows compatibility and resolution problems of the saved plots may arise**. Test different formats to find the optimal solution to the current computer architecture.

ⓘ **Warning/Information:** There is no information showing whether the results shown are obtained with the current values or not. For example, if a new MCR-ALS model was generated, or a new number of clusters supplied, the Segmentation Plot will not be updated until the "k-Means Clustering" button was not clicked again, which will overwrite the previous results.

ⓘ **Warning/Information:** Only the plot will be saved, the data matrix to generate the plot will NOT be included. To save the data matrix, use the "Save Segment Results" button instead (see **Save Segmentation Results**).

ⓘ **Warning/Information:** To change the colour scheme of the plot, use the "Colours" drop-down list in the Visualisation section of the GUI (see **Change Colour**).

ⓘ **Warning/Information:** The Segmentation Map is also added to the list of "Visualisation" plots, thus it can also be saved by the buttons in the Visualisation section of the GUI (see **Save Intensity Map** and **Save Visualisation Plots**). Use these buttons if the Pure Markers (see **Mark Purest on Map**) or Class Markers (see **Mark Classes**) need to be visible on the saved Segmentation Map, as those are not shown in the Clustering and Segmentation section of the GUI.

ⓘ **Warning/Information:** Since the Segmentation Map is also added to the list of "Visualisation" plots, it can be refolded (see **Change Dimensions (Refold)**) and saved with the new dimensions using the "Save Intensity Map" button in the Visualisation section of the GUI (see **Save Intensity Map**). Since refolding does not affect the Clustering and Segmentation section of the GUI, use this option if the plot needs to be saved in a different dimension setting than provided with the original dataset.

7.5. Save Centroid Plot

Click on the "Save Centroid Plot" button in the Clustering and Segmentation section of the GUI to save the Centroid Plot as shown in the GUI. A standard dialog window appears where the folder and filename can be provided.

By default, the GUI suggests the same folder from which the original dataset was loaded, and suggests a name consisting of the dataset name, appended with the text "_CentroidMapXComp_YClusters" at the end, where X is the number of Components the MCR-ALS model had and Y is the number of Clusters used for k-means clustering.

The default format is .pdf, but it can be changed to all major formats MATLAB allows for export (.jpg, .tif, .png, .gif, .eps).

ⓘ **Warning/Information: Mac and Windows compatibility and resolution problems of the saved plots may arise**. Test different formats to find the optimal solution to the current computer architecture.

ⓘ **Warning/Information:** There is no information showing whether the results shown are obtained with the current values or not. For example, if a new MCR-ALS model was generated, or a new number of clusters supplied, the Segmentation Plot will not be updated until the “k-Means Clustering” button was not clicked again, which will overwrite the previous results.

ⓘ **Warning/Information:** Only the plot will be saved, the data matrix to generate the plot will NOT be included. To save the data matrix, use the “Save Segment Results” button instead (see **Save Segmentation Results**).

ⓘ **Warning/Information:** While the Segmentation Map is also added to the list of “Visualisation” plots, thus it can also be saved by the buttons in the Visualisation section of the GUI (see **Save Intensity Map** and **Save Visualisation Plots**), the Centroid Plot can only be saved by this button.

8. Closing the GUI

The GUI can be closed by simply closing the main GUI window or by exiting MATLAB. In both cases, a warning dialog appears, informing the potential loss of all unsaved data. Proceed accordingly.

ⓘ **Warning/Information:** If only the main GUI window is closed but MATLAB is not, some data might still be recovered. In particular, the variables that are placed in the MATLAB Workspace after an MCR-ALS modelling has been performed will remain there after the main GUI window is closed (see **Perform MCR-ALS**). These variables are easily accessible. Other variables may be possible to salvage, but they would require expert computer skills.

ⓘ **Warning/Information:** Closing the main GUI window does NOT close any potential popup window opened by the GUI (such as Component Maps or Reference Matching results, etc.). Thus, the contents of these windows is possible to save even after the main GUI window is closed, until the user manually closes them, or exits MATLAB.

ⓘ **Warning/Information:** Closing the main GUI window does NOT clear the variables in the memory. Type "clear all" in the MATLAB Command Window to clear all variables from the memory. There are no warnings, but this will clear all variables permanently, with no possible recovery options.

ⓘ **Warning/Information:** Closing the main GUI window does NOT clear the MATLAB Command Window, where results of the MCR-ALS optimisation may be visible (see Perform MCR-ALS). Type "clc" in the MATLAB Command Window to clear it completely. This action cannot be undone.

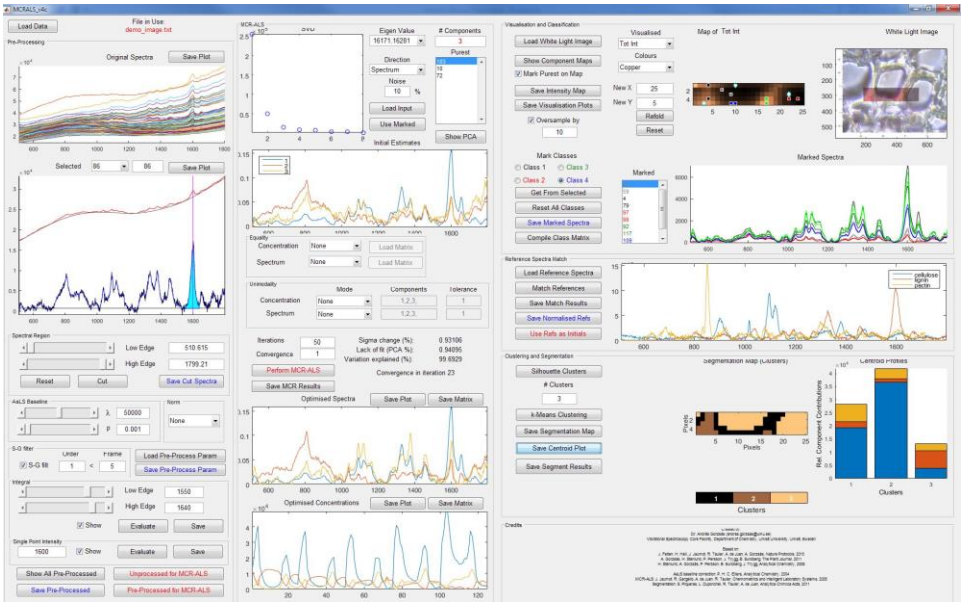

***Figure 36**. The fully populated GUI. The example uses the "demo_image.txt" file after MCR-ALS imaging (as described in **Figures 16** and **26**), with the "demo_references.xlsx" auxiliary dataset loaded. Classes are marked as in **Figure 31**, and segmentation performed after silhouette clustering with 3 clusters (**Figure 35**).*

Acknowledgements

The Vibrational Spectroscopy Core Facility is supported by the Chemical Biological Centre and the Department of Chemistry at Umeå University.

The contributions of A. de Juan, J. Felten, T. Gillgren, K. Hagvall, H. Hall, J. Kenney, J. Jaumot, and P. Persson to writing, designing and/or testing parts of this GUI are acknowledged. Discussions with A. de Juan and her comments have been especially valuable and highly appreciated for the MCR-ALS part of the GUI. Discussions with J. Kenney and her comments were very helpful when designing and implementing the added functionalities of the GUI compared to the version published in Nature Protocols in 2015.