



# User Manual



**Fastosh, version 1.0.10**

2018-2026

Fastosh v. 1.0.10 - User Manual *(last update: 26 November, 2025)*

# Welcome !

Fastosh is a free standalone program that provides unique functionalities to process X-ray Absorption Fine Structure (XAFS) spectroscopy data. This program was created for Users of SAMBA beamline, SOLEIL Synchrotron, but can be employed to process data generated at any XAFS beamline since it can open data files in ASCII format. The name of the program is pronounced [*fast-osh*] as the French word « fastoche », which familiarly means « easy ». The program indeed allows users to easily and rapidly perform a number of operations such as chunking and averaging a large number of scans, auto-deglitching, 2D filtering, 3D plotting, data processing using chemometric methods, and assessing the progress in data acquisition using a fully automatic data viewer tool. Additionally, functions specifically compatible with SAMBA data notably allow to average HDF files and extract new fluorescence spectra from MCA patterns processed to minimize acquisitions artefacts, such as diffraction peaks. These functions exclusive to SAMBA data are listed in the following Table of Content with “SAMBA data only” next to their names.

Fastosh should be helpful for a wide range of research applications where XAFS is employed, including Geochemical and Environmental studies where the characterization of complex mixtures by chemometric approaches, and analyses of very diluted samples using XAFS spectra acquired in fluorescence mode, may be required. Some elements of Fastosh include algorithms from existing codes whose credits fully belong to their respective authors, including the Wavelet transform codes from Muñoz *et al.* [1] & Funke *et al.* [2], the MCR-ALS Matlab Toolbox from Joaquim Jaumot *et al.* [3, 4], as well as the Larch functions from Matthew Newville [5]. For sake of simplicity, this manual mostly describes how to use all functionalities that are unique to Fastosh. Detailed XAFS data treatment procedures and tutorials can be found elsewhere.

The program is available for Windows, Mac, & Linux platforms. Please refer to the “Installation” section of this manual to successfully install Fastosh on your computer. Latest release of the program is available for free download at SAMBA beamline’s website:

<https://www.synchrotron-soleil.fr/en/beamlines/samba>

Cheers!



# Table of Content

<b>1</b>	<b>Getting started</b>	<b>7</b>
1.1	Needs and solutions proposed by Fastosh	7
1.2	Installation / Uninstallation	8
1.2.1	Only one file is needed to install Fastosh on Windows, MacOS, or Linux	8
1.2.2	Installation on Windows	8
1.2.2.1	Installing Fastosh for the first time on your PC	8
1.2.2.2	Installing an updated version of Fastosh on Windows	9
1.2.2.3	Uninstalling Fastosh on Windows	10
1.2.3	Installation on Mac	11
1.2.3.1	Installing Fastosh for the first time on a Mac	11
1.2.3.2	Installing an updated version of Fastosh on Mac	12
1.2.3.3	Uninstalling Fastosh on Mac	13
1.2.4	Installation on Linux	13
1.2.4.1	Installing Fastosh for the first time on Linux	13
1.2.4.2	Installing an updated version of Fastosh on Linux	15
1.2.4.3	Uninstalling Fastosh on Linux	16
<b>2</b>	<b>Basic Functionalities</b>	<b>17</b>
2.1	Purpose of Main GUI	17
2.2	Two methods to find E0	18
2.3	Importing data to the Main GUI	18
2.3.1	Drag and Drop	18
2.3.2	Importing any file containing XAFS data, or other types of data	19
2.3.2.1	XAFS data from ASCII files	19
2.3.2.2	Non-XAFS data	20
2.3.3	Importing data from any HDF file, other than those generated at SAMBA	20
2.3.3.1	Direct selection of the absorption	20
2.3.3.2	Specifying the numerator & denominator of the absorption expression	22
2.3.3.3	Importing hyperspectral data	24
2.3.3.4	Trick for succesfull MCR-ALS processing of hyperspectral data in Fastosh	26
2.3.4	Importing data from Athena project	28
2.3.5	Importing SAMBA HDF file (SAMBA data only): many benefits over SAMBA ASCII file	28
2.4	Basic data treatment	31
2.4.1	Data handling in Main GUI	31
2.4.1.1	Multiple selection in the sample list	31
2.4.1.1.1	Check all, Uncheck all, Invert	31
2.4.1.1.2	Lasso	32
2.4.1.2	Plotting data of one or multiple sample(s)	32
2.4.1.2.1	Viewing one scan (the highlighted one in the sample list)	32
2.4.1.2.2	Viewing multiple scans in 2 or 3D	32
2.4.1.2.3	Coloring multiple scans in 2 or 3D	34
2.4.1.2.4	Stacking multiple scans in 2D	36
2.4.1.3	Normalization (Larch method)	37
2.4.1.3.1	Visually estimating the normalization	38
2.4.1.3.2	E0 readjustment	39
2.4.1.3.3	Victoreen pre-edge function	39
2.4.1.3.4	Polynomial pre-edge function	40
2.4.1.3.5	Post-edge function	41
2.4.1.4	Alternative normalization methods applied to multiple spectra	42
2.4.1.4.1	Fastosh simple method: subtract offset and locally equalize spectra	42
2.4.1.4.2	PyMCA method 1: "Subtract offset and normalize to maximum"	42
2.4.1.4.3	PyMCA method 2: "Subtract offset and normalize to counts"	42
2.4.1.4.4	PyMCA method 3: "Subtract offset and normalize to integrated area"	42
2.4.1.4.5	PyMCA method 4: "Normalize to maximum"	43
2.4.1.5	Background removal	43

2.4.1.5.1 Principle of background subtraction in Larch (Python & Matlab versions)	43
2.4.1.5.2 Details on how “background” in Expression (1) is obtained by autobk	43
2.4.1.5.3 Specificity of Larch autobk adapted to Fastosh	46
2.4.1.5.4 $R_{bkg}$	46
2.4.1.5.5 Spline k-weight	46
2.4.1.5.6 Spline $k_{min}$ & $k_{max}$	46
2.4.1.5.7 Clamps	46
2.4.1.6 Fast Fourier Transform	47
2.4.1.6.1 k range: $k_{min}$ and $k_{max}$	47
2.4.1.6.2 Window	47
2.4.1.6.3 dk	47
2.4.1.6.4 FT plot window, with or without theoretical chi models created by FEF	48
2.4.1.7 Reverse Fourier Transform: $R_{min}$ , $R_{max}$ , Window, and dR	49
2.4.1.8 Plot k weight	50
2.4.1.9 Wavelet Transform	50
2.4.1.9.1 Presentation	50
2.4.1.9.2 Wavelet Plot options	51
- Plot type: “WT of current (+ model, if created)”	51
- Plot type: “WT of current + chi + $\sum_i WT(k_i)$ ”	53
- Plot type: WT of current + chi + wavelet	54
- Plot type: WT of current + Reverse WT	55
2.4.1.9.3 Cauchy-Morlet WT Comparator	56
2.4.1.9.4 Cauchy Wavelet Transform- Fourier Transform Comparator	56
2.4.2 Viewing the spectrum of a sample or associated reference	58
2.4.2.1 Viewing the reference data	58
2.4.2.2 Viewing the references associated to multiple samples	59
2.4.3 Data calibration & alignment	59
2.4.4 Chunk Merging	61
2.4.5 Data truncation	66
2.4.6 Interpolation	67
2.4.6.1 Usefulness	67
2.4.6.2 Interpolation methods	68
2.4.6.2.1 Plot options	69
2.4.6.2.2 E0 value	69
2.4.6.2.3 Transfer options to main GUI	69
2.4.7 Deglitching strategy: manual or automatic?	70
2.4.8 Manual deglitching	70
2.4.8.1 Presentation of manual deglitching methods & main window	70
2.4.8.2 Manual deglitching procedures	71
2.4.8.2.1 Interactively selecting a set of data point	71
2.4.8.2.2 Editing data point values	74
2.4.9 Automatic deglitching	74
2.4.9.1 Presentation	74
2.4.9.1.1 Principle	74
2.4.9.1.2 Caveat on using Autodeglitching to multiple spectra	75
2.4.9.1.3 Description of the Auto-deglitching window	75
2.4.9.1.4 Parameters	76
2.4.9.2 Examples	77
2.4.10 Two-D Filtering	80
2.4.10.1 Usefulness & principle	80
2.4.10.2 Presentation of the 2D filtering window	81
2.4.10.3 Setting & testing filter parameters	82
2.4.10.4 Viewing and transferring 2D filtering results	82
2.4.11 Useful tips!	82
2.4.11.1 Displaying progressive difference	82
2.4.11.2 Displaying the stack tile number in a 3D plot of highlighted scan in Main GUI	83
2.4.11.3 Moving up and down sample names in list of Main GUI	84
2.4.11.4 Applying one specific parameter of the highlighted scan to other scans	84
2.4.11.5 Modifying, saving, and reimporting preferences of data plotter's look	84
2.4.12 Pre-data treatment of non-XAFS data	85
<b>3 Live Viewer</b>	<b>87</b>

3.1	Presentation .....	87
3.1.1	Usefulness .....	87
3.1.2	Description of Live Viewer window .....	87
3.2	How it operates.....	89
3.2.1	Generalities .....	89
3.2.1.1	Principle, in a nut shell .....	89
3.2.1.2	Importing an XAFS scan .....	89
3.2.1.2.1	Opening a SAMBA HDF file (SAMBA data only).....	89
3.2.1.2.2	Opening an ASCII file .....	90
3.2.1.2.3	Opening the reference data collected along with sample data and auto alignment.....	90
3.2.2	Description of all automatic steps performed during an update .....	91
3.2.3	Selection of the highlighted sample .....	92
3.3	Functionalities.....	93
3.3.1	Constraining the E0 value.....	93
3.3.2	“Edit Mode”: modifying default parameters.....	94
3.3.2.1	Principle .....	94
3.3.2.2	Apply all parameters to sample list .....	95
3.3.3	Plotting options.....	95
3.3.3.1	Four ways to display data relative to the current highlighted sample .....	95
3.3.3.2	Plot boundaries .....	95
3.3.3.3	Displaying data in 2D or 3D .....	95
3.3.3.4	3D plot rotation.....	96
3.3.3.5	Shifts to scatter multiple spectra .....	96
3.3.3.5.1	Y shifts: for 2D plot .....	96
3.3.3.5.2	Z shifts for 3D plot.....	96
3.3.3.6	Displaying first derivative .....	96
3.3.3.7	Plot color options.....	97
3.3.4	Viewing sample of reference data relative to a single iteration .....	97
3.3.4.1	Viewing sample data of a single iteration and unselecting it .....	98
3.3.4.2	Viewing the aligned reference spectrum of current iteration .....	98
3.3.5	Displaying signal-to-noise ratio and random noise estimation .....	98
3.3.6	Transferring merge to the Main GUI.....	100
4	<b>Preliminary data treatment prior to PCA, TT, MCR-ALS, or LCF .....</b>	<b>101</b>
5	<b>PCA &amp; Target Transformation (TT) .....</b>	<b>102</b>
5.1	Usefulness.....	102
5.2	Preparation of Data Matrix & references in Main GUI .....	102
5.3	Module window at opening .....	102
5.4	PCA.....	103
5.4.1	Principle.....	103
5.4.2	SVD method: centering or no pre-processing .....	104
5.4.3	Interpreting PCA results .....	104
5.4.3.1	Eigenvalue methods .....	105
5.4.3.1.1	The cumulated relative eigenvalues (default plot) .....	105
5.4.3.1.2	The eigenvalues listed in the result table of the PCA window .....	105
5.4.3.1.3	Eigenvalue plotted against component number .....	105
5.4.3.1.4	Variance fraction plotted against component number .....	106
5.4.3.1.5	Logarithmic first difference of the eigenvalues .....	106
5.4.3.2	Other methods .....	106
5.4.3.2.1	Eigenvalues, excluding the first one (XANES spectra only).....	106
5.4.3.2.2	IND.....	107
5.4.3.2.3	Singular values .....	108
5.5	Target Transformation .....	110
5.5.1	Principle.....	110
5.5.2	Comparison with SPOIL values reported in Malinowski (1978) .....	111
5.5.3	Procedure.....	111
5.5.3.1	Selecting reference spectra to target-transform .....	111

5.5.3.2 Results .....	111
5.5.3.2.1 SPOIL .....	113
5.5.3.2.2 Plot of target array vs projected vector .....	113
<b>6 MCR-ALS .....</b>	<b>114</b>
6.1 Usefulness .....	114
6.2 Principle of MCR-ALS .....	114
6.3 Pre MCR-ALS .....	115
6.3.1 Main steps .....	115
6.3.2 PCA & Multiple SIMPLISMA .....	116
6.3.2.1 PCA .....	116
6.3.2.2 Multiple SIMPLISMA .....	117
6.3.2.2.1 Usefulness .....	117
6.3.2.2.2 Description of the Multiple SIMPLISMA window .....	119
6.3.2.2.3 Choice of COO expression .....	120
6.4 Jaumot <i>et al.</i> MCR-ALS Toolbox .....	120
6.4.1 Overall procedure .....	120
6.4.2 PCA, prior to MCR-ALS .....	121
6.4.3 Initial estimation: SIMPLISMA or EFA .....	121
6.4.3.1 SIMPLISMA .....	121
6.4.3.1.1 Principle .....	121
6.4.3.1.2 Procedure .....	123
6.4.3.2 EFA .....	123
6.4.3.2.1 Principle .....	123
6.4.3.2.2 Procedure: case of an individual kinetics dataset .....	124
6.4.3.2.3 Procedure: case of an “augmented” data matrix (multiple kinetics datasets) .....	125
6.4.4 Constraints .....	126
6.4.4.1 Automatic setting of constraints .....	126
6.4.4.2 Non-negativity .....	127
6.4.4.2.1 Force to zero .....	127
6.4.4.2.2 nnls .....	127
6.4.4.2.3 fnnls .....	127
6.4.4.3 Unimodality .....	127
6.4.4.4 Closure .....	129
6.4.4.4.1 Principle .....	129
6.4.4.4.2 Application of closure constraint to XAFS data .....	130
6.4.4.5 Equality .....	130
6.4.4.5.1 Principle .....	130
6.4.4.5.2 Application of Equality Constraint to [Pure concentrations] .....	131
6.4.4.5.3 Application of Equality Constraint to [Pure spectra] .....	131
6.4.5 Advanced toolbox functionalities .....	132
6.4.6 Fitting .....	133
6.4.6.1 ALS optimization: general parameters & window .....	133
6.4.6.1.1 Number of iterations .....	133
6.4.6.1.2 Convergence Criterion .....	133
6.4.6.1.3 ALS optimization window .....	133
6.4.6.2 Handling MCR-ALS results .....	134
6.4.6.2.1 Saving as a text file .....	134
6.4.6.2.2 Transferring pure spectra to main GUI .....	134
6.4.7 Displaying maps of pure species, for datasets acquired in 2D .....	135
6.4.7.1 Displaying the GUI .....	135
6.4.7.2 Added features for displaying 2D maps, non available in the original toolbox .....	135
6.4.7.2.1 Homing on each axis during data acquisition .....	135
6.4.7.2.2 Composite map of all pure species .....	136
6.5 Post MCR-ALS .....	136
6.5.1 Generalities .....	136
6.5.1.1 Principle .....	136
6.5.1.2 Procedure .....	136
6.5.2 Amplitude correction factor (EXAFS only) .....	136

6.5.3 Result table .....	137
<b>7 Linear Combination Fitting .....</b>	<b>138</b>
7.1 Usefulness.....	138
7.2 LCF procedure.....	138
7.3 Fitting options .....	139
7.3.1 Fit constrains.....	139
7.3.1.1 Non-negativity on all floated.....	139
7.3.1.2 Last inferred, not floated .....	140
7.3.2 Free fit: scale sum to 100% .....	140
7.3.3 Multiple fits .....	140
7.4 Results .....	140
7.4.1 Goodness of fit: R Factor, chi-square, & red. chi-square .....	140
7.4.1.1 Definition .....	140
7.4.1.2 Interpretation of R, $\chi^2$ , and reduced $\chi^2$ .....	141
7.4.2 Result list.....	141
7.4.3 Red chi square comparison .....	142
7.4.4 Show residual and/or components .....	142
7.4.5 EXAFS fit: view Fourier transform results.....	143
7.4.6 Saving results.....	143
7.4.7 Transferring normalized mu or chi to main GUI.....	144
7.5 LCF on multiple samples using a specific set of references .....	144
7.5.1 Guidelines .....	144
7.5.2 Visualization of results .....	144
7.5.3 Usefulness of LCF on multiple samples belonging to a kinetic dataset .....	146
<b>8 Quick EXAFS fitting using single scattering paths modelled by FEFF .....</b>	<b>147</b>
8.1 Presentation .....	147
8.1.1 Purpose .....	147
8.1.2 Main functionalities.....	147
8.2 Single scattering path creation using FEFF8L.....	147
8.2.1 Example of a path creation .....	147
8.2.2 Parameters required to build the theoretical chi .....	149
8.2.2.1 Parameters whose values are obtained by FEFF8L.....	150
8.2.2.2 Parameters set to specific values, other than the amplitude term .....	150
8.2.2.3 The amplitude term, User-defined as $SO_2 \times N$ , $SO_2$ , or N.....	150
8.3 Manual modification of a model parameter value .....	151
8.4 Fitting a model EXAFS to the experimental EXAFS .....	152
8.4.1 Principle.....	152
8.4.2 Choice of minimization approach: Fourier Transform (FT) or Wavelet Transform (WT) .....	152
8.4.3 Fit preparation: checking all parameters involved in the fit .....	153
8.4.3.1 Parameters exclusive to the experimental spectrum that affect the fit results .....	153
8.4.3.2 Parameters applied during a fit to both the experimental and theoretical spectra .....	154
8.4.3.2.1 Generalities.....	154
8.4.3.2.2 Case where fitting minimization is done using the FT .....	154
8.4.3.2.3 Case where fitting minimization is done using the WT .....	154
8.4.3.3 Parameters relative to the theoretical spectrum only.....	155
8.4.4 Condition to allow the fitting functionality.....	156
8.4.5 Fitting report.....	157
8.4.5.1 Goodness of fit.....	157
8.4.5.2 Uncertainty .....	157
8.4.5.3 Correlation.....	157
8.4.6 Color code reflecting the physical likelihood of model parameters .....	158
8.4.7 Determining what is the best fit between two fits .....	158
8.4.7.1 Three possible cases .....	158
8.4.7.2 Hamiton Test.....	159

8.4.8 Example: fitting of thorite EXAFS and Hamilton Test.....	159
8.5 Multiple fits at once .....	162
8.5.1 Principle.....	162
8.5.2 Fitting results.....	163
8.5.2.1 Table summarizing the fitting results .....	163
8.5.2.2 Bar plot relative to each model parameter or R factor .....	163
<b>9 Modules to exploit SAMBA files (SAMBA data only) .....</b>	<b>165</b>
9.1 HDF Processor (SAMBA data only) .....	165
9.1.1 Usefulness .....	165
9.1.2 How to use this module.....	166
9.1.2.1 Specify repertory of data folder .....	166
9.1.2.2 Data visualization .....	166
9.1.2.3 Data averaging .....	167
9.1.2.3.1 Best practice to select scan iterations to average .....	167
9.1.2.3.2 Creation of TXT or HDF merge file .....	167
9.1.2.3.3 Operations done during merging .....	167
- Creation of merge text file (Fast method) .....	168
- Creation of merge HDF file (Slow method) .....	168
9.1.2.4 Data conversion from HDF to TEXT file while redefining ROI and pixel selection .....	168
9.1.2.4.1 Preliminary step: manual or automatic pixel selection on a single iteration .....	170
9.1.2.4.2 Preliminary step: Automatic pixel selection on all iterations of a sample .....	172
9.1.2.4.3 Preliminary step: Modifying ROI .....	173
9.1.2.4.4 Second step: creating TXT files.....	173
9.2 MCA Viewer (SAMBA data only).....	173
9.2.1 Usefulness .....	173
9.2.2 Importing an HDF file .....	174
9.2.3 Main Window.....	175
9.2.3.1 Presentation .....	175
9.2.3.2 Browsing MCA tiles .....	176
9.2.4 Operations.....	176
9.2.4.1 Viewing and removing pixels .....	176
9.2.4.2 Modifying ROI min & max .....	179
9.2.4.3 MCA background subtraction.....	180
9.2.4.4 Diffraction corrector.....	181
9.2.4.4.1 Principle .....	181
9.2.4.4.2 Parameters .....	181
9.2.4.4.3 Example .....	183
9.2.4.5 Method number .....	183
9.3 Information Plot (SAMBA data only) .....	184
9.4 ascan, dscan reader (SAMBA data only) .....	185
<b>10 Acknowledgements .....</b>	<b>186</b>
<b>11 Troubleshooting.....</b>	<b>186</b>
<b>12 References .....</b>	<b>187</b>

# 1 Getting started

## 1.1 Needs and solutions proposed by Fastosh

The purpose of Fastosh is to provide solutions to several needs in XAFS data analysis based on a practical experience with Users of SAMBA, which is a beamline dedicated to bulk XAFS in the hard X-ray domain (i.e. 5 to 40 keV) at Synchrotron SOLEIL [6]. Some of the initial needs that motivated the creation of the program and their corresponding solutions proposed by Fastosh are described in this section.

- **Need for tools to automatically visualize data when a rapid XAFS scan acquisition mode is employed**

Rapid acquisition modes are becoming common at XAFS beamlines. They can be notably helpful in determining whether beam damage has occurred when the sample is exposed to X-rays. When a rapid acquisition mode is employed to analyze diluted samples requiring long acquisition time periods, or follow chemical reactions with slow kinetics, a large number of repeated scans (i.e. > 50 scans) can be acquired. Determining when enough data has been collected is not always straightforward if several dozens of scans are acquired for a given sample. Until the creation of Fastosh, this was typically done at SAMBA by loading all repeated scans in an XAFS data treatment software to individually visualize them and eventually averaging them. The acquisition was stopped once no significant change was observed in the signal-to-noise ratio of the average spectrum for a given time period. Therefore, this approach implied to iteratively perform multiple time-consuming operations on a computer during the acquisition of a sample. The first goal of Fastosh was then to propose a tool that enables to automatically visualize the data being collected and display the improvement of the signal-to-noise ratio of the mean spectrum.

A spectrum collected using a rapid acquisition mode may feature across its entire energy grid a constant energy step. A lot of data points can be then collected for a single XAFS spectrum. For example, if such rapid acquisition mode is employed to collect data on a 1500 eV energy range, a 0.5 eV step size results in an XAFS spectrum featuring 3000 data points. The deglitching tools of several XAFS data treatment software require to manually select each problematic data point in the  $\mu$  or EXAFS spectrum. This may represent a tedious task if many data points must be manually deglitched in the  $\mu$  spectrum. Therefore, another goal of Fastosh was to provide an auto-deglitching tool to facilitate the process of removing data points from  $\mu$  spectrum, including those collected with a rapid acquisition mode.

- **Need for a tool to efficiently perform Target Transformation**

The most common tool available to Users to perform Target Transformation (TT) requires to successively target transform each individual reference spectrum from a reference library. For example, if the user's reference library consists of 30 reference spectra, 30 TT must be successively performed subsequently to PCA analysis of the experimental data. However, this method can be very time-consuming. For example, if the upper limit in  $k$  space of the experimental EXAFS spectra is modified to test the potential effect of this parameter on the TT results, one must perform all over again the PCA and 30 subsequent individual Target Transforms. One goal of Fastosh was then to propose a tool that could instantaneously do target transformation on an entire group of spectra (e.g. a large library of references), instead of individual spectrum, following the TT procedure described in the 1978 paper of Edmund Malinowski [7].

- **Need for a tool to conveniently handle XAFS data with the MCR-ALS Matlab Toolbox from Jaumot *et al.* [3]**

The Matlab Toolbox from Jaumot *et al.* can be employed to perform MCR-ALS on a given XAFS data set consisting of XANES/ $\mu$  or EXAFS spectra. This implies, however, to preprocess the data set, group it as a data matrix, and import it to the toolbox. For example, suppose that a set of EXAFS spectra, from 3 to 14  $\text{\AA}$  in  $k$  space and  $k$  weighed to 3, must be processed by MCR-ALS. The spectra of this data set must be firstly obtained, grouped together, and imported to the Jaumot *et al.* toolbox. One goal of Fastosh was then to propose a user-friendly interface to easily preprocess the XAFS data and transfer it to the toolbox, given that Fastosh is also written in Matlab. Another goal was also to propose a post-data treatment tool allowing to compare the spectra

extracted by MCR-ALS corresponding to the pure phases with each standard spectrum of a user's reference library. This functionality may help identify the pure phases present in the system.

- **A specific need at SAMBA beamline: a tool to handle multi-pixel fluorescence detector data**

SAMBA beamline is notably equipped with a 36 pixel Ge detector and, since 2021, a 13 pixel Si drift detector. The XAFS data collected at SAMBA with a rapid, "continuous" acquisition mode is saved as both ASCII and HDF files. The ASCII file notably features the averaged fluorescence data corresponding to all pixels of the fluorescence detector. The HDF file contains the raw data corresponding to each pixel of the fluorescence detector, such as the MCA spectrum or input and output count rates collected at each energy value of the scan, for each pixel. This file is also provided to SAMBA User to perform post-processing of the fluorescence data if needed. However, until the creation of Fastosh, the means to extract and process the data saved in the SAMBA HDF files was entirely left to the beamline users. One goal of Fastosh was then to provide to SAMBA Users a friendly graphical interface to easily extract the data saved in the HDF files and perform operations such as visualizing MCA spectra and extracting new XAFS spectrum using a specific pixel selection.

## 1.2 Installation / Uninstallation

### 1.2.1 Only one file is needed to install Fastosh on Windows, MacOS, or Linux

Fastosh is written and compiled using MATLAB. Consequently, to run Fastosh as a standalone program on your computer under Windows, MacOS, or Linux operating system, the program "MATLAB Runtime" must be installed in addition to the program Fastosh. **Please note that "MATLAB Runtime" is not the software "MATLAB". The former is used to run compiled Matlab codes as standalone programs while the latter is used to create Matlab codes. Therefore, it is not required to have MATLAB installed on the computer to run Fastosh!** Both programs (Matlab & "Matlab Runtime") work independently from each other. This also implies that if MATLAB is installed on the computer, it is not necessary to uninstall it, in order to properly install "MATLAB Runtime".

It is also not necessary to download "MATLAB Runtime" from the web since an installer of this program is included in the Fastosh Installer file. Therefore, both Fastosh and MATLAB Runtime can be installed on your computer using the Fastosh installer file, following the directions described in the next paragraphs. The Fastosh Installer is really all you need to fully install Fastosh on your computer and run all functionalities of Fastosh.

### 1.2.2 Installation on Windows

#### 1.2.2.1 Installing Fastosh for the first time on your PC

The following installation procedure was successfully tested on Windows 7 & 10:

**Step 0:**

Get the latest installer file of Fastosh. This is an ".exe" file, e.g.:  
Fastosh\_Windows\_v0p10p1\_Installer.exe

**Step 1:**

Double click on the file to launch the Installation process. You can launch it from any location on your PC, e.g. Download, Documents, or Desktop folder.

**Step 2:**

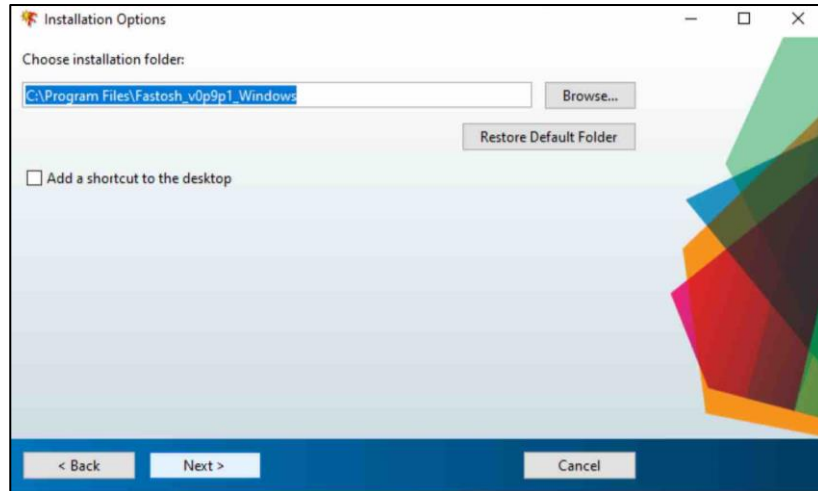
After double clicking on the Installer file, a window appears to assist you with the installation. Firstly, the version of Fastosh that will be installed is displayed in the window. Click on the button "Next" at the bottom of the window.

**Step 3:**

The window now displays “Choose installation Folder”. Here the folder location where Fastosh will be installed in your PC must be specified. The default location should be similar to:

“C:\Program Files\Fastosh”

This default location should be adequate. If you’re OK with it, just click on the button “Next” at the bottom of the window.

**Step 4:**

The folder location where MATLAB Runtime will be installed in your PC must be specified. The default location should be similar to:

“C:\Program Files\MATLAB\MATLAB\_Runtime”

This default location should be adequate. If you’re OK with this choice, click on the button “Next” at the bottom of the window.

**Step 5:**

The window now displays the locations that you specified where Fastosh & Matlab Runtime will be installed. To confirm, click on the button “Install” at the bottom of the window. Wait for the installation to complete. When all done, click on the button “Finish” at the bottom of the window.

**Step 6:**

The program is now installed on your PC in the directory that you specified at Step 3. For example, if you specified this directory “C:\Program Files\Fastosh”, the installation created three folders at this location:

“C:\Program Files\Fastosh\appdata”

“C:\Program Files\Fastosh\application”

“C:\Program Files\Fastosh\uninstall”

In this example, the execute file of Fastosh can be found in “C:\Program Files\Fastosh\application”. Double click on this file to start the program.

Also, you can right click on this execute file to:

**-Add a shortcut to the desktop:** “Create a shortcut”

**-Add a shortcut to the start menu:** “Pin to start”

**-Add a shortcut to the taskbar:** “Pin to taskbar”

That’s it!

### 1.2.2.2 Installing an updated version of Fastosh on Windows

Follow these steps to install a new version of Fastosh on your PC:

**Steps 0, 1, 2, & 3:**

Same as Steps 0, 1, 2, & 3 described above in Section 1.2.2.1.

#### Step 4:

This step can vary depending on the Fastosh update. If the update is relatively small, it is likely that the MATLAB Runtime will be the same between the old and new versions of Fastosh. If it is a major update, the required MATLAB Runtime may be different between the two versions. For example, Fastosh versions 0.9.0 and 0.9.1 are both running with MATLAB Runtime 2020a, v98. Therefore, if you upgraded from 0.9.0 to 0.9.1, you would not need to upgrade the MATLAB Runtime.

At this step, the installer window displays what is the MATLAB Runtime version required to run the new version of Fastosh that you are installing. Also, it informs you whether this version of MATLAB Runtime is already installed on your PC. If it is the case, the installer window displays a message similar to: "MATLAB Runtime version XXX is already installed at this location:

C:\Program Files\MATLAB\MATLAB\_Runtime\vXXX".

In that case, there's nothing to do: just click on "Next" button at the bottom of the screen. Alternatively, if Fastosh requires a new version of Matlab Runtime, you must specify at this step where the new Matlab Runtime will be installed. The default location displayed in the window should be similar to:

"C:\Program Files\MATLAB\MATLAB\_Runtime"

This default location should be adequate. If you're OK with this choice, click on the button "Next" at the bottom of the window.

#### Steps 5 & 6:

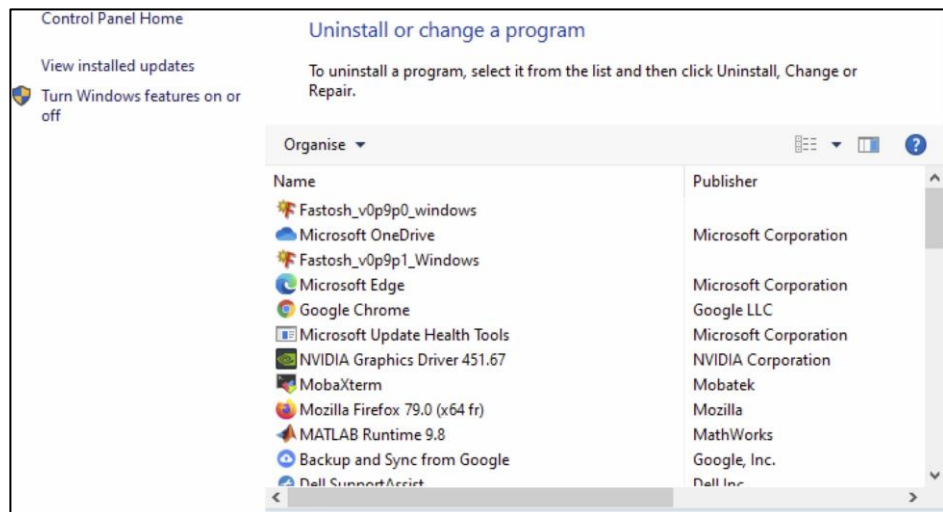
Same as Steps 5 & 6 described above in Section 1.2.2.1.

#### Step 7:

Uninstall from your PC the previous version of Fastosh. On Windows10, go to:

Start Menu > Windows System > Control Panel > Uninstall programs

In the example below, one can see that MATLAB Runtime 9.8 as well as Fastosh versions 0.9.0 and 0.9.1 are installed on the PC. In this case you would then need to uninstall only Fastosh 0.9.0. Right click on the program that you wish to remove and select "uninstall".



Also, you may have to delete the previous MATLAB Runtime in case you just have installed a new MATLAB Runtime version, as discussed at Step 4 above. That's it!

#### 1.2.2.3 Uninstalling Fastosh on Windows

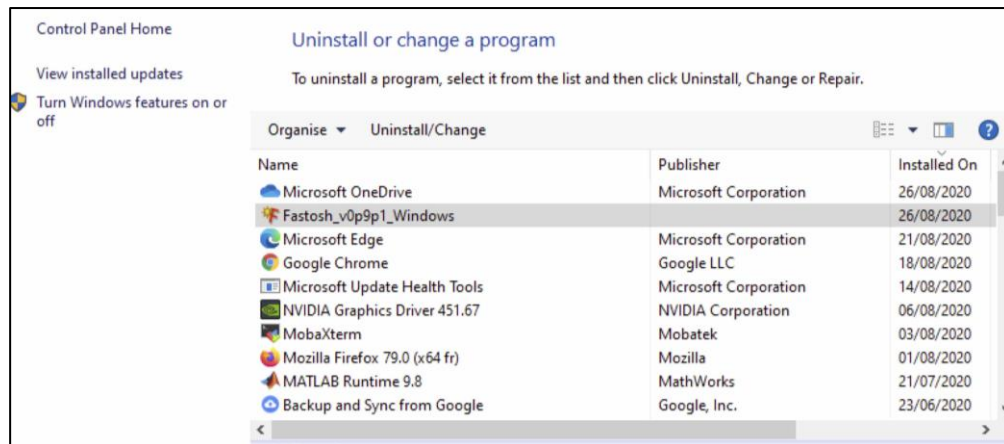
As mentioned in paragraph 1.2.1, running Fastosh on your computer implies having both the programs Fastosh and "MATLAB Runtime" installed. Therefore, these two programs must be

removed to entirely uninstall Fastosh. However, if you wish to install a newer version of Fastosh later on, you may not need to remove your MATLAB Runtime if the latest release of Fastosh uses the same version of MATLAB Runtime than the one already installed on your computer.

Go to the Windows system's Program Manager tool. For example, in Windows10, go to:

[Start Menu > Windows System > Control Panel > Uninstall programs](#)

In the example below, if Fastosh v 0.9.1 & MATLAB Runtime 9.8 have to be removed, right click on each program and select “uninstall”.



## 1.2.3 Installation on Mac

### 1.2.3.1 Installing Fastosh for the first time on a Mac

The following installation procedure was successfully tested on MacOS Catalina v10.15.3 (Intel chip) and MacOS Sonoma v 14.5 (Silicone chip):

#### Step 0:

Get the latest installer file of Fastosh. Unzip it. The unzipped file is an “.app” file, e.g.:  
Fastosh\_Mac\_v0p9p1\_Installer.app

#### Step 1:

Double click on the .app file to launch the Installation process. You can launch it from any location on your Mac, e.g. Download, Documents, or Desktop folder.

#### Step 2:

After double clicking on the Installer file, a window appears to assist you with the installation. Firstly, the version of Fastosh that will be installed is displayed in the window. Click on the button “Next” at the bottom of the window. Wait for a while as a small message window displays: “Assembling product list...”

#### Step 3:

The window now displays “Choose installation Folder”. Here the folder location where Fastosh will be installed in your Mac must be specified. The default location should be similar to:

“/Applications/Fastosh”

This default location should be adequate. If you’re OK with this choice, click on the button “Next” at the bottom of the window.

#### Step 4:

The folder location where MATLAB Runtime will be installed in your Mac must be specified. The default location should be similar to:

“Applications/ MATLAB\_Runtime”

This default location should be adequate. If you’re OK with this choice, click on the button “Next” at the bottom of the window.

**Step 5:**

The window now displays the locations that you specified where Fastosh & Matlab Runtime will be installed. To confirm, click on the button “Install” at the bottom of the window. Wait for the installation to complete. When all done, click on the button “Finish” at the bottom of the window. Fastosh and “MATLAB Runtime” are now installed on your Mac in the directories that you specified at Step 3 & 4, respectively. For example, if you specified this directory “Applications/Fastosh”, the installation created three folders at this location:

“Applications/Fastosh/appdata”  
 “Applications/Fastosh/application”  
 “Applications/Fastosh/uninstall”

Also, if you specified this directory “Applications/MATLAB/MATLAB\_Runtime”, the installation created a folder RXXX (e.g. R2023a) containing the Runtime codes, for example: “Applications/MATLAB/MATLAB\_Runtime/R2023a”

**Step 6 (Optional):**

Open a terminal (Terminal can be found in Applications/Utilities), and type the following command after replacing MR by the required information (see examples below):

***export DYLD\_LIBRARY\_PATH=/MR/RXXX/runtime/maci64:/MR/RXXX/sys/os/maci64:/MR/RXXX/bin/maci64***

In this command you must replace MR by the directory where MATLAB is installed on your computer:

*Example 1:*

*If R2023a is located in:*

*/Users/landrot/Applications/application/R2023a*

*Then MR is: « Users/landrot/Applications/application »*

*Example 2:*

*If MR is the one shown in Example 1, then the command to type in the terminal*

*is:*

***export DYLD\_LIBRARY\_PATH=/Users/landrot/Applications/application/R2023a/runtime/maci64:/Users/landrot/Applications/application/R2023a/bin/maci64:/Users/landrot/Applications/application/R2023a/sys/os/maci64***

The .app file of Fastosh can be found in “Applications/Fastosh/application”. Double click on this file to start the program.

Also, to pin the program to the taskbar, just manually drag the Fastosh .app file, from the “Applications/Fastosh/application” folder to the taskbar.

That’s it!

### 1.2.3.2 Installing an updated version of Fastosh on Mac

To install a new version of Fastosh, it is not mandatory to firstly uninstall the previous version of Fastosh that you wish to replace. Please follow these steps to install a new version of Fastosh on your Mac:

**Steps 0, 1, 2, & 3:**

Same as Steps 0, 1, 2, & 3 described above in Section 1.2.3.1.

**Step 4:**

This step can vary depending on the Fastosh update. If the update is relatively small, it is likely that the MATLAB Runtime will be the same between the old version of Fastosh that you want to replace and the one that you want to install. If it is a major update, the required MATLAB Runtime may be different between the two versions. For example, Fastosh versions 0.9.0 and 0.9.1 are both running with MATLAB Runtime 2020a, v98.

Therefore, if you upgraded from 0.9.0 to 0.9.1, you would not need to upgrade the MATLAB Runtime.

At this step, the installer window displays what is the MATLAB Runtime version required to run the new version of Fastosh that you are installing. Also, it informs you whether this version of MATLAB Runtime is already installed on your Mac. If it is the case, the installer window displays a message similar to: “MATLAB Runtime version vXXX is already installed at this location:

“Applications/vXXX”.

In that case, there’s nothing to do: just click on “Next” button at the bottom of the screen. Alternatively, if Fastosh requires a new version of Matlab Runtime, you must specify at this step where the new Matlab Runtime will be installed. The default location displayed in the window should be similar to:

“Applications/MATLAB\_Runtime”

This default location should be adequate. If you’re OK with this choice, click on the button “Next” at the bottom of the window.

#### **Step 5**

Same as Step 5 described above in Section 1.2.3.1.

#### **Steps 6, 7, & 8**

Do these steps only if the Fastosh update required installing a new version of MATLAB Runtime. They are the same as Steps 6, 7, 8 described above in Section 1.2.3.1.

#### **Step 9:**

Uninstall from your Mac the previous version of Fastosh. Go to its directory and simply drag its corresponding folders to the trash.

Also, you may have to uninstall, using the same method, the previous MATLAB Runtime in case you just have installed a new Runtime version, as discussed at Step 4 above.

That’s it!

### **1.2.3.3 Uninstalling Fastosh on Mac**

As discussed in paragraph 1.2.1, running Fastosh on your computer implies having both the programs Fastosh and “MATLAB Runtime” installed. Therefore, these two programs must be removed to entirely uninstall Fastosh. However, if you wish to install a newer version of Fastosh later on, you may not need to remove your MATLAB Runtime if the latest release of Fastosh uses the same version of MATLAB Runtime than the one already installed on your computer.

Go to Fastosh’s directory (most likely Applications/Fastosh), and simply drag its corresponding folders to the trash. Discard also the folders corresponding to the Matlab Runtime if necessary.

### **1.2.4 Installation on Linux**

Fastosh on Linux should be compatible on several Linux platforms, including Ubuntu, Linux Mint, and Cent OS.

#### **1.2.4.1 Installing Fastosh for the first time on Linux**

The following installation procedure was successfully tested on Ubuntu 18.04.4:

##### **Step 0:**

Get the zip file of the latest installer file of Fastosh. Unzip it. The unzipped file is an “.install” execute file, e.g.:

Fastosh\_Linux\_v0p9p1\_Installer.install

##### **Step 1:**

In a root terminal, go to the directory containing the installer. Then, type:

Fastosh v. 1.0.10 - User Manual - Getting started

```
./Fastosh_Linux_v0p9p1_Installer.install
```

**Step 2:**

A window appears to assist you with the installation. Firstly, the version of Fastosh that will be installed is displayed in the window. Click on the button “Next” at the bottom of the window.

**Step 3:**

The window now displays “Choose installation Folder”. Here the folder location where Fastosh will be installed on your computer must be specified. The default location should be similar to:

“/usr/Fastosh”

This default location should be adequate. If you’re OK with it, just click on the button “Next” at the bottom of the window.

**Step 4:**

The folder location where MATLAB Runtime will be installed on your computer must be specified. The default location should be similar to:

“/usr/local/MATLAB/MATLAB\_Runtime”

This default location should be adequate. If you’re OK with it, just click on the button “Next” at the bottom of the window.

**Step 5:**

The window now displays the locations that you specified where Fastosh & Matlab Runtime will be installed. To confirm, click on the button “Install” at the bottom of the window. Wait for the installation to complete. When all done, click on the button “Finish” at the bottom of the window.

**Step 6:**

Fastosh and “MATLAB Runtime” are now installed on your computer in the directories that you specified at Step 3 & 4, respectively. For example, if you specified this directory “/usr/Fastosh”, the installation created three folders at this location:

“/usr/Fastosh/appdata”  
“/usr/Fastosh/application”  
“/usr/Fastosh/uninstall”

Also, if you specified this directory “/usr/local/MATLAB/MATLAB\_Runtime”, the installation created a folder RXXX (e.g. R2023a) containing the Runtime codes, for example: “/usr/local/MATLAB/MATLAB\_Runtime/RXXX”

In the terminal, type the following command after replacing MR by the required information (see examples below):

```
export XAPPLRESDIR=/MR/RXXX/X11/app-defaults
```

In this command you must replace MR by the directory where MATLAB is installed on your computer:

*Example 1:*

*If R2023a is located in:*

*/usr/local/MATLAB/MATLAB\_Runtime/R2023a*

*Then MR is: «usr/local/MATLAB/MATLAB\_Runtime»*

*Example 2:*

*If MR is the one shown in Example 1, then the command to type in the terminal*

*is:*

```
export XAPPLRESDIR=/usr/local/MATLAB/MATLAB_Runtime/R2023a/X11/app-defaults
```

Next, type the following command after replacing MR by the required information (see example below):

```
export LD_LIBRARY_PATH=/MR/RXXX/runtime/glnxa64:/MR/RXXX/
```

***bin/glnxa64:/MR/RXXX/sys/os/glnxa64:/MR/RXXX/sys/opengl/lib/glnxa64***

In this command you must replace MR by the directory where MATLAB is installed on your computer.

*Example 3:*

*If MR is the one shown in Example 1, then the command to type in the terminal is:*

```
export LD_LIBRARY_PATH=/usr/local/MATLAB/  
MATLAB_Runtime/R2023a/runtime/glnxa64:/usr/local/MATLAB/MATLAB_Runtime/  
R2023a/bin/glnxa64:/usr/local/MATLAB/MATLAB_Runtime/R2023a/sys/os/glnxa64:/u  
sr/local/MATLAB/MATLAB_Runtime/R2023a/sys/opengl/lib/glnxa64
```

Now the program should be able to start.

To start the program, go to Fastosh/application folder in the terminal. Then type “.” followed by the name of the execute file in the folder. For example:

`./Fastosh_Linux_v0p9p1`

#### 1.2.4.2 Installing an updated version of Fastosh on Linux

To install a new version of Fastosh, it is not mandatory to firstly uninstall the previous version of Fastosh that you wish to replace. Please follow these steps to install a new version of Fastosh on your computer:

##### **Steps 0, 1, 2, & 3:**

Same as Steps 0, 1, 2, & 3 described above in Section 1.2.4.1.

##### **Step 4:**

This step can vary depending on the Fastosh update. If the update is relatively small, it is likely that the MATLAB Runtime will be the same between the old version of Fastosh that you want to replace and the one that you want to install. If it is a major update, the required MATLAB Runtime may be different between the two versions. For example, Fastosh versions 0.9.0 and 0.9.1 are both running with MATLAB Runtime 2020a, v98. Therefore, if you upgraded from 0.9.0 to 0.9.1, you would not need to upgrade the MATLAB Runtime.

At this step, the installer window displays what is the MATLAB Runtime version required to run the new version of Fastosh that you are installing. Also, it informs you whether this version of MATLAB Runtime is already installed on your computer. If it is the case, the installer window displays a message similar to: “MATLAB Runtime version vXXX is already installed at this location:

“usr/local/MATLAB/MATLAB\_Runtime/vXXX”.

In that case, there’s nothing to do: just click on “Next” button at the bottom of the screen. Alternatively, if Fastosh requires a new version of Matlab Runtime, you must specify at this step where the new Matlab Runtime will be installed. The default location displayed in the window should be similar to:

“usr/local/MATLAB/MATLAB\_Runtime/”

This default location should be adequate. If you’re OK with this choice, click on the button “Next” at the bottom of the window.

##### **Step 5**

Same as Step 5 described above in Section 1.2.4.1.

##### **Step 6**

Do this step only if the Fastosh update required installing a new version of MATLAB Runtime. This is the same as Step 6 described above in Section 1.2.4.1.

##### **Step 7**

Uninstall from your computer the previous version of Fastosh. For example, if the old version of Fastosh that you wish to delete is located in “/usr/Fastosh\_v0.9.0”, open a root terminal and type:

```
rm -r /usr/Fastosh_v0.9.0
```

Also, you may have to uninstall, using the same method (for example: `rm -r /usr/local/MATLAB/MATLAB_Runtime/v97`), the previous MATLAB Runtime in case you just have installed a new Runtime version, as discussed at Step 4 above.

That's it!

#### 1.2.4.3 Uninstalling Fastosh on Linux

As discussed in paragraph 1.2.1, running Fastosh on your computer implies having both the programs Fastosh and "MATLAB Runtime" installed. Therefore, these two programs must be removed to entirely uninstall Fastosh. However, if you wish to install a newer version of Fastosh later on, you may not need to remove your MATLAB Runtime if the latest release of Fastosh uses the same version of MATLAB Runtime than the one already installed on your computer.

For example, if Fastosh is located in `/usr/Fastosh`, open a root terminal and type:

```
rm -r /usr/Fastosh
```

Also, if Matlab\_Runtime is located in `/usr/local/MATLAB/MATLAB_Runtime`, type in the root terminal:

```
rm -r /usr/local/MATLAB/MATLAB_Runtime/
```

## 2 Basic Functionalities

### 2.1 Purpose of Main GUI

The Main GUI appears once the program is started. As mentioned in the first part of this manual, the main purpose of Fastosh is to provide new useful functionalities to process XANES or EXAFS data. These functions are available in individual modules accessible via the menu bar on top of the main GUI of Fastosh (Figure 1). The raw XAFS spectra collected at the beamline are firstly imported to the Main GUI and pre-processed, i.e. normalized and background subtracted to extract chi spectra featuring the EXAFS. The Main GUI can also be used to visualize the data in multiple ways, including 3 dimensions.

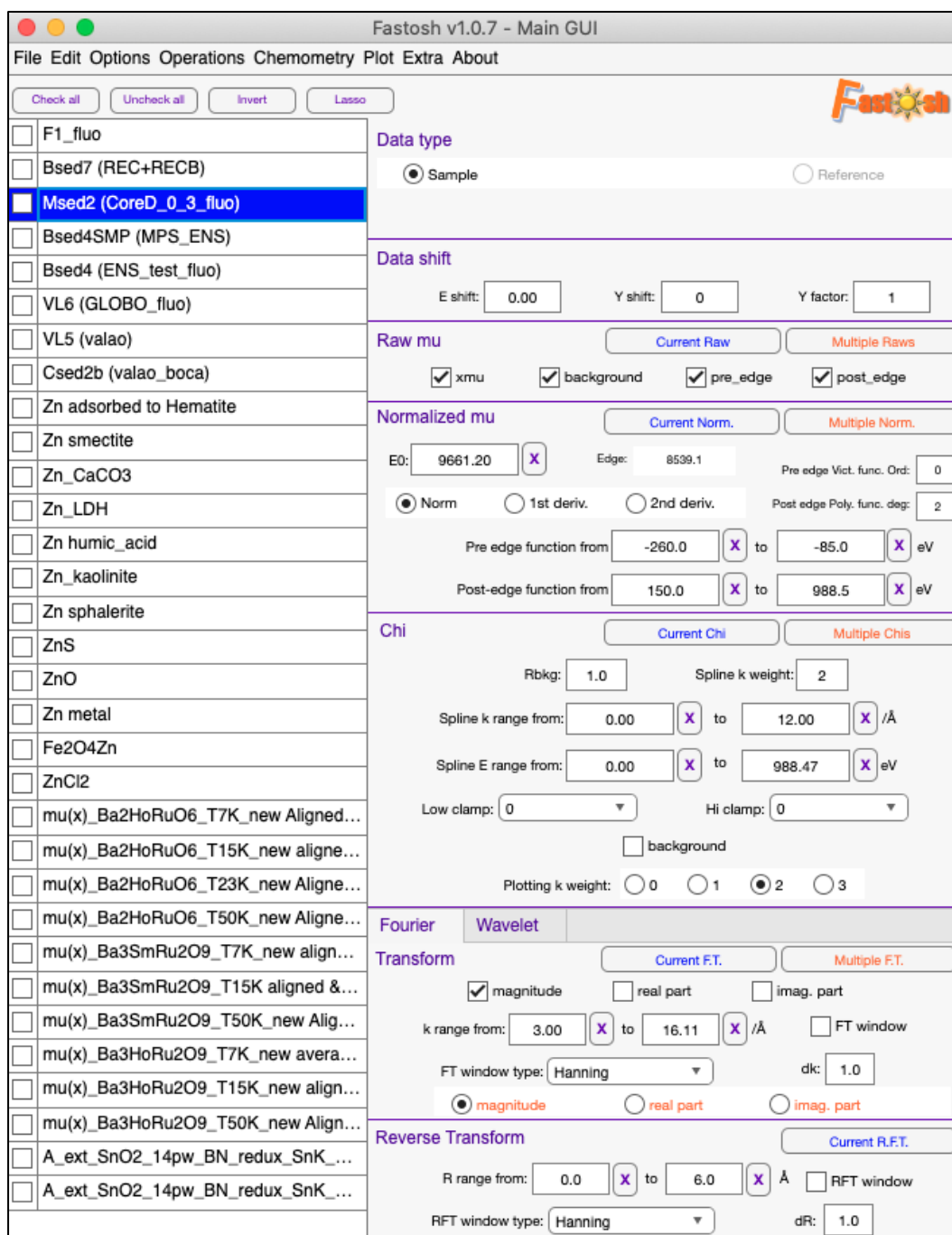


Figure 1 -Main GUI of Fastosh

All parameters displayed on the right side of the Main GUI in the fields “Data Type”, “Data Shift”, “Normalization”, “Background Removal”, “Fourier Transform”, “Reverse Fourier Transform”, and “Wavelet Transform” are scan-specific and correspond to the scan whose name is highlighted in blue in the sample list on the left side of the main GUI. For example, all scan parameters displayed on the right side of the window shown in Figure 1 correspond to the scan “sample\_inc2\_KT6\_ac”. The next section provides information on each parameter and functionally featured in the Main GUI.

## 2.2 Two methods to find E0

When importing an XAFS spectrum to the Main GUI, the e0 value can be found via two methods: it can be automatically determined using the Larch function “find\_e0” (i.e. automatic method) or constrained based on a user-defined theoretical value, which can be set before data importation (i.e. manual method).

In most cases, the automatic method is suitable to find the e0 value. In a few cases, however, it is not. For example, if an XAFS spectrum features a very small edge-jump and poor signal-to-noise ratio, the automatic method may find the e0 value at an absurd position in the XAFS spectrum. Of course, the e0 value automatically found can be manually readjusted after data importation to the Main GUI. However, if data similar to this spectrum is repeatedly imported to the Main GUI, it is convenient to constrain the e0 to a specific value before importing the data. This avoids manually readjusting the e0 of each spectrum after importation.

- The automatic method (Larch method, Default choice) can be selected via:

Main GUI Menu > Options > E0 Determination > “Automatic”

With this method, the vector dmu is firstly calculated:

$$dmu = \frac{\text{1st derivative of mu array}}{\text{1st derivative of energy array}}$$

Then, the maximum value in dmu is found, excluding the first and last data points in the dmu array. The index (position in the array) of the maximum value in dmu corresponds to the index of e0 in the energy array:

$$E0 = \text{energy}(\text{index}_{\max dmu})$$

- The manual method can be selected via:

Main GUI Menu > Options > E0 Determination > “Manual”

If this menu option is selected, a window appears where E0 can be constrained to a specify value. The final E0 value of a spectrum imported to the Main GUI corresponds to the closest energy value, among all energy values of the spectrum, to this constrained E0 value.

## 2.3 Importing data to the Main GUI

### 2.3.1 Drag and Drop

Since Fastosh version 1.0.6, a “drag and drop” functionality enables to directly take a spectrum file from a folder and throw it to the program to open it. There are two possibilities:

-If a file is dragged & dropped for the first time, the Drag & Drop’s memory is empty, then as usual the program will request specifying the nature of the data, in the ASCII file opener window (case of text file), or HDF opener window (case of HDF file).

-If a file has already been dragged & dropped, then the next dragged & dropped file will be imported following the same way as the one of previous data importation. If the data needs to be

respecified for next drag and drop action, select “Clear Drag and drop memory” in Main GUI’s menu>File> “Drag & Drop: clear memory”.

## 2.3.2 Importing any file containing XAFS data, or other types of data

As described in the following sections, Fastosh can import to the Main GUI data saved in ASCII files, HDF files (since Fastosh v1.0.10, it can be hdf5 file generated anywhere, not only SAMBA), Fastosh project file, or Athena project file. In any cases, go to the bar menu and select the file to import via:

“Main GUI Menu \ File \ Open file(s)...”

### 2.3.2.1 XAFS data from ASCII files

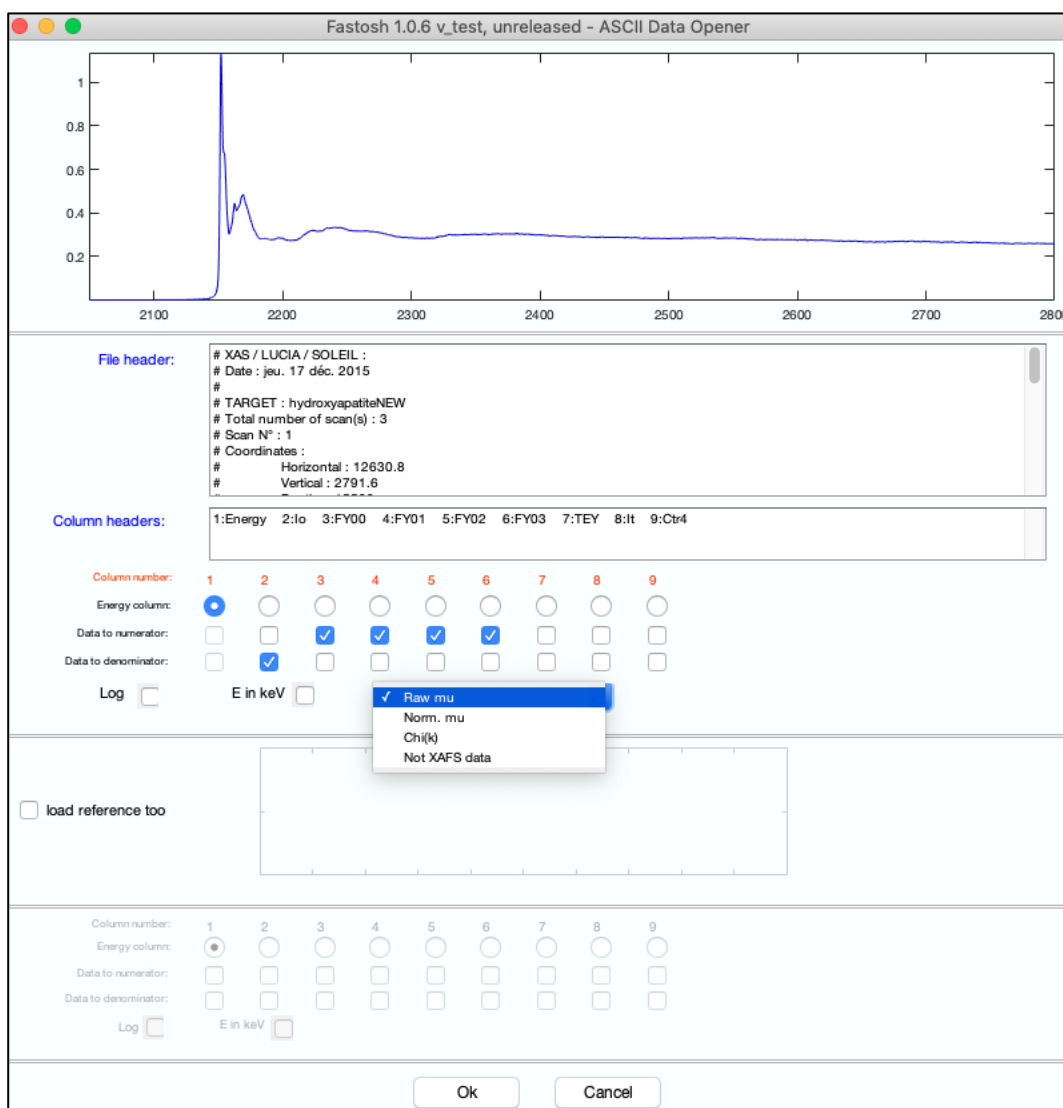


Figure 2 – Importation of ASCII data generated at LUCIA beamline, Synchrotron SOLEIL

Fastosh can open ASCII file generated at any XAFS beamline as long as the file extension is “.dat” or “.txt”. When importing this type of file, a Data Opener GUI appears (Figure 2). In this window, firstly select the data type: ‘Raw mu’ (the raw data collected at the beamline without post-acquisition data treatment applied to it), ‘norm. mu’ if the mu has already been treated to achieve an edge-jump equal to 1, or chi spectra. Secondly, select the appropriate channel(s) to open the data. The “Column headers” of the ASCII file shown in the window may help identify find the correct channels to select. For example, Figure 2 shows the opening of data collected at LUCIA beamline, Synchrotron SOLEIL. This data was collected with a 4 pixel Si drift

detector. In this file, the energy array corresponds to the first data column, which is indeed selected by default as the energy column by the program. The fluorescence collected by the first, second, third, and fourth pixels corresponds to columns 3, 4, 5, and 6, respectively. The I0 channel corresponds to column 2. Therefore, to open the total normalized fluorescence data (total  $I_f/I_0$ ), the data found at columns 3, 4, 5, 6 must be checked as numerator, and data found at column 2 must be checked as denominator, as shown in Figure 2. Once the correct channels are selected, click OK.

In this window, the energy is by default in eV, but can be alternatively defined as keV, for example to open ASCII files generated at BM30B/FAME beamline, ESRF.

To also import the spectrum of the reference associated with the sample (this must imply that a reference, e.g. metal foil, was simultaneously analyzed along with the sample and thus the data corresponding to the reference is also available in your ascii file), click in the box "load reference too". This activate a set of data columns relative to the reference at the bottom of the window. Check the box(es) in these columns corresponding to the data of the reference.

When all done, click on "OK" to close the window. At this point, Larch functions automatically perform a number of operations, including normalization and extraction of the EXAFS using the data corresponding to the sample and optionally its associated reference if the latter is simultaneously imported. If both sample & reference data are imported, the data related to the sample will be shown by default in the Main GUI of the program. To display the data related to the reference, see Section 2.4.2.1.

### 2.3.2.2 Non-XAFS data

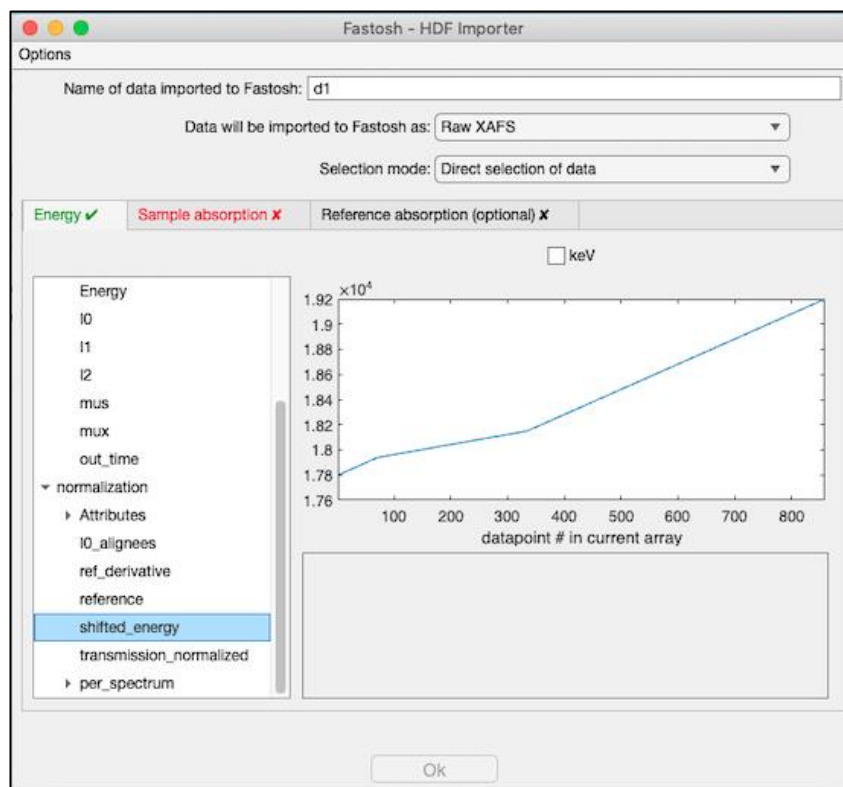
Since Fastosh v.1.0.6, the program can import dataset that are not necessarily XAFS spectra: RAMAN, FTIR, XRD data can be imported as long as the ASCII files are similar to the XAFS one: there must be a data column for the X axis (e.g. absorbance or 2theta) and Y axis (e.g. intensity of raman, FTIR, or diffraction spectra). Once imported to the main GUI, the data can be pre-processed (see Section) and then treated by MCR-ALS.

## 2.3.3 Importing data from any HDF file, other than those generated at SAMBA

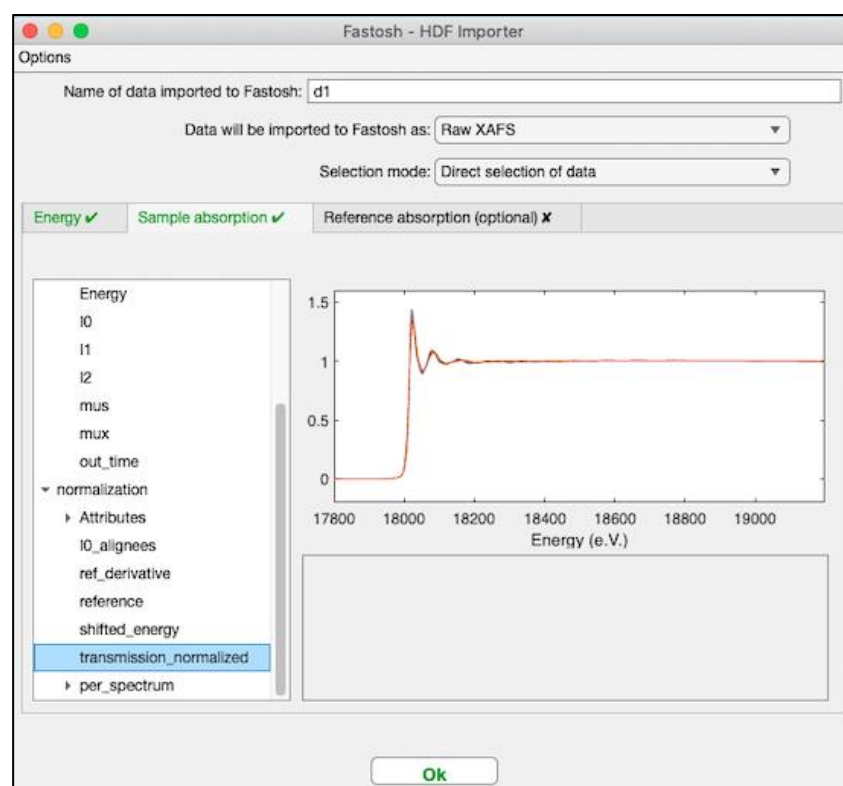
Since Fastosh v.1.0.10, data saved in HDF files generated anywhere can be imported to Fastosh, as long as they have a HDF5 format (i.e. .h5; .hdf5; .he5, or .hdf). Two modes are available to select the absorption, as detailed in the following two sections. If the HDF file contains hyperspectral data, corresponding to a 2D area probed via a spectroscopic approach (e.g. hyperspectral XAFS mapping), it can be also imported to Fastosh; this case is described in the third section below.

### 2.3.3.1 Direct selection of the absorption

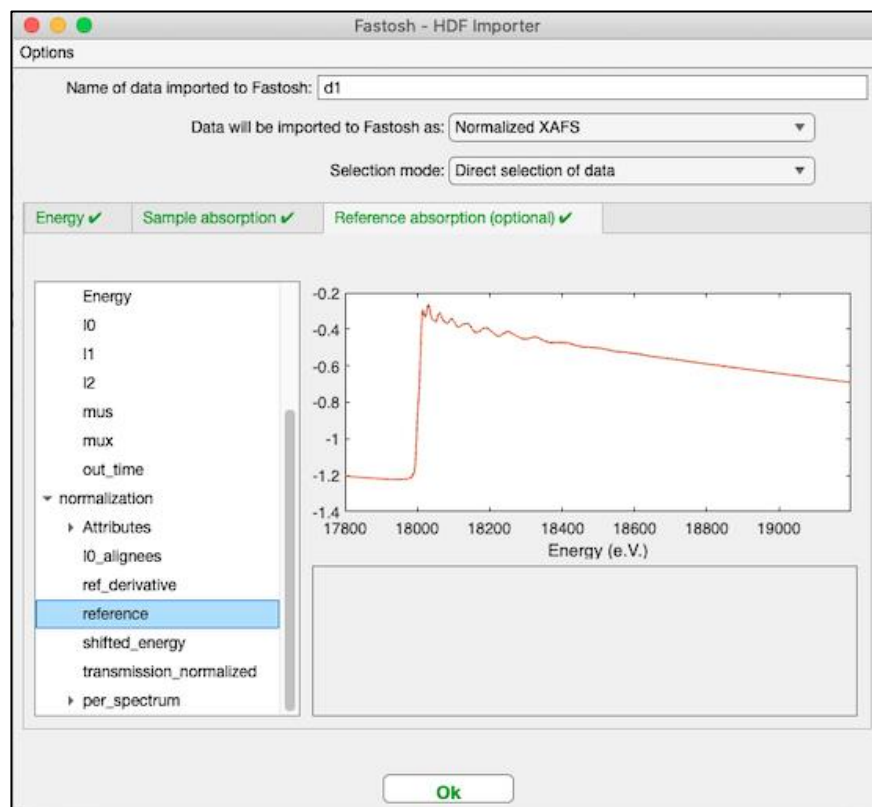
If the absorption saved in the HDF file is already normalized by I0 (i.e.  $\log(I_0/I_1)$  or  $I_{\text{fluor}}/I_0$  in case of transmission or fluorescence data, respectively), select the selection mode "Direct selection of data" (Default choice) featured in the HDF Importer Window. Then, simply click on the specific field within the HDF directory corresponding to the energy array of the XAFS spectrum to import (for example, this is the field "shifted\_energy" in the example below):



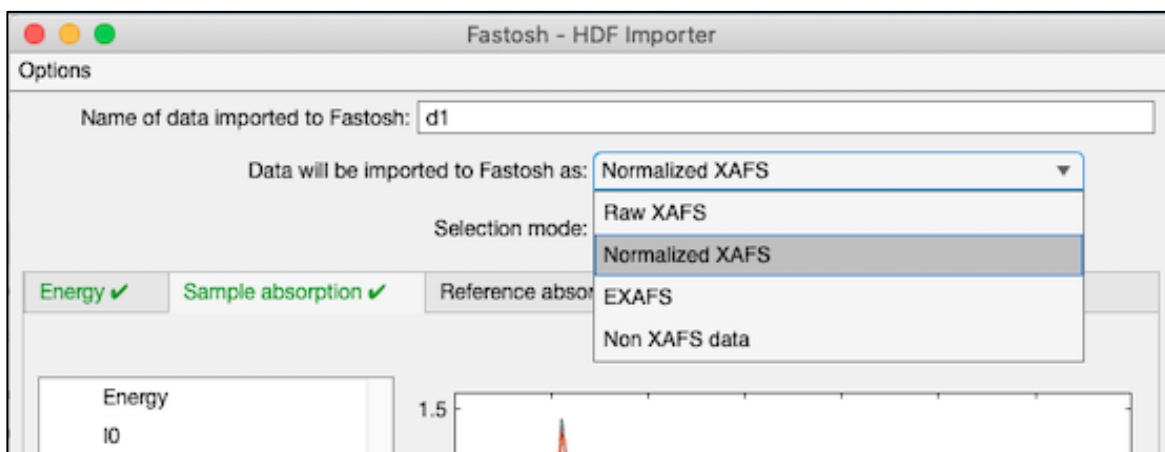
Then, in the tab “Sample absorption”, select the field corresponding to the absorption of the XAFS spectra to import. In the case below, the field called “transmission\_normalized” corresponds to multiple XAFS spectra, hence all of them will be imported to the Main GUI:



Optionnally, the reference data associated with the sample data can be imported as well:

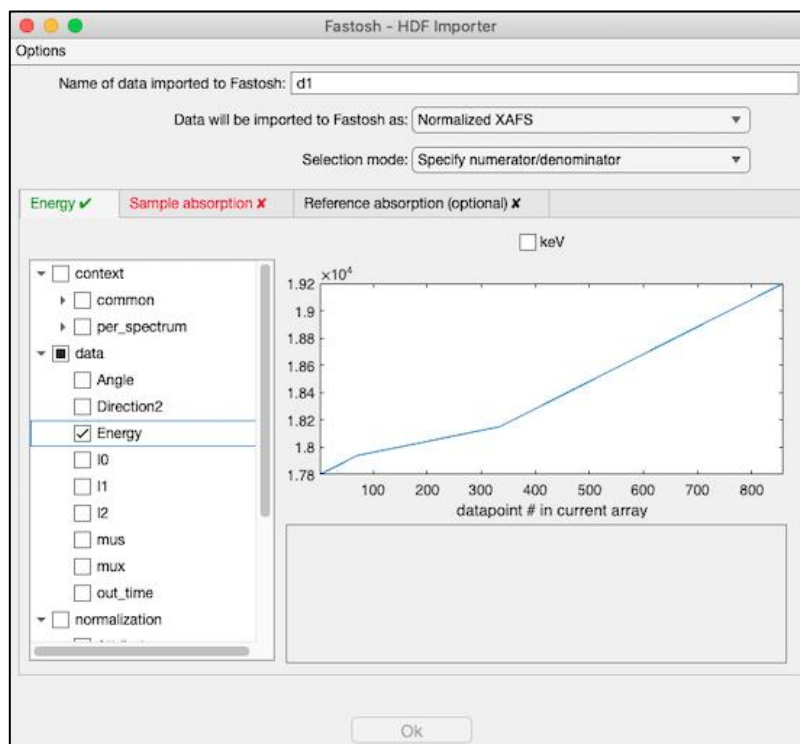


Also, the data type to import to Fastosh can be specified, given that it is declared as “Raw XAFS” data by default:

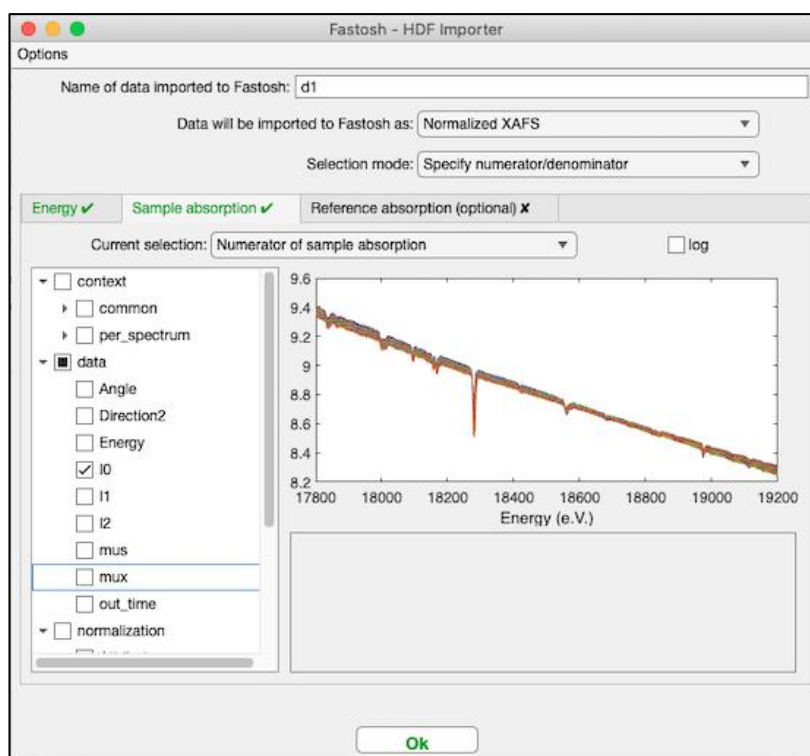


### 2.3.3.2 Specifying the numerator & denominator of the absorption expression

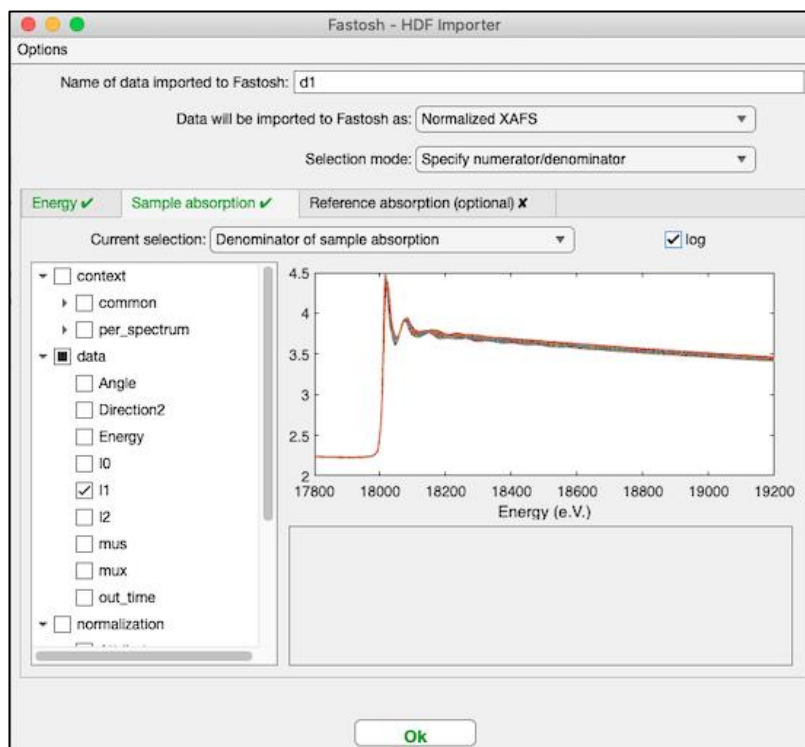
If only I0, I1, and the fluorescence are saved in the HDF file, the absorption can be defined using the Selection mode “Specify Numerator/Denominator”. Firstly, in the energy tab, **check** the box corresponding to the field of the HDF file where the energy array is saved:



Then in the “Sample absorption” tab, specify firstly the numerator of the absorption expression. For example, in case of transmission, check the field corresponding to I0:



Next, change “Current Selection” to “Denominator of sample absorption” in order to define the denominator of the absorption expression. Also, check the box “log”, so that the absorption, collected in the example below in transmission, can be defined as  $\log(I0/I1)$ :

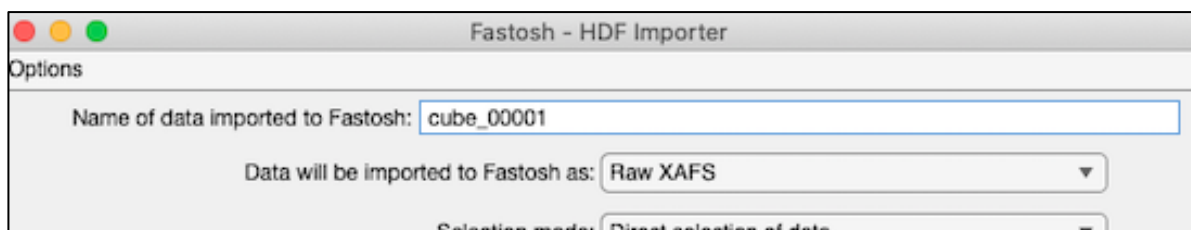


### 2.3.3.3 Importing hyperspectral data

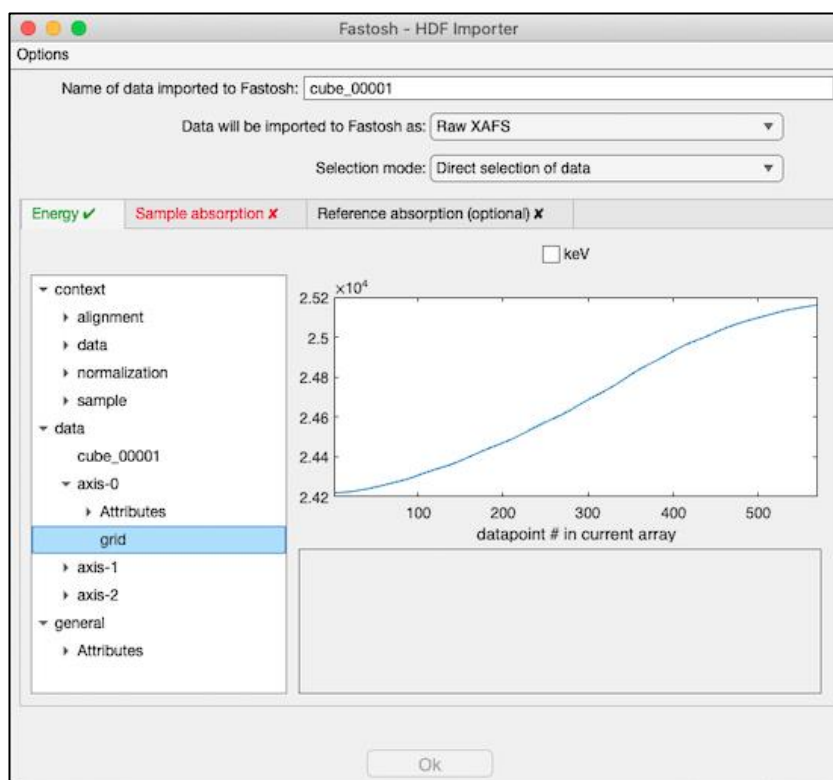
Below is an example of an HDF file generated at ROCK beamline (Synchrotron SOLEIL) containing hyperspectral XAFS mapping data. The name of this file is very long:



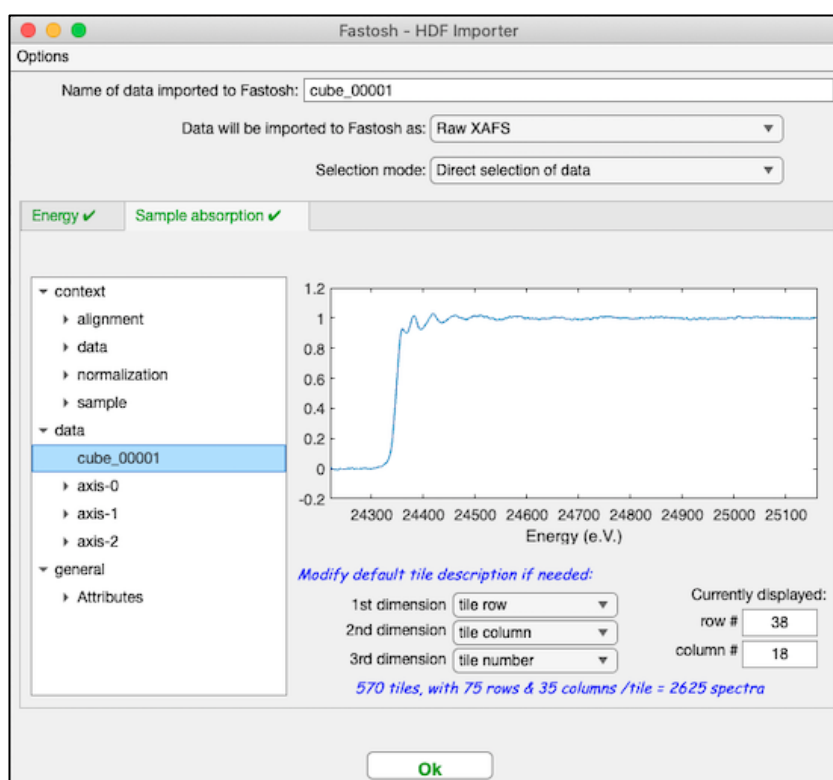
This name can be redefined to a shorter version (e.g. "cube\_0.0001"), which will be more convenient to display in the sample list of Fastosh Main GUI, once data importation will be completed:



To select the hyperspectral data, firstly the energy array is selected from the Energy tab:



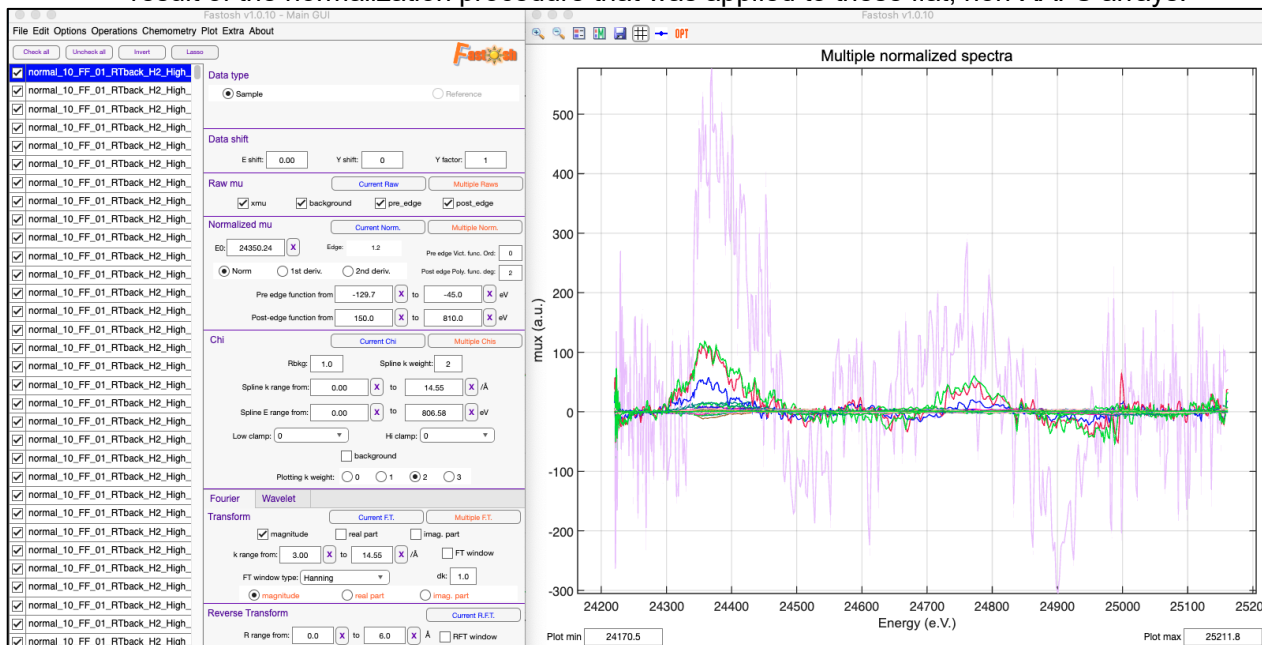
Then, the absorption, saved as 3D array, is selected from the tab “Sample absorption”:



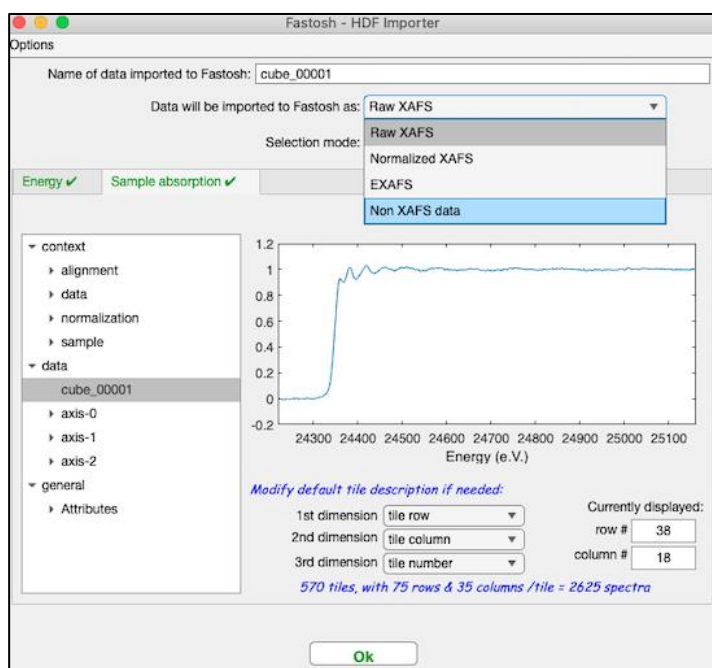
For example, in the example below, the property “cube\_00001” corresponds to the absorption, and it is saved as 3D matrix in the HDF file. By default, Fastosh defines the third dimension as the tile direction; the first and second dimensions are then the width and height of the 2D probed area, respectively. This can be refined in the HDF importer window. Also, the window displays by default the data corresponding to a pixel located in the middle of the 2D probed area. For example, if the 2D probed area was 75x35 (HxV) in size (i.e. the example above), the program shows by default the pixel 38 and 18 on the horizontal and vertical directions of the 2D area, respectively.

### 2.3.3.4 Trick for successful MCR-ALS processing of hyperspectral data in Fastosh

The hyperspectral data may have been preprocessed prior to its saving into the HDF file. For example, the hyperspectral XAFS dataset was normalized (edge-jump equal to 1) before being saved in the HDF file of ROCK beamline shown in the previous example. This pre-processing procedure was applied to **all individual pixels** featured within the 2D area probed by hyperspectral XAFS mapping, even when no XAFS spectrum was actually collected at some of these pixels (typically, those corresponding to the background on the edges of the 2D area, outside of the object). As a result, some of the hyperspectral spectra imported to Fastosh may be totally aberrant, as a result of the normalization procedure that was applied to these flat, non-XAFS arrays:

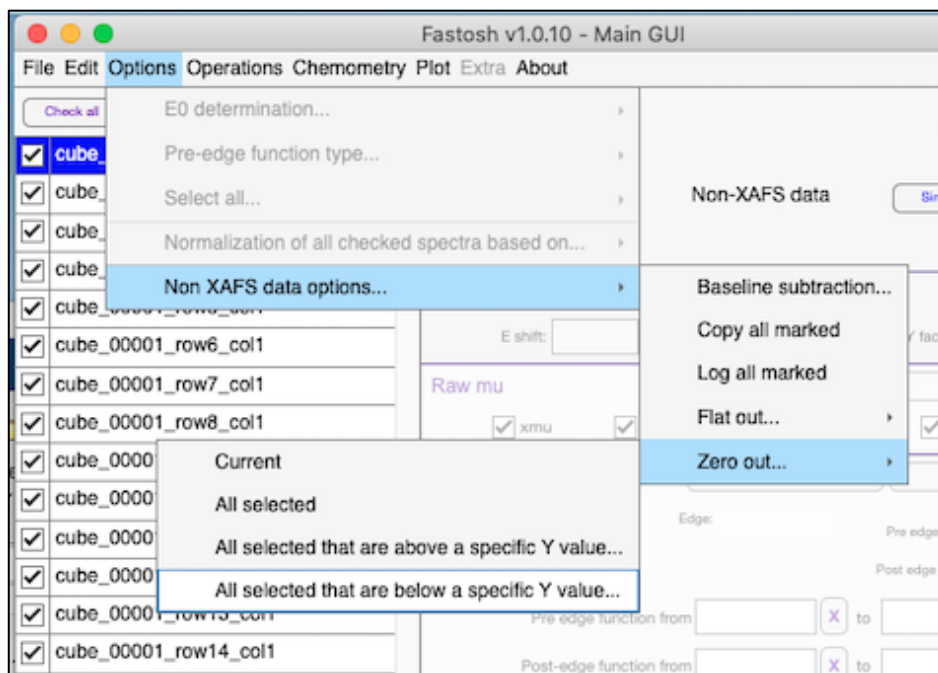


If the goal was to perform MCR-ALS on this dataset, **the aberrant spectra should not be deleted from the Main GUI**, since exactly 2625 (75 x 35) spectra must be provided to the MCR-ALS toolbox, even if some of these spectra did not correspond to actual normalized XAFS spectra. Otherwise, the MCR-ALS toolbox would not be capable to extract the 75 x 35 distribution map of the pure species, since some of the pixel data would be missing. Therefore, **the trick is to import this dataset not as XAFS data, but as “Non XAFS data”**, which is possible in the HDF importer window:

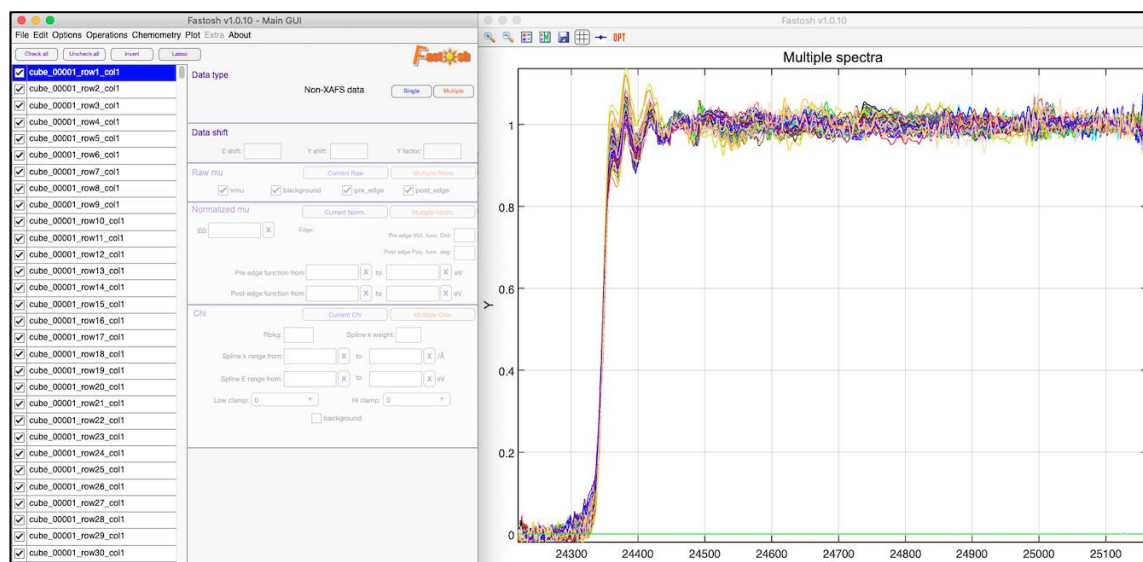


Once all these “Non-XAFS data” spectra are imported to Fastosh, select via the Menu bar of the Main GUI:

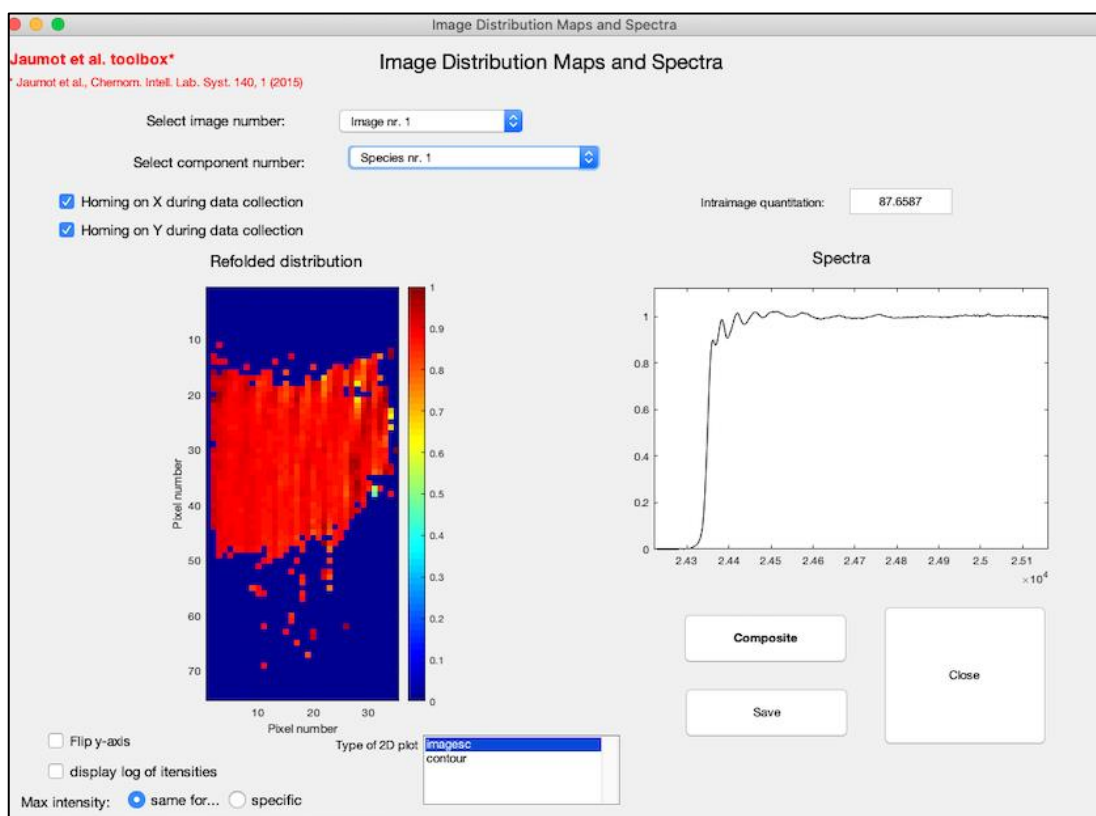
“Main GUI Menu \ Options \ Non XAFS data options \ Zero out \ All selected that are below a specific Y value...”



This will allow to determine a threshold value for which all spectra, where at least one of its data points is below this threshold, will be entirely zeroed-out. For example, setting the threshold to -0.02 results with the dataset previously shown:



With this threshold, some spectra (the aberrant ones) were now equal to zero throughout the entire energy range, whereas the number of spectra in the sample list were still 2625 since no spectrum was deleted. This subsequently allowed in Fastosh to successfully extract the 75 x 35 pixel map by MCR-ALS:



### 2.3.4 Importing data from Athena project

Since Fastosh v.1.0.9, data saved in an Athena project can be imported to the Main GUI. When importing such file, a small GUI appears enabling to import all files saved in the Athena project or a selection of them.

### 2.3.5 Importing SAMBA HDF file (SAMBA data only): many benefits over SAMBA ASCII file

Data files generated at SAMBA beamline currently come in ASCII and HDF formats. Therefore, to import SAMBA data to Fastosh, one could open the SAMBA ASCII files following the method described in Section 2.3.2. However, opening SAMBA HDF files instead of ASCII gives multiple benefits to SAMBA Users employing Fastosh:

- The overall time spent to upload data in Fastosh can be faster when opening SAMBA data from HDF files compared to ASCII files.
- When uploading, one does not have to specify what is the transmission or fluorescence data corresponding to the sample or reference among all data columns in the file since Fastosh already knows where to find the data in the SAMBA HDF file. When uploading a SAMBA HDF file, one can directly select the transmission or fluorescence data of the sample, or upload the reference as a sample (Figure 3). If uploading the transmission or fluorescence data of the sample, a checkbox at the bottom of the Window Data Opener, "Include reference", allows to import the data corresponding to the reference in addition to the sample data (Figure 3).
- When importing data to Fastosh from "Menu>file>Open file(s)..." one can select multiple ASCII or HDF files to upload at the same time. This requires, if the files to open are in ASCII format, to manually select all files at the same time, including all iterations corresponding to a given sample. In contrast, with SAMBA HDF files, one only has to select a single iteration of each sample to import and then click on the checkbox "Open all scans in data folder associated to each selected

file” (Figure 3). As explicitly stated, this automatically opens all iterations associated to each sample selected.

For example, 100 and 20 scans were collected for “Sample\_A” and “Sample\_B”, respectively. To upload all these data using ASCII files, one must manually select all 120 scans. In contrast, with SAMBA HDF files, one only has to select a single scan of Sample\_A and Sample\_B (whatever iteration of each sample, e.g. Sample\_A\_0042.hdf & Sample\_B\_007.hdf). If the first box in the opening window is checked, all 120 scans will be imported into the Fastosh Main GUI.

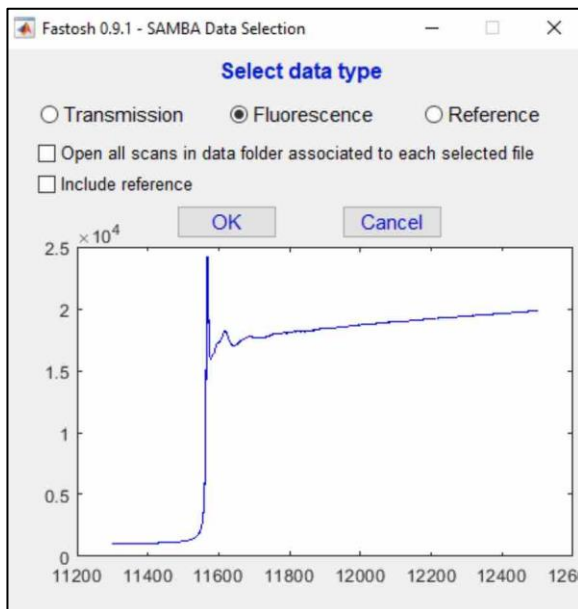


Figure 3– Importing data from a SAMBA HDF file

- if transmission or fluorescence data is imported to Fastosh using SAMBA HDF file, the type of acquisition mode will be displayed in the Main GUI in the field “Data Type” (Figure 4A). In contrast, the field “Data Type” in the Main GUI will simply display “Sample” for data imported to Fastosh as ASCII file, regardless of the type of acquisition mode that was employed to collect the data (Figure 4B).

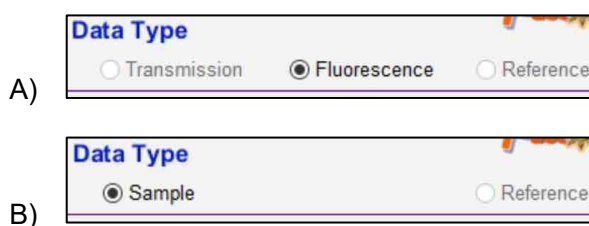


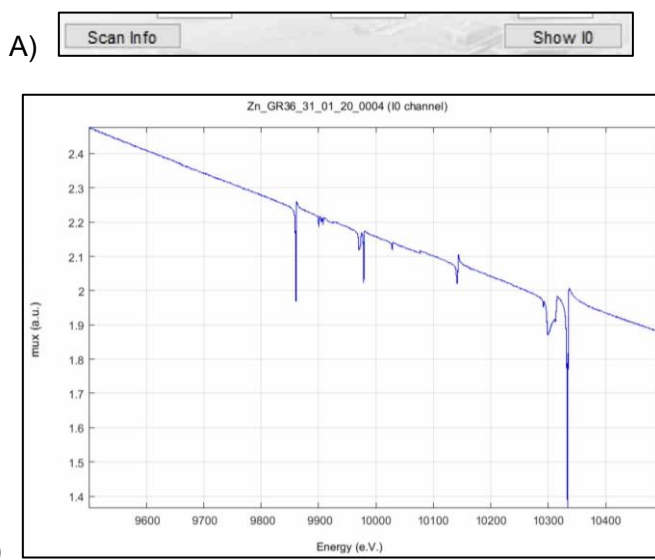
Figure 4– Fluorescence data of a sample uploaded in Fastosh from A) SAMBA HDF file and B) ASCII file

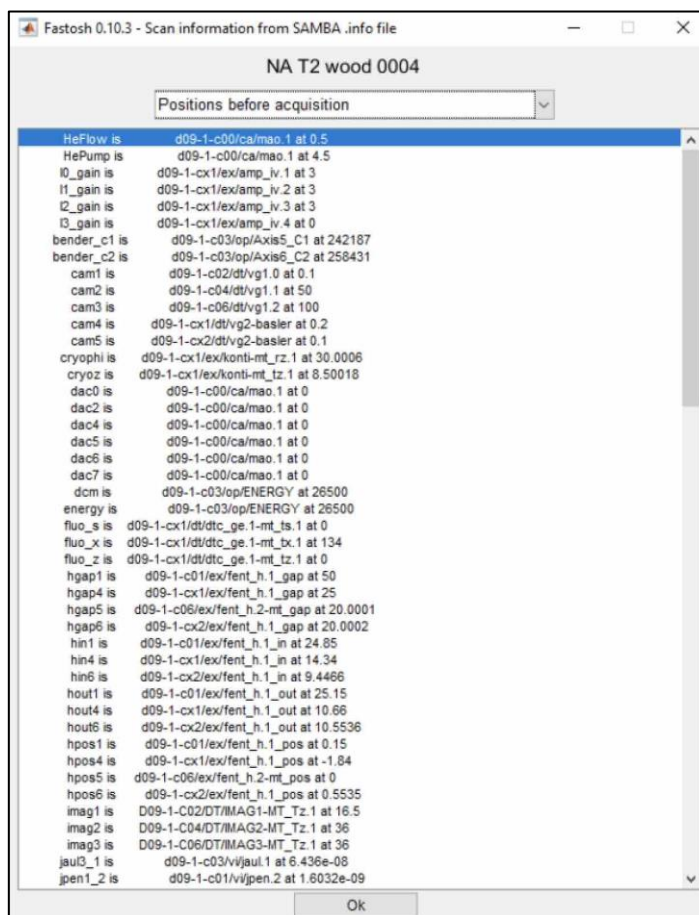
- if sample data is imported to Fastosh from a SAMBA HDF file, a button called “Show I0” will appear in the field “SAMBA information” of the Main GUI (Figure 5A). This button allows to display, in a new window, the I0 channel data of the selected sample (Figure 5B). Secondly, an HDF file allows to access to all contextual information related to SAMBA beamline setup. The type of information available in the HDF file depends on the type of data acquisition code employed at SAMBA on the day of beamtime. The SAMBA acquisition code underwent a major upgrade in fall 2020. Since Summer 2020, a new acquisition code called “Pulse” has been used at the beamline. With data generated at SAMBA prior to mid 2020, a button called “Scan info” will appear at the top-left corner of the synchrotron picture in the Main GUI (Figure 5A) only if the SAMBA INFO file generated at the beamline is present in the directory containing the HDF file of the current scan.

This button allows to display the content of the SAMBA INFO file, which gathers all motor positions at the onset of data acquisition for the selected sample (Figure 5C).

If the HDF file was collected at SAMBA in Fall 2020 or afterwards, the button called “Scan info” appears in the field “SAMBA information” of the Main GUI (Figure 5A). This button opens up a window that can display the...

- ...positions of all motors before the beginning of the scan
- ...positions of all motors after the scan is completed
- ...scan parameters
- ...dcm parameters (for local contact)
- ...mostab parameters (for local contact)





C) Figure 5— A) Buttons in main GUI, to visualize B) I0 channel of the sample or C) Contextual information related to SAMBA beamline

## 2.4 Basic data treatment

When importing a scan to Fastosh following the steps described in Section 2.3, Larch functions automatically normalize the imported raw mux spectrum, extract the EXAFS spectrum following a background subtraction procedure, and estimate random noise. The scan parameters obtained from these operations are then displayed in the Main GUI at the very last step of data importation. These scan parameters include E0 value, energy limits of pre-edge, post-edge, and spline functions, as well as a suggested kmax value to do the Fourier Transform of the EXAFS. The next sections describe how to visualize the data imported to the Main GUI and how to refine each scan parameter displayed in the Main GUI.

### 2.4.1 Data handling in Main GUI

#### 2.4.1.1 Multiple selection in the sample list

Four buttons are available on the top left corner of the Main GUI to select/unselect multiple spectra in the sample list. Note that these four buttons are available next to the sample list in many modules of Fastosh.

##### 2.4.1.1.1 Check all, Uncheck all, Invert

The first three buttons on top of sample list are self-explanatory: press “check all” button to select all spectra in the sample; “Uncheck all” to unselect all spectra in the sample list; or “invert” to invert the spectra selection in the list.

#### 2.4.1.1.2 Lasso

The “Lasso” button enables to select a group of spectra in the sample list, by checking the first and last spectra of the group.

##### Example

Suppose a group of 30 spectra has to be selected from a sample list featuring 90 spectra.

- Press the button “Lasso”. This button now displays “Check 1<sup>st</sup>”.
- Select in the list the first spectrum of the group (spectrum 1 of the 30 spectra). Once selected, the name of the spectrum becomes pink as a reminder that this one was selected. The “lasso” button now displays “Check 2<sup>nd</sup>”.
- Select in the list the last spectrum of the group (30<sup>th</sup> spectrum of the 30 spectra). All 30 spectra are now selected in the list.

#### 2.4.1.2 Plotting data of one or multiple sample(s)

##### 2.4.1.2.1 Viewing one scan (the highlighted one in the sample list)

To visualize a particular scan imported to the program, click on its name in the sample list featured on the left side of the Main GUI. This should highlight with a blue color the scan name that you are interested in. From there you can show its:

##### - raw $\mu(x)$ :

Firstly, make sure “xmu” is checked in the “Raw  $\mu$ ” field of the Main GUI (Figure 1). Then, to show the data, click on the blue button “Current Raw” in the Main GUI.

##### - normalized $\mu(x)$ :

Click on the blue button “Current Norm.” in the “Normalized  $\mu$ ” field of the Main GUI (Figure 1).

##### - 1<sup>st</sup> derivative of $\mu(x)$ :

Check “1<sup>st</sup> deriv.” in the “Normalized  $\mu$ ” field of the Main GUI (Figure 1). Then, to show the data, click on the blue button “Current Norm.” in the Main GUI.

##### - 2<sup>nd</sup> derivative of $\mu(x)$ :

Check “2<sup>nd</sup> deriv.” in the “Normalized  $\mu$ ” field of the Main GUI (Figure 1). Then, to show the data, click on the blue button “Current Norm.” in the Main GUI.

##### - Chi (featuring the EXAFS):

Click on the blue button “Current Chi.” in the “Chi” field of the Main GUI (Figure 1).

##### - Fast Fourier Transform:

Click on the blue button “Current F.T.” in the Fourier Transform tab of the Main GUI.

##### - Reverse Fourier Transform:

Click on the blue button “Current R.F.T” in the Fourier Transform tab of the Main GUI.

##### - Wavelet Transform:

Click on the blue button “Current W.T.” in the Wavelet Transform tab of the Main GUI.

##### 2.4.1.2.2 Viewing multiple scans in 2 or 3D

To plot multiple scans imported to the program, check each scan in the sample list that you wish to simultaneously plot. Once all scans to be plotted are checked, you can show their:

##### - raw $\mu(x)$ :

Click on the orange button “Multiple Raws” in the Main GUI.

##### - normalized $\mu(x)$ :

Click on the orange button “Multiple Norm.” in the Main GUI.

##### - 1<sup>st</sup> derivative of $\mu(x)$ :

Check “1<sup>st</sup> deriv.” in the “Normalized  $\mu$ ” field of the Main GUI. Then, to show the data, click on the orange button “Multiple Norm.” in the Main GUI.

- **2<sup>nd</sup> derivative of  $\mu(x)$ :**

Check "2<sup>nd</sup> deriv." in the "Normalized  $\mu$ " field of the Main GUI. Then, to show the data, click on the orange button "Multiple Norm." in the Main GUI.

- **Chi:**

Click on the orange button "Multiple Chis" in the Main GUI.

- **Fast Fourier Transform:**

Click on the orange button "Multiple F.T.." in the Fourier Transform tab of the Main GUI.

## Multiple scans shown in 2D

By default, if any orange plot button (Figure 1) is pressed, multiple data is displayed in 2 dimensions. The "2D plot (Multiple Scans)" option is then checked by default in the menu:

Main GUI Menu > Plot > "2D plot (Multiple Scans)"

For example, if the orange button "Multiple Norm." (Figure 1) is pressed to display multiple normalized spectra, and the 2D plotting option is selected in the menu, one gets the following figure (Figure 6):

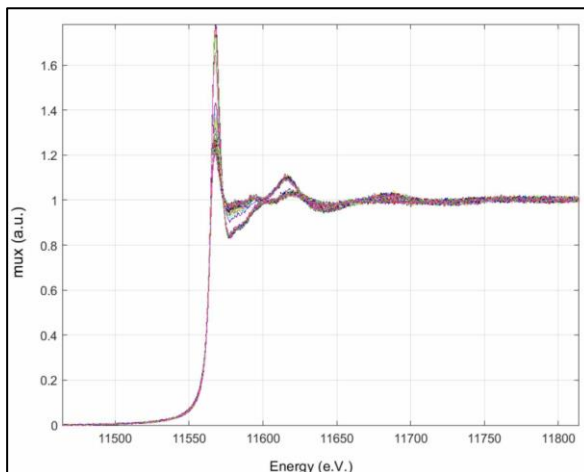


Figure 6- Example of multiple normalized spectra displayed in 2D

## Multiple scans shown in 3D

To display multiple scans, firstly select

Main GUI Menu > Plot > "3D plot, surface & lines (Multiple Scans)"

to show the data in 3D with a surface created, or select

Main GUI Menu > Operations > "3D plot, lines only (Multiple Scans)"

to show the data in 3D without creation of a surface.

**Note:** At this point, the data won't be displayed. Any orange plot button (Figure 1) must be pressed again to display in 3D the data corresponding to the checked scans.

For example, clicking on the orange button "mux" (Figure 1) with the surface 3D plotting option selected, one gets the figure below (Figure 7):

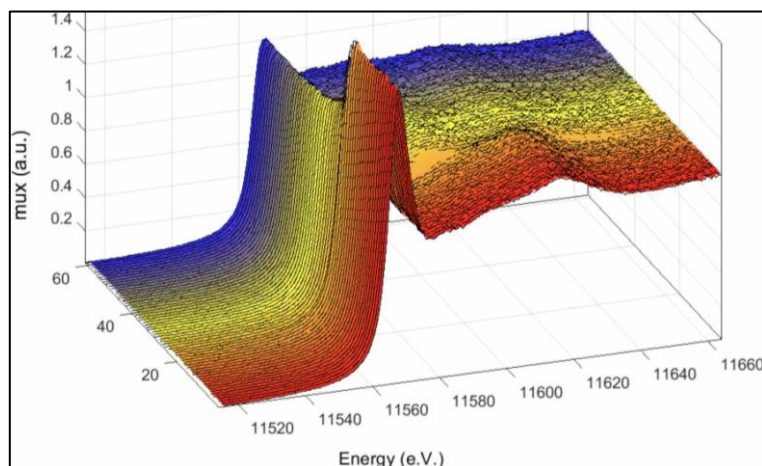


Figure 7- Example of multiple normalized spectra (same as those shown in Figure 6) displayed in 3D, using a three-color plot option

#### 2.4.1.2.3 Coloring multiple scans in 2 or 3D

##### -Number of colors

When plotting multiple scans in 2D or 3D, the data is displayed in rainbow colors (default option). The “Rainbow color plot in (Multiple Scans)” is then checked by default in the menu of Main GUI:

Main GUI Menu > Plot > “Rainbow color plot (Multiple Scans)”

The Rainbow color option is useful to display data in 2D, but is usually not the best option to display data in 3D. For 3D plots, a blend of two, three, or four colors is usually the best choice. To display multiple scans in 2, 3, or 4 colors, select in the menu of Main GUI, respectively:

Main GUI Menu > Plot > “Two color plot (Multiple Scans)”

Main GUI Menu > Plot > “Three color plot (Multiple Scans)”

Main GUI Menu > Plot > “Four color plot (Multiple Scans)”

##### -Type of colors

The color types used in the plot can be also modified from their default RGB values. To open the RGB selector window for 2, 3, or 4 colors, select in the Menu, respectively:

Main GUI Menu > Plot > “Two color selection... (Multiple Scans)”

Main GUI Menu > Plot > “Three color selection... (Multiple Scans)”

Main GUI Menu > Plot > “Four color selection... (Multiple Scans)”

This will open a window to select colors in Red Green Blue (RGB) format. For example, the window below appears when “Four color selection” is selected (Figure 8):

	1	20	40	60
R	1	0.2	1	0.2
G	0	1	1	0.1
B	0	0.3	0	1

Figure 8 - Four-color selection window

In this window, each box column corresponds to the R, G, B values of a color.

The tile number above each column specifies which specific scan among the set of scans (i.e. tile # in the scan stack) the color is applied to. The first and last colors are always the first and last tiles. If, for example, the first color is tile #1, and the second color is tile #20, the colors of tile #2 to 19 will be a blend of the first and second colors.

- To edit a specific color, firstly select the corresponding column in the field "Apply to".
- Any color button on top of the window can be pressed to quickly select a color (black, blue, red, yellow, orange, etc...). For example, press "Red" to apply the R, G, B, values corresponding to the red color (i.e. R=1, G=0, B=0) to the selected column. Alternatively, you can edit each RGB value to customize a color.

#### Example:

A stack of XAFS spectra corresponding to an operando experiments carried out at the beamline is shown above Figure 7. In this stack, tile #1 to 20 were taken before the chemical reaction started. The reaction started at tile #21, and finished at tile #60. The choice of colors in this Figure is not ideal as these colors do not help differentiate between the time periods where the chemical reaction took place and before it started. To improve this figure, one can use the window shown in Figure 8 and do the following setting modifications:

	1	20	21	60
R	0.2	0.2	1	0.2
G	1	0.4	0	0.1
B	0.3	0.3	0	1

#### Procedure to get the plot below:

##### Step 0: Select 3D plot

In the Main GUI menu, select:

Menu > Plot > "3D plot, surface & line (Multiple Scans)"

##### Step 1: Select a four-color plot

In the Main GUI menu, select:

Menu > Plot > "Four color plot (Multiple Scans)"

**Step 2: Open the window to edit the colors and tile numbers**

In the Main GUI menu, select:

Menu > Plot > "Four color plot (Multiple Scans)"

**Step 3: Modify the RGB values of each column**

-Select the first column, and press "Green" button to apply RGB values corresponding to the green color to the first column.

-Select the second column, and press "Green" button to apply RGB values corresponding to the green color to the second column (you can make the second color a bit darker than the first color by setting G=0.4).

-Select the third column, and press "Red" button to apply RGB values corresponding to the red color to the third column.

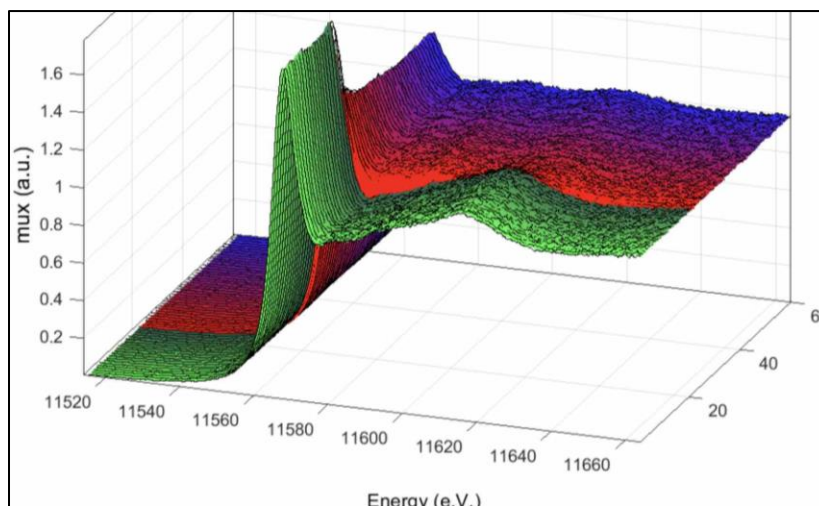
-Select the fourth column, and press "Blue" button to apply RGB values corresponding to the blue color to the fourth column.

**Step 4: Modify the tile numbers**

-Select the third column, and edit the tile number to 21.

**Step 5: Refresh the plot with the new plot settings**

-Press the orange button "Multiple Norm." (Figure 1) to refresh the plot and show the data with the new plot settings. The figure now looks like that:



**2.4.1.2.4 Stacking multiple scans in 2D**

To vertically scatter a set of scans in 2D (Figure 9B), firstly check all scans in the sample list to scatter and then go to:

Main GUI Menu > Plot > "2D stack plot... (Multiple Scans)"

. The vertical distance between each scattered scan must be specified in the field "Scatter selected by:" (Figure 9A). To vertically display two different sets of scattered scans, with a distinct space between the two sets to visually differentiate them, a vertical shift can be applied to all scans belonging to the second set. This value can be specified in the field "Shift selected by:" (Figure 9A).

A)

Scatter selected by	<input type="text" value="0.4"/>
Shift selected by	<input type="text" value="2"/>
<input type="button" value="Apply to selected"/>	

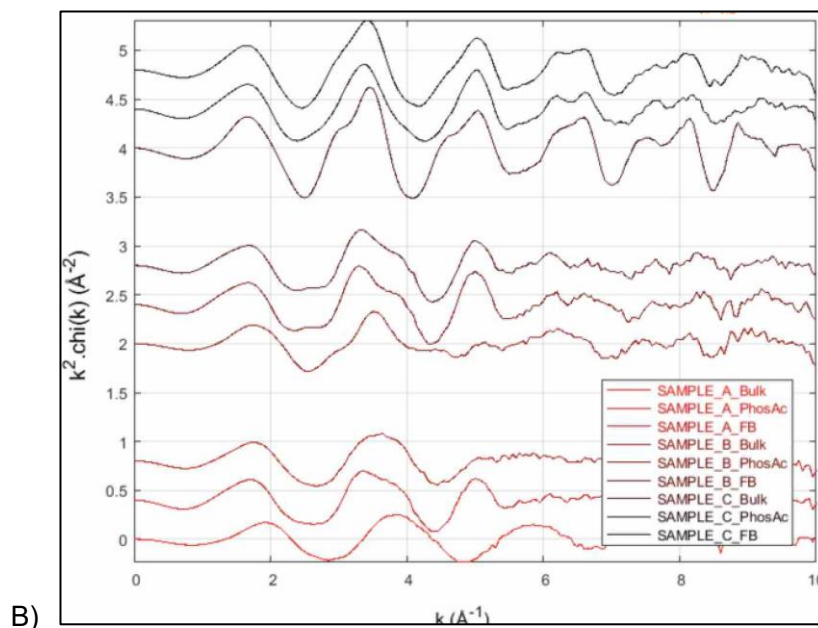


Figure 9- A) Stack plot options and B) an example of 2D data stack. NOTE: the legend can be moved anywhere in the plot by just clicking/holding the mouse on the legend and dragging it anywhere in the plot.

**Example:** The example of 2D stack plot showed above (Figure 9B) features 9 spectra. This data corresponds to three samples A, B, & C with three spectra per sample. This figure was obtained after doing the following steps:

- Select « stack » plot options
- Check the three scans of SAMPLE A in the sample list
  - Change the scatter plot options to these setting:
    - “Scatter select by:” 0.4
    - “Shift select by:” 0
  - Once these values are typed, press below them the button “Apply”
- Uncheck the three scans of SAMPLE A in the sample list
  - Check the three scans of SAMPLE B in the sample list
  - Change the scatter plot options to these setting:
    - “Scatter select by:” 0.4
    - “Shift select by:” 2
  - Once these values are typed, press below them the button “Apply”
- Uncheck the three scans of SAMPLE B in the sample list
  - Check the three scans of SAMPLE C in the sample list
  - Change the scatter plot options to these setting:
    - “Scatter select by:” 0.4
    - “Shift select by:” 4
  - Once these values are typed, press below them the button “Apply”
- Refresh the plot with the new plot settings
  - Check all spectra of samples A, B, and C
  - Press the orange button “Multiple Norm.” (Figure 1) to refresh the plot and show the data with the new plot settings.

#### 2.4.1.3 Normalization (Larch method)

The raw  $\mu(x)$  spectrum imported to the Main GUI must be normalized as a first step of the EXAFS extraction procedure. This enables, as mentioned in Section 2.4.1.5 on “Background removal”, to flatten the extracted  $\chi$  spectrum featuring the EXAFS and estimate the edge-step of the  $\mu(x)$  spectrum whose value is employed to normalize the  $\chi$  spectrum. The EXAFS is indeed broadly defined as *“the **normalized** oscillatory part of the absorption coefficient above the absorption edge”*.

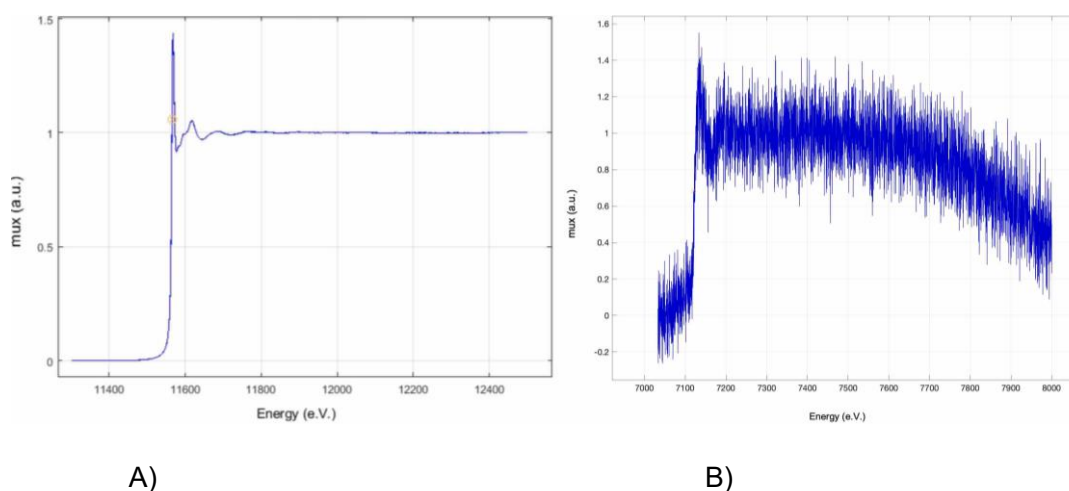
Normalizing the  $\mu(x)$  spectrum consists in fitting a pre-edge function (generally linear) and post edge function (generally a quadratic polynomial) to the part of the  $\mu(x)$  spectrum below and above the absorption edge, respectively. The latter is represented in the program by the E0 parameter. The edge-step is equal to the difference between the post-edge and pre-edge function at the energy of E0.

In many cases, the default normalization parameters automatically obtained following data importation to the Main GUI should be satisfactory, especially when the element of interest is not too diluted in the sample. However, in some cases the automatic normalization is not satisfactory, and the default normalization parameters must be corrected. It is then a good practice to systematically check the normalization right after importing a  $\mu(x)$  spectrum to the Main GUI, as described in the next section.

#### 2.4.1.3.1 Visually estimating the normalization

If a correct normalized is achieved, the pre-edge region of the normalized spectrum should be flat and equal to 0 throughout the Y axis. Similarly, the oscillating and flat parts of the post-edge region of the normalized spectrum should evenly oscillate around 1 and be equal to 1 on the Y axis, respectively. To verify that, firstly check the normalized spectrum by pressing the blue button "Current Norm." in the Main GUI. Examples of acceptable and unacceptable normalized spectra are shown in Figure 10A & B, respectively. The normalized spectrum shown in Figure 10B is unacceptable as its pre-edge part is not equal to 0 throughout the Y axis. Instead, the pre-edge part keeps on increasing above 0 with increasing energy. Additionally, the flat part of the post-edge region is not equal nor oscillate around 1 throughout the Y axis. It keeps on decreasing from 7600 eV to the end of the spectrum.

Visualizing the raw, unnormalized  $\mu(x)$  spectrum along with the fitted pre-edge and post-edge functions can help understand the cause of the normalization issue. Ideally, the pre-edge and post-edge functions should both fit well the experimental spectrum and be roughly parallel to each other, especially at the energy of E0 where the edge-jump is assessed. For example, these functions were more or less parallel to each other (Figure 11A) to obtain the well-normalized spectrum shown in Figure 10A. This was not the case, however, with the pre-edge and post-edge functions employed to obtain the spectrum not well normalized shown in Figure 10B. The pre-edge function indeed dropped with increasing energy instead of fitting the data of the unnormalized spectrum (Figure 11B). Similarly, the post-edge function was too curvy and far from following the data trend towards the end of the unnormalized spectrum (Figure 11B). This typically represents the case where the extraction parameters should be modified to optimize the normalization. The modifications of normalization parameters are discussed in the next section.



A)

B)

Figure 10 - Examples A) acceptable and B) problematic normalized spectra

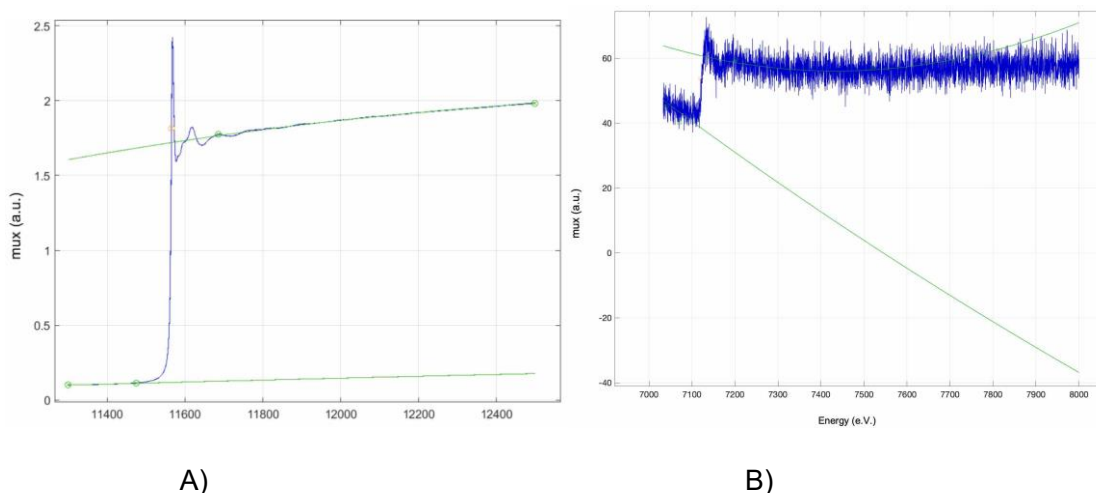


Figure 11 - Example of A) correct or B) problematic pre-edge and post-edge functions, which were used to obtain the normalized spectra shown in the previous Figure

#### 2.4.1.3.2 E0 readjustment

The minimum and maximum energy values of the pre-edge & post-edge functions are not defined as absolute energy values but relative to the energy position of the edge jump, which is represented in the program as the E0 value. One must make sure that the E0 value is adequate before adjusting the parameters relative to the pre-edge & post-edge functions. To do the normalization, the E0 value can be set anywhere between the beginning and end of the edge-jump, given that it may be necessary to fine-tune this parameter at a latter step of the data analysis, notably to perform linear combination fitting.

If the “automatic” E0 determination method (Section 2.2) is employed to find e0, the E0 value can be found at the maximum of the 1<sup>st</sup> derivative of the mux spectrum, which falls somewhere around the middle of the edge-jump. This E0 value should be ok to perform the normalization. It is, however, also possible that the normalization algorithm (“automatic” E0 determination method discussed in Section 2.2) does not correctly find the edge-jump and erroneously set the E0 value at an absurd position in the XAFS spectrum, e.g. in the middle of the EXAFS. This can be the case notably when the XAFS spectrum features monochromator glitches or diffraction peaks. It can also occur when the edge-jump is too small, which may imply that the element of interest is too diluted in the sample.

To readjust the E0, either type a new E0 value directly in the Main GUI or interactively choose it from the plot. This can be done by pressing the cross button next to the E0 value in the Main GUI and clicking on the desired energy position in the plot. There is a ten second-countdown to do this operation.

#### 2.4.1.3.3 Victoreen pre-edge function

A Victoreen function is used by default to fit the pre-edge region of the mux spectrum. This function is selected in the menu of the Main GUI when opening Fastosh:

Main GUI Menu > Options > Pre-edge function type... > “Victoreen”

The expression of a Victoreen function is:

$$\text{Pre\_edge} = a * E + b * E^{-\text{nvict}}$$

where E is the full energy array of the mux spectrum, nvict is the order of the Victoreen function, which is a user-defined parameter in the Main GUI window, and a & b are coefficients of a linear function fitted to satisfy this relationship:

$$a * E_{iEmin \text{ to } iEmax} + b = \text{mu}_{iEmin \text{ to } iEmax} * E_{iEmin \text{ to } iEmax}^{\text{nvict}}$$

where  $iE_{\min}$  &  $iE_{\max}$  are the array indexes in E of  $E_{\min}$  &  $E_{\max}$ , which are the minimum and maximum energy value of the pre-edge region, respectively.

In the Main GUI, the  $E_{\min}$  &  $E_{\max}$  values of the pre-edge region are defined in the fields “Pre-edge function from:” and “to:”. These are not defined as absolute energy values but relative to the energy position of the edge jump, which is represented in the program as the  $E_0$  value.

By default, the order of the Victoreen is equal to 0 (linear function), as shown in the field of the Main GUI: “Pre-edge Vict. func. ord.:”. This degree value should be adequate for any spectrum where the pre-edge region is quite flat. The Victoreen degree should be modified only if the pre-edge region is so curvy that a linear function cannot fit the data. This may occur for example when the tail of the Compton peak significantly leaks into the ROI of MCA where the fluorescence spectrum is extracted. In this case, the Victoreen degree should be increased; if this is still not satisfactory, a polynomial pre-edge function should be tested (see next section).

If the Victoreen pre-edge function does not fit well the data, a common cause is that the min and/or max values of the fitted pre-edge region are/is not optimized. The best approach is to empirically find in the pre-edge region the min and max values for which the function fits well the data. A good fit is obtained when the pre-edge region of the resulting normalized spectrum is flat and equal to 0 throughout the Y axis. For example, the default min and max of the pre-edge function used to obtain the unacceptable normalized spectrum shown above were -80 and -50 eV, respectively. After a few trial and errors, it was empirically found that slightly modifying the min & max to -100 & -30 eV, respectively, enabled to get a good fit of the pre-edge (Figure 12 & Figure 14B).

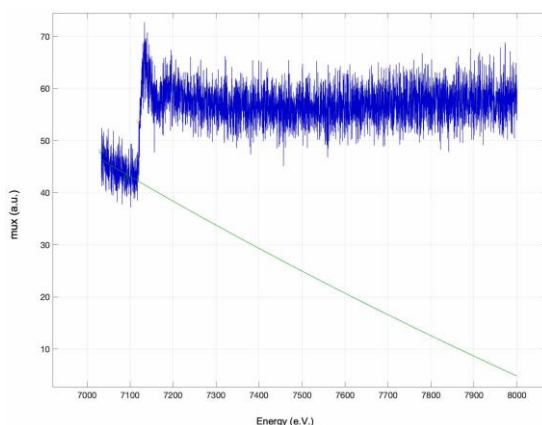


Figure 12 - Example of a linear function that fits well the pre-edge region.

To modify the min or max value of the pre-edge in the fields “Pre-edge function from:” and “to:”, respectively, either type a new value directly in the Main GUI or interactively choose it from the plot. This can be done by pressing the cross button next to the value in the Main GUI and clicking on the desired energy position in the plot. There is a ten second-countdown to do this operation.

#### 2.4.1.3.4 Polynomial pre-edge function

The pre-edge part of a mu spectrum acquired in fluorescence mode can be sometimes extremely curvy, such as the mu spectrum shown as an example in Figure 13A. This can occur when the tail of peak corresponding to the scattered emission leaks into the ROI of the MCA from where the mu spectrum is extracted, and especially when the scattered emission is much higher than the sample fluo emission reaching the detector. In this case, a victoreen function may not properly fit the pre-edge part of the mu spectrum, even with a function order higher than 0. A polynomial function can be then a better choice. For instance, the pre-edge part of the mu spectrum shown in Figure 13A was well fitted using a polynomial function of order 6, as demonstrated by the decent normalized spectrum obtained (Figure 13B). This type of pre-edge function can be selected from the menu of the Main GUI:

Main GUI Menu > Options > Pre-edge function type... > “Polynomial”

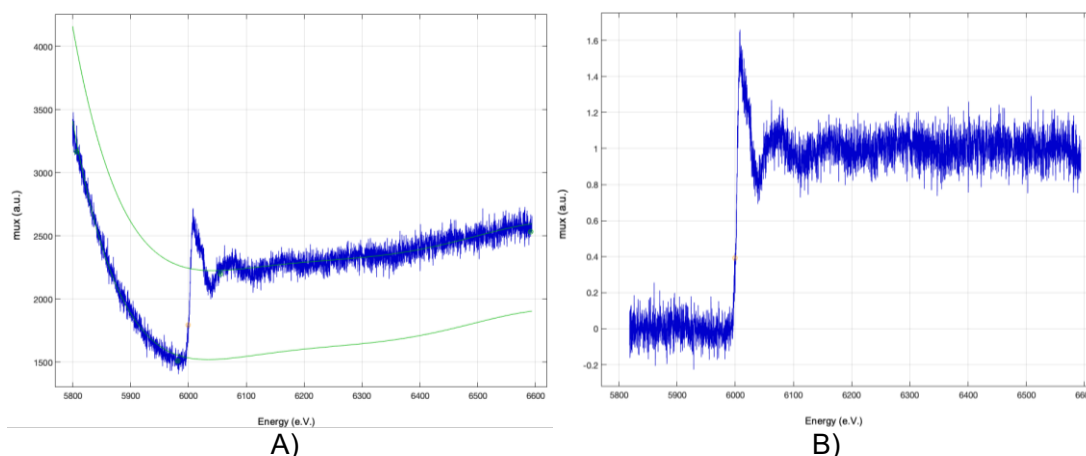


Figure 13 – (a) Example of polynomial pre-edge function of order 6 applied to a mu spectrum and (b) resulting normalized spectrum

By default, the order of the polynomial is equal to 1 (linear function). The order value is displayed in the field of the Main GUI: “Pre-edge Poly. func. ord.:”, when “Polynomial” function is selected from the menu. It should be increased until a good fit of the pre-edge part is achieved. Additionally, the post edge part must be relatively flat and not too curvy, as shown in Figure 13A. With a pre-edge polynomial function, the fit of the mu pre-edge part is done not only between  $E_{\min}$  and  $E_{\max}$  in the pre-edge region of the mu spectrum, but also on the part of the mu spectrum above  $E_0$ , after subtracting from this part the edge jump. This helps keep the pre-edge polynomial function relatively flat towards the end of the mu spectrum.

#### 2.4.1.3.5 Post-edge function

A polynomial function is used to fit the post-edge region of the mux spectrum. The min and max values of the fitted part of the post-edge region are defined in the fields “Post-edge function from:” and “to:” in the Main GUI. These are not defined as absolute energy values but relative to the energy position of the edge jump, which is represented in the program as the  $E_0$  value.

By default, the degree of the polynomial is equal to 2 (quadratic function), as shown in the field of the Main GUI: “Post-edge Poly. func. deg.:”. In most cases, this should be adequate to fit the post-edge region of an XAFS scan. It can be sometimes necessary to decrease the degree of the polynomial to 1 when for instance the EXAFS part is too short and a polynomial function with a degree of 2 fails from normalizing the data.

Like the fitting of the pre-edge region, it is common that the post-edge function does not fit well the data because the min and/or max values of the fitted post-edge region are not optimized. The best approach is to empirically find in the post-edge region the min and max values for which the function fits well the data. A good fit is obtained when the oscillating and flat parts of the post-edge region featured in the resulting normalized spectrum evenly oscillate around 1 and equal to 1 on the Y axis, respectively.

For example, the default min and max of the post-edge function used to obtain the unacceptable normalized spectrum shown above in Figure 11B were 80 and 400 eV, respectively. After a few trial and errors, it was empirically found that slightly modifying the min & max to 80 & 782 eV, respectively, enabled to get a better fit of the post-edge (Figure 14A & Figure 14B).

To modify the min or max value of the post-edge in the fields “Post-edge function from:” and “to:”, respectively, either type a new value directly in the Main GUI or interactively choose it from the plot. This can be done by pressing the cross button next to the value in the Main GUI and clicking on the desired energy position in the plot. There is a ten second-countdown to do this operation.

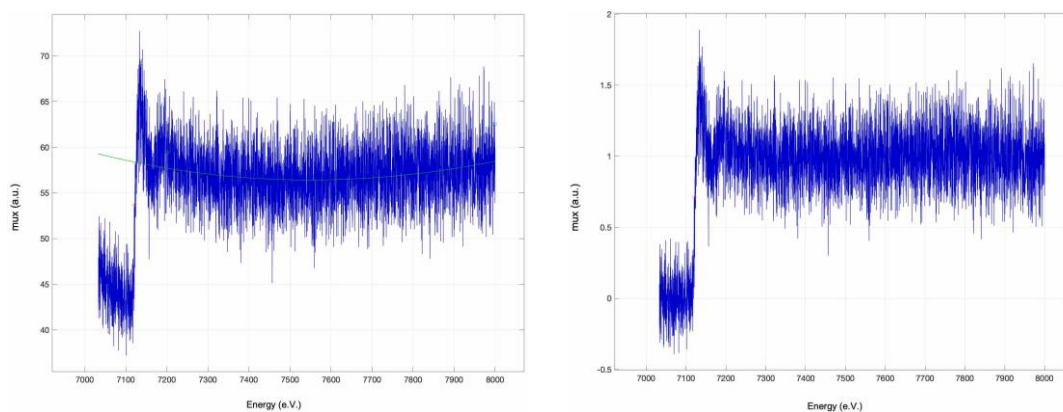


Figure 14 - Example of a quadratic function that fits well the post-edge region & B) New normalized spectrum obtained using the optimized pre-edge and post-edge functions shown in Figure 13 and 14A, respectively

#### 2.4.1.4 Alternative normalization methods applied to multiple spectra

Since Fastosh v1.0.6, the program proposes five simple normalization methods, as an alternative to Larch's normalization method to normalize raw mu spectra imported to the program. The first one is unique to Fastosh, the others are from the program PyMCA [8]. All these alternative normalization methods must be applied to multiple spectra, not one at a time: to use any of them, it is compulsory to have at least two scans selected in the sample list of the Main GUI. Then, select the desired method in:

Main GUI Menu > Options > “Normalization of all checked spectra based on...”

When any of these methods is employed, the raw mu data is interpolated based on the energy grid of the first selected raw mu spectrum.

##### 2.4.1.4.1 Fastosh simple method: subtract offset and locally equalize spectra

With this method, the minimum absorption value is subtracted from each raw mu spectrum, so that its minimum value is equal to zero. Then, each spectrum is divided by its last absorption value. Finally, a user-defined part of the raw mu spectra is vertically equalize via a least square fitting method. By default, the equalized part is equal to the last 5 eV of the spectra.

##### 2.4.1.4.2 PyMCA method 1: “Subtract offset and normalize to maximum”

With this method, the minimum absorption value is subtracted from each raw mu spectrum, so that its minimum value is equal to zero.

Then, each spectrum is divided by its maximum absorption value, so that its maximum value is equal to 1.

##### 2.4.1.4.3 PyMCA method 2: “Subtract offset and normalize to counts”

With this method, the minimum absorption value is subtracted from each raw mu spectrum, so that its minimum value is equal to zero.

Then, each spectrum is divided by the integral of its absorption measured across its entire energy range. The integral is performed via a basic rectangular “mid point” rule.

##### 2.4.1.4.4 PyMCA method 3: “Subtract offset and normalize to integrated area”

With this method, the minimum absorption value is subtracted from each raw mu spectrum, so that its minimum value is equal to zero.

Then, each spectra is divided by the integral of its absorption measured across its entire energy range. The integral is performed via a trapezoidal rule, as opposed to the basic rectangular “mid point” rule.

#### **2.4.1.4.5 PyMCA method 4: “Normalize to maximum”**

With this method, each spectra is simply divided by its maximum absorption value, so that its maximum value is equal to 1.

### **2.4.1.5 Background removal**

The principle of the background subtraction procedure to extract, from the raw data, the chi spectrum featuring the EXAFS is detailed below in the first part of this section. This procedure is based on the autobk algorithm included in Larch. The Matlab code employed in Fastosh has been slightly modified from the original Python code of Larch. The difference between the two versions is explained in this section. The principle of the background removal procedure is described below so that the meaning of all user-defined background subtraction parameters displayed in the Main GUI can be understood. These parameters are individually detailed at the end of this section.

#### **2.4.1.5.1 Principle of background subtraction in Larch (Python & Matlab versions)**

The post-edge region of the unnormalized mu spectrum possesses multiple oscillations mixed together, which can be classified into two categories: the background oscillations and those belonging to the chi featuring the EXAFS. Among all oscillations mixed in the post-edge region, the ones having the lowest frequencies correspond to the absorption background. This variation in absorption is not dependent on the atomic environment around the absorber. It would then still occur in the post-edge region of a mu spectrum corresponding to an isolated, single atom ideally present in a vacuum with no neighboring atoms around it. In contrast, all other oscillations mixed with the background oscillation, a few wavenumbers above the E<sub>0</sub> energy in the post-edge region of mu spectrum, are related to the photoelectron travelling back and forth from the absorber to the neighboring atoms. These oscillations, which inform on the chemical structure around the absorber, constitute the EXAFS and are featured in the chi spectrum, whose expression is shown in Equation (1):

$$(1) \text{ Chi}_{(\text{from } E_0 \text{ to } E_{\text{max}})} = \frac{\text{raw mux}_{(\text{from } E_0 \text{ to } E_{\text{max}})} - \text{background}_{(\text{from } E_0 \text{ to } E_{\text{max}})}}{\text{edge step}_{(\text{measured at } E_0)}}$$

Accordingly, the contribution of the background must be removed from the post-edge region of the mu spectrum to isolate the EXAFS. This step of the data treatment is called the “background subtraction”. It is performed automatically by Larch functions when importing a raw mu spectrum to Fastosh, just like the normalization.

When importing a mu spectrum, the normalization is firstly done to flatten the post-edge region of the mux spectrum from where the chi is extracted and measure the edge-step at the energy of E<sub>0</sub>. The autobk function then uses this edge-step value and the background (from E<sub>0</sub> to E<sub>max</sub>) to extract the chi spectrum featuring the EXAFS, following the expression shown in Equation (1). This is done at the final stage of the background subtraction procedure, after autobk automatically calculates the background via a fitting procedure described in the next paragraph. This fitting procedure must be understood to grasp the meaning of the background subtraction parameters displayed in the main GUI of Fastosh, i.e. Rbkg, spline kmin, kmax, and clamps.

#### **2.4.1.5.2 Details on how “background” in Expression (1) is obtained by autobk**

In autobk, the background is represented by a spline function. It is essentially a series of 2<sup>nd</sup> degree polynomials that successively join at knots spreading throughout the post-edge region of the mu spectrum. The number of knots of the spline is given by Equation (2):

$$(2) \text{ Total number of knots in spline} = \frac{2 \cdot R_{\text{bkg}} \cdot (\text{spline kmax} - \text{kmin})}{\pi} + 1$$

where spline kmax and kmin are the limits of the spline in k space and user-defined parameters. The spline is a b-spline type; the first and last knots are degenerated and the second and the one before the last one are removed. The meaning of  $R_{\text{bkg}}$  featured in Equation (2) is explained below. Since the total numbers of knots of the spline is known, the horizontal positions of all spline knots are constrained and constant. In contrast, the vertical positions of all knots (called “coefs” in autobk) are floated when a least-square fit is performed to obtain the background spectrum. This is a non-linear fitting procedure whose steps are described in the next paragraph.

For the first iteration of the fitting procedure, an initial guess of the background spectrum is obtained by setting the value of each coef of the spline equal to the value of the unsubtracted chi at each knot position, i.e.  $\text{coefs} = \text{chi}(\text{knots})$ . This initial background is subtracted from the post-edge region of the mu spectrum, hence “subtracted chi” is obtained. This chi is then interpolated to a 0.05 /Å grid using a smoothing spline, and eventually Fourier-transformed. This Fourier Transform (FT) is done employing an apodization window (“Hanning”) and a user-defined spline k-weight (set to 2 by default). Then, a portion of the chi FT is used for the minimization procedure, from 0 Å to the user defined  $R_{\text{bkg}}$  value (set to 1 Å by default) as the lower and upper limits of the FT part used in the fit, respectively. Therefore, the fit minimization is done by default using the part from 0 to 1 Å of the FT. The values corresponding to the real and imaginary values at each point of this FT portion are then alternatively put onto a single array, called “out” in autobk. Then, “out” is used to calculate “scale” following the expression shown in Equation (3).

$$(3) \text{ scale} = 1 + 100 \cdot \text{mean}(\text{out}^2)$$

Finally, “out” and “scale” are both used to calculate “residual”, following the expression shown in (4). Residual is essentially a single array of numbers constituted of three sub arrays concatenated one after another: “out”, the low-end part of chi spectrum weighed by the clamps, and the high-end part of chi spectrum weighed by the clamp (4).

$$(4) \text{ Residual} = [\text{out}], [\text{low\_clamp} \cdot \text{scale} \cdot \text{chi}_{(\text{first three data points})}], [\text{hi\_clamp} \cdot \text{scale} \cdot \text{chi}_{(\text{last three data points})}]$$

In (4), low\_clamp and hi\_clamp are user-defined values that are employed to weight the low-end and high-end of the chi spectrum, respectively. **The fit then iteratively modifies the vertical positions of the spline’s knots (coefs) in order to minimize “Residual”.** This means that for each iteration of the fit, the whole procedure described above (i.e. a spline is obtained after modifying coefs, the chi and its FT are computed, and residual is finally calculated) until fit convergence has been reached. When this is the case, the final coefs of the last fit iteration are used to get the final spline, which is eventually employed to extract the chi spectrum following the expression shown in Equation (1). A summary of this entire procedure is shown in Figure 15.

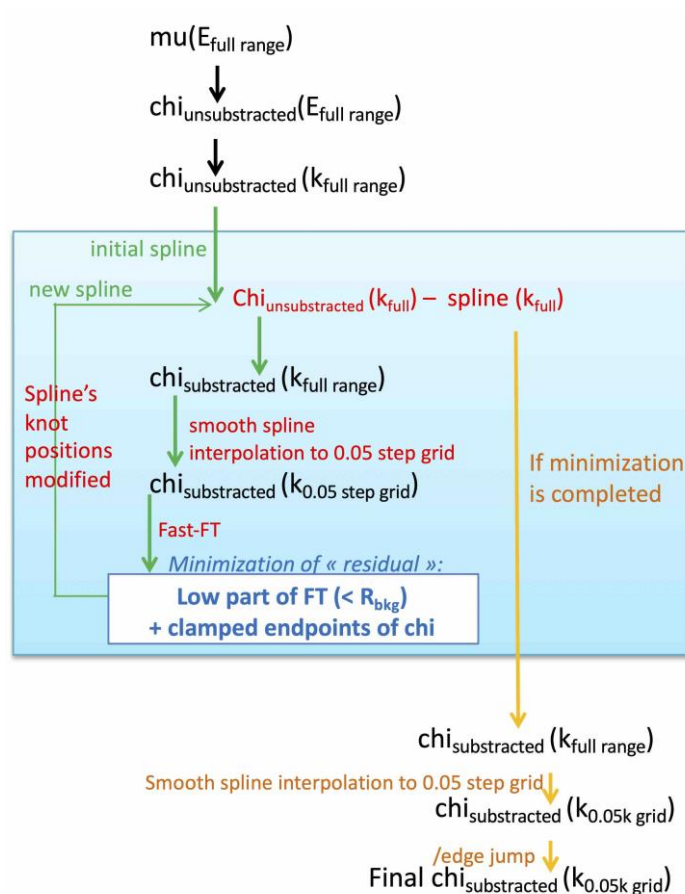


Figure 15 – Background fitting procedure performed by Larch function

After autobk function is done with the background subtraction step, it does one last operation: the chi spectrum of the background obtained from the fit and used to extract the chi spectrum of the sample (Eq. (1)) is converted into a full background mu spectrum. To do that, the background chi spectrum obtained from the autobk fit, which starts from  $E_0$  to  $E_{\text{max}}$  in the full energy array of the sample data, is concatenated (i.e. “stitched”) to the part of the mu spectrum corresponding to the sample (i.e. the raw spectrum imported to Fastosh) from  $E_{\text{min}}$  to  $E_0$  on the full energy array.

After all automatic operations are completed when importing data into the Main GUI, the mu spectrum corresponding to the sample and its associated background mu spectrum are plotted in blue and red colors, respectively (Figure 16).

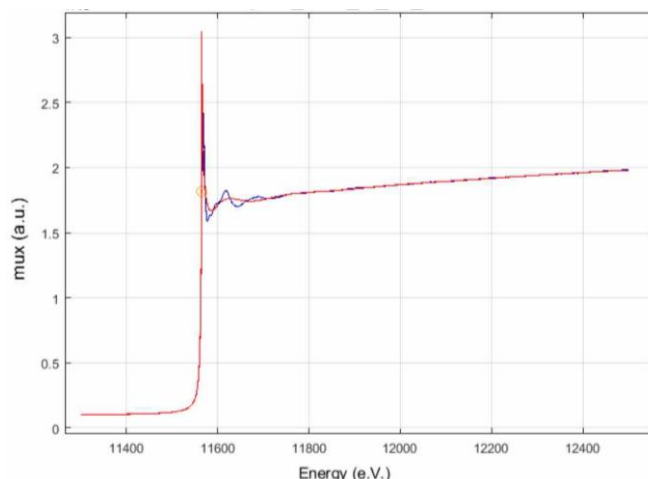


Figure 16 - mu spectra corresponding to the sample (spectrum in blue color) and its associated background (spectrum in red color). The yellow dot indicates the position of  $E_0$ .

#### 2.4.1.5.3 Specificity of Larch autobk adapted to Fastosh

Since Fastosh v1.0.1, the nature of the mathematical function used to represent the background is exactly the same between the original Larch version written in Python and the one adapted to Matlab featured in Fastosh. It is a b-spline in both versions. Therefore, the outcome of the automatic chi extraction should be the same between the two versions.

#### 2.4.1.5.4 $R_{bkg}$

This parameter defines the upper limit of the Fourier-transformed chi portion that is used to calculate “residual” during the fit (Expression (4)). A general rule of thumb is that the  $R_{bkg}$  value should be roughly half the distance of the first shell around the absorber. If  $R_{bkg}$  value is too low, the oscillations having the lowest frequencies featured in the post-edge region of the mu spectrum, hence the oscillations of the background, won't be filtered enough and will “leak” into the Fourier Transform of the chi. In that case, the Fourier Transform may feature peaks below the one corresponding to the first shell, including at 1 Å or below (such small interatomic distances are physically impossible). In contrast, if  $R_{bkg}$  value is too high, some of the EXAFS signal may be erroneously considered as background signal. As a result, the signal in the FT related to the atomic shells around the absorbed may be cut.

The default value of  $R_{bkg}$  (1 Å) is adequate in many instances. Also, there might be more than one correct value to choose from. For example, if 1 Å is correct, 0.9 or 1.1 Å may give the same results and thus could be also adequate values to employ. To check the effect of  $R_{bkg}$  on the FT, one can copy the current spectrum (right click on its name to make a copy), apply a new  $R_{bkg}$  value to the copied spectrum, and plot together the FT of the duplicate samples to compare them. To summarize, a good  $R_{bkg}$  value in the FT should not leak some too-low frequencies ( $R_{bkg}$  value too small) and should not cut the signal you care about ( $R_{bkg}$  value too high).

#### 2.4.1.5.5 Spline k-weight

This parameter represents the k-weight of the Fourier-transformed chi spectrum used during the Autobk optimization of the Spline. It allows to optimize the background subtraction at a specific part of the chi function. A low value and high value, i.e. 1 and 3, will emphasize the lower and higher of the data in the chi spectrum, respectively.

A method to set this parameter is discussed in Kelly *et al.* 2008 [9]. Basically, a suitable spline k-weight should result in a chi spectrum that is equally balanced below and above the zero line of the EXAFS oscillations when a plotting k-weight of 1, 2, or 3 is employed. A good Spline k-weight should then produce a chi spectrum that is independent of plot k-weight, or at least the part of chi spectrum that you are interested in. This implies that the lower part of the chi spectrum  $\sim <3$  Å may be not considered when assessing the suitability of a given spline k weight value.

#### 2.4.1.5.6 Spline $k_{min}$ & $k_{max}$

Spline  $k_{min}$  and  $k_{max}$  are the lowest and highest values of the spline, respectively. Keeping  $k_{min}$  equal to zero should be suitable in most cases, if not all cases. Regarding  $k_{max}$ , one should keep the value of this parameter as high as possible, even if the chi spectrum is mostly noise at high k values, as long as the spectrum does not feature distortions such as monochromator glitches that cannot be suppressed using deglitching tools. If the spectrum features large distortions that cannot be removed, the signal may be cut by either truncating it or lowering Spline  $k_{max}$ . Also, slightly lowering Spline  $k_{max}$  may be helpful in achieving a good background subtraction when the spline is not fitting the data at high k values and modifying the high clamp value has no effect (see next section).

#### 2.4.1.5.7 Clamps

The low and high clamps are essentially integer values that are used to weigh the low- and high-end parts of the chi spectrum in the calculation of Residual, respectively (Expression (4)). Residual is the quantity minimized during the least-square fit performed by autobk to extract the background (Section 2.4.1.5.2). The higher the clamp value, the more the background spline will

be forced to fit the experimental chi spectrum in the region where the clamp is applied. From a practical standpoint, a misfit between the experimental spectrum and spline is likely to occur mostly towards the end of the mu spectrum. The high clamp is then often more relevant than the low clamp, which is rarely used. A misfit typically occurs when the end of the experimental mu spectrum is flat while the end of the background mu spectrum appears wavy. As a result, the end of the chi spectrum may dramatically drop or rise instead of reaching a value around zero. Modifying the high clamp current value to a higher one may overcome this problem. If modifying the high clamp value does not help improve the background spline, one could try slightly decreasing the value of Spline  $k_{\max}$ , as discussed in the previous section.

## 2.4.1.6 Fast Fourier Transform

### 2.4.1.6.1 $k$ range: $k_{\min}$ and $k_{\max}$

The endpoints of the chi spectrum part that is Fourier Transformed correspond to  $k_{\min}$  and  $k_{\max}$ . Since the Fourier Transform is performed on the EXAFS featured in the chi spectrum,  $k_{\min}$  shouldn't be too small to exclude the end part of the XANES, which corresponds to the first part of the chi spectrum. The XANES ends in the chi spectrum about 50 eV above the E0 value, which corresponds to  $\sim 3.6$  /Å in  $k$  space. Therefore, a  $k_{\min}$  equal to 3 /Å proposed by the program represents a default value not too conservative to start with. This value may be decreased if the goal is, for instance, to widen the EXAFS range in order to increase the resolution of the Fourier Transform. However, this should be done only if the new  $k_{\min}$  value does not significantly modify the shape of the Fourier Transform.

Regarding  $k_{\max}$ , its value should enable to include as much signal of the chi spectrum as possible in order to maximize the resolution of the Fourier Transform. However, it should be before the end part of the chi spectrum mostly dominated by random noise. This is summarized in Shelly Kelly *et al.* 2008 [9]: *"optimizing the  $k$  range for EXAFS data analysis involves a tradeoff between increasing resolution of neighboring atomic shells and decreasing the contribution of spectral noise to the Fourier transform"*.

### 2.4.1.6.2 Window

This functionality allows to specify the type of apodization window employed to perform the Fourier Transform. It is in fact a "Fast Fourier Transform" as the processed EXAFS signal is finite and bounded by  $k_{\min}$  and  $k_{\max}$  endpoints. Fast Fourier Transforming a finite EXAFS signal featuring non-zero values at  $k_{\min}$  and  $k_{\max}$  endpoints causes ripples in the resulting Fourier Transform spectrum. These ripples appear as small curved peaks at all  $R$  values in the Fourier Transform. They are thus problematic as they could be potentially mistaken for peaks that we care about, i.e. those due to photoelectron single or multiple scatterings. The ripples can be minimized by multiplying an apodization function to the chi spectrum, which brings the signal to zero values at the two endpoints  $k_{\min}$  and  $k_{\max}$ .

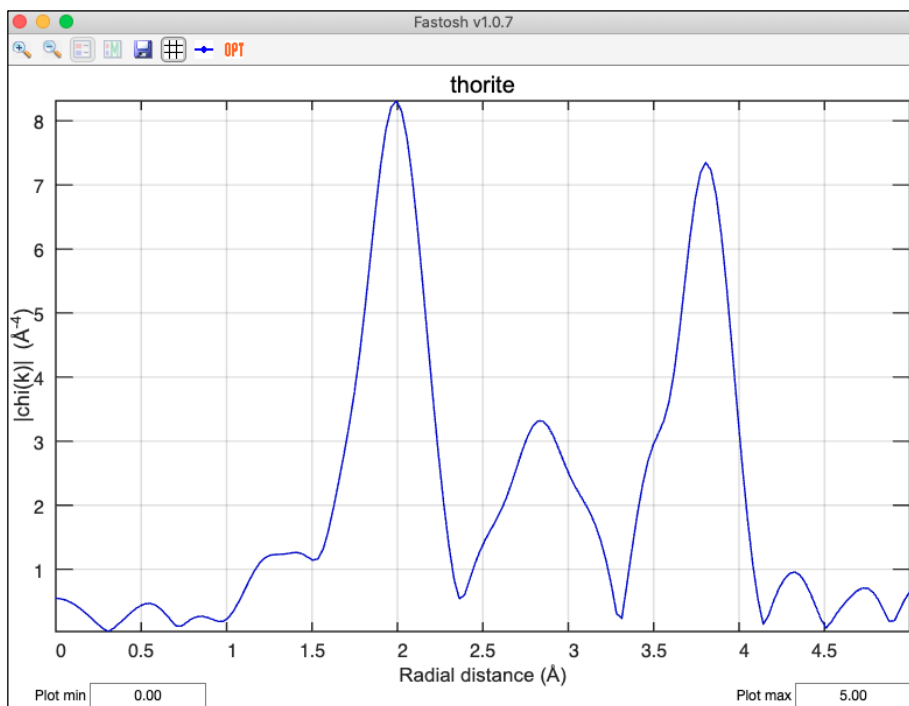
There are six types of apodization windows proposed: Hanning, Parzen, Welch, Gaussian, Sine, & Kaiser. The chosen apodization window can be plotted along with the experimental chi spectrum to observe the window shape. Go to Plot Options at the bottom-left of the Main GUI, select "chi" in the menu tab, check "window", and then replot the chi spectrum by pressing the "chi" plot button. Each type of apodization window has a specific shape and thus effect on ripple minimization in the Fourier Transform. However, the main information that Users get from the Fourier Transform spectrum should not significantly depend on apodization window type. It is generally recommended that a whole data set should be treated using only one type of apodization window, the choice in window type being left to the User.

### 2.4.1.6.3 $dk$

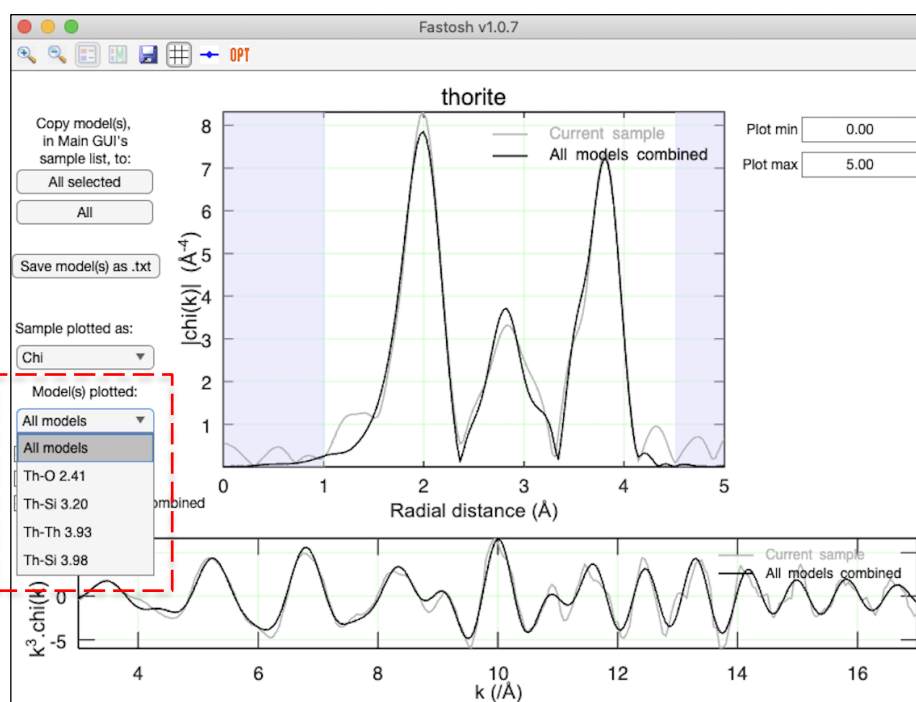
The value of  $dk$  parameter corresponds to the width of the left and right sills of the apodization window. The left and right sills are centered at  $k_{\min}$  and  $k_{\max}$ , respectively. For example, if  $k_{\max}=12$  /Å and  $dk=1$ , the right sill of the apodization window will start to decrease at 11.5 /Å and reach a zero value at 12.5 /Å. Increasing the value of  $dk$  may enhance the minimization of ripples featured in the Fourier Transform spectrum (see previous section). However, the  $dk$  value should not be too high either to minimize the filtering by the window of the structural oscillations featured in the EXAFS signal.

#### 2.4.1.6.4 FT plot window, with or without theoretical chi models created by FEFF

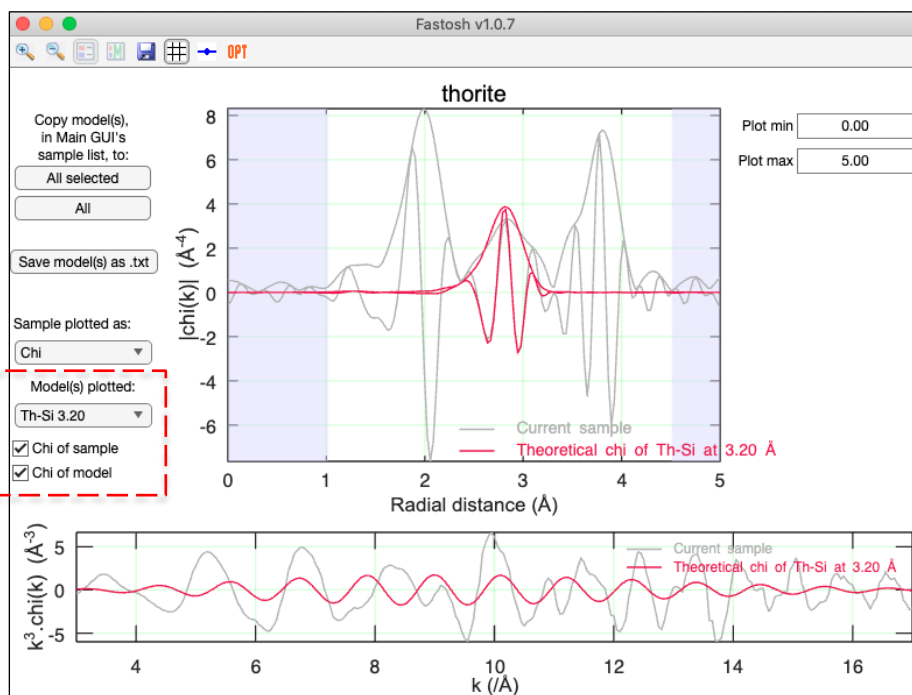
If no theoretical models have been created using the Fastosh module “Quick EXAFS modelling” (see Section 8 of this User Manual), the plot window will display only the FT corresponding to the current sample, when the blue button “Current FT” is pressed in the Main GUI (Figure 17a). If theoretical chi models have been created, the Fourier Transform of the highlighted sample is displayed along with the Fourier Transform of the theoretical data: all models combined (Figure 17b), a specific single scattering path (Figure 17c), or all individual single scattering paths (Figure 17d), using the plot options available on the left side of the plot window (Figure 17).



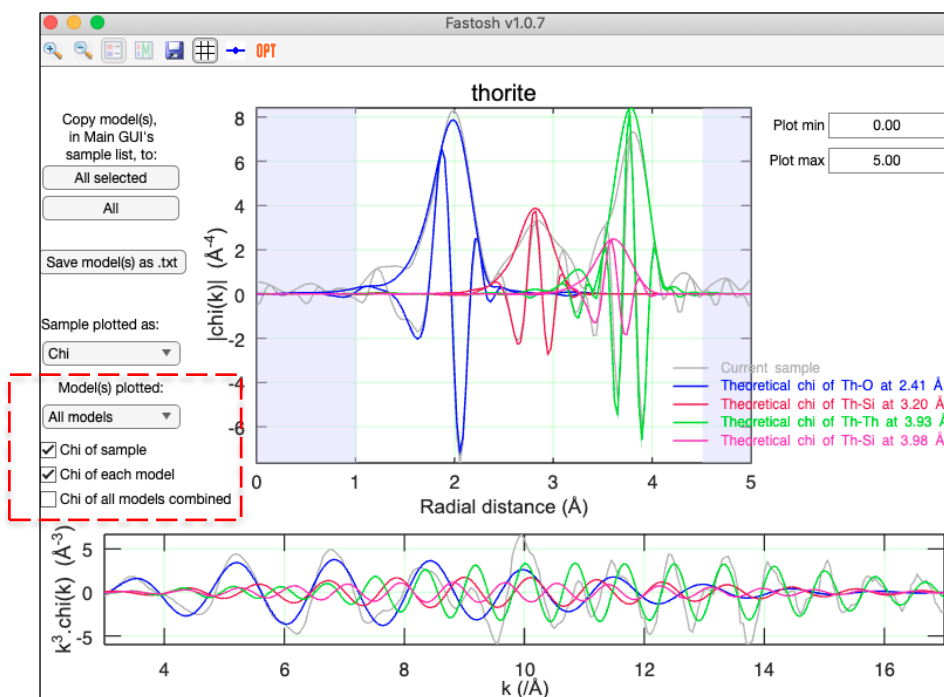
(a)



(b)



(c)



(d)

Figure 17 - Fastosh plot window when the blue button “Current FT” is pressed in the Main GUI to display the FT data relative to the current sample: if no theoretical chi models have been created using the Fastosh Module “Quick EXAFS fitting” (Section 8 of the User Manual), only the Fourier Transform of the highlighted sample is displayed (a). If theoretical chi models have been created using the Fastosh Module “Quick EXAFS fitting”, the Fourier Transform of the highlighted sample is displayed along with the Fourier Transform of the theoretical data: all models combined (b), a specific single scattering path (c), or all individual single scattering paths (d)

#### 2.4.1.7 Reverse Fourier Transform: $R_{\min}$ , $R_{\max}$ , Window, and $dR$

Performing a reverse Fourier Transform can be helpful in isolating a specific oscillatory part of the chi spectrum that contributes to a discrete region of the Fourier Transform whose boundaries are defined by  $R_{\min}$  and  $R_{\max}$  values. As discussed in the previous section, the chi spectrum is multiplied by an apodization window to minimize ripples in the Fourier Transform spectrum. Similarly, the Fourier Transform is multiplied by an apodization window to minimize ripples in the reverse Fourier Transform spectrum.

To do a reverse Fourier Transform, firstly specify the values for  $R_{\min}$  and  $R_{\max}$ , i.e. the endpoints of the specific region of the Fourier Transform to be reverse Fourier Transformed, as well as the type of apodization window and its sill width (dR). Then, click on the “kq” plot button to display the original chi spectrum (in blue color) and the resulting reverse Fourier Transform spectrum (in red color).

#### 2.4.1.8 Plot k weight

The plot k weight value allows to emphasize a specific part of the chi or Fourier Transform spectrum. The k-weighted chi is equal to:

$$k\text{-weighted chi}(i) = \text{chi}(i) \cdot k(i)^{\text{plot k weight}}$$

where “k” is the k array associated to chi array, and “i” is the position of a data point in the arrays. The higher the plot k weight value, the higher the emphasis towards the end of the spectrum (Figure 26). The plot k weight value can be also modified to determine the optimal value of the spline k weight parameter as mentioned in Section 2.4.1.5.5. Additionally, the plot k weight can be used as a mean to identify whether two atomic shells observed at two different R values in the magnitude of the Fourier Transform are similar or significantly different from each other in nature. If an element corresponding to a given shell has an atomic number at least three times higher than the element corresponding to the other shell, then the magnitude ratio between the two shells should increase with increasing plot k weight [9]. In contrast, if the elements corresponding to the two shells have similar atomic numbers (i.e. those within one or two atomic numbers), then modifying the plot k weight should not significantly modify the relative difference in magnitude of the two shells.

A suitable value of k plot weight to use by default should result in a Fourier Transform where all observable atomic shells would provide a relative even contribution to the spectrum. The magnitude of the chi spectrum should have a symmetrical envelop along the k space axis. For example, this corresponds to plot k weight=2 for the data shown in Figure 26.

#### 2.4.1.9 Wavelet Transform

##### 2.4.1.9.1 Presentation

Since Fastosh v1.0.7, the EXAFS data can be processed via a Wavelet Transform approach, as an alternative to the Fourier Transform approach. The type of Wavelet can be either the Cauchy or Morlet Wavelet. For more details about each of them, refer to the original studies that introduced them, i.e. in Muñoz *et al.* [1] and Funke *et al.* [2], respectively. In this User Manual, most of the figures featuring some Wavelet Transform examples were made using the EXAFS spectrum corresponding to the mineral phase thorite. This spectrum was featured in the original paper of Muñoz *et al.* [1], which represents the first study that introduced the use of Wavelet Transform of the EXAFS. The structure of thorite is suitable for showcasing the usefulness of wavelet transform of the EXAFS as it features notably a:

- Th-Si shell at 3.16 Å, with 2 Si atoms
- Th-Si and Th-Th shells both at 3.9 Å, with 4 Si and 4 Th atoms in the shell, respectively
- Th-Th shell at 5.95 Å, with 8 Th atoms

Muñoz *et al.* [1] showed that while only a single peak can be observed at 3.9 Å in the magnitude of the FT spectrum of thorite, the WT map does suggest the presence of Th-Si and Th-Th shells at 3.9 Å (Figure 18). Therefore, compared to FT, WT may allow to better identify the presence of more than one shell existing at a specific distance.

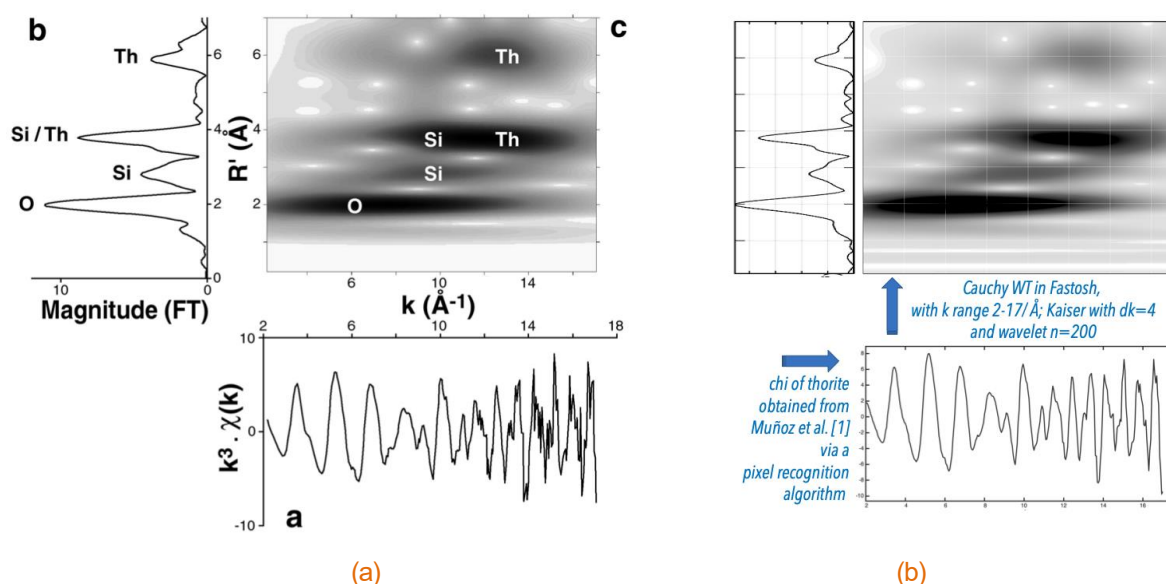


Figure 18 - EXAFS spectrum of thorite, and its corresponding Cauchy Wavelet Transform and Fourier Transform featured in Figure 1 of Muñoz et al. [1] (a), and the WT and FT data reproduced in Fastosh (b). This EXAFS spectrum will be employed in other WT examples featured in this User Manual

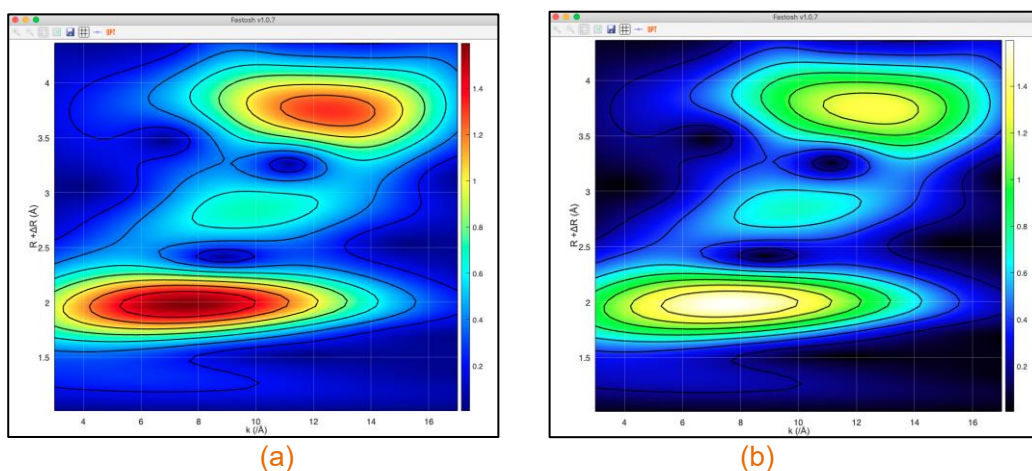
#### 2.4.1.9.2 Wavelet Plot options

Several plot options are available to display the Wavelet transform, via the “Plot type” scrollable menu in the Wavelet Transform tab of Fastosh Main GUI:

- Plot type: “WT of current (+ model, if created)”

When this plot option is selected, the wavelet transform of the EXAFS, corresponding to the sample currently highlighted in the program Main GUI can be displayed, with or without theoretical models created with FEFF (Figure 19). Additional plot options are available from the Wavelet Transform tab:

- “Colors”: Several color options are available to display in color the WT map, including a colormap called “T and K 2009”, as a tribute to the 2009 manuscript by Timoshenko and Kuzmin [10] (Figure 19b), as well as several built-in or modified Matlab colormaps, including the default color choice “Jet” (Figure 19a).
- “WT map style”: The current WT map can be displayed either as surface and contours (Figure 19 a & b), surface only (Figure 19c), or contours only (Figure 19d).



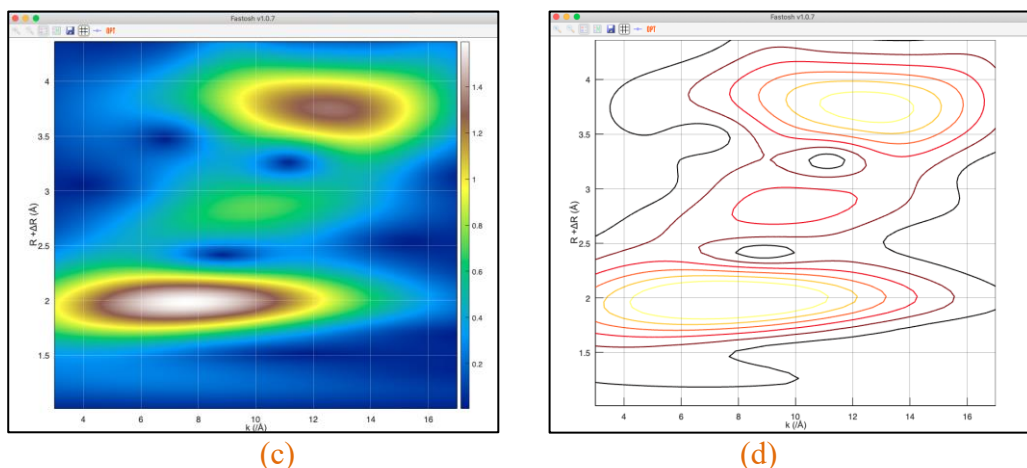
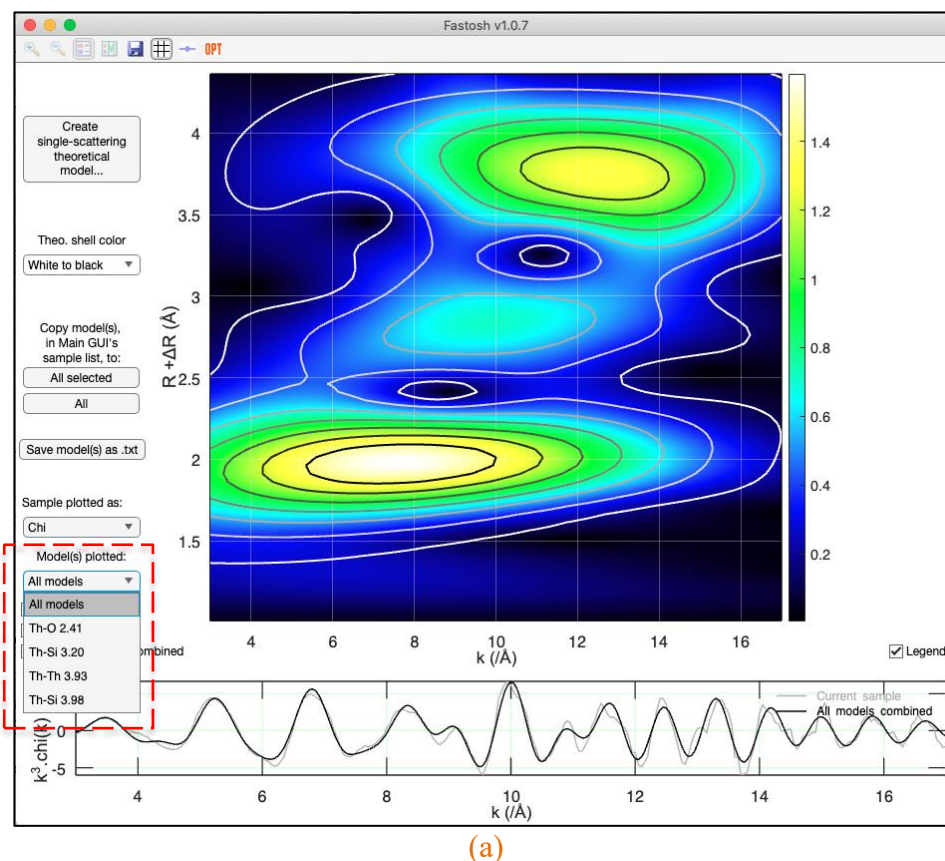
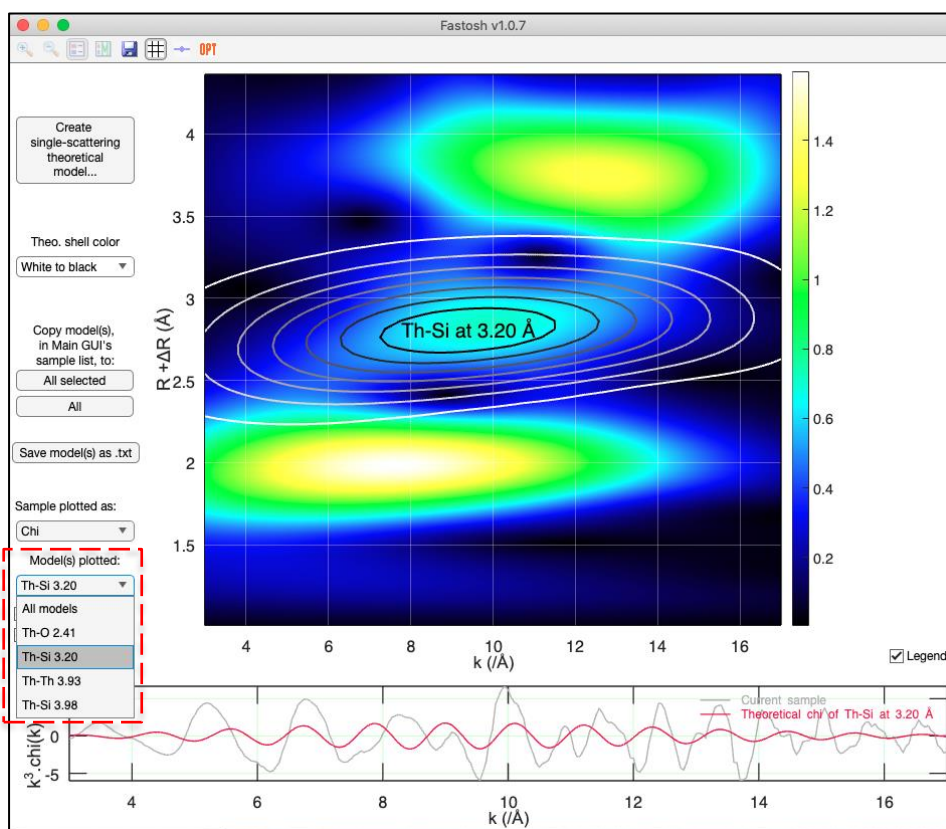


Figure 19 - A wavelet transform map displayed using different plot options. No theoretical FEFF model has been created.

If theoretical models have been created using the Fastosh module “Quick EXAFS modelling” (see Section 8 of this User Manual), the sample map will be automatically displayed as “surface only”. Then, the contours, stacked on top of the sample WT map shown as “surface only”, correspond to the sum of all theoretical single scattering paths combined (Figure 20a), or a specific theoretical single scattering path (Figure 20b).

Additionally, if theoretical models have been created using the Fastosh module “Quick EXAFS modelling” (see Section 8 of this User Manual), the plot window will display, below the WT map, the chi spectra of the sample and the theoretical model (Figure 20a&b). Alternatively, the reverse fourier transform corresponding to the sample/theoretical model spectra can be plotted, by choosing “RFT” instead of “chi” in the scrollable menu called “Sample plotted as” located on the left side of the plot window.





(b)

Figure 20 -WT map, displayed as surface, corresponding to the sample (thorite), along with WT map, displayed as contour, corresponding to the sum of all four single scattering paths combined together (a) or a specific single scattering path, i.e. Th-Si at 3.2 Å (b)

- Plot type: “WT of current + chi +  $\sum_i WT(k_i)$ ”

When this plot option is selected, the chi of the highlighted sample is shown (horizontal plot at the bottom of the plot window), as well as the wavelet transform corresponding to the chi (2D map in the middle of the plot window) and the sum of the R arrays found at all k values of the WT map, referred as “ $\sum_i WT(k_i)$ ”, i.e. essentially all the pixel columns of the WT map summed together (vertical plot on the left of the figure). Optionnally, on top of the latter plot, the magnitude of the FT can be added via a checkbox available at the bottom left of the plot window (Figure 21). This functionality is useful in comparing the resolution in R between the WT and FT. For example, in the example shown in Figure 21, the Cauchy Wavelet Transform was done using the n parameter equal to 200 (default n value). Based on the obvious difference between the  $\sum_i WT(k_i)$  and FT spectra, the resolution in R of the WT seems lower than the R resolution of the FT.

Two additional information on the  $\sum_i WT(k_i)$  / FT vertical plot are given below:

- To conveniently display both the  $\sum_i WT(k_i)$  and magnitude of FT spectra in the same plot as shown in Figure 21, the amplitude of the  $\sum_i WT(k_i)$  spectrum is, in fact, modified using a fitting function where an amplitude correction factor is floated to minimize the difference between the magnitude of FT spectrum and the  $\sum_i WT(k_i)$ .
- To make the  $\sum_i WT(k_i)$  and FT trully comparable, the magnitude of FT spectrum displayed is obtained using the current chi range min & max values, apodization window type, and dk, *specific to the current Wavelet Transform* (values specified in the Wavelet Transform tab of the Main GUI), not the current chi range min & max values, apodization window type, and dk, *of the current*

## Fourier Transform (values specified in the Fourier Transform tab of the Main GUI)

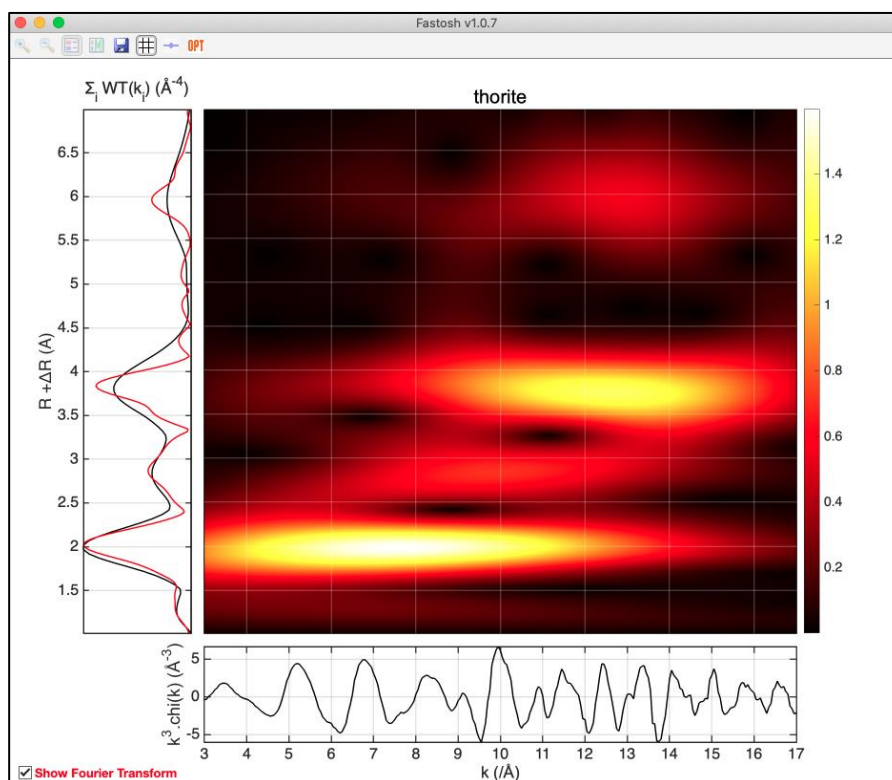


Figure 21 - Plot window when the “WT of current + chi +  $\sum_i WT(k_i)$ ” plot option is selected. In this example, a Cauchy WT was employed with the  $n$  parameter equal to 200 (default  $n$  value). Based on the obvious difference between the  $\sum_i WT(k_i)$  and magnitude of FT spectra, the resolution in  $R$  of this WT seems lower than the FT resolution.

### - Plot type: WT of current + chi + wavelet

When this plot option is selected, the chi of the highlighted sample (in black color) as well as the specific wavelet that is employed to obtain each pixel of the WT map (in red color) are displayed together in  $k$  space in a horizontal plot located at the bottom of the plot window (Figure 22). Optionally, the apodization window applied to the chi, whose type and  $dk$  value are both specified in the Wavelet tab of the Main GUI, can be displayed (in blue color). Uncheck “Chi window”, located in the Wavelet tab of the Main GUI, to mask the apodization window in this plot.

Above this plot, the WT map is displayed. A red cross inside the WT map indicates the pixel location corresponding to the current wavelet shown in  $k$  space in the bottom plot (Figure 22). If a new cross position in the WT map is interactively chosen, by clicking anywhere in the WT map, or if any wavelet parameter (Cauchy's  $n$ , or Morlet's  $\eta$  and  $\sigma$ ) is modified from the Main GUI, the current wavelet will be modified accordingly. Similarly, if any EXAFS-related parameter (i.e.  $k_{min}$ ,  $k_{max}$ , apodization window, or  $dk$ ) is modified from the Wavelet Transform tab of the Main GUI, the EXAFS used to perform the WT, and consequently the WT map, will be modified accordingly. To summarize, this plot option can be used as an educational tool to visually understand:

- the relationship between the specific  $k$  and  $R$  values at each pixel of the WT map and the wavelet's frequency and center position in  $k$  space
- the effects of the wavelet parameter (Cauchy's  $n$ , or Morlet's  $\eta/\sigma$ ) on the shape of the wavelet used to process the EXAFS.

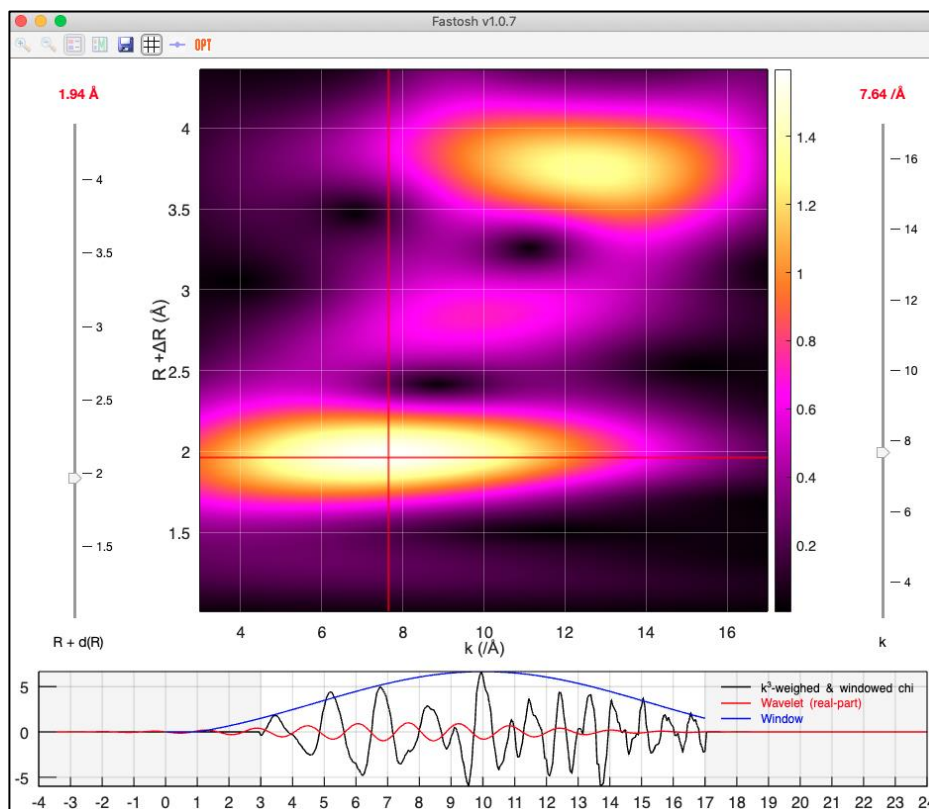


Figure 22 -Plot window when the “WT of current + chi + wavelet” plot option is selected.

#### - Plot type: WT of current + Reverse WT

When this plot option is selected, the WT map is shown, along with another plot at the bottom of the window that features the chi of the highlighted sample (in black color) and the reverse wavelet transform (in red color). The code to calculate the reverse wavelet transform spectrum is a Matlab adaptation of the reverse WT function from the Python code “wtEXAFS” by Zhihang Ye [11].

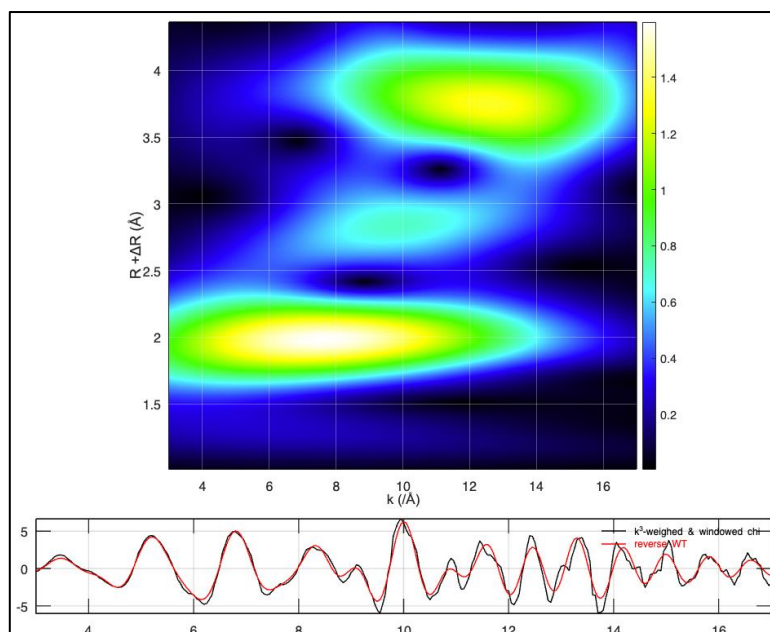


Figure 23- Plot window when the “WT of current + Reverse WT” plot option is selected.

#### 2.4.1.9.3 Cauchy-Morlet WT Comparator

For a given EXAFS spectrum, the Cauchy Wavelet Transform looks very similar to the Morlet Wavelet Transform, but does not look like exactly the same as the Morlet Wavelet Transform, when the default wavelet parameters are employed for each wavelet type (i.e. Cauchy's default n value= 200, and Morlet's default eta and sigma values = 15 & 1, respectively) (Figure 24 a & b). The mathematical expression of the Cauchy and Morlet wavelet transforms, which were described in Muñoz *et al.* [1] and Funke *et al.* [2] respectively, are not the same:

$$\text{Cauchy } \Psi(k') = \left(\frac{i}{k' + i}\right)^{n+1} \quad \text{Morlet } \Psi(k') = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{2i\eta k'} \cdot e^{\left(\frac{k'^2}{2\sigma^2}\right)}$$

Therefore, is the small difference observed in Wavelet Transform results, shown as an example in Figure 24 a&b, mainly due to the difference in nature of these two Wavelet types, or because the default values of the wavelet parameters, colored in red in the two expressions above, are not exactly equivalent? To address this issue, a Fastosh functionality enables to minimize the difference between the Cauchy and Morlet WT maps, by floating the n parameter value of the Cauchy, while the Morlet WT parameters (eta and sigma) are fixed during the fit to their values specified in the Main GUI. To perform such operations, go to:

Main GUI Menu > Extra > “Float “n” of Cauchy wavelet to minimize the difference between...>Cauchy and Morlet wavelet transforms of current sample”

This fitting procedure does lower even more the difference between the Cauchy and Morlet wavelet transforms (Figure 24 c). Therefore, since the two types of wavelet transforms provide very similar results when applying equivalent wavelet parameters, Cauchy is the default choice of wavelet type in Fastosh as the users only has to handle one wavelet parameter instead of two.

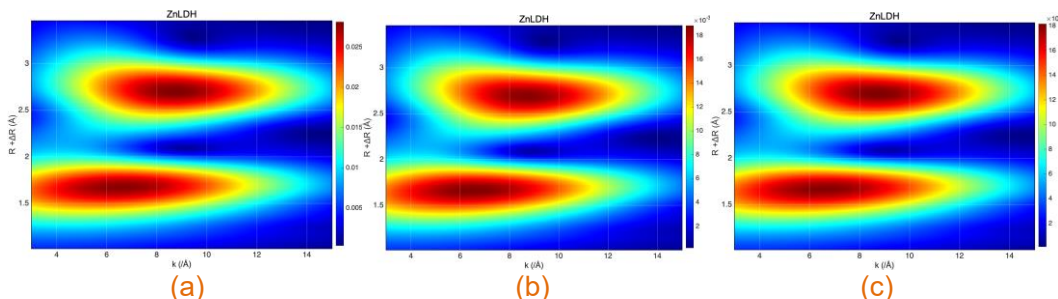


Figure 24 -Morlet Wavelet Transform of an EXAFS spectrum with eta=15 and sigma=1 (default values) (a), Cauchy Wavelet transform of the same EXAFS with n=200 (default value) (b), and Cauchy Wavelet transform of the same EXAFS with n=223, after fitting this parameter to minimize the difference between Cauchy and Morlet Transforms

#### 2.4.1.9.4 Cauchy Wavelet Transform- Fourier Transform Comparator

As demonstrated previously (Figure 21), the Cauchy Wavelet Transform resolution in R can be lower than the Fourier Transform resolution in R when the Cauchy's n parameter is set to its default value (n=200). A Fastosh functionality then enables to automatically increase the Cauchy Wavelet Transform resolution in R and equalize it to the Fourier Transform resolution in R. This is achieved by minimizing the difference between the magnitude of the Fourier Transform spectrum and the sum of the R arrays found at each k value of the Cauchy WT map ( $\sum_i WT(k_i)$ ) while floating the value of the Cauchy n parameter. To perform such operations, go to:

Main GUI Menu > Extra > “Float “n” of Cauchy wavelet to minimize the difference between...>Cauchy Wavelet and Fourier transforms of current sample”

Although this function improves the resolution in R of the WT map and equalize it to the Fourier Transform resolution, as demonstrated in the example shown in Figure 25, it also inevitably decreases the resolution in k of the WT map, since R and k resolutions of the WT map are inversely proportional. Nevertheless, even with the decreased resolution in k space, one can still spatially distinguish in the WT map the Th-Si shell at 3.16 Å, the two Th-Si and Th-Th shells at 3.9 Å, and the Th-Th shell at 5.95 Å (Figure 25).

Two additional information on this fitting procedure are provided below:

- In addition to the Cauchy n value, an amplitude correction factor is floated during the fit to minimize the difference between the FT spectrum and the  $\sum_i WT(k_i)$ .
- To make the  $\sum_i WT(k_i)$  and FT spectra trully comparable, the FT spectrum used in the fitting procedure is obtained using the current chi range min & max values, apodization window type, and dk, *specific to the current Wavelet Transform* (values specified in the Wavelet Transform tab of the Main GUI), not the current chi range min & max values, apodization window type, and dk, *specific to the current Fourier Transform* (values specified in the Fourier Transform tab of the Main GUI)

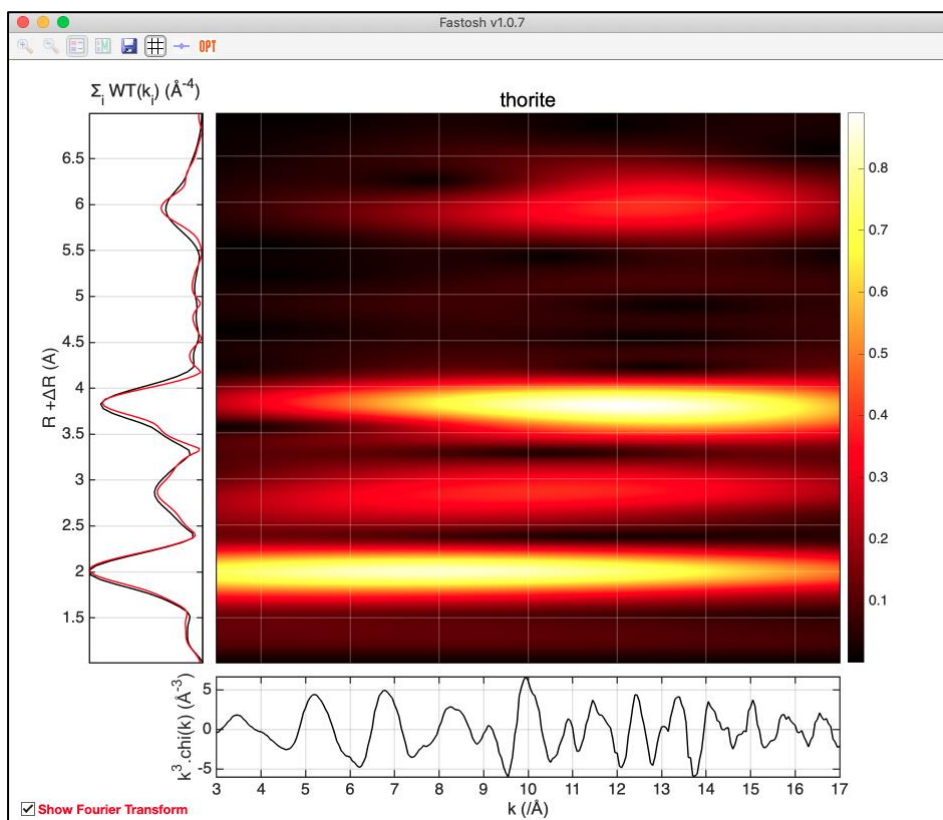


Figure 25 - Plot window when the “WT of current + chi +  $\sum_i WT(k_i)$ ” plot option is selected, with the Cauchy n parameter equal to 1046, after floating this parameter to equalize the difference between the Cauchy WT and magnitude of the FT

## 2.4.2 Viewing the spectrum of a sample or associated reference

### 2.4.2.1 Viewing the reference data

To view the data corresponding to the reference (e.g. a metal foil) associated to a sample, firstly click on the sample name in the sample list featured in the Main GUI. Make sure that the button “Reference” is activated in the field “Data Type” of the Main GUI. If it is not activated as shown in Figure 27A, the reference data was not imported to the Main GUI along with the sample data. To import both data, please refer to Section 2.3. If the Reference button is activated as shown in Figure 27B, the reference data was imported along with the sample data. Click on “Reference” button, and click on any black plot button (e.g. mux, Chi, FT, or kq) to visualize the reference data.

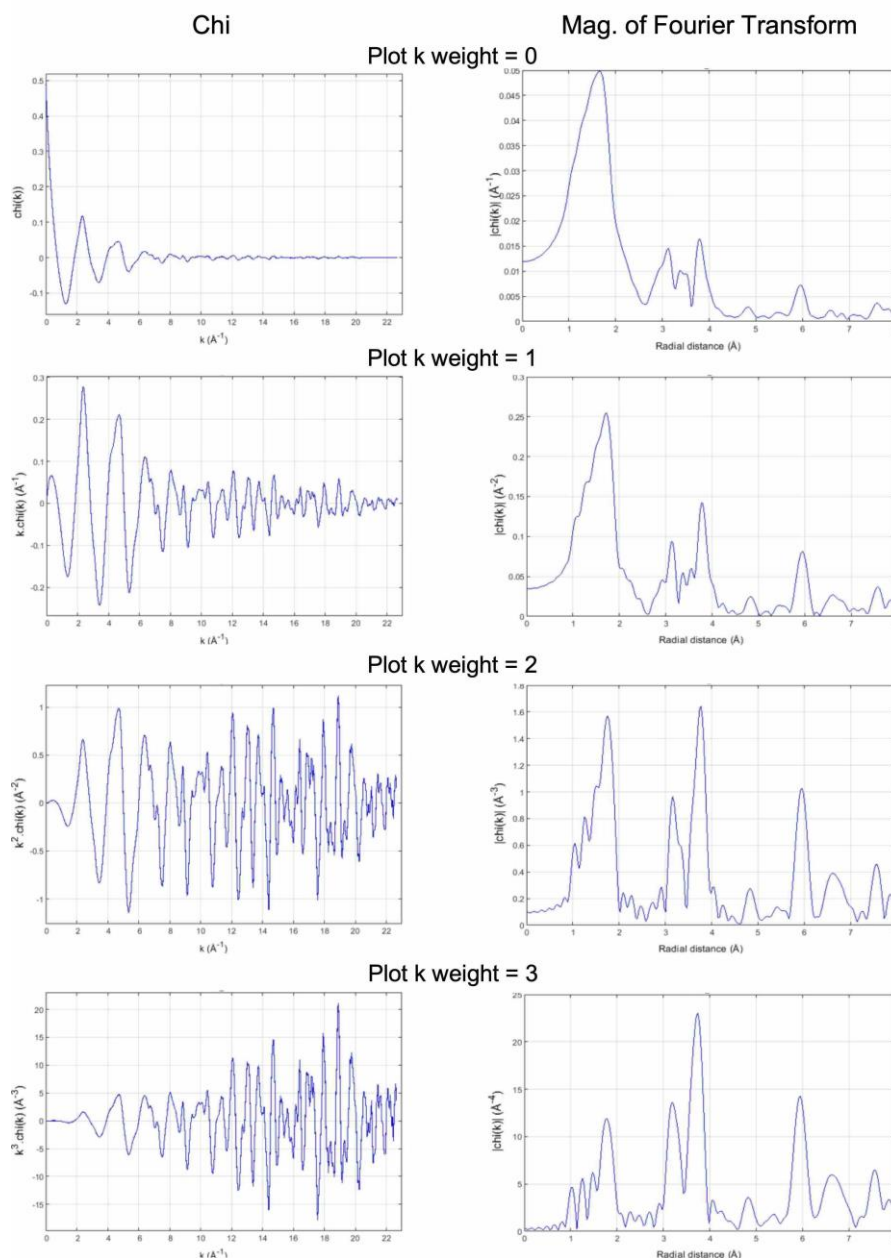


Figure 26 - Effect of plot k weight on the chi or magnitude of Fourier Transform spectrum obtained from a single mu spectrum

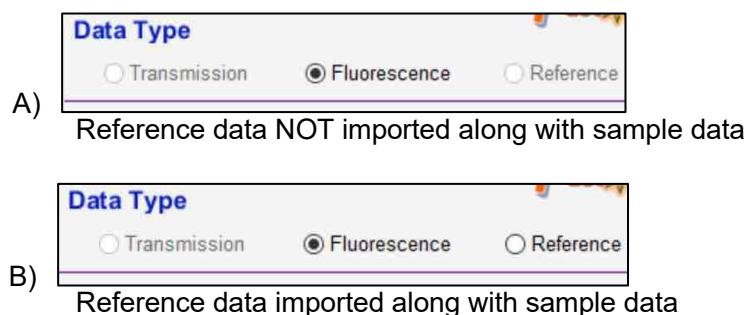


Figure 27 – “Data Type” in Main GUI: A) reference data was not imported along with sample data; and B) reference data was imported along with sample data

#### 2.4.2.2 Viewing the references associated to multiple samples

To view the reference data associated to multiple samples, the button “Reference” (Figure 27B) must be selected for each reference to display. There are two approaches to select multiple references.

The first approach, which is not convenient when plotting many references, is to manually select each individual reference to display following the method described in Section 2.4.1.2. For example, this method must be repeated 30 times to display 30 references !

A faster approach consists in selecting this option in the Menu bar of the Main GUI:

Main GUI Menu > Plot > “Data type: show Reference for all sets of sample/reference”

As explicitly stated, this will select all references imported to the Main GUI.

Once all references to display are selected, and their associated sample names are checked in the sample list of the Main GUI, click on any orange plot button (e.g. Mux, Chi, FT) to plot the references.

Once you are done looking at the reference, you can unselect all reference data and select instead their associated sample data by selecting in the Menu:

Main GUI Menu > Plot > “Data type: show Sample for all sets of sample/reference”

#### 2.4.3 Data calibration & alignment

The “Data Calibration & Alignment” module of Fastosh allows to calibrate a set of samples. This is done using the references (e.g. metal foil) associated to all samples to calibrate. Accordingly, the reference data must be imported to the main GUI along with the sample data to perform the calibration of the data set. To import to the main GUI both reference and sample data, please refer to Section 2.3 “Importing Data”. To calibrate a sample data set, go to:

Main GUI Menu > Operations > “Data Calibration & Alignment”

The calibration & alignment window appears (Figure 6). The list shows all reference data imported to the main GUI, not the sample data. The plot shows the reference of the purple-highlighted reference name in the list.

**To calibrate a whole data set, follow these two steps:**

- One reference is firstly calibrated among all uploaded references**

To calibrate this reference, an E shift is calculated based on the energy position of one particular data point in the reference spectrum, which is typically the maximum of the first inflection point of the spectrum corresponding to a metal foil. The E shift is equal to the theoretical value (known) of this data point minus the current value of this data point found in the experimental spectrum. The calibration then consists in shifting the whole energy array of the reference by this E shift value.

- Once a reference is calibrated, the remaining references are aligned to it.**

Each remaining reference is aligned to the calibrated reference by a least-square fitting approach. The fit minimizes the difference between the calibrated reference (A) and the reference to align (ref B) by floating an E shift added to the E array of ref B. For each reference aligned, the E shift value obtained from alignment is applied to both reference and sample data.

#### Calibration:

- Choose a reference to calibrate. The chosen reference is highlighted with a purple color in the reference list of the window. In the example above (Figure 6), the calibration is done on the first reference in the list: 'E2\_MS\_Noeads\_4mois\_0001'. The data was collected at the Cd K edge.
- Choose how the reference is plotted to do the calibration. You can plot either the 1<sup>st</sup> derivative (default choice) or mu spectrum.
- Make sure that the blue cursor in the plot is at the position where you want to do the calibration. In the example above (Figure 28), the calibration should be done at the maximum of the first inflection point in the spectrum. The cursor is indeed at this position (Figure 28). In terms of energy, this cursor position is at 26710.9 eV as shown in the window.
- In theory, the maximum of the first inflection point of a Cd metal foil should be at 26711 eV based on the literature. Since it is found at 26710.9 eV, a shift of 0.1 eV must be applied to the reference to calibrate it. You can then either write 26711 in the field "Modify cursor energy to:", or 0.1 in the field "shift applied:". Either way, press ENTER to validate the new value.
- Press the button "Apply 0.1 eV shift to highlighted" to do the actual calibration of the highlighted reference.

#### Alignment:

- Checked all references that you wish to calibrate in the reference list of the window.

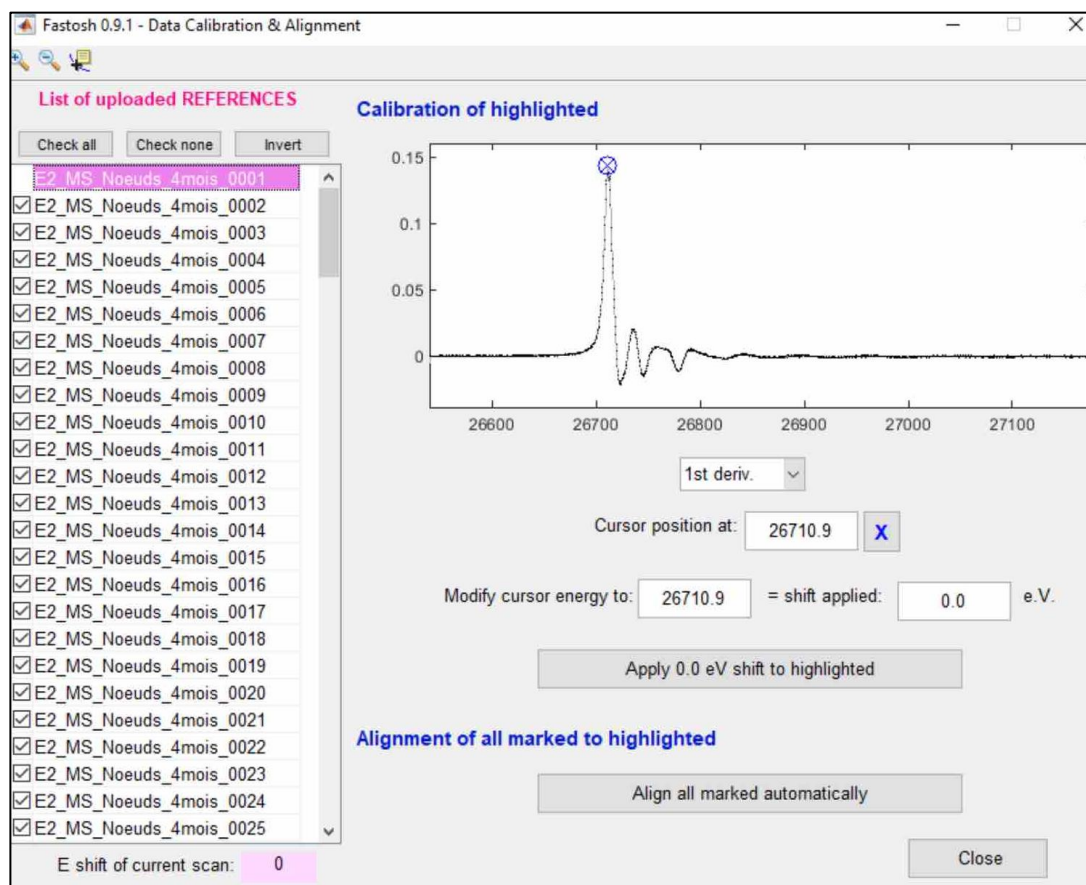


Figure 28 – Calibration & Alignment window

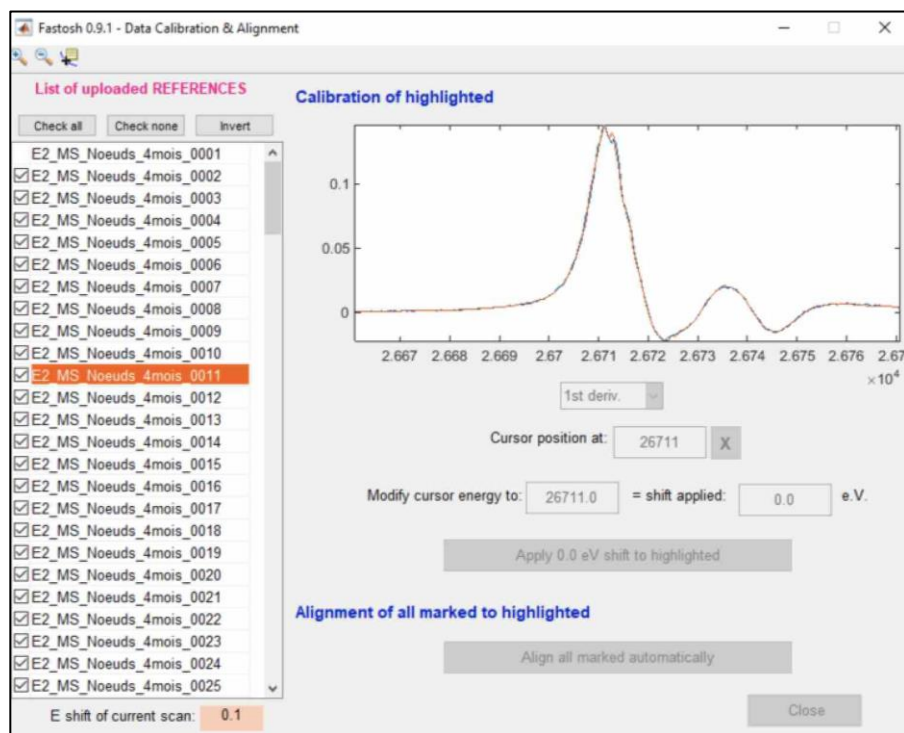


Figure 29 – Calibration & Alignment window. The references to calibrate are spectra corresponding to a Cd metal foil.

- Once all references to calibrate are checked in the list, press on the button “Align all marked automatically”. A fit is successively done on each checked reference. The current reference being fitted is shown with an orange color. The resulting E shift applied to the reference obtained from the fit is shown in the box “Shift applied” at the bottom left of the window. For example, for “E2\_MS\_Noeads\_4mois\_0011”, a 0.1 eV has been applied to the E array of this reference, as shown below (Figure 29).

Once all checked references are aligned, you are done with calibrating your data set. Close the window by pressing the button “Close”. You can notice in the Main GUI that the calibration was done since the E shift for each set of sample/reference is now displayed in the field “E shift” (Figure 30).

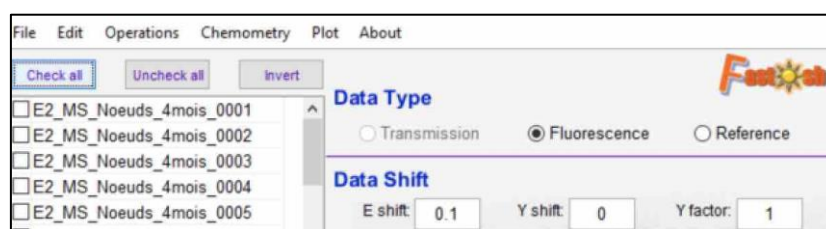


Figure 30 – After calibration, the E shift is displayed in the Main GUI for each set of sample/reference. Here, a 0.1 eV shift was applied to “E2\_MS\_Noeads\_4mois\_0011”

## 2.4.4 Chunk Merging

To open the “chunk Merging” module, go to:

Main GUI Menu > Operations > “Chunk Merging”

This module can be used for two main purposes:

- To reduce the total amount of data to analyze (main purpose)

When a slow chemical reaction is followed at the beamline for a long time period using a rapid acquisition mode, a high number of individual scans may be collected. In that case, it may be unnecessary to treat each scan individually. For example, if an acquisition of a slow chemical reaction followed at the beamline resulted in 200 individual scans, scan #1 may be very similar to scan #5. Also, scan #196 may be very similar to scan #200. One could then create 40 chunks with 5 individual scans per chunk to reduce the total number of scans to handle in the data analysis.

- **To rename a set of individual scans having a specific name and increasing file increment**

For example “ExperimentA\_001, ExperimentA\_002, ..., ExperimentA\_050” can be renamed to: “New\_name\_1, New\_name\_002,..., New\_name\_050”.

The Chunk Merging window features three large vertical lists (Figure 31):

- **“Spectra Selection”**: the first list on the left side of the window corresponds to the sample list, i.e. the names of all scans uploaded to the main GUI. To add more scans to this list, close the Chunk Merging window, import more data to the Main GUI, and reopen the Chunk Merging window.
- **“Chunks to merge”**: the second list, in the middle of the window, groups the names of individual scans that are selected in the first list. Each group is displayed with a specific color. The number of individual scans per group is defined by the parameter “Spectra per chunk”. Its value is equal to 2 by default and can be modified in the Chunk Merging window in the field “Spectra per chunk” (Figure 31). This implies that the color of the scan names displayed in the second list changes by default every 2 scans.  
For example, if 22 scans are selected in the first list, and “Spectra per Chunk” is equal to 5, then 4 groups will be displayed in the second list using the first 20 selected scans. The names of the last two selected scans (i.e. scans 21 & 22) will be shown in a small list entitled “Leftover spectra” at the bottom of the Chunk Merging window. These two scans won’t be used to chunk the data when the button “Merge and Transfer Data” is pressed.
- **“All merges to be created”**: the third list, on the right side of the window, corresponds to the names of the chunk averages that will be created once the button “Merge and Transfer Data” at the bottom right of the window is pressed. The specific color used in the second list to group each individual scan is also shown in the third list next to each chunk average.

**NOTE:** When opening the Chunk merging window, the content of second and third lists may be empty. This is because no scan is selected in the first list on the left side of the window. Individually select scans in the sample list, or use “All” or “Lasso” button for multiple scan selection. This will automatically refresh all related contents of the Chunk Merging window.

#### **To chunk merge a data set:**

- Firstly, check all scans to chunk in the sample list featured on the left side of the window. To check all scans in the list, click on the button “All” on top of the list. The color grouping of individual scans in the second list and names of chunk averages in the third list are automatically redefined.
- Specify the number of scans per chunk in the box “Spectra per chunk”, and click ENTER. The color grouping of individual scans and names of chunk averages are then automatically redefined based on the new “Spectra per chunk” number provided.

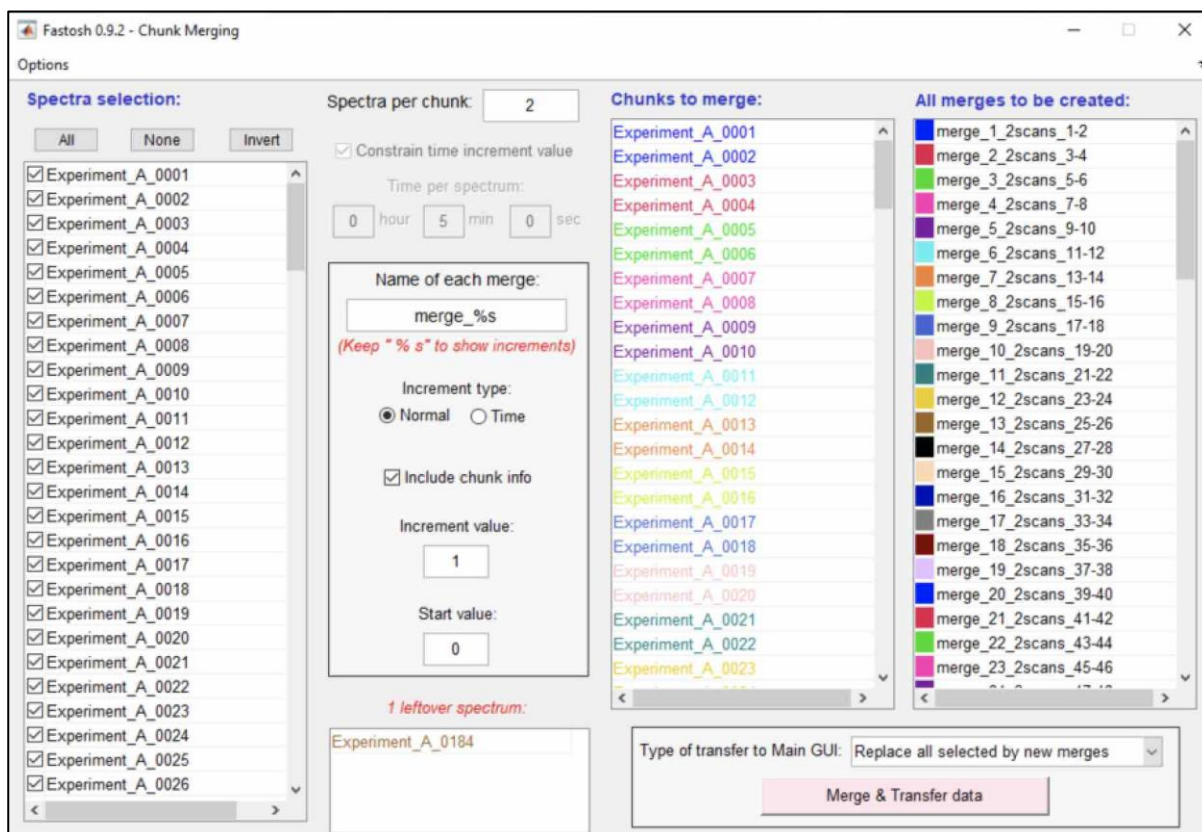


Figure 31 - Chunk merging window

In the example shown in Figure 32, the number per chunk was changed to 5 (it was set to 2 in Figure 31, for the same data set). One can notice that each color group in the second list ("Chunks to merge") is indeed composed of 5 individual scans. Sometimes there's not enough selected scans to create a last group. In that case the names of the scans that could not be used to create a last group are shown in the small box "Leftover spectra". For example, with "Spectra per chunk" set to 2 (Figure 31) and 5 (Figure 32) for the same data set, there was 1 (Figure 31) and 2 (Figure 32) leftover spectra, in other words there was  $(2-1)=1$  and  $(5-2)=3$  scans missing to create a last group, respectively.

- Before chunking the data (i.e. pressing the button "Merge & Transfer data" at the bottom right corner), the names of the chunk averages to create, which are displayed in the third list on the left side of Chunk Merging window, can be customized.

**There are different options to customize the name of each chunk merge:**

- **The increment type** of all chunk merge names can be either normal (default choice) with increasing numbers (e.g. merge 1, merge 2, merge 3, etc...) as in Figure 32, or increasing time periods as shown in Figure 34. To select the increment type to "normal" or "time", use the selector "Increment Type", which is found in the middle of the Chunk merging window (Figure 33A & B).
- **The name of each chunk average**, which is "merge" by default followed by the increment of each chunk average (Figure 32 & Figure 34 for chunk average names displayed with normal & time increment type, respectively). This can be modified in the field "Name of each merge" found in the middle of the Chunk Merging window (Figure 33 A & B).

If the Chunk Merging module is only used as a tool to rename each individual scan and not as an averaging tool, set the "Scans per chunk" parameter to 1 and provide a new name in the field "Name of each merge".

**Note:** to keep the increment at the end of each name, the part “\_%s” must be kept at end of the name specified in the field “Name of each merge” (Figure 33 A & B).

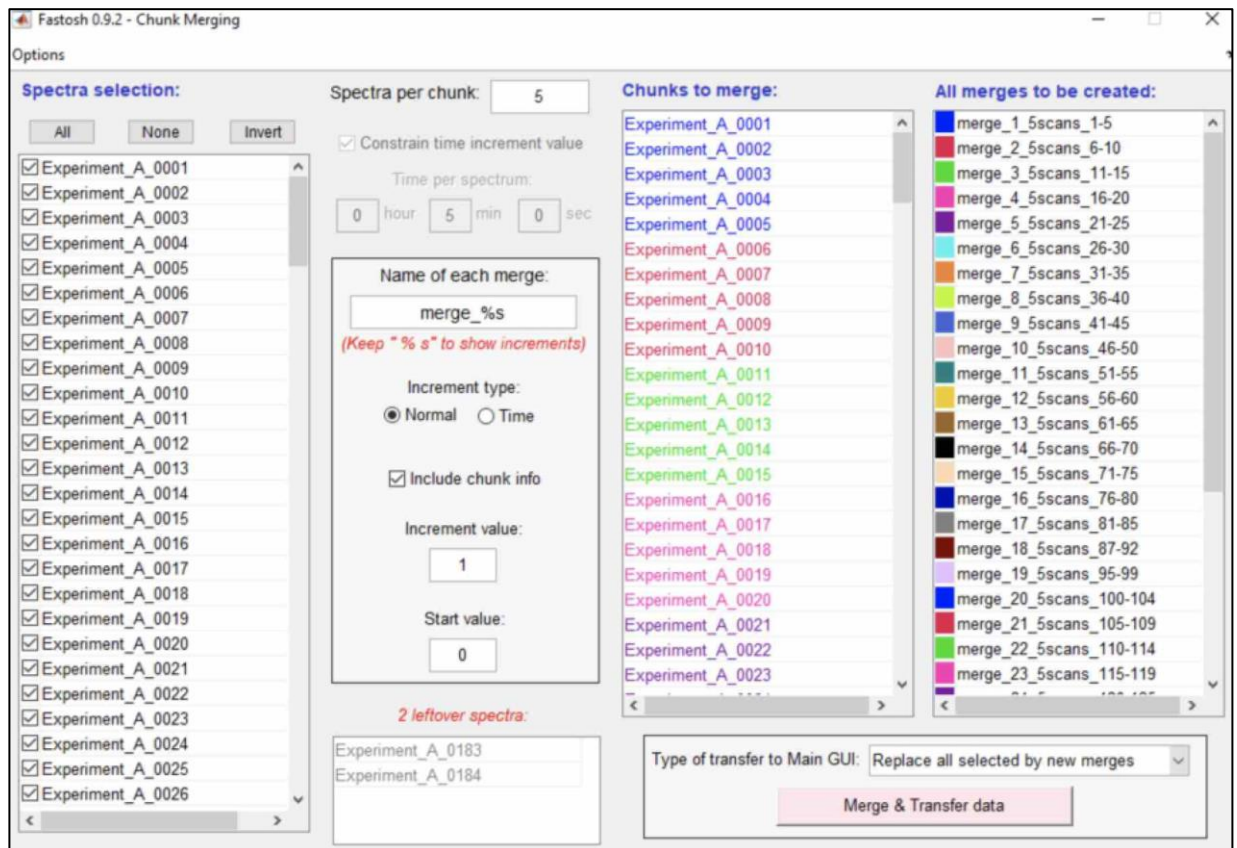


Figure 32 - Chunk merging window with the same sample list as the one shown in the previous Figure but with “Spectra per chunk” set to 5 in its dedicated field at the top of the window

- **The increment starting value** is set to 0 if the increment type is normal (Figure 33A) or time, which is in that case given in seconds (Figure 33B). Modifying this starting value is possible in the field “Start Value” (Figure 33A & B). Additionally, if the increment type is set to normal, the increment value is set to 1 by default. It can be modified in the field “Increment value” (Figure 33A).
- **Chunk info (only if “increment type” is set to “normal”)** The “chunk info” option seen in the middle of Figure 33A will be active by default. This displays at end of each chunk merge name the number of Spectra per chunk and the iteration of all individual scans used to create the chunk:  
merge\_1\_5scans\_1-5  
merge\_2\_5scans\_6-10  
merge\_3\_5scans\_11-15  
etc...

if this option is deactivated, then only the iteration of the merge names will be shown:

merge\_1  
merge\_2  
merge\_3  
etc...

Figure 33 shows two panels, A and B, of the 'Chunk merging' window. Both panels have a checkbox 'Constrain time increment value' which is checked. Panel A has 'Time per spectrum' set to 0 hour, 5 min, 0 sec. Panel B has 'Time per spectrum' set to 0 hour, 2 min, 10 sec. Both panels have a text field 'Name of each merge:' with the value 'merge\_%s' and a red note '(Keep " % s" to show increments)'. Panel A has 'Increment type:' with 'Normal' selected (radio button). Panel B has 'Increment type:' with 'Time' selected (radio button). Both panels have a checkbox 'Include chunk info' which is checked. Panel A has an 'Increment value:' field with the value '1'. Panel B has an 'Increment value:' field with the value 00 hour, 10 min, 50 sec. Both panels have a 'Start value:' field with the value '0'.

Figure 33 - Middle part of the Chunk merging window when “Increment Type” selector is on A) “Normal” or B) “Time”

- **Time increment value (only if “increment type” is set to “time”).** There are two ways to set the time increment value. It can be set based on the time length of a single scan (default option), or one scan group. To use the former and latter approaches, check and unchecked the box “Constrain time increment value”, respectively (Figure 33B).

If the box “Constrain time increment value” is checked, the time length corresponding to one scan can be specified in the field “Time per spectrum” (Figure 33B). For example, if this value is set to 1 minute and 30 seconds, and there are 3 scans per group (i.e. “Scan per chunk” parameter set to 3), then the displayed time increment value of each chunk merge will be 4 minutes and 30 seconds.

If the box “Constrain time increment value” is unchecked, the “Time per spectrum” field will be disabled and the “Increment Value” field will be enabled (Figure 33B). The time length corresponding to one scan group can be specified in the field “Increment Value”. The displayed time increment value of each chunk merge will be then equal to this value.

- **Time format (only if “increment type” is set to “time”).** To display the time increment value of each chunk merge, different type formats are possible:
  - hours/ minutes/ seconds (default format)
  - seconds only
  - minutes only
  - hours only
  - hours/ minutes

Select any of these time format using the menu bar “Time Format” in the middle of the Chunk Merging window (Figure 33B).

- Once all parameters are set, the data can be processed. Before proceeding to the data treatment, a processing option can be selected at the bottom right of the Chunk Merging window (Figure 34):
  - **Replace all selected by new merges:** in the sample list of the Main GUI, all scans corresponding to those selected in the Chuck Merging window will be replaced by the new chunk merges.
  - **Add new merges to list:** the new chunk merges will be added to the sample list already existing in the Main GUI.

- **Delete all, then add new merges:** All content of the sample list in the Main GUI will be deleted. Then, the new chunk merges will be added to it.

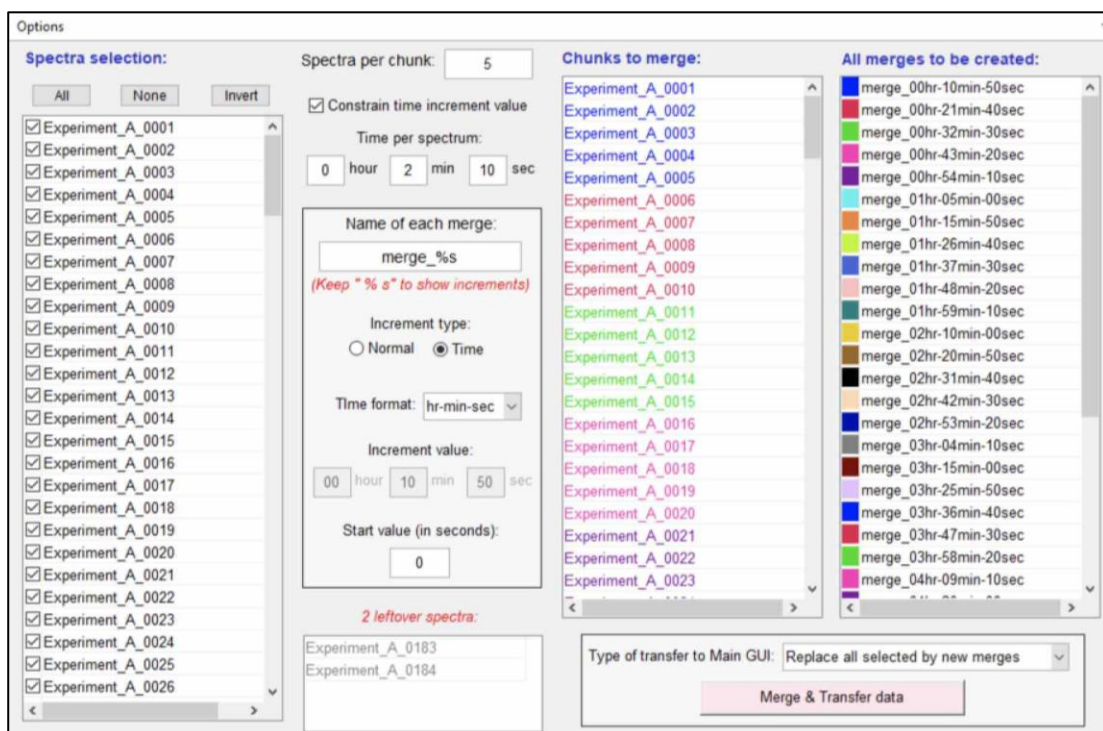


Figure 34 - Chunk merging window with the same sample list as the one shown in the previous Figure but the names of all merges shown with time increments instead of normal increasing increments

- Then, to finally process the data, press the button “Merge & Transfer data” at the bottom right corner of the Chunk Merging window (Figure 34).
- Once all chunk merges have been transferred to the Main GUI, close the Chunk Merging window.

## 2.4.5 Data truncation

This module allows to truncate an unwanted part of the mu or chi spectrum. For example, the end part of the mu spectrum shown below (Figure 35) features a second edge-jump. Either the highlighted sample or all selected samples in the sample list in the Main GUI can be truncated. In any case, the spectrum that is shown in the truncation module corresponds to the highlighted sample in the Main GUI. To open the “Data truncation” module, go to:

Main GUI Menu > Operations > “Data truncation”

The spectrum will be truncated relative to a specific energy value defined by the user, whose position in the plot is shown by a vertical dashed line (Figure 35). The part of the spectrum below or after this energy value will be deleted.

### Procedure to truncate:

- Firstly, specify if the mu or chi spectrum will be truncated, using the “mu” or “chi” selector located on top of the Truncation Window (Figure 35).
- Provide the energy value where the truncation will be performed. Either type a new value directly in the field dedicated to it in the Data Truncation window or interactively choose it from the plot. This can be done by pressing the cross button next to the energy value in the Data Truncation window and clicking on the desired energy position in the plot. There is a ten second-countdown to do this operation.

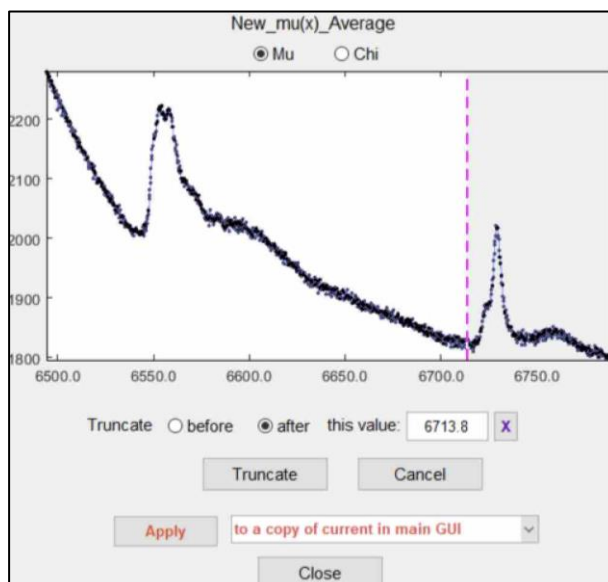


Figure 35 – Data truncation window: example of scan to be truncated

- Choose if the spectrum will be truncated below or above this value, using the “before” or “after” selector located in the middle of the Truncation Window (Figure 35).
- Click on “Truncate” to delete the unwanted part of the spectrum.
- Select one option to transfer the data to the Main GUI from the four-option menu bar located in the middle of the Truncation Window (Figure 35). The names of the four options are self-explanatory:
  - Apply truncation...
    - ...to a copy of current in Main GUI
    - ...to current (no copy) in Main GUI
    - ...to a copy of all checked scans in Main GUI
    - ...to all checked scans (no copy) in Main GUI
- Press the button “Apply” to transfer data to the main GUI. Once all data has been transferred, close the Data Truncation window.

## 2.4.6 Interpolation

Fastosh enables to interpolate data imported to the main GUI, either the highlighted sample or all selected samples in the sample list. In any case, the spectrum that is shown in the interpolation module corresponds to the highlighted sample in the Main GUI. To open the “Interpolation” module, go to:

Main GUI Menu > Operations > “Interpolation”

The Interpolation window is shown in Figure 36.

### 2.4.6.1 Usefulness

Interpolation allows modifying the energy array of an XAFS while preserving the absorption trend of the original spectrum. This could be done in Fastosh for two main purposes:

- **Data reduction:** Interpolation can reduce the size of an XAFS spectrum, for example from several thousand data points to only a few hundred data points.

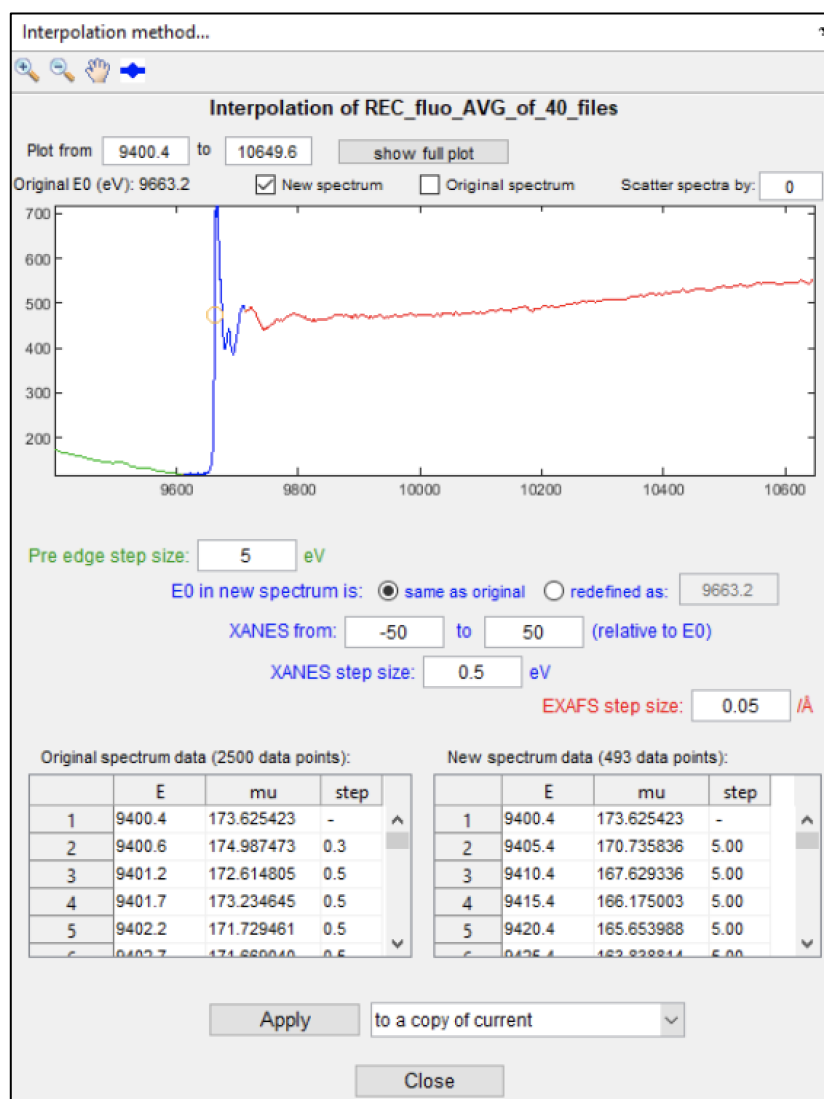


Figure 36 - Interpolation window

- **To process a set of EXAFS spectra:** Performing an operation, such as PCA, linear combination fitting, or MCR-ALS, on a set of EXAFS spectra may require to firstly employ the same e0 value in the XAFS to extract the EXAFS. Sometimes, a specific e0 energy value cannot be applied to multiple XAFS spectra when the value is not available in the XAFS energy arrays considered. It may be then required to interpolate all XAFS data to a common energy array before extracting the EXAFS.

**Note:** Whenever the program performs an operation on a set of XANES/mu spectra, such as spectral averaging, PCA, linear combination fitting, or MCR-ALS, the code firstly interpolates all spectra of the data set, except the first one, based on the energy array of the first spectrum of the data set.

#### 2.4.6.2 Interpolation methods

Two methods are available to interpolate an XAFS spectrum.

- With the first one (default method), specific parameters are applied to the pre-edge, XANES, and EXAFS regions:

Interpolation Menu > Interpolation method... > "Specific interpolations in pre-edge, XANES, and EXAFS regions"

**-1<sup>st</sup> region: pre-edge** This region starts at the beginning of the XAFS spectrum and finishes before the beginning of the XANES. This part of the XAFS should be featureless and ideally flat. A rough energy step size can be then used in this region, such as 5 or 10 eV. This part of the interpolated XAFS is displayed with a green color.

**-2<sup>nd</sup> region: XANES** The beginning and final energies of the XANES part are relative to the energy of  $e_0$ . A fine energy step size should be used in this part. Use the plot options “New spectrum” and “Original spectrum” to plot the data before and after interpolation, respectively. The chosen energy step size for the XANES should not result in a decrease in resolution of the sharp features present in the XANES. This part of the interpolated XAFS is displayed with a blue color in the interpolation window.

**-3<sup>rd</sup> region: EXAFS** The step size of this part of the XAFS is expressed as a wave number in k space ( $\text{\AA}^{-1}$ ). It can be converted into energy value (e.V.) following the expression below:

$$\text{Step size}_{\text{k space } (\text{\AA}^{-1})} = \sqrt{0.2625 * \text{Step size}_{\text{energy (e.V.)}}}$$

Because of the root square relationship in the above expression, an EXAFS with a constant step size value in k space ( $\text{\AA}^{-1}$ ) is equivalent to an EXAFS with increasing step size values in energy (e.V.). The EXAFS part of the interpolated XAFS is displayed with a red color in the interpolation window.

- With the second interpolation method, a constant step size in energy (e.V.) is applied throughout the entire spectrum:

Interpolation Menu > Interpolation method... > “Same step size throughout spectrum”

#### 2.4.6.2.1 Plot options

- The spectra before and after interpolation can be displayed by checking the boxes “Original spectrum” and “New spectrum”, respectively.
- The spectra can be plotted with user-defined  $E_{\min}$  and  $E_{\max}$ . To zoom back to the original full spectra, press the button “Show full plot”
- The spectra before and after interpolation can be vertically scatter from each other, using a value specify in the field “Scatter spectra by:”.

#### 2.4.6.2.2 $E_0$ value

Whether the first or second interpolation is chosen, one option enables to freely define the  $E_0$  value for which the XANES  $E_{\min}$  and  $E_{\max}$  of the interpolated spectrum is based on. This allows to make sure that this specific value is available in the energy array of all interpolated XAFS spectra to be processed as one data set.

To define a specific  $e_0$  value, choose the option “redefine  $e_0$ ” and specify the new energy value. The interpolated spectrum and interpolated data shown in the table “New spectrum data” refresh automatically. Otherwise, choose the option “same as original” (Default option). In that case, the  $e_0$  energy value of the original spectrum will be preserved in the new spectrum, and the interpolation will be based on this value.

#### 2.4.6.2.3 Transfer options to main GUI

Once the interpolation method and parameters have been defined, one can transfer the new spectrum to the Main GUI for further data processing. There are four options to transfer the data to the Main GUI:

- **To a copy of current:** a copy of the highlighted spectrum in the main GUI will be created
- **To current, no copy:** the highlighted spectrum in the main GUI will be replaced by its corresponding interpolated spectrum

- **To a copy of all selected scans:** a copy of all selected spectra in the Main GUI will be created
- **To all selected scans, no copy:** all selected spectra in the main GUI will be replaced by their corresponding interpolated spectra

Choose any of these options in the menu bar at the bottom of the Interpolation window, then press on the “Apply” button next to this menu bar (Figure 36) to proceed to the data transferring to the main GUI.

## 2.4.7 Deglitching strategy: manual or automatic?

Fastosh proposes multiple choices of deglitching methods, which could be classified in two main categories: manual (Section 2.4.8) or automatic (2.4.9) deglitching approaches.

**If the goal is to deglitch the chi spectrum, it is recommended in Fastosh to firstly try the automatic deglitching method as it requires no or minimum efforts to do the deglitching. It then allows to quickly find out whether it represents a suitable method to deglitch the chi spectrum.** For spectra featuring not many data points, e.g. scans collected using a step-by-step acquisition mode, a manual deglitching method may be the most appropriate approach. It indeed allows to exactly target the values corresponding to the problematic data points in the spectrum. In contrast, if many data points have to be processed, typically in XAFS spectra collected using a rapid acquisition mode, the automatic deglitching method may represent the best approach.

## 2.4.8 Manual deglitching

### 2.4.8.1 Presentation of manual deglitching methods & main window

The spectrum displayed & treated in this module corresponds to the current scan highlighted in blue color in the list of spectra uploaded to the main GUI. Therefore, to deglitch a specific scan, firstly select it in the Main GUI before opening the Manual Deglitching module. Then, to open the module, go to:

Main GUI Menu > Operations > “Manual Deglitching”

There are two different methods to manually deglitch the data:

- The first method consists in selecting a group of problematic data points in the spectrum by interactively drawing a box around them. All data points inside the box will be deleted. See procedure described in Section 2.4.8.2.1.
- The second method consists in editing one by one the raw data values corresponding to the glitches in the mu spectrum. The values of the problematic data points are then not deleted but modified to user-defined values. See procedure described in Section 2.4.8.2.2.
- Either way, the deglitching can be performed on the chi (default choice) or mu spectrum. Select in the menu:

Manual Deglitching Menu > Options > Mu data selection

or

Manual Deglitching Menu > Options > Chi data selection

When opening the Manual Deglitching module of Fastosh, a window appears featuring four figures (Figure 37):

- **The two figures on top of the figure** (mu and chi spectra) are used to deglitch the data. The deglitching is done on one of these two spectra, depending on the user choice set in Menu>Options. The default choice is the chi spectrum,

which is displayed without data interpolation onto a constant 0.05/Å step grid, meaning that all collected data points are shown.

- **The two figures at the bottom of the figure** correspond to the mu and chi spectra before (if “Original” box is checked) and after (if “New” box is checked) deglitching. The mu/chi spectra before and after deglitching are displayed with a purple and black color, respectively. Also, the chi spectra displayed at the bottom right corner of the window (before and after deglitching) are shown with a data interpolation onto a constant 0.05/Å step grid.

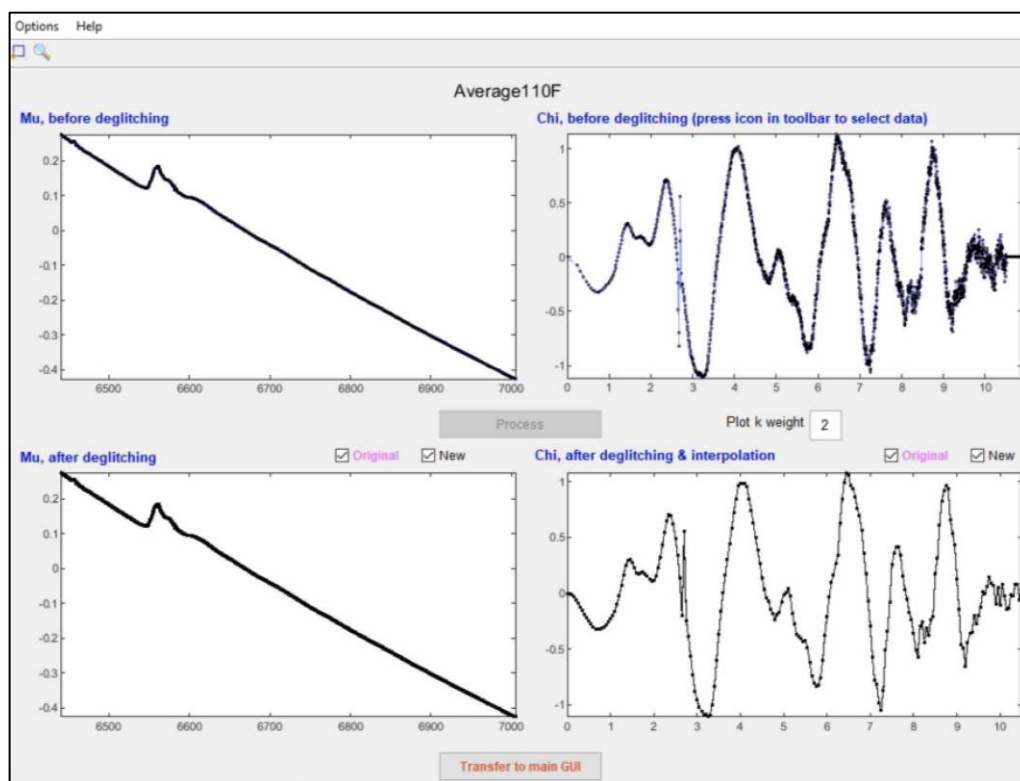


Figure 37 - Manual deglitching window at opening

## 2.4.8.2 Manual deglitching procedures

### 2.4.8.2.1 Interactively selecting a set of data point

#### Deglitching from chi spectrum

- Make sure the deglitching will be carried out on the chi spectrum:  
Menu>Options>“Chi data selection” must be checked
- Zooming step (optional): before interactively selecting the problematic data points to deglitch, it is possible to zoom inside the part of the chi spectrum where these data points are located. In the Menu bar, on the top left corner of the window, click on the magnifier icon (Figure 38A). Draw a box inside the chi spectrum in the region to zoom. **When done, click again on the magnifier icon to deactivate the zooming option.**
- Data selection step: click on the box icon in the menu bar (Figure 38B). Clicking on this button deactivates all figures of the window, except the one showing the chi spectrum to deglitch in the top right corner of the window. Draw a box in this figure that will contain all problematic data points to remove (Figure 38). All data points inside the box are then displayed with a red color.

- If not satisfied with the data point selection, reset the selection with the menu option: Menu>Options>"Reset Data Selection".  
If satisfied, press the "Process" button located in the middle of the window to deglitch the spectrum (Figure 38). The mu and chi spectra before and after deglitching are shown at the bottom of the window in black and purple colors, respectively (Figure 39).
- If not satisfied with the deglitched spectra, reset the whole window with the menu option: Menu>Options>"Reset deglitching".  
If satisfied, the new, deglitched spectrum can be exported to the Main GUI of Fastosh (where it can be saved) by pressing the button "Transfer to main GUI" at the bottom of the window.
- When all done, close the Manual Deglitching window.

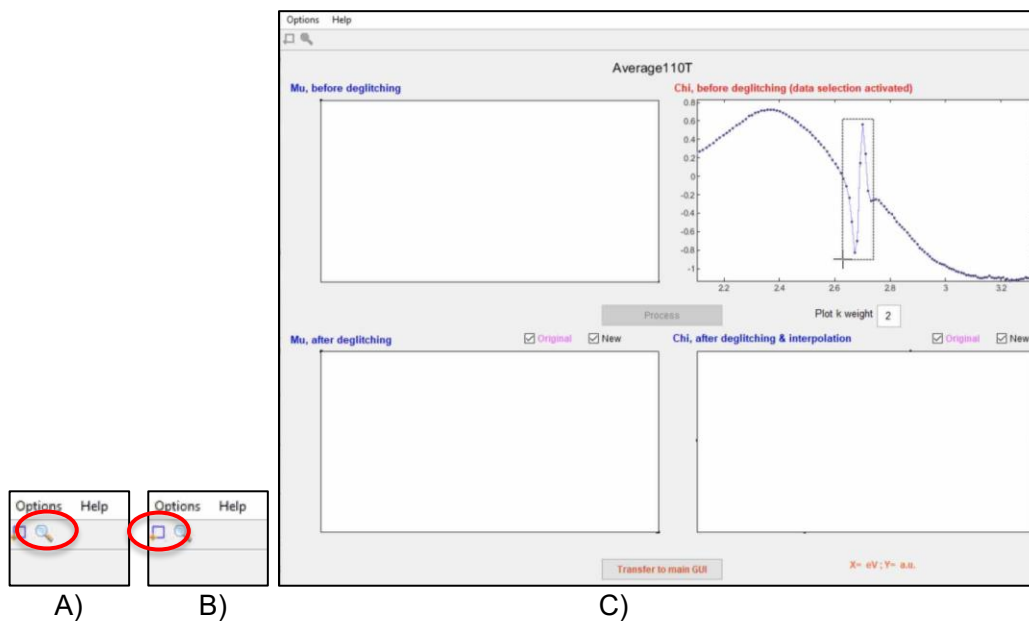


Figure 38 - To interactively deglitch a set of data points in the chi, firstly zoom in the region of interest in the chi spectrum using the magnifier icon in the menu bar (A), then select the little box icon in the menu bar (B) to draw in the chi spectrum a box containing the data points to remove (C)

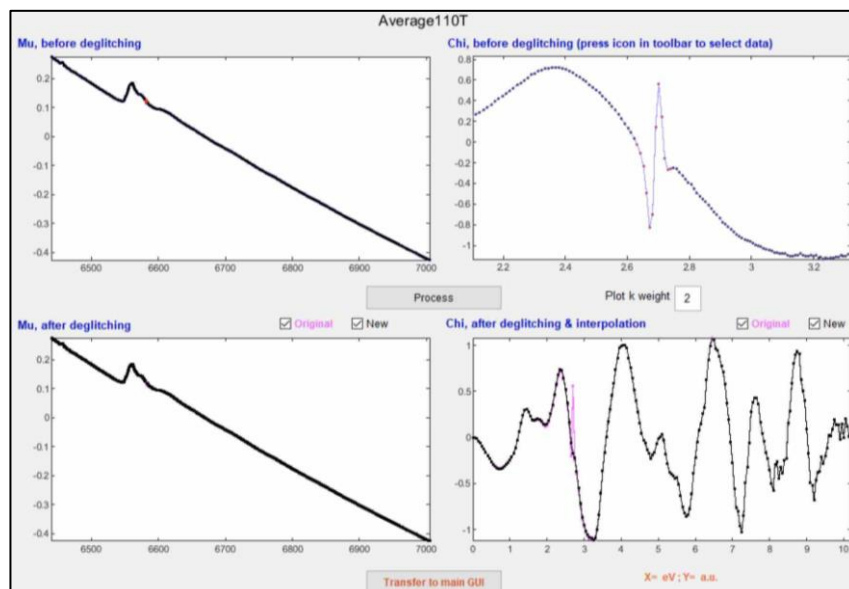


Figure 39 - The interpolated chi spectrum (bottom right corner of the window) was successfully deglitched based on the spectrum before (purple color) and after (black color) data processing

### Deglitching from mu spectrum

- Make sure the deglitching will be carried out on the mu spectrum:  
Menu>Options>"mu data selection" must be checked
- From there, it is the same procedure than the one mentioned above to deglitch the chi spectrum: firstly, zoom in the mu spectrum using the magnifier icon (Figure 40A), click on the box icon (Figure 40B) to draw, inside the mu spectrum figure, a box that will contain the problematic data points (Figure 40C), and then press the "Process" button. One can notice that the new chi spectra obtained by deglitching the chi (Figure 39) and mu (Figure 41) are similar to each other.

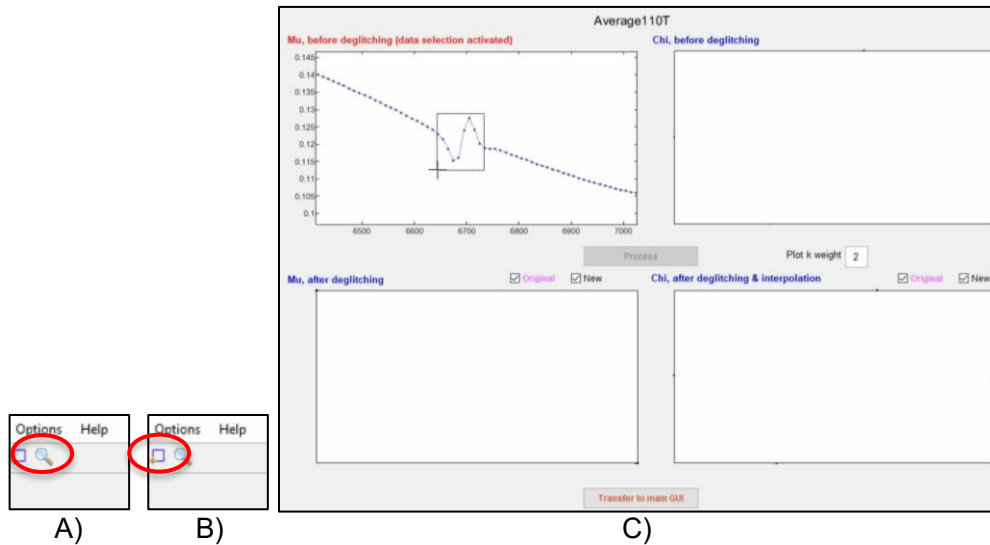


Figure 40 - To interactively deglitch a set of data points in the mu, firstly zoom in the region of interest in the mu spectrum using the magnifier icon in the menu bar (A), then select the little box icon in the menu bar (B) to draw in the mu spectrum a box containing the data points to remove (C)

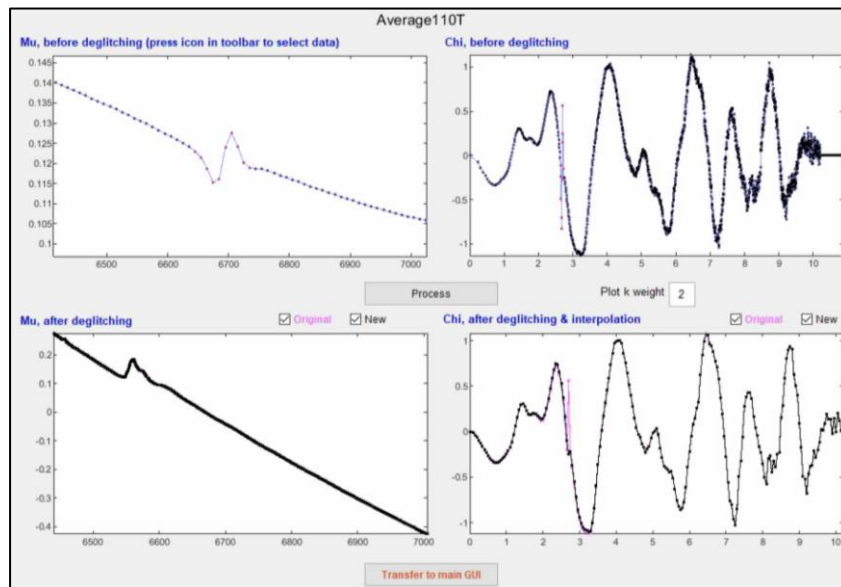


Figure 41 - The interpolated chi spectrum (bottom right corner of the window) before (purple color) and after (black color) deglitching the mu spectrum

1048	26918.52	2633.60
1049	26918.93	2552.02
1050	26919.32	2498.53
1051	26919.72	2619.66
1052	26920.10	2526.80
1053	26920.50	2587.16
1054	26920.91	2584.84
1055	26921.30	2580.82
1056	26921.70	2579.87
1057	26922.09	2517.83
1058	26922.50	2478.94
1059	26922.90	2524.79
1060	26923.30	2527.79
1061	26923.71	2482.10
1062	26924.10	2623.09
1063	26924.49	2514.52
1064	26924.88	2573.35
1065	26925.28	2500
1066	26925.69	2574.18
1067	26926.11	2539.56
1068	26926.53	2568.51
1069	26926.94	2544.59
1070	26927.33	2581.01
1071	26927.73	2499.00
1072	26928.13	2522.14
1073	26928.51	2490.90
1074	26928.91	2532.97
1075	26929.29	2482.23
1076	26929.70	2653.95

Figure 42 – Window to edit one at a time the data points in the mu spectrum

#### 2.4.8.2.2 Editing data point values

##### Procedure

- In the menu bar of the Manual Deglitching window, select:  
Menu>Options>"Edit Mu"
- A window appears, vertically displaying on the left and right side the energy and absorption arrays corresponding to the mu spectrum, respectively (Figure 42). Clicking on any absorption value automatically highlights in red color the position of the chosen data point in the mu figure, and possibly in the chi figure (if the chosen data point is above the E0 value) in the Manual Deglitching Window.
- Edit the value of the chosen data point. The new mu and chi are then automatically showed at the bottom of the Manual Deglitching Window.
- If not satisfied with the new mu & chi spectra, reset the whole window with the menu option: Menu>Options>"Reset deglitching".  
If satisfied, the new mu spectrum can be exported to the Main GUI of Fastosh (where it can be saved) by pressing the button "Transfer to main GUI" at the bottom of the window.
- When all done, close the Manual Deglitching window.

### 2.4.9 Automatic deglitching

#### 2.4.9.1 Presentation

##### 2.4.9.1.1 Principle

The spectrum treated in this module corresponds to the current scan highlighted in blue color in the list of spectra uploaded to the main GUI. Therefore, to deglitch a specific scan, firstly select it in the Main GUI before opening the auto-deglitching module. To open this module select:

Main GUI Menu > Operations > "Auto-Deglitching"

This module proposes two auto-deglitching approaches, which can be chosen using the menu of the Auto-deglitching window:

Auto-Deglitching Menu > Type > “Auto-Deglitching from chi” (Default choice)

Auto-Deglitching Menu > Type > “Auto-Deglitching from I0”

The principle of each approach is the following:

- **Auto-deglitching of the chi spectrum: for ASCII data generated at any beamline**  
The chi spectrum not background subtracted and featuring all data points (i.e. before chi is interpolated into a constant 0.05/Å step grid in k space) is fitted with a smoothing spline function. At each k value of the chi spectrum, the modulus of the experimental value is subtracted by the value of the fitted function. If the resulting value is above a defined threshold value, the experimental data point is considered as a glitch and removed from the mu spectrum. This approach can only identify glitches located in the region of the mu spectrum above the E0 value, as it employs the upper part of the XAFS spectrum to find them.
- **Auto-deglitching of the entire mu spectrum: for data saved in SAMBA HDF file only**  
To entirely deglitch the mu spectrum including its part below E0 (the part below chi), the data corresponding to the I0 channel is fitted with a smoothing spline function. At each E value of the spectrum, the modulus of the experimental value is subtracted by the value of the fitted function. If the resulting value is above a defined threshold value, the experimental data point is considered as a glitch and removed from the mu spectrum.

With both methods, the deglitched mu spectrum and its corresponding chi spectrum (background subtracted & interpolated to a 0.05/Å grid) are eventually displayed so that one can visually determine the effectiveness of the deglitching.

#### 2.4.9.1.2 Caveat on using Autodeglitching to multiple spectra

Since Fastosh version 1.0.0, two data options, which are available at the bottom of the autodeglitching window, allow to either deglitch only the sample that is highlighted in the Main GUI of Fastosh (“Transfer to a copy of current in Main GUI”) or all samples that are marked in the Main GUI of Fastosh (“Transfer to a copy of all checked in Main GUI”). In the latter case, the deglitching parameters that are defined in the Autodeglitching module for the highlighted sample are applied to a copy of all samples that are marked in the Main GUI of Fastosh. This option should be used only when glitches energy positions and intensities are reproducible from scan to scan. If they are not similar to each other, deglitching should be done individually on each sample. In this case, after deglitching a sample, close the autodeglitching window, select a new highlighted sample in the Main GUI, and reopen the autodeglitching window to deglitch the new sample.

#### 2.4.9.1.3 Description of the Auto-deglitching window

- **Auto-deglitching of the chi spectrum**

*1<sup>st</sup> window: “Fitting”*

The chi spectrum not background subtracted and featuring all data points (i.e. before chi is interpolated into a constant 0.05/Å step grid in k space) is displayed in black color, and its corresponding fit (a smoothing spline function) is shown in red color.

*2<sup>nd</sup> window: “Threshold selection”*

The values shown in black color in this figure correspond to the modulus of the experimental chi value subtracted by the value of the fitted function, at each k value of the chi. These values are scaled to %, with the highest value being equal to 100 %. The threshold level (in %) is also displayed with a horizontal dotted line. All values above this threshold are considered as glitches and are encircled in red color.

*3<sup>rd</sup> window: “Full data, before background subtraction and deglitching”*

The chi spectrum not background subtracted and featuring all data points (i.e. before chi is interpolated into a constant 0.05/Å step grid in k space) is displayed in black color and its data points identified as glitches are encircled in red color.

*4<sup>th</sup> window: "Mu, before deglitching"*

The original mu spectrum is displayed in black color and its data points identified as glitches are encircled in red color.

*5<sup>th</sup> window: "Chi"*

Chi spectrum (background subtracted & interpolated to a 0.05/Å grid) before and after deglitching in purple and black colors, respectively.

*6<sup>th</sup> window: "Mu"*

Mu spectrum before and after deglitching in purple and black colors, respectively.

- **Auto-deglitching of the entire mu spectrum: for data saved in SAMBA HDF file only.**

*1<sup>st</sup> window: "I0"*

The data of the I0 channel (ionization gas chamber before the sample) is displayed in black color, and its corresponding fit (a smoothing spline function) is shown in red color.

*2<sup>nd</sup> window: "Threshold selection"*

The values shown in black color in this figure correspond to the modulus of the experimental I0 value subtracted by the value of the fitted function at each E value of the I0 spectrum. The highest value is used to normalize all values in %, the highest value being 100%. The threshold level (in %) is also displayed with a horizontal dotted line. All values above this threshold are considered as glitches and are encircled in red color.

*3<sup>rd</sup> window: "Full data, before background subtraction and deglitching"*

The chi spectrum not background subtracted and featuring all data points (i.e. before chi is interpolated into a constant 0.05/Å step grid in k space) is displayed in black color and its data points identified as glitches are encircled in red color. If glitches were identified at energies below the E0 value (i.e. beginning E of the chi), they won't be displayed in this figure.

*4<sup>th</sup> window: "Mu, before deglitching"*

The original mu spectrum is displayed in black color and its data points identified as glitches are encircled in red color.

*5<sup>th</sup> window: "Chi"*

Chi spectrum (background subtracted & interpolated to a 0.05/Å grid) before and after deglitching in purple and black colors, respectively.

*6<sup>th</sup> window: "Mu"*

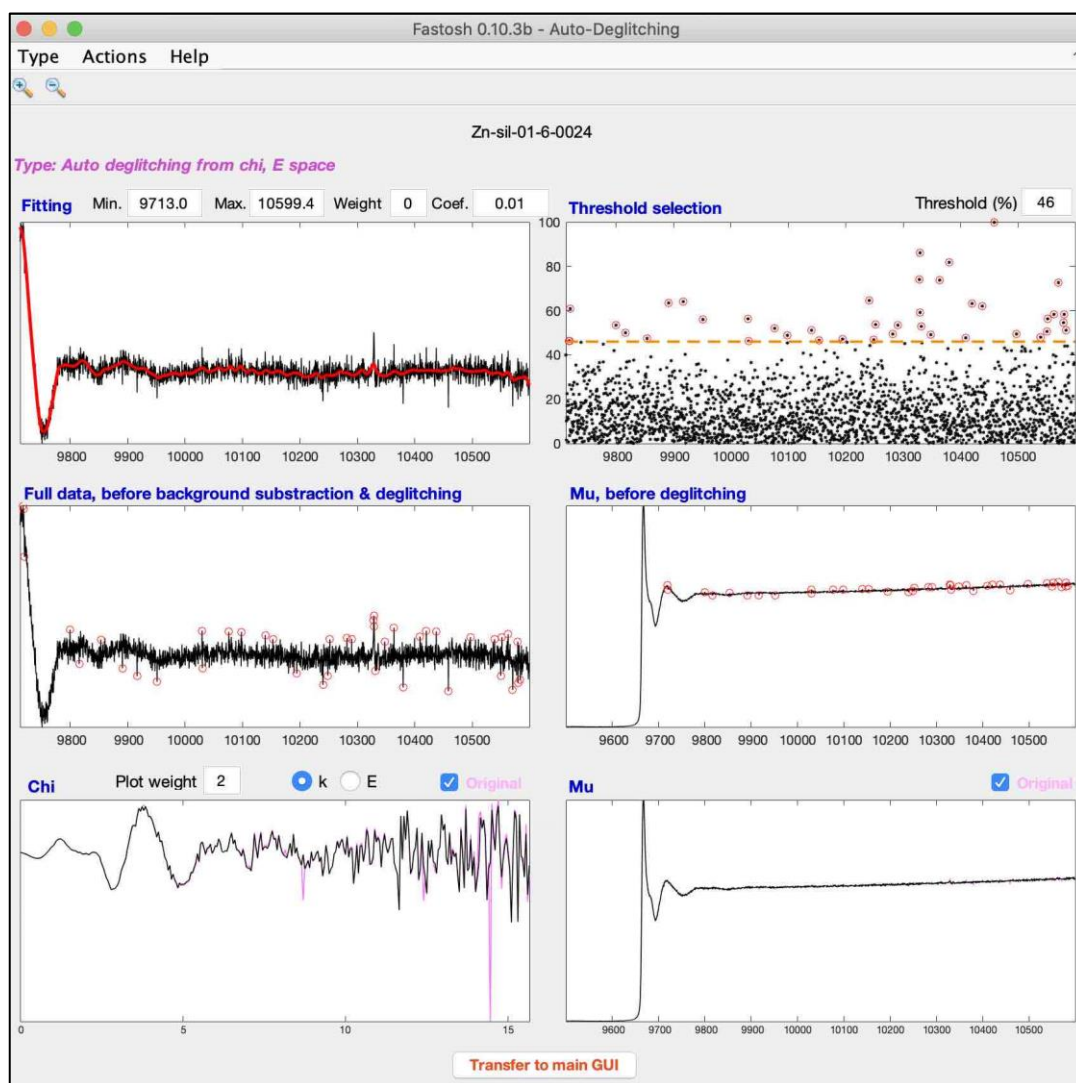
Mu spectrum before and after deglitching in purple and black colors, respectively.

#### **2.4.9.1.4 Parameters**

The two main parameters employed in both chi and I0 auto-deglitching approaches are the coefficient of the fitted function and the glitch threshold:

- **"Coef"** is the degree of the smoothing spline function used to fit the experimental spectrum. The lower its value, the stiffer the fit, thus the less likely abnormal values are reproduced by the fit. The default value of this parameter is set to 0.01. One can visually estimate whether this default value is satisfactory from the fit results displayed in the first figure of the auto-deglitching window. Ideally, the fit should exactly follow the variations in the experimental spectrum except those corresponding to the glitches.
- **"Threshold"**: any values above it will be considered as glitch in the spectrum. Its value is set by default to four times the standard deviation of all data shown in the second figure. This data corresponds to the modulus of the experimental spectrum subtracted by the value of the fitted function, at each position of the experimental spectrum.

#### **Example 1: No ajustement**

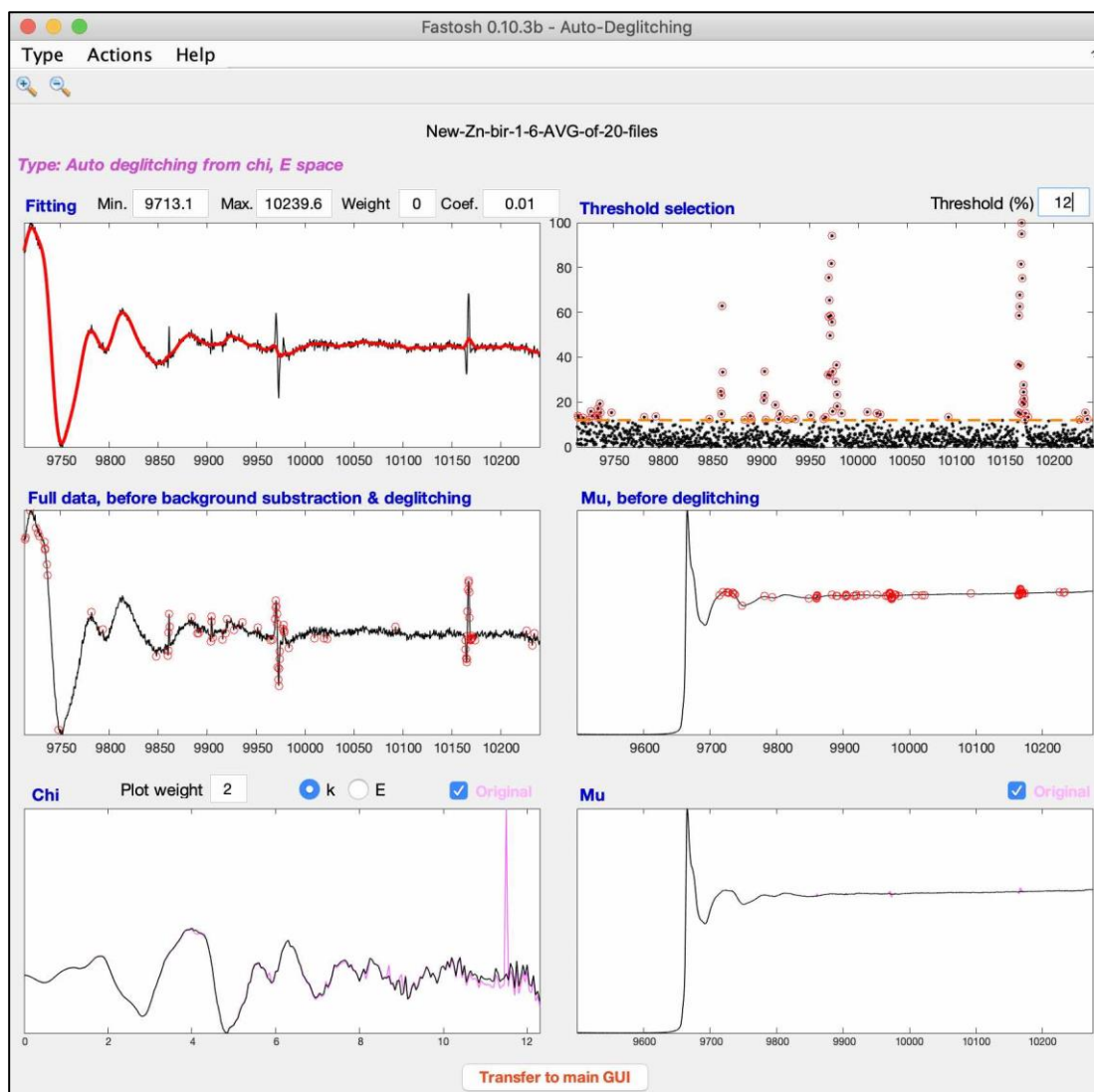


## 2.4.9.2 Examples

Below are examples of chi spectra deglitched with the auto-deglitching module of Fastosh. Sometimes, the spectrum is fully deglitched just by opening the module, implying that the default values of Threshold and Coef are adequate to obtain satisfactory deglitching results (Example 1). This can be the case when glitches do not significantly impact the experimental spectrum and/or the spectrum has a poor signal-to-noise ratio that masks the spectral imperfections. In other cases, however, it is necessary to decrease the value of Coef or Threshold (Example 2), or the values of both parameters (Example 3) to get satisfactory deglitching results. An example of mu spectrum deglitched using 10 channel data is also provided (Example 4).

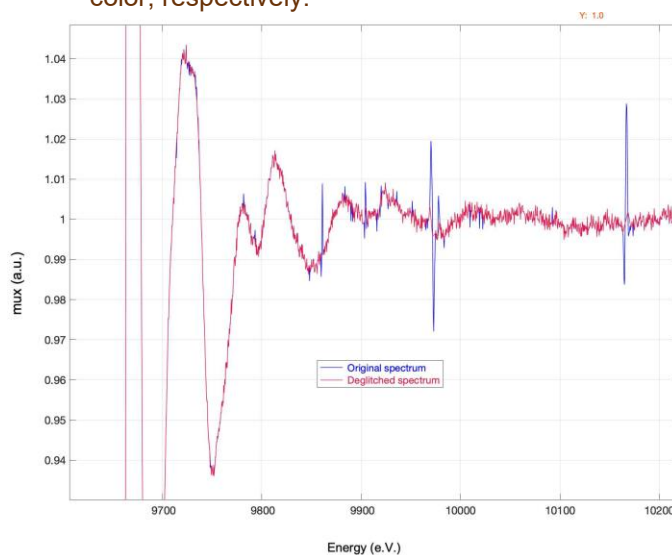
- Based on the original chi and new spectra shown at the bottom left corner in purple and black color, respectively, all obvious glitches were removed with the default values of Coef. (0.01) and Threshold (46 %).
- Simply press the button “Transfer to main GUI” at the bottom of the window so that the new deglitched spectrum can be saved or handled for further analyses in Fastosh.

### Example 2: Adjusting one parameter

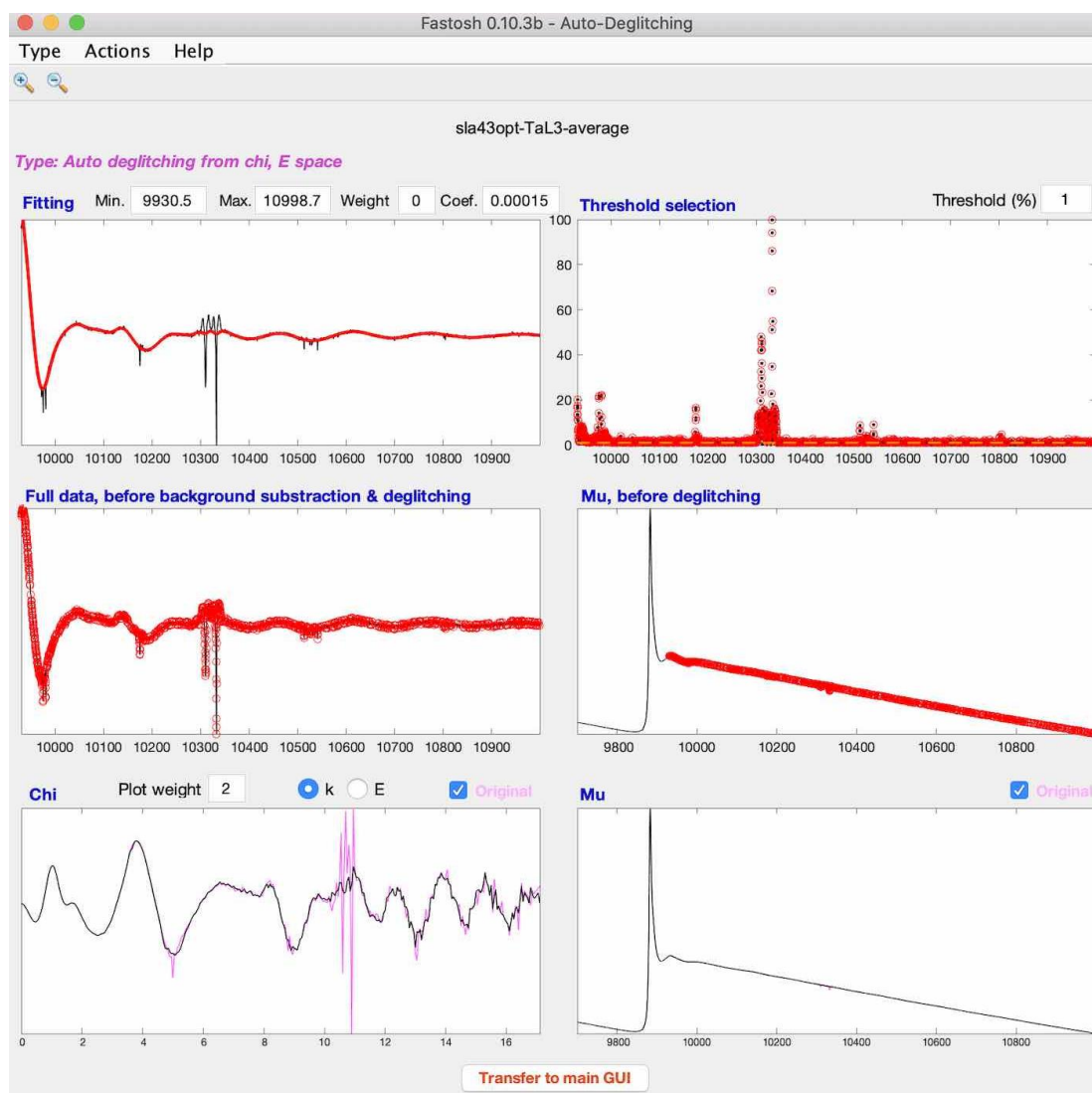


- Only the Threshold value was modified to get a satisfactory deglitched chi spectrum: the best value was empirically found at 12%.

These are the original and new spectra in the Main GUI, shown in blue and red color, respectively:



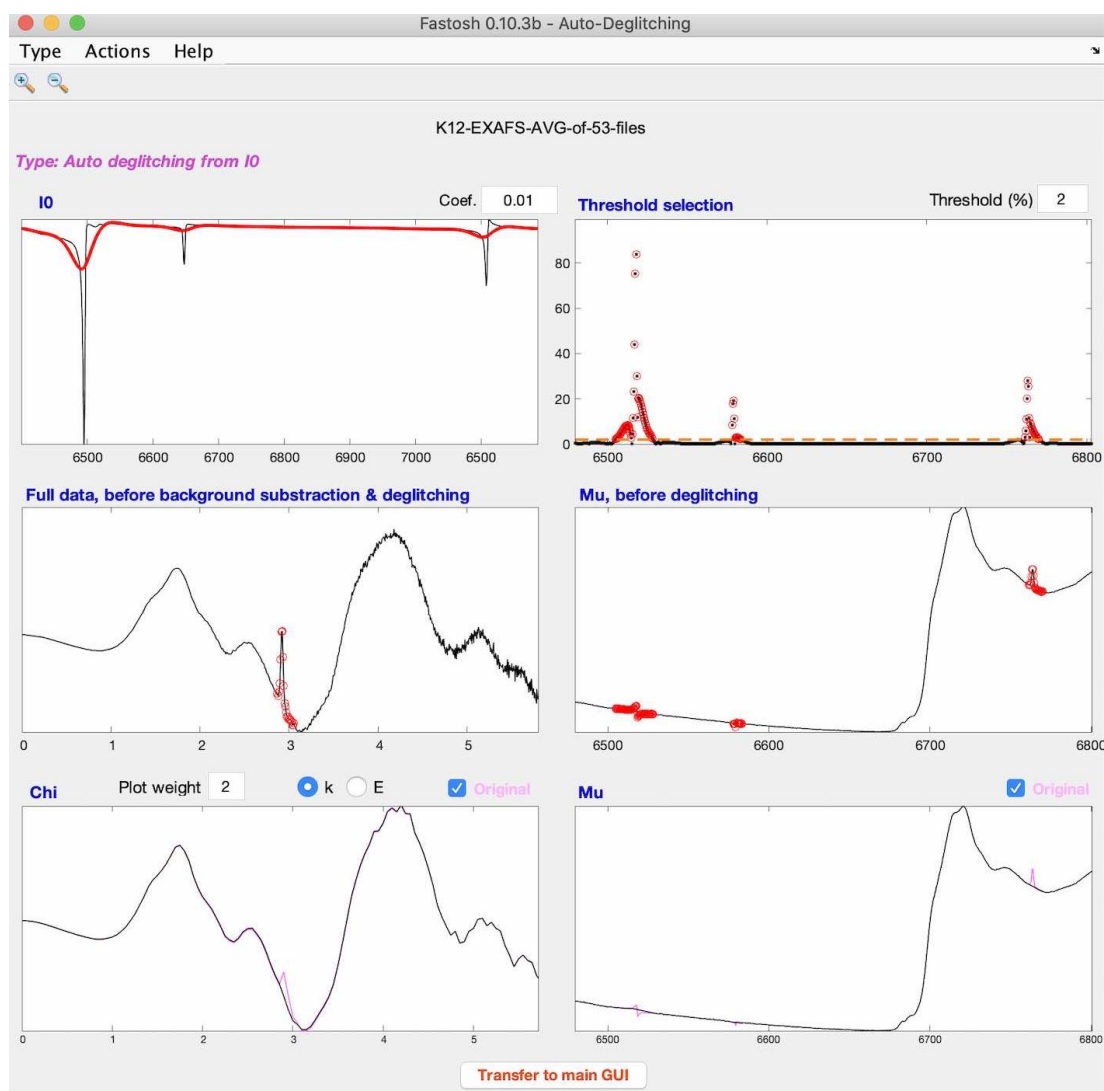
### Example 3: Adjusting two parameters



- Coef. was set at 0.00015. This enable to obtain a fit that only followed all real oscillations in the spectrum. If Coef. was set to a value above 0.00015, the fit was following the real oscillations but also some glitches around 10300-10350 eV in the original spectrum.
- Additionally, threshold was set to 1% to get a satisfactory deglitched spectrum.

### Example 4: mu deglitched using I0 channel (SAMBA data only)

- Here, only the Threshold value was modified to get a satisfactory deglitched mu spectrum. The best value was empirically found at 2%. This threshold value was applied to the I0 channel data to deglitch the mu spectrum.



## 2.4.10 Two-D Filtering

### 2.4.10.1 Usefulness & principle

A set of spectra evolving with time, for example corresponding to a chemical reaction followed *in-situ* at the beamline, can be filtered in 2D (energy + tile directions), instead of 1D (energy direction only) as shown in Figure 43 A. This may increase the accuracy of the filtering since noise is considered in both directions. An example of a data set treated with a 2D filter is shown in Figure 43 B & C.

Since Fastosh v1.0.7, the 2D filtering function applies at once a 2D Savitzky-Golay filter to the data set, instead of a 1D Savitzky-Golay filter applied successively in the energy and tile direction. The 2D filter can be applied either to the normalized mu spectra or EXAFS spectra not interpolated to a smooth 0.05 /Å grid (so that the filter is applied to all data points collected in the EXAFS).

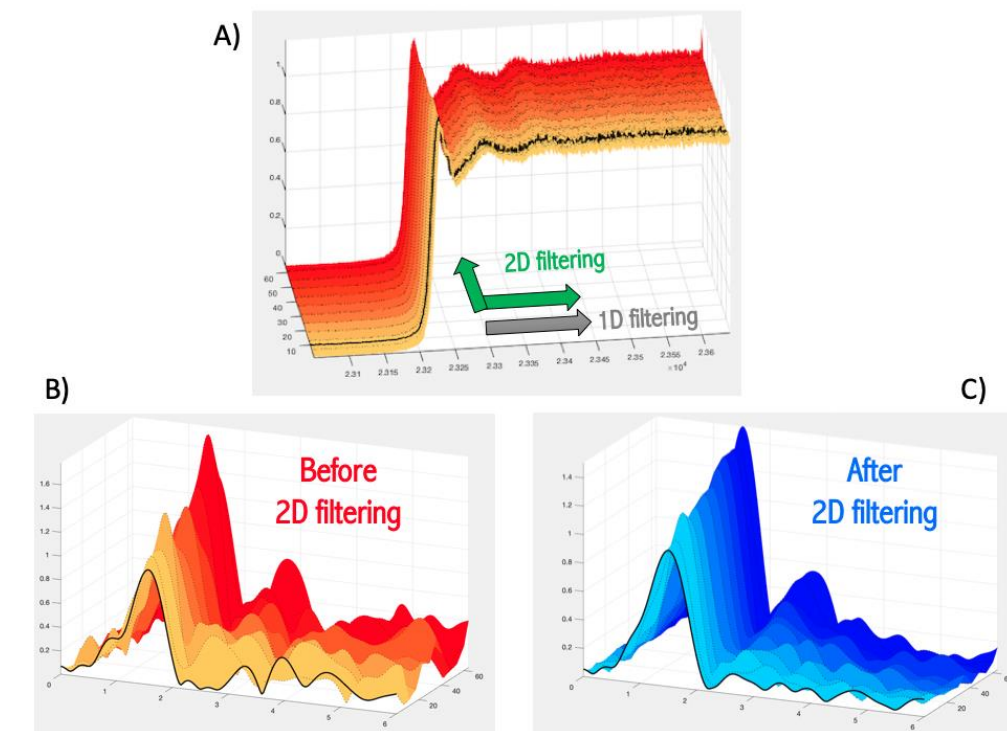


Figure 43 - A) Direction differences between 2D (energy & tile directions) vs 1D (energy direction only) filtering. The bottom of the figure features an example of a data set B) before and C) after 2D filtering

#### 2.4.10.2 Presentation of the 2D filtering window

Firstly, in the Main GUI, check all spectra of the data set to filter. Then, open the 2D filtering module via the Menu bar on top of the Main GUI window:

Main GUI Menu > Operations > “2D Filtering”

The 2D filtering window is three main parts, which can be seen in Figure 44.

- Filtering options:
  - Specify whether the 2D filter is applied to the normalized mu spectra or EXAFS spectra not interpolated to a smooth 0.05 /Å grid
  - Specify the parameters of the 2D Savitzky-Golay filter
  - Apply filter to all the data set
- Plotting options
  - Display the original data, or filtered data, or original + filtered data
  - Display the entire dataset or the current spectrum, in the spectral or tile direction. Displaying the current spectrum enables to visualize locally on the data the effect of the Savitzky-Golay filter
- Data plot and transfer of results to Main GUI

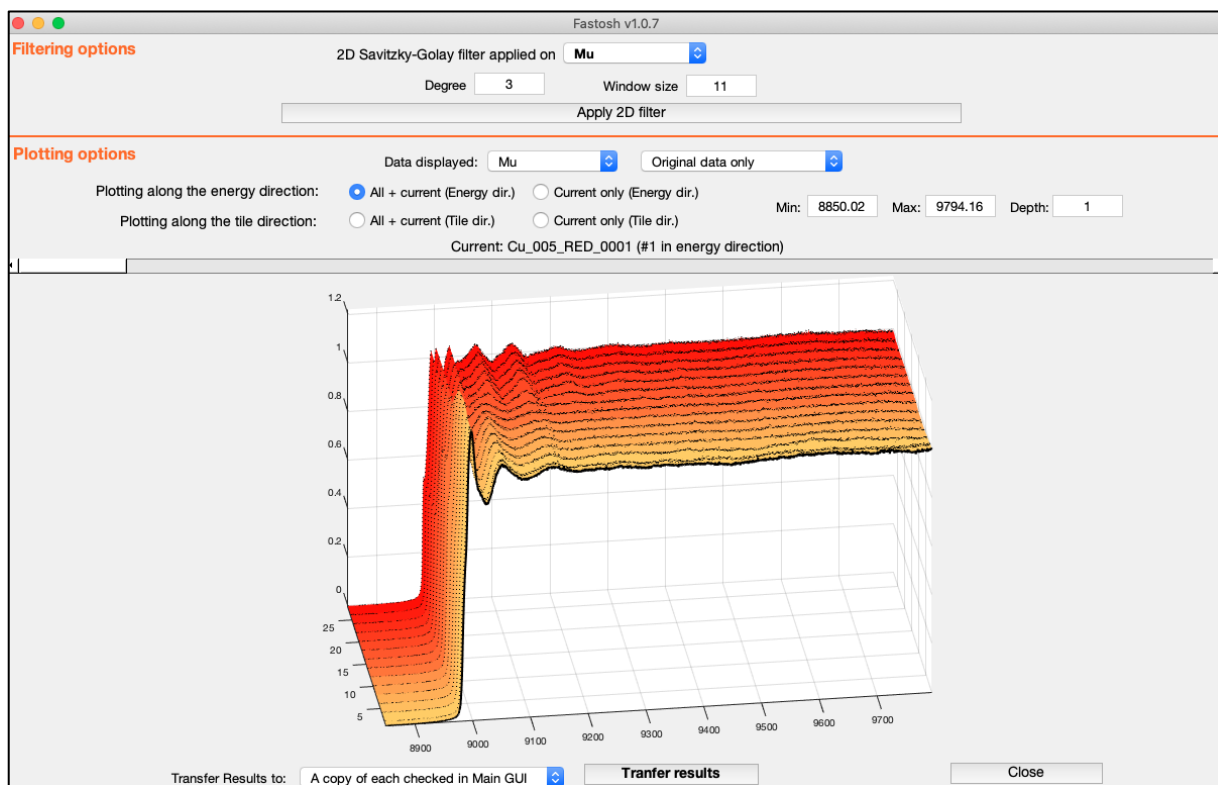


Figure 44 - Two-D filtering window. Different parts of the window are shown with a specific color (description in the text)

#### 2.4.10.3 Setting & testing filter parameters

The 2D Savitzky-Golay filter has two parameters to specify: filter degree & frame length. A filter degree of 3 (default value) should be suitable in most cases. The frame length value must be an odd number and cannot be below the filter degree. Increase the frame length, apply the filter to all, and visually assess the filtering efficiency on the highlighted spectrum (energy direction) or energy value (tile direction), or the entire dataset.

#### 2.4.10.4 Viewing and transferring 2D filtering results

Multiple plotting options are available to display the original data set and/or filtering results. When satisfied with the results, press the “transfer results” button, after specifying the type of transfer from the scrollbar next to the transfer button. This will transfer the new filtered spectra to the Main GUI, where the data can be further analyzed or saved.

### 2.4.11 Useful tips!

#### 2.4.11.1 Displaying progressive difference

Small spectral variations occurring in a set of XAFS spectra sometimes cannot be visually detected in a regular 2 or 3D plot of the data set. Interesting findings can then remain unnoticed. Displaying a progressive difference may be useful to observe these small variations. Each spectrum of a progressive difference plot (PDP) is essentially each spectrum of a data set (DS) subtracted by the first spectrum of the data set:

1<sup>st</sup> spectrum of PDP = 1<sup>st</sup> spectrum of DS - 1<sup>st</sup> spectrum of DS  
 2<sup>nd</sup> spectrum of PDP = 2<sup>nd</sup> spectrum of DS - 1<sup>st</sup> spectrum of DS  
 3<sup>rd</sup> spectrum of PDP = 3<sup>rd</sup> spectrum of DS - 1<sup>st</sup> spectrum of DS  
 4<sup>th</sup> spectrum of PDP = 4<sup>th</sup> spectrum of DS - 1<sup>st</sup> spectrum of DS  
 etc...

For example, a set of XAFS corresponding to a chemical reaction followed *in-situ* at the beamline is displayed as a regular 3D plot in Figure 45. The green and red/blue spectra were taken before and after the beginning of the reaction, respectively. All red spectra look similar to each other, beside an obvious progressive decrease in their white-line intensity (Figure 45). In contrast,

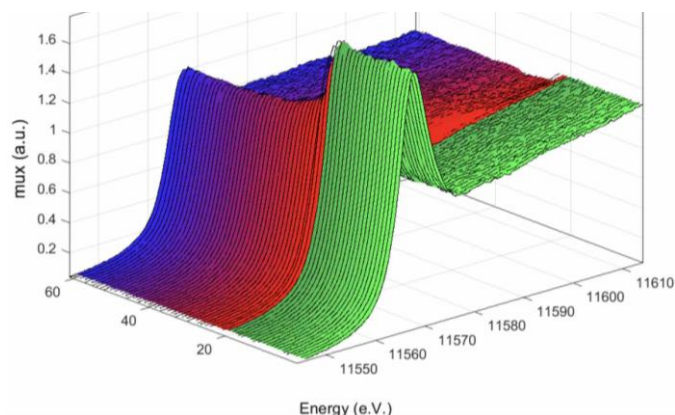


Figure 45 - Regular 3D plot of a data set corresponding to a chemical reaction followed *in-situ* at the beamline. No intermediate species can be observed from these spectra.

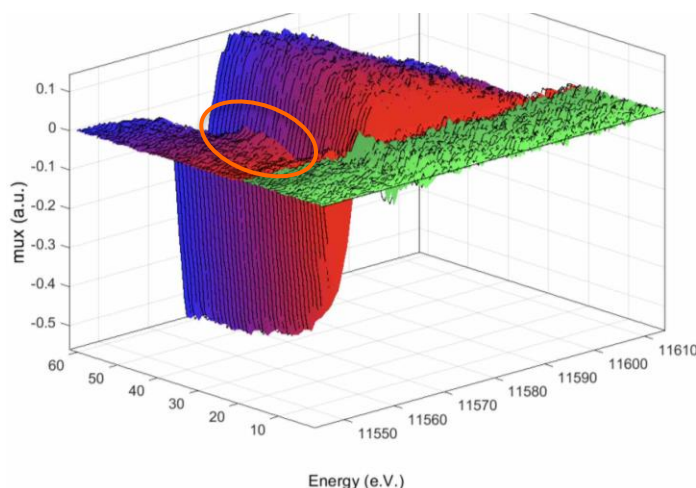


Figure 46 - Progressive difference plot of the data set shown in the previous figure. This reveals a temporal spectral modification around the edge-jump energy in some of the red spectra, which remained unnoticed when displaying the data with a regular 3D plot.

the progressive difference plot of this data set reveals a temporal spectral modification around the edge-jump energy in some of the red spectra (Figure 46), which suggests the presence of reaction intermediates. Applying an MCR-ALS fitting approach on this set data provided further evidences that intermediate species could have existed during the chemical reaction.

This plotting option can be activated via the menu of the Main GUI :

Main GUI Menu > Plot > “Progressive Difference (Multiple Selection)”

Then, press a multiple-scan plot button in the Main GUI to display the progressive plot.

#### 2.4.11.2 Displaying the stack tile number in a 3D plot of highlighted scan in Main GUI

This functionality enables to identify in the 3D plot what is the spectrum corresponding to the current scan in the Main GUI, whose name is highlighted in blue color in the list of all spectra uploaded to the program, and its corresponding tile number in the stack plot. For example, the current scan in the Main GUI shown in Figure 47 is “A\_08\_red\_Pt\_L3\_0040”. This spectrum is

highlighted in black color in the 3D plot, while all other spectra are colored in grey (Figure 47). Its corresponding tile number is also displayed at the bottom of the plot ("Tile #39").

This plotting option can be activated via the menu of the Main GUI:

Main GUI Menu > Plot > "Show Highlighted Scan (3D)"

Then, press a multiple-scan plot button in the Main GUI to display the highlight the current scan. As explicitly stated, this function is available only if the 3D plotting option is activated in Menu>Plot.

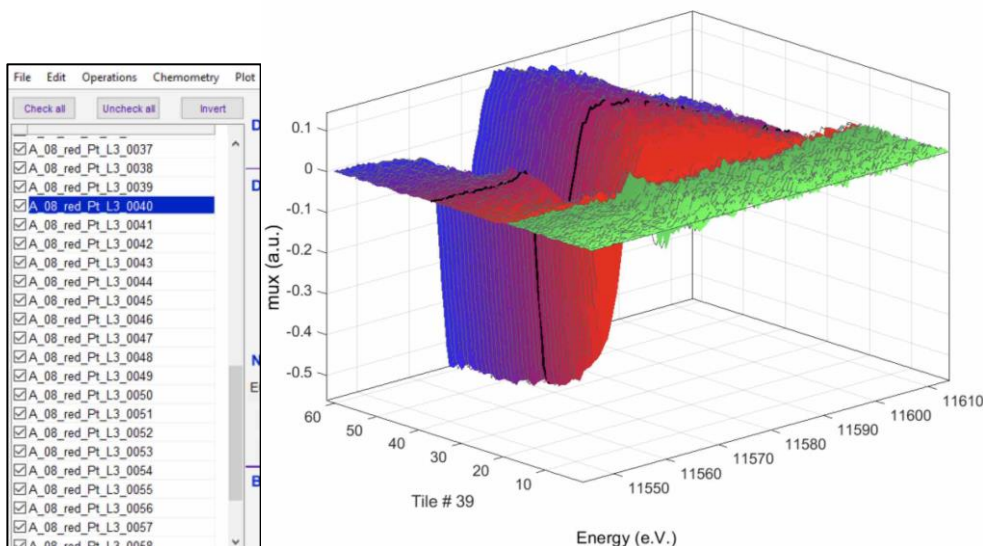


Figure 47 – The current scan in the main GUI ("A\_08\_red\_Pt\_L3\_0040") is highlighted in black color in the 3D plot while all other spectra are displayed in grey color. Its corresponding tile number in the stack is also shown at the bottom of the plot.

#### 2.4.11.3 Moving up and down sample names in list of Main GUI

The position of a sample listed in the sample list of the Main GUI can be moved up or down in the list. Press the key "q" on your keyboard to move up the sample in the list, or "w" to move down the sample.

#### 2.4.11.4 Applying one specific parameter of the highlighted scan to other scans

To apply only one specific parameter of the highlighted scan to other scans, right click on the name of the parameter next to the field where its value can be edited. For example, to apply only the E0 value of the highlighted scan to other scans, right click on the text "E0:" on the left side of the E0 value displayed in a box in the middle of the Main GUI. A menu appears where "Apply this E0 to all selected" or "Apply this E0 to all in list" can be selected.

To apply all parameters of the highlighted scan to other scans, right click or go to Menu>Edit and select "Apply Current to all Selected" or "Apply Current to all in list".

#### 2.4.11.5 Modifying, saving, and reimporting preferences of data plotter's look

Since Fastosh v.1.0.6, it is possible to modify the look of the data plotter. Click on the orange icon "OPT" in the Data Plotter Menu bar to open the plot preference window (Figure 48). In this window your personal preferences (including the size of the window can be saved in a ".Fplot" file, and can be reloaded anytime. For example, this can be potentially useful for quickly obtaining small figures, with a look of your taste, to add in written reports or oral presentation slides.

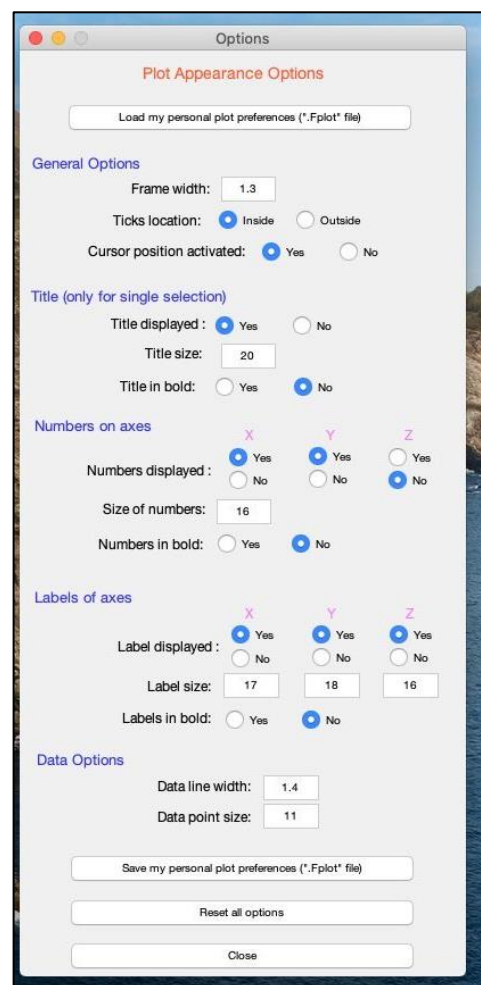
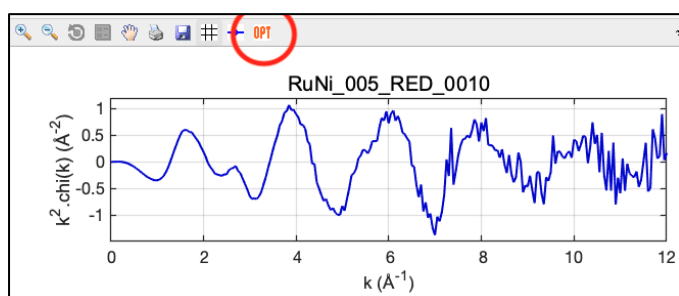


Figure 48- To open the plot preference window (right), press the button “OPT” in the menu bar of the Data Plotter window (left)

## 2.4.12 Pre-data treatment of non-XAFS data

Non-xafs data imported to Fastosh can be pre-treated before being processed by MCR-ALS. The available pre-treatment functionalities are:

- Spectrum truncation: see Section 2.4.5.

Main GUI Menu > Operations > “Tuncating”

- Baseline subtraction, in:

- Main GUI Menu > Options > Non-xafs data options> « Baseline subtraction »

This tool employs the baseline subtraction function of Mazet et al. 2005 [12].

- Zeroing-out problematic spectra

This functions enables to substitutes the spectrum Y data for zeros values. This can be notably useful for removing problematic spectra belonging to a 2D map: if the problematic spectra were simply deleted from Fastosh, some pixels of the map would be missing in the data matrix to be processed by MCR-ALS. Zeroing-out the data then enables to “preserve” the number pixels, hence the dimensions of the map are unmodified: the MCR-ALS results relative to the map can then be plotted in 2D. One can zero-out the current spectrum highlighted in the sample list, all selected spectra in the sample list, or all selected spectra in the sample list that have at least one Y value above a user-defined value. This function is available in:

- Main GUI Menu > Options > Non-xafs data options> « Zero-out... »

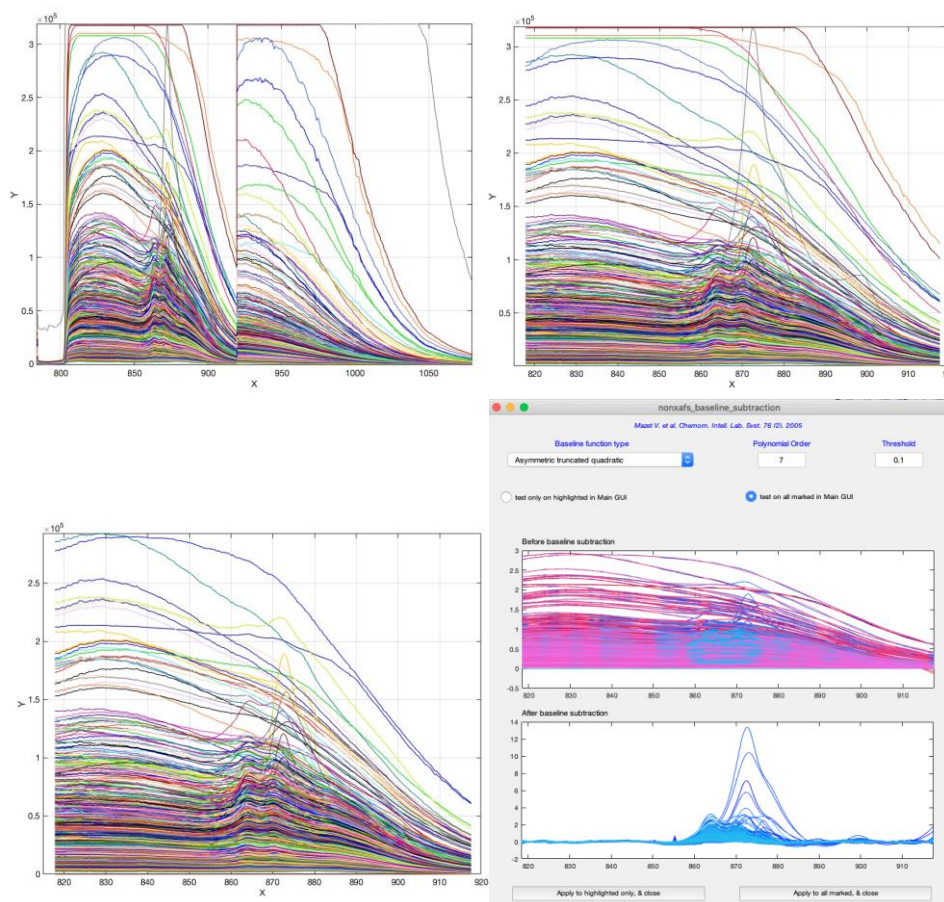


Figure 49 - Example of a Raman dataset, consisting of 1250 spectra belonging to two RAMAN 2D maps of 25x25 pixel in size, processed using the pre-data treatment tools of Fastosh specific for non-XAFS data

### Example

- Two Raman maps were collected, each of them were 25x25 pixels in size (625 RAMAN spectra were collected per map).
- All 1250 RAMAN spectra were imported to Fastosh as “non-XAFS” data; the 625 spectra of the first map, then the 625 spectra of the second map (Figure 49, top left).
- Using the data truncation tool, the spectra were truncated to about 820-920 (Figure 49, top right).
- The Y values of all spectra that have at least one Y value above 300000 were set to zero (Figure 49, bottom left).
- The RAMAN spectra were then background subtracted using the baseline subtraction tool (Figure 49, bottom right). The resulting data can be then treated by MCR-ALS.

## 3 Live Viewer

This module of Fastosh can be open via the menu of the Main GUI:

Main GUI Menu > Operations > “Live Viewer”

### 3.1 Presentation

#### 3.1.1 Usefulness

This module enables to rapidly visualize and average spectra saved in a specific data folder. It can also show the progressive improvement of the estimated random noise and signal-to-noise ratio for a set of spectra corresponding to a specific sample. Therefore, this module can be used:

- directly at the beamline, as a fully-automatic tool to display all XAFS spectra or their corresponding average spectrum for the sample being analyzed. It can be also helpful in determining when enough data has been collected for the sample being analyzed.
- after the beamtime, as a post-acquisition tool to glimpse all scan iterations collected for a given sample, and merge them into one average spectrum, which is transferrable to the Main GUI.

#### 3.1.2 Description of Live Viewer window

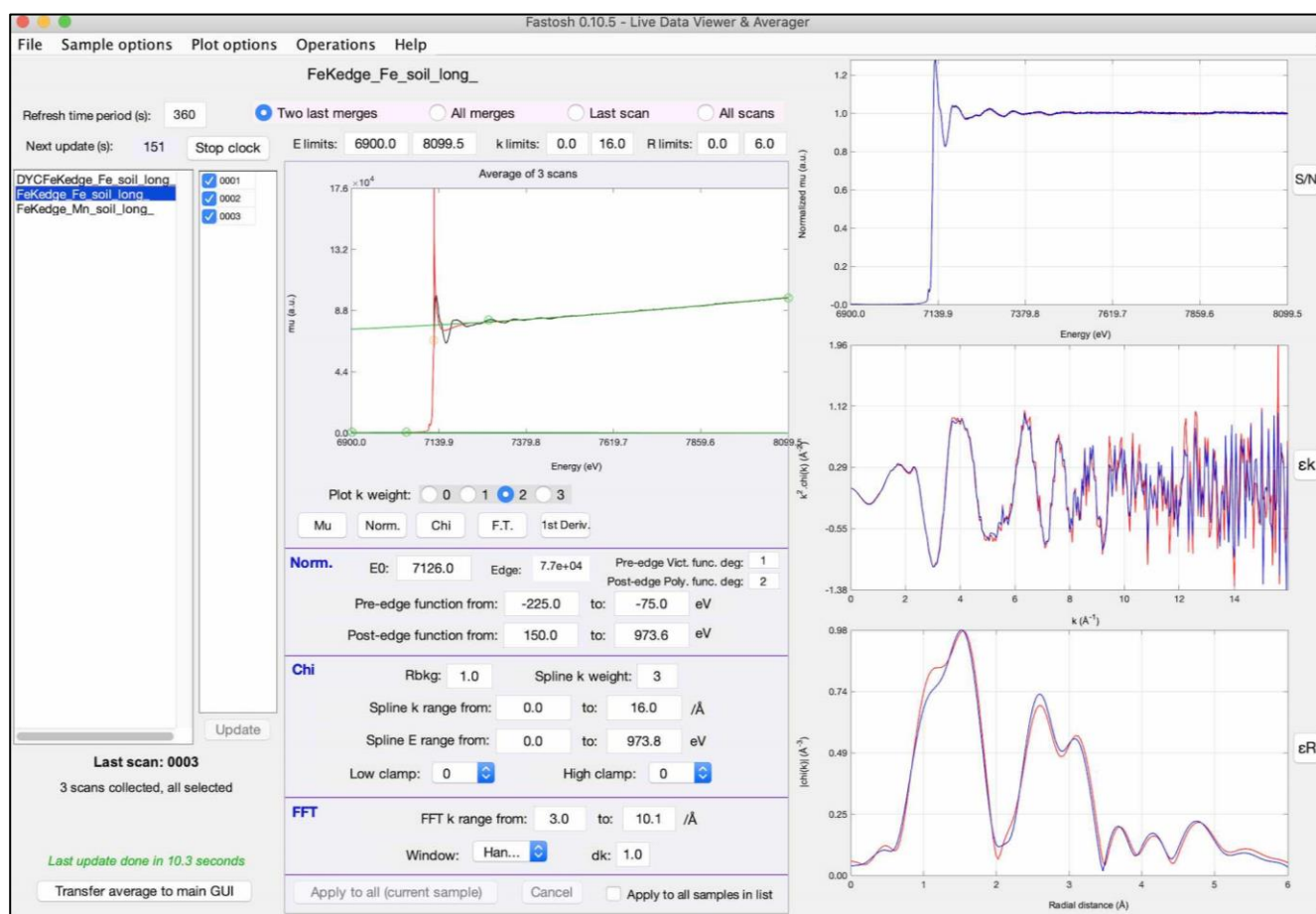


Figure 50 - Live Viewer window

An example of a data set saved in a specific folder and processed by the Live Viewer is shown in Figure 50. To fill out the content of this window, only one spectrum saved in a data folder

was imported to the module as the chosen spectrum; the program automatically found in the data folder all iterations relative to the chosen spectrum, and all other sample names whose spectra are similar in nature than the chosen spectrum (details on how the module operates are provided in the next section). The Live Viewer window is divided in seven parts, which are highlighted in Figure 51 in specific colors:

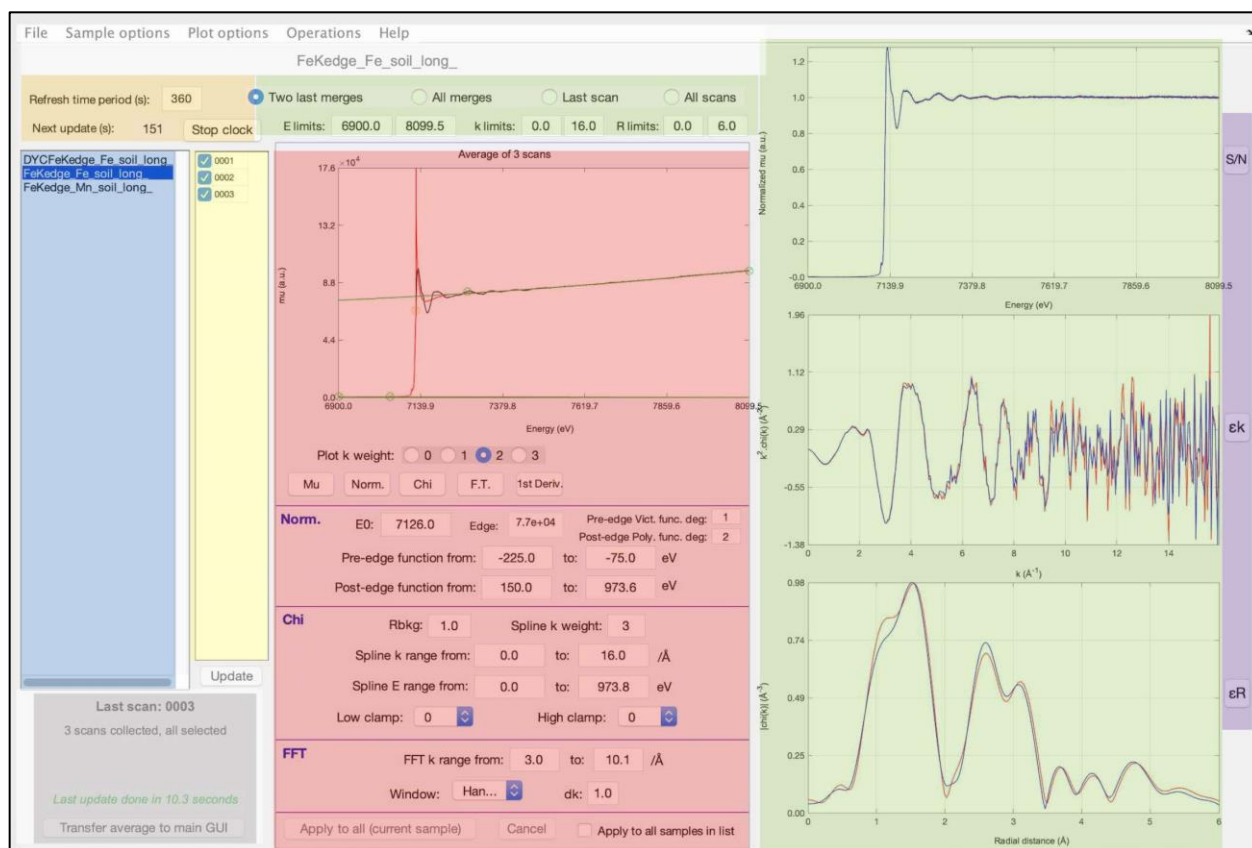


Figure 51 – Seven different parts of the Live Viewer window, highlighted with specific colors (see text for description of each part)

- **Blue** – list of all samples found in the data folder. These correspond to specific file names whose first iteration spectra are similar in nature to the spectrum imported to Live Viewer. The sample names found in the data folder are shown in this list without the multiple scan iterations associated to each of them. One of these sample names is highlighted in blue color. The rest of the data shown in the window (i.e. scans parameters displayed in the middle of the window, and plots displayed in the right side of the window) corresponds to this highlighted sample.
- **Yellow** – list of all scan iterations found in the data folder **for the current highlighted sample**. The spectrum associated to each of these iterations is similar in nature to the spectrum imported to Live Viewer.
- **Green** – On the right side of the window, three plots feature the data relative to **the current highlighted sample**: normalized  $\mu$ ,  $\chi$ , and Fourier Transform. In these plots, the data relative to the highlighted sample can be shown in multiple ways depending on the Plotting options available on the top middle part of the window (Section 3.3.3), and those available from the menu bar.
- **Purple** – Three buttons ( $S/N$ ,  $\epsilon_k$ ,  $\epsilon_R$ ) are located on the very right side of the Live Viewer window. Pressing any of them opens up a secondary window that show the signal-to-noise ratio and estimated random noise for all merges related to the current highlighted sample (Section 3.3.5).

- **Red** – This box features all Larch function input parameters that are used to process all XAFS spectra relative to the current highlighted sample. These spectra must be processed using the same input parameters (e.g.  $E_0$  value, plot  $k$  weight, spline range) so that they can be conveniently compared to each other in the Live Viewer window. The XAFS data displayed in the box corresponds to the average spectrum of all iterations of the highlighted sample, from which the default parameters are obtained during the first automatic update. All parameters in the box can be manually modified by the user. If one parameter is modified, the entire box switches to an edit mode and the countdown to the next automatic update is set on pause. The edit mode is no longer active and the countdown resumes when the button “Apply to all (current sample)” is pressed at the bottom of the box.
- **Orange** – Time period between two automatic updates (set to 360 seconds by default), and countdown to the next update, in seconds.
- **Gray** – Information related to the processing of the current highlighted sample. If the highlighted sample has more than 20 checked iterations, the iterations are split into groups so that the automatic update can be always rapidly completed regardless of total iteration number (Section 3.2.2). The number of groups created, scan iterations per group, and left-over iterations are displayed in this part of the window.  
At the left bottom side of the Live Viewer window, a button allows to transfer to the Main GUI the average spectrum of all checked iterations corresponding to the current highlighted sample.

## 3.2 How it operates

### 3.2.1 Generalities

#### 3.2.1.1 Principle, in a nut shell

The Live Viewer module automatically identifies, in a specific data folder, all file names whose corresponding spectra are similar to each other in nature (i.e. similar  $E_{\min}$  and  $E_{\max}$ ). All these file names are then listed on the left side of the Live Viewer window (Figure 50). One file name present in this list is highlighted in blue color. This highlighted sample can be manually or automatically defined, as mentioned in Section 3.2.3. All scan iterations relative to this specific file name are displayed in the middle of the Live Viewer window (Figure 50). The XAFS spectra displayed in the Live Viewer window then correspond to these iterations relative to the highlighted sample. The program automatically performs this file identification in the data folder on a regular basis, based on a countdown in seconds. If a new file name, or a new scan iteration relative to the highlighted sample, has been added to the data folder, the program may automatically reprocess all XAFS data displayed in the Live Viewer Window at the next cyclic update.

#### 3.2.1.2 Importing an XAFS scan

To identify all spectra that are similar to each other in the data folder, one of them must be imported to the Live Viewer module. The  $E_{\min}$  and  $E_{\max}$  of this particular spectrum will be used as reference values during each automatic update, whose steps are described below in Section 3.2.2.

The file to import can be a SAMBA HDF file or an Ascii file (.txt, .dat, .DAT,) or. If it is an Ascii file, an iteration number must be featured at the end of the file name, for example:

MySample\_0001.txt  
MySample1.txt  
MySample\_\_1.txt  
etc...

##### 3.2.1.2.1 Opening a SAMBA HDF file (SAMBA data only)

The data opener window for SAMBA HDF file is very simple as it only requires to specify whether transmission or fluorescence data should be imported to the Live Viewer window. After

data importation, the chosen acquisition mode is displayed in the middle of the Live Viewer Window as a reminder of what data type was chosen. Optionnally, one can also import the reference spectrum. If this option is selected, the average data exported from the Live Viewer module to Fastosh Main GUI will be automatically aligned.

### 3.2.1.2.2 Opening an ASCII file

Opening an ASCII file in the Live Viewer module is similar to opening an ASCII file in the Main GUI, which is discussed in Section 2.3.2. The columns relative to the energy and mu data saved in the ASCII file must be specified in a window similar to the one shown in Figure 2. In this window, the column(s) in the ASCII file relative to the reference data can be also specified. Additionally, it is possible to specify on top of the ASCII file opener window whether the sample data being imported was collected in transmission or fluorescence acquisition mode. If any of these two modes is specified, it will be displayed in the middle of the Live Viewer Window as a reminder of what data type was chosen.

### 3.2.1.2.3 Opening the reference data collected along with sample data and auto alignment

An option available in the ASCII or SAMBA HDF file opener window enables to import to the Live Viewer the reference data collected along with the sample data. If this option is selected, the reference data is not displayed in the Live Viewer window. However, when an average spectrum is generated from the Live Viewer and transferred to the Main GUI, the merge sample and reference data are both included in the transferred spectrum, just like when the sample and associated reference data are imported to the Main GUI.

Additionally, since version 1.0.3 of Fastosh, the merge sample and reference data are auto aligned during transfer to the Main GUI. If this option is selected, the reference associated with the chosen sample must be calibrated after data importation. Calibration is done from a window (Figure 52) that appears right after closing the ASCII or SAMBA HDF file opener window. In this window, move the cursor position to the location of the spectrum where the energy can be redefined to a known value, if the default cursor position is not suitable. The default cursor position is found at the first inflection point of the mu spectrum. Then, redefine the energy at this data point of the spectrum in the field "Modify cursor energy to". Finally, press the button "Apply" to do the actual calibration of the reference. Finally, press "Done" to close the window.

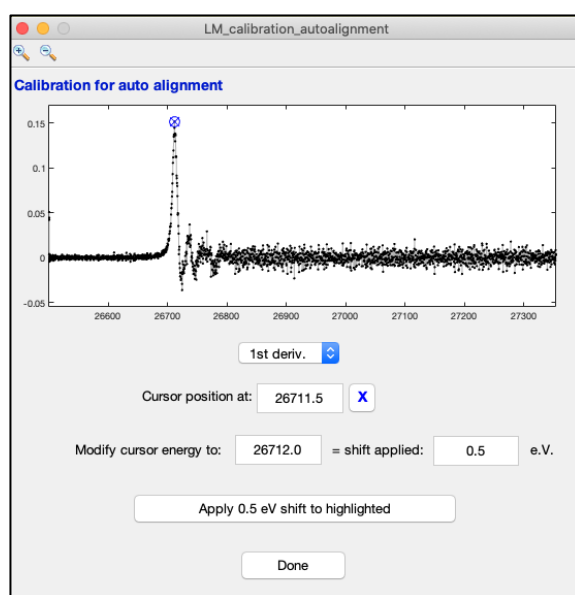


Figure 52- Calibration of reference used for auto alignment

### 3.2.2 Description of all automatic steps performed during an update

After importing a spectrum to the Live Viewer as described in Section 3.2.1.2, a number of automatic operations are performed following these five successive steps:

- **Step 0 - Only if an update has already been done: checking whether a new update is necessary**

All file names in the data folder are compared to those present in the data folder at the previous automatic update. If the content of the data folder is identical between the current and previous automatic updates, the current update does not proceed to the next step and stops right away.

- **Step 1 : Identifying in the data folder all file names similar in nature to the imported file**

At the very first automatic update, each file in the data folder with a specific file name (if multiple file iterations exist for a given file name, only the first iteration is used) is open to test whether its corresponding energy array has the same  $E_{\min}$  (with a  $\pm 2$  eV difference tolerated),  $E_{\max}$  (with a  $\pm 2$  eV difference tolerated), and total number of data points than those in the energy array of the imported file. For example, if the data folder has ten files having a specific file name, the Live Viewer window will display this message when the first file is checked:

*'Updating data, please wait (1/10 sample files tested)'*

For any automatic update other than the very first update, only the new files with specific file name that were not present in the data folder at the previous update are tested. Therefore, the first automatic update may take longer time to complete than the following ones.

- **Step 2 : Identifying in the data folder all scan iterations associated to the current highlighted sample that are similar in nature to the imported file**

At the very first automatic update after selecting the current highlighted sample (more on highlighted sample selection in next Section 3.2.3), each scan iteration in the data folder associated to the current highlighted sample is open to check whether its corresponding energy array has the same  $E_{\min}$  (with a  $\pm 1$  eV difference tolerated),  $E_{\max}$  (with a  $\pm 1$  eV difference tolerated), and total number of data points than those in the energy array of the imported file. For example, if the current highlighted sample has ten associated scan iterations saved in the data folder, the Live Viewer window will display this message when the first file is checked:

*'Updating data, please wait (1/10 scan iterations tested)'*

For any automatic update other than the first update after selection of the current highlighted sample, only the new iterations that were not present in the data folder at the previous update are tested. Therefore, the first automatic update after selection of the current highlighted sample may take longer time to complete than the following ones.

- **Step 3 : Processing of all scan iterations associated to the current highlighted sample**

The iterations associated to the highlighted sample identified in the previous step are processed following these successive operations:

-In the energy array of each iteration, the energy must always increase in value, otherwise the data point is deleted.

-Data of all iterations, except the first one, are interpolated (linear interpolation) based on the energy array of the first iteration.

- **Before processing the data using Larch functions, if more than 20 iterations were identified at Step 2, they are evenly split into groups so that the total number of groups does not exceed 20, and the iterations within each group are merged together.** This limits the total number of spectra to process with Larch functions, as the latter represents the most time-limiting step of the entire update process. The grouping thus guarantees a rapid automatic update regardless of the number of iterations. For example, if 56 iterations were found, 18 groups are created; each group consists in three iterations merged together. Since  $18 \times 3 = 54$ , the last two iterations won't be used in the final steps below.

-All iterations are merged, and the average is processed using Larch functions. If these operations are done at the first update after the current highlighted sample was defined, the scan parameters obtained from these operations (e.g.  $e_0$ , pre-edge function and post-edge functions parameters) are used as input parameters in the Larch functions employed to process each iteration/group below, unless some of these parameters have been manually constrained by the user (more on default parameter modifications in Section 3.3.2).

-Each iteration (if  $< 20$  iterations), or group (if  $> 20$  iterations) is processed using Larch functions with the same input parameters= normalization, background subtraction, Fourier Transform, and random noise estimation. The signal-to-noise ratio of the spectrum is also assessed (more on this estimation in Section 3.3.5).

For example, if 54 iterations of the highlighted sample were found in the data folder, the Live Viewer window will display this message when the first iteration is processed:

*'Updating data, please wait (1/54 operations)...'*

Also, in this example, the processing will slightly slow down every three operations since this corresponds to the creation of a complete group, which is then processed with Larch Functions.

- **Step 4 : Creation of 3D plots relative to the current highlighted sample**

3D plots are created using the data relative to all iterations (if  $< 20$  iterations) or groups (if  $> 20$  iterations). This is not instantaneous as it implies interpolating the data to create surfaces used for 3D rendering. The Live Viewer window will display this message:

*'Last step: creating 3D plots...'*

### 3.2.3 Selection of the highlighted sample

Right after importing a file (Section 3.2.1.2) into Live Viewer, the sample highlighted in the sample list of the Live Viewer window corresponds to the name of the imported file. The XAFS data displayed in the Live Viewer window then corresponds to the data relative to this file and all its associated scan iteration files found in the data folder.

The highlighted sample can be set to a new one manually or automatically:

- **Manuel selection of the highlighted sample:** a new highlighted sample can be set by interactively clicking, in the sample list featured on the left side of the Live Viewer Window (Figure 50), on the new sample name to highlight. Since this operation changes the current highlighted sample, the Live Viewer window is automatically updated following Steps 2, 3, and 4 described in the previous section so that the data relative to the new highlighted sample can be shown in the window.

- **Automatic selection of the highlighted sample:** the highlighted sample can be automatically changed in the sample list of the Live Viewer window during an update if two conditions are met:
  - 1) A new file name was found in the data folder during Step 1 (see previous Section) of an update. This file name was not present in the data folder at the previous update.
  - 2) The option “Always display a new sample” is checked in the Live Viewer menu (this option is checked by default):

Live Viewer menu > Sample options > “Always display a new sample”

If these two conditions are met, the new sample name found in the data folder during Step 1 of the update will then appear in the sample list of the Live Viewer window, and will become the current highlighted sample. After Step 1, the update will proceed to Steps 2, 3 and 4 since a new highlighted sample has been defined. At the end of this update process, the data relative to the new highlighted sample is shown in the Live Viewer window.

To avoid this possible automatic modification of the highlighted sample, one can select in the menu:

Live Viewer menu > Sample options > “Always stay on the current sample”

In that case, if a new sample name is found in the data folder during Step 1 of the update process, only the sample list of the Live Viewer is updated, i.e. the new sample name is added to the sample list. The update stops at the end of Step 1 and does not proceed to Steps 2, 3, and 4 because the highlighted sample has not been changed.

## 3.3 Functionalities

### 3.3.1 Constraining the E0 value

When importing a file to the Live Viewer, the e0 value can be set via two methods: it can be automatically determined using the Larch function “find\_e0” (i.e. automatic method) or constrained based on a user-defined theoretical value that is set before data importation (i.e. manual method).

- The automatic method (Larch method) is selected by default in the Live Viewer menu bar:

Live Viewer menu > Sample Options > E0 Determination > “Automatic”

- To constrain e0 to a user-defined value, set the “E0 Determination” method to “Manual”:

Live Viewer menu > Options > E0 Determination > “Manual”

If this menu option is selected via the Live Viewer menu bar, a window appears where the theoretical E0 value can be specified. This must be done before importing a file to the Live Viewer module. If any spectrum is imported to Live Viewer after constraining e0, the e0 will be set to the closest value, among all values available in the raw energy array of the sample, to the user-defined E0 value.

### 3.3.2 “Edit Mode”: modifying default parameters

#### 3.3.2.1 Principle

During the first update after the current highlighted sample has been defined, the average spectrum of all scan iterations of the highlighted sample is processed using Larch functions (normalization, autobk background-subtraction, and Fourier Transform functions). The parameters obtained from these operations (e.g.  $E_0$ , pre-edge function and post-edge functions parameters) are displayed in a box in the middle of the Live Viewer window (Figure 51). These parameters are used as input parameters to process each iteration/group of the highlighted sample. Therefore, all XAFS spectra shown in the Live Viewer window can be conveniently compared to each other as they were processed using the same input parameters (e.g.  $E_0$  value, plot k weight, spline range). Although the default values of these parameters are automatically obtained during the first update, they can be modified later on by the user, by directly modifying their values in the central box of the Live Viewer window. This central box of the Live Viewer also shows the data (raw  $\mu$ , normalized  $\mu$ ,  $\chi$ , Fourier Transform, or first-derivative spectrum) relative to the average of all checked iterations of the current highlighted sample.

If any parameter is modified in the central box of the Live Viewer window, the box switches to an Edit mode: all parameters in the box are colored with a blue color (Figure 53) and the countdown to the next automatic update of the Live Viewer window is paused. No automatic update can thus occur when the parameters are being modified in “Edit mode”.

**As long as the central box is in “Edit mode”, all data relative to the current highlighted sample, displayed on the right side of the Live Viewer window (Figure 51), will not be reprocessed using the new parameters. The new parameters are only applied to the average spectrum displayed in the box.** This is to avoid a complete window update every time the value of a single parameter is modified.

If the new parameters are satisfactory, based on their effects on the average spectrum displayed in the parameter box in “Edit mode”, press on the button “Apply to all (current sample)” at the bottom of the box (Figure 53), to switch back to normal mode. This initiates a complete window update: all data relative to the highlighted sample is processed using the new parameters. The countdown to the next automatic update resumes.



Figure 53 - Parameter box of the Live Viewer, in “Edit mode”

### 3.3.2.2 Apply all parameters to sample list

If the current highlighted sample has been manually or automatically changed (Section 3.2.3), the Live Viewer window undergoes a complete update (Section 3.2.2). The parameters used for the previous highlighted sample can be kept and used to process the data of the new highlighted sample, by clicking on the option “Apply to all sample in list” at the bottom of the parameter box (Figure 53), before the change in highlighted sample. Otherwise, new parameters will be automatically obtained during the first update relative to the new highlighted sample.

### 3.3.3 Plotting options

#### 3.3.3.1 Four ways to display data relative to the current highlighted sample

The right side of the Live Viewer window features three plots where the normalized, chi, and Fourier Transform spectra relative to the current highlighted sample are displayed, respectively (Figure 50). The contents of these plots are defined using four plotting options that are available on top of the Live Viewer Window:

**- Last two merges (2D only) / Default option**

Two spectra are displayed per plot: the first one, shown in blue color, corresponds to the average of all checked scan iterations in the scan iteration list of the Live Viewer window. The second one, shown in red color, corresponds to the average of all checked scan iterations except the last checked iteration. This enables to visually assess from the plot whether collecting the last iteration has improved the quality of the merge. If not, this could suggest that collecting the last iteration for this sample was unnecessary, thus the acquisition for this sample may be stopped at this point.

**- All merges (2D or 3D possible)**

All merges relative to the current highlighted sample are displayed per plot. For example, if the scan iteration list features 5 iterations and all of them are checked, there will be 4 average spectra displayed: merge of the first two iterations, first three iterations, first four iterations, and all five iterations. Optionally, these spectra can be shown in 3D.

**- Last scans/groups (2D only)**

The last checked scan iteration (if less than 20 iterations are checked) or group (if more than 20 iterations are checked) relative to the current highlighted sample is displayed in each plot.

**- All scans/groups (2D or 3D possible)**

All checked scan iterations (if less than 20 iterations are checked) or groups (if more than 20 iterations are checked) are displayed in each plot. Optionally, these spectra can be shown in 3D.

#### 3.3.3.2 Plot boundaries

The boundaries of the plots featured on the right side of the Live Viewer window can be modified at the top part of the window (Figure 50). These are the “E limits” to modify the  $E_{\min}$  or  $E_{\max}$  of the normalized spectra plot, “k limits” to modify the  $k_{\min}$  or  $k_{\max}$  of the chi spectra plot, and “R limits” to modify the  $R_{\min}$  or  $R_{\max}$  of the Fourier Transform spectra plot.

#### 3.3.3.3 Displaying data in 2D or 3D

When the plotting option “all merges” or “all scans/groups” (described in Section 3.3.3.1) is selected, the data is displayed by default in 2D on the right side of the Live Viewer window. This is because this option is checked by default in the menu:

Live Viewer menu > Plot options > “2D plots”

If the plotting option “all merges” or “all scans/groups” is selected, a menu option enables to display the data in 3D:

Live Viewer menu > Plot options > “3D plots”

**Note that if the plotting option “last two merges” or “last scan/group” (described in Section 3.3.3.1) is selected, the data will be always displayed in 2D even if the “3D plots” option is selected in the menu.**

#### 3.3.3.4 3D plot rotation

If the plotting option “all merges” or “all scans/groups” is selected, and the “3D plots” is selected in the menu, the data is plotted in 3D in the Live Viewer window. The three plots can be individually rotated. By default, the 3D rotation is active on the normalized mu plot as this option is checked in the menu:

Live Viewer menu > Plot options > “3D rotation to E plot”

To specifically rotate in 3D the chi plot, this option must be firstly selected in the menu:

Live Viewer menu > Plot options > “3D rotation to k plot”

Lastly, to specifically rotate in 3D the Fourier Transform plot, this option must be selected in the menu:

Live Viewer menu > Plot options > “3D rotation to R plot”

#### 3.3.3.5 Shifts to scatter multiple spectra

##### 3.3.3.5.1 Y shifts: for 2D plot

If multiple spectra are shown in 2D in the Live Viewer window, they can be vertically scatter by applying a shift to each spectrum. A specific value can be provided for the normalized mu, chi, or Fourier Transform plot via:

Live Viewer menu > Plot options > “E plot Y shift (2D)”

Live Viewer menu > Plot options > “k plot Y shift (2D)”

Live Viewer menu > Plot options > “R plot Y shift (2D)”

##### 3.3.3.5.2 Z shifts for 3D plot

If multiple spectra are shown in 3D in the Live Viewer window, the depth of the 3D figure can be extended by applying a shift to each spectrum. A specific value can be provided for the normalized mu, chi, or Fourier Transform plot via:

Live Viewer menu > Plot options > “E plot Z shift (3D)”

Live Viewer menu > Plot options > “k plot Z shift (3D)”

Live Viewer menu > Plot options > “R plot Z shift (3D)”

#### 3.3.3.6 Displaying first derivative

By default, the normalized mu spectra relative to the current highlighted sample are shown on the right side of the Live Viewing window (Figure 50). This is because the following option is checked in the menu:

Live Viewer menu > Plot options > “Show normalized mu”

Instead of displaying the normalized mu spectra, the first derivative of mu spectra relative to the current highlighted sample can be displayed by selecting this option in the menu:

Live Viewer menu > Plot options > “Show mu first-derivative”

Displaying the first derivative can notably help determine that all scan iterations are relatively aligned to each other in energy.

### 3.3.3.7 Plot color options

By default, the newest and oldest spectra featured in all plots of the Live Viewer are displayed in blue and red color, respectively. This is following option is indeed checked by default in the menu:

Live Viewer menu > Plot options > “Color plots”

Alternatively, the newest and oldest spectra featured in all plots of the Live Viewer can be displayed in black and gray color, respectively, via this option in the menu:

Live Viewer menu > Plot options > “Black & White plots”

### 3.3.4 Viewing sample of reference data relative to a single iteration

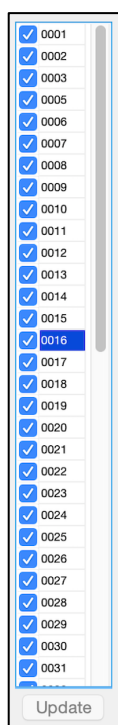
Since version 1.0.3 of Fastosh, one can display either the XAFS spectrum corresponding to a single scan iteration whose number is listed in the Live Viewer window, or the aligned sample reference along with the calibrated reference. The first option is selected by default in the menu:

Live Viewer menu > Sample options > “Clicking on a iteration to display the... > ... sample”

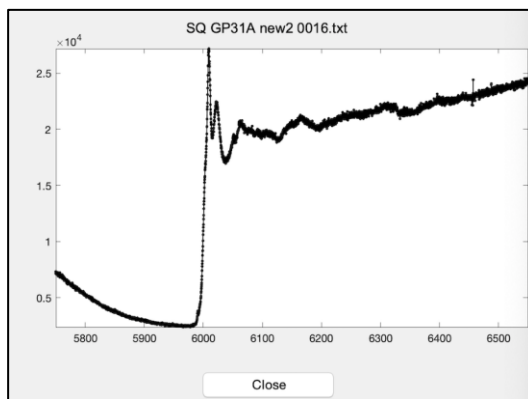
The second option can be selected from the menu. Note that this fonctionnality is activated only when the reference is imported along with the data:

Live Viewer menu > Sample options > “Clicking on a iteration to display the... > ... aligned sample reference and calibrated reference”

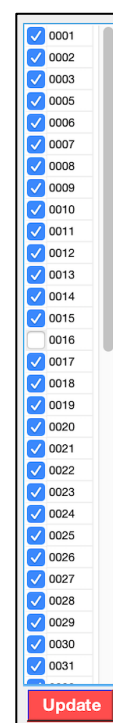
Each of these fonctionnalities are discussed in the next two sections.



a)



b)



c)

Figure 54 - a) Example of an iteration list; b) secondary window displaying the mu spectrum of the selected iteration c) After unselecting an iteration, the button “Update” at the bottom of the window must be pressed to refresh the entire window

### 3.3.4.1 Viewing sample data of a single iteration and unselecting it

The raw sample data of each iteration listed in the iteration list of the Live Viewer window (Figure 51) can be accessed by simply double-clicking on the specific iteration in the list. For example, in Figure 54a, a double-click was done on iteration “0016” to check the raw data of this scan. This opens up a secondary window displaying the mu spectrum corresponding to this iteration (Figure 54b). Simply press the upper or lower key on the keyboard to navigate through the iteration list to automatically refreshes the secondary window and display the raw data of the iteration currently selected. For example, the raw data corresponding to iteration 0016 in Figure 54b is problematic since the XAFS spectrum is distorted. This iteration is thus unselected from the iteration list (Figure 54c). After all problematic iterations have been unselected from the list, the button “Update” at the bottom of the window (Figure 54c) must be pressed to refresh the entire window, using only the data of the remaining checked iterations.

### 3.3.4.2 Viewing the aligned reference spectrum of current iteration

The aligned reference associated to each iteration listed in the iteration list of the Live Viewer window can be viewed by simply double-clicking on the specific iteration in the list. Firstly, this option must be selected from the menu, as mentioned in Section 3.3.4. A window appears (Figure 55), where the aligned reference of the current iteration is displayed in orange color. The calibrated reference is displayed too, in blue color. The energy shift required to align the reference of the current iteration to the calibrated reference is shown on top of the figure, in e.V.

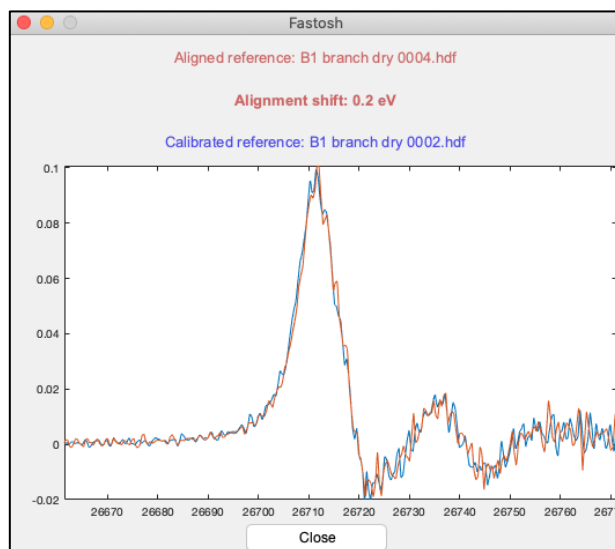


Figure 55 - Secondary window displaying the aligned reference of the current iteration and calibrated reference

### 3.3.5 Displaying signal-to-noise ratio and random noise estimation

The right edge of the Live Viewer window features three buttons: S/N,  $\epsilon_k$ ,  $\epsilon_R$  (Figure 50). Pressing any of them opens up a secondary window where the signal-to-noise ratio (S/N) and estimated random noise  $\epsilon_k$  and  $\epsilon_R$  are plotted as a function of total number of scan iterations per average associated to the current highlighted sample (Figure 56). **This secondary window is automatically refreshed at every Live Viewer update.** Therefore, if used during an acquisition,

this secondary window can be left open on the computer screen next to the Live Viewer window, to visually follow the progressive improvement in spectral quality.

The S/N,  $\mathcal{E}_k$ ,  $\mathcal{E}_R$  are calculated for all averages associated to the current highlighted sample. For instance, if four scan iterations were collected for the highlighted sample, there are four associated averages: the first iteration itself (no merge in this case), the first two iterations, the first three iterations, and all four iterations merged together.

While the S/N value is calculated by a Fastosh function and provided as a supplementary information on noise estimation, the values for  $\mathcal{E}_k$  and  $\mathcal{E}_R$  are calculated by Larch functions. As mentioned below,  $\mathcal{E}_k$  is directly proportional to  $\mathcal{E}_R$  and thus does not provide any significant additional information on data noise. It is reported along with S/N and  $\mathcal{E}_R$  in Live Viewer only for sake of completeness.

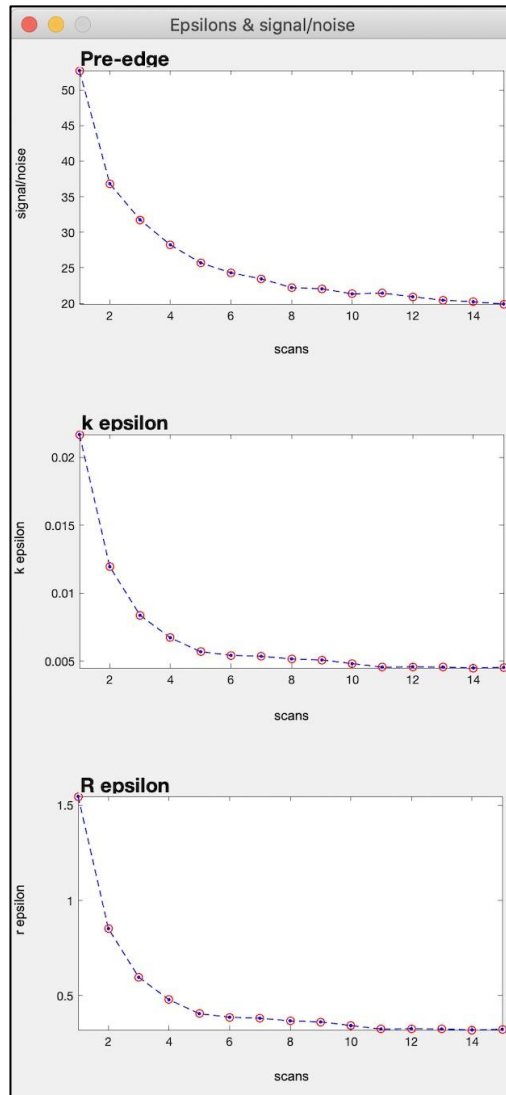


Figure 56 - S/N,  $\mathcal{E}_k$ ,  $\mathcal{E}_R$  as a function of number of scan iterations averaged together, relative to the current highlighted sample

- Signal-to-noise ratio (S/N)

The signal-to-noise ratio is estimated in the first part of the mu spectrum, from the beginning of the mu spectrum to 100, 150, 200, and 250 eV below the  $e_0$  energy value, if the latter is <10000, <20000, <30000, and >30000 eV, respectively. This part of the mu spectrum should be flat (noise aside) as it occurs before the rise in edge-jump. It is fitted using a polynomial function of degree 2. The fit is used to represent the ideal, perfect signal devoid of noise. The estimated signal-to-noise

ratio is equal to the average of the absolute difference between the experimental spectrum and the ideal, noise-free signal measured at each point in the early part of the mu spectrum.

- R epsilon ( $\epsilon_R$ ): estimated random noise in  $\chi(R)$

The noise in  $\chi(R)$  is estimated by calculating the root mean square average of  $\chi(R)$  between 15 and 30 Å. It is assumed that this part of the  $\chi(R)$  contains no meaningful structural information and its signal only corresponds to “white noise” (noise independent of R). Since these assumptions could be challenged,  $\epsilon_R$  may be inaccurate and underestimate the actual noise level in  $\chi(R)$ .

- K epsilon ( $\epsilon_k$ ): estimated random noise in  $\chi(k)$

The noise in  $\chi(R)$  is converted into noise in  $\chi(k)$  using the Parseval’s theorem:

$$\epsilon_k = \epsilon_R * \sqrt{\frac{(2 * \pi * (2 * kweight + 1))}{0.05 * (kmax^{(2*kweight+1)} - kmin^{(2*kweight+1)})}}$$

Given this equation,  $\epsilon_k$  is directly proportional to  $\epsilon_R$ , thus their variations should mirror each other, as shown in Figure 56.

### 3.3.6 Transferring merge to the Main GUI

The mu spectrum corresponding to the average of all checked iterations of the current highlighted sample can be transferred to the Main GUI by pressing the button “Transfer average to main GUI” at the bottom left of the Live Viewer window (Figure 50). The transfer is instantaneous if only data relative to the sample has been imported to the Live Viewer. If the reference data has been also imported to the module, the transfer takes a few moments to complete because all data relative to the reference is processed at this point.

Before transferring a merge spectrum to the Main GUI, it is important to make sure that the energy values of all iterations are well calibrated. If not sure about that, open both sample and reference data of each scan iteration in the Main GUI and adjust the energy calibration/alignment of all scan iterations using the Calibration and Alignment module of Fastosh (Section 2.4.3). Merge the sample data only when the energy calibration of the reference is consistent in all scan iterations.

## 4 Preliminary data treatment prior to PCA, TT, MCR-ALS, or LCF

Principal Component Analysis (PCA)/Target Transformation (TT), Multi Curve Regression-Alternate Least Square (MCR-ALS) fitting, and Linear Combination Fitting (LCF) are described in Sections 5, 6, & 7 of this manual, respectively. All these data treatment methods involve the simultaneous processing of multiple spectra, grouped as a Data Matrix (PCA, TT, MCR-ALS) or a set of experimental and reference spectra (LCF). The grouped spectra are normalized  $\mu$ , 1<sup>st</sup> derivative of normalized  $\mu$ , or  $\chi$  (EXAFS). A prerequisite to process these spectra by MCA/TT, MCR-ALS, or LCF is that all spectra must be firstly imported to the Main GUI and consistently normalized, and also background-subtracted if there are EXAFS:

- **Consistent normalization:**
  - the baseline in the pre-edge region before the edge-jump of all  $\mu$  spectra should be equal to 0. For example, a set of  $\mu$  spectra with inconsistent and consistent normalization in the pre-edge region is shown in Figure 57A and B, respectively.
  - the end of all normalized  $\mu$  spectra should be equal to 1, or oscillate around 1. Even if only the XANES region is studied, the raw XAFS spectrum collected at the beamline should extend far enough beyond the XANES region to conveniently & unambiguously normalize the post-edge region of the spectra.
- **Consistent background subtraction (EXAFS spectra):**

If the redox states of the element of interest are not too different in the samples corresponding to the EXAFS spectra, it is recommended to use the same E0 value to extract all EXAFS spectra. In some cases, it is not possible to apply the same E0 to all raw  $\mu$  spectra as the latter may not share the same energy array (e.g.  $\mu$  spectra collected with different scan parameters). If it is not possible to apply the same E0 to all spectra, the raw  $\mu$  data can be firstly interpolated to a common energy array using the Interpolation module of Fastosh (Section 2.4.6). Also, the spline parameters, especially spline  $k_{\max}$  &  $k_{\text{weight}}$ , should be the same to extract all EXAFS spectra.

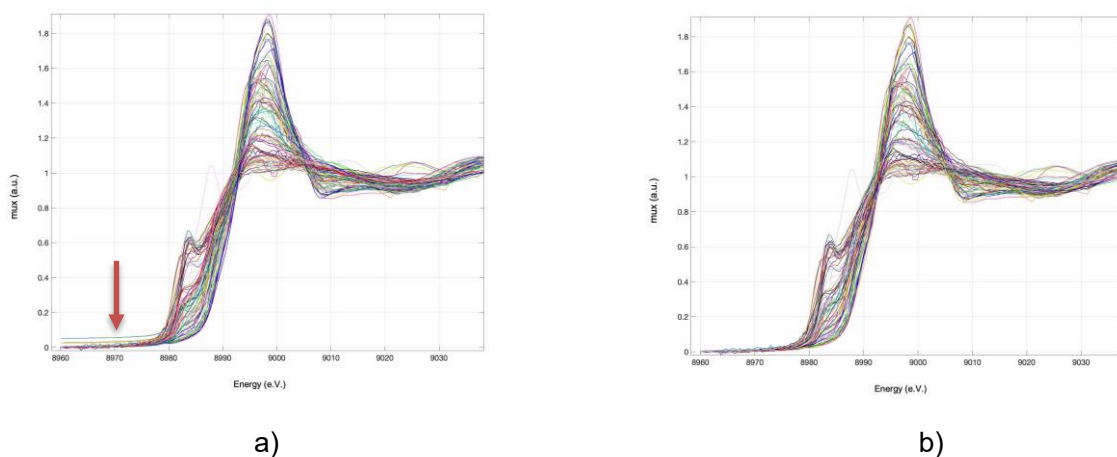


Figure 57 - normalization of 99  $\mu$  spectra belonging to a Data Matrix to process by PCA: a) inconsistent normalization and b) consistent normalization in the pre-edge region between 8960 and 8975 eV (data from Manceau *et al.* 2014).

## 5 PCA & Target Transformation (TT)

### 5.1 Usefulness

The principles of Principal Component Analysis (PCA) and Target Transformation are specifically detailed below (Sections 5.4.1 & 5.5.1). These methods are applied to a data set that is essentially a group of either normalized  $\mu$  or EXAFS experimental spectra (i.e. the "Data Matrix"). It is assumed that the main chemical forms of the element of interest are the same in all samples corresponding to the spectra of the Data Matrix. These species thus represent the "principal components" of the sample mixture. Performing a PCA on the Data Matrix can be helpful in determining how many principal components are present in the sample mixture. Target transformation can be performed to identify the nature of each principal component: a spectrum that represents a specific chemical form of the element of interest (i.e. typically a reference spectrum among a library of standards) is target-transformed to determine whether this species is one of the principal components present in the sample mixture.

The relative quantities of the principal components in the sample mixture are sample-specific. They can be determined for any sample that is part of the Data Matrix using MCR-ALS (Section 6) or more traditionally using a Linear Combination Fitting approach (Section 7). In the latter case, PCA, Target Transformation, and even MCR-ALS should be performed before LCF, as they help determine the number and types of references to choose from the standard library to perform LCF.

### 5.2 Preparation of Data Matrix & references in Main GUI

The data processed by Principal Components Analysis (PCA) is essentially a group of normalized  $\mu$  or EXAFS experimental spectra. The number of spectra in the Data Matrix must be at least equal to the number of principal components: if they are equal, the spectra of the samples must correspond to the spectra of the pure components (in this case, there would be no need to identify the nature of the principal components). Before opening the PCA/TT module, the data must be consistently preprocessed in the Main GUI, as mentioned in Section 4.

After all data has been consistently pre-processed in the Main GUI (i.e. all spectra of the Data Matrix to process by PCA, and optionally all reference spectra to target transform after PCA), select:

Main GUI Menu > Chemometry > PCA/Target Transformation

All spectra of the Data Matrix must have similar energy range. If not, the shortest energy range is used as the one for all spectra of the Data Matrix in the PCA module. This means that all spectra of the Data Matrix, whose energy ranges are longer than the shortest one, are truncated so that all spectra have the same energy range. Additionally, all XANES/ $\mu$  spectra of the Data Matrix, except the first one, are interpolated based on the energy array of the first spectrum.

Similarly, all reference spectra to be Target Transformed after PCA should have an energy range similar to the one employed to process the Data Matrix by PCA. If some of the reference spectra have longer energy range than the PCA E range, they will be adequately truncated. If some of them have shorter energy range than the PCA E range, it will be impossible to perform the Target Transformation. Either exclude these reference spectra from the reference list to target transform, or redo the PCA by shortening the energy range of the Data Matrix where PCA is performed, as explained in the next section.

### 5.3 Module window at opening

Once the PCA/Target Transformation window is open, all spectra of the Data Matrix are plotted on the top right corner of the window (Figure 58)

The minimum and maximum values of the energy range of the Data Matrix to process by PCA are shown in the fields "PCA min and max", below the sample list (Figure 58).

Specify the E or k range of the data to process, type of SVD method (see next part of this manual), and then start the PCA by pressing the button "Do PCA" (Figure 58).

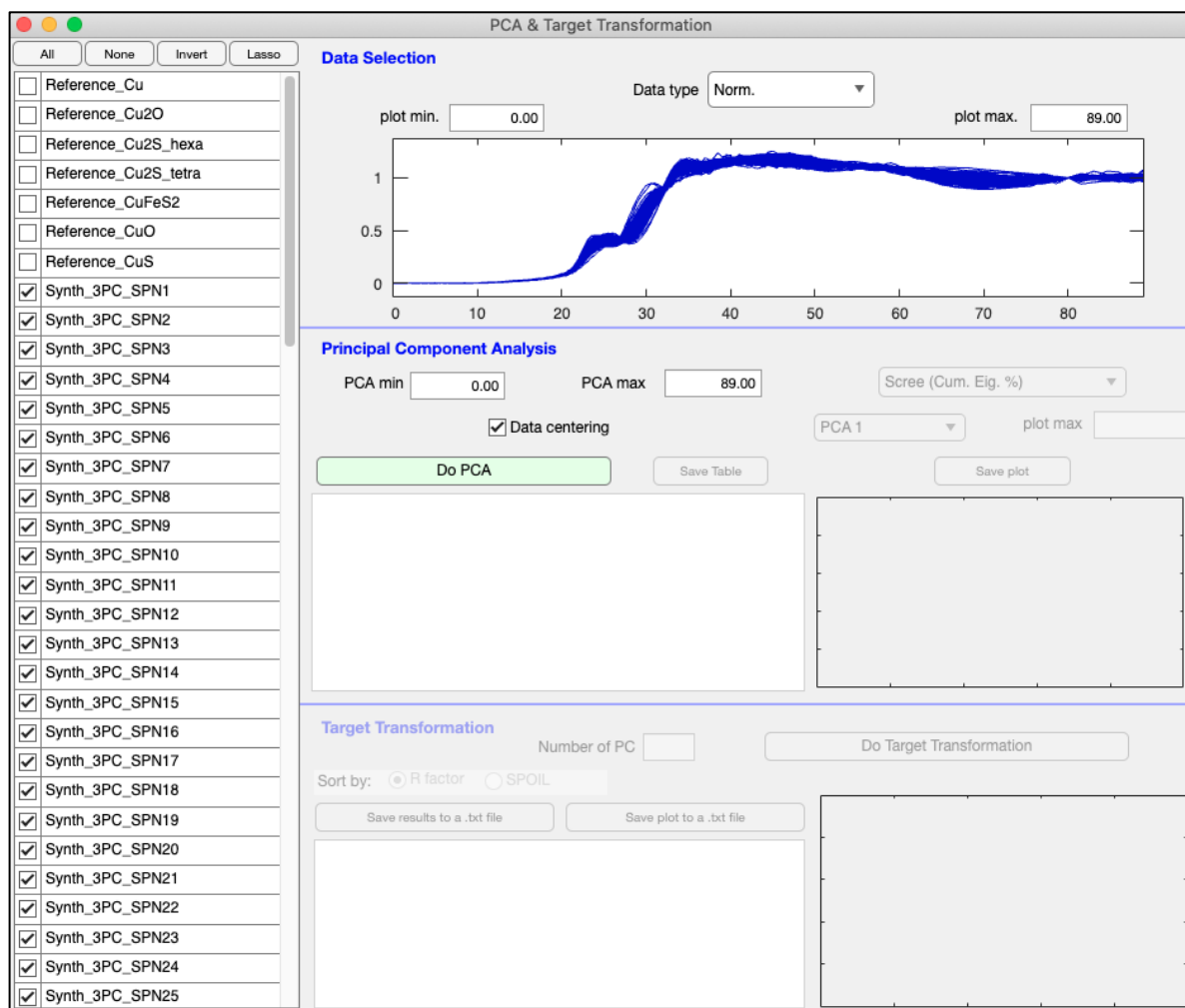


Figure 58 - PCA/Target Transformation module at opening: the EXAFS spectra of the Data Matrix selected in the Main GUI are plotted and their names are shown in the sample list

## 5.4 PCA

### 5.4.1 Principle

All spectra to process by PCA are grouped into a  $n \times m$  data matrix  $[D]$ , where  $n$  is the total number of data points per spectrum and  $m$  is the total number of spectra. At the beginning of PCA,  $[D]$  can be preprocessed to center the data. The specificities of these SVD methods are presented in the next section. The PCA decomposes the data matrix  $[D]$  into singular values following a matrix factorization (5):

$$(5) [D] = [U] \times [S] \times [V]^t$$

In Equation (5),  $[U]$ ,  $[S]$ , and  $[V]$  is a  $n \times n$ ,  $n \times m$ , and  $m \times m$  matrix, respectively. The [score] matrix, which is equal to  $[U] \times [S]$ , corresponds to the representation (coordinates) of the spectra in the new data space. The first column of [score] represents the first component of  $[D]$  that expresses the maximal variance in  $[D]$ , the 2<sup>nd</sup> column of [score] represents the 2<sup>nd</sup> component of  $[D]$  that expresses the residual variance orthogonal to the 1st component, the 3<sup>rd</sup> column of [score] represents the 3<sup>rd</sup> component of  $[D]$  that expresses the residual variance orthogonal to the 2<sup>nd</sup> component, and so on... These correspond to apparent, not true, fractions of the variance if XAFS data is not centered [13]. The total variance in  $[D]$  is equal to the sum of the eigenvalues, which correspond to the square of the singular values [7]. The latter are the elements of the diagonal of  $[S]$ . The eigenvalues are in descending importance along the diagonal of  $[S]$ , starting from the top left corner of  $[S]$ .

In some cases, one can determine, by summing one by one the eigenvalues from left to right along the diagonal of [S], what is the minimum number of eigenvalues required to reproduce most of the total variance. This number of eigenvalues corresponds to the number of principal components present in [D]. The remaining eigenvalues correspond to the secondary components present in [D], i.e. noise of the data. However, this approach based on the eigenvalues does not always enable to accurately determine the number of principal components in the mixture as demonstrated in the few examples below. Alternative approaches are then also proposed in the program and are described in the next section.

#### 5.4.2 SVD method: centering or no pre-processing

Prior to performing the Singular Value Decomposition (SVD), the dataset can be optionally centered. Centering [D] consists in subtracting the spectral mean (a.k.a. “observation mean”) of [D] from the dataset [D]. As a result, the spectra are vertically centered to zero (Figure 59).

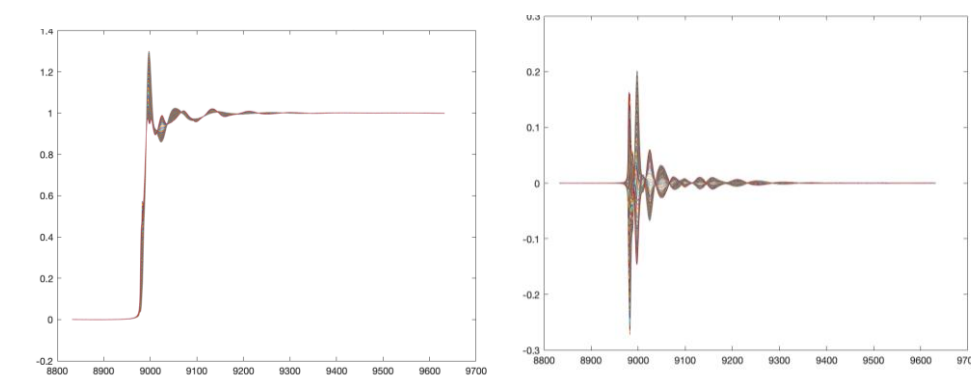


Figure 59 – A dataset [D] not centered (a) and the same dataset centered (i.e. [D] subtracted by the spectral mean of [D])

The first principal component of an uncentered dataset [D] is proportional to the spectral mean of [D]. Therefore, when the dataset [D] consists of normalized mu spectra/XANES spectra, the first eigenvalue is overwhelmingly high compared to the remaining eigenvalues as the mean edge-jump contributes a lot to the overall data variance, thus making the PCA results difficult to interpret. One way to “mask” the edge-jump contribution of the mean is then to center the dataset [D] prior to SVD. In contrast, when the dataset [D] consists of EXAFS spectra, the first principal component does not feature an edge-jump; hence its corresponding variance fraction does not overwhelm the remaining ones when no centering is performed to the dataset prior to SVD. This is why the option “centering” is checked by default in the PCA/module of Fastosh when the dataset consists in normalized mu spectra, and it is unchecked by default when the dataset consists in EXAFS.

**Because centering essentially consists in removing the first principal component of the dataset, one must add an additional principal component to the total number of principal components inferred from the PCA results to compensate for this removal.** For example, if the scree plot (see following section), obtained from PCA, which was performed on a centered dataset consisting in normalized mu spectra, suggests two principal components, they are three principal components in the original dataset.

#### 5.4.3 Interpreting PCA results

Once the Data Matrix has been processed by PCA, the number of principal components can be determined using different approaches available in Fastosh: eigenvalues, logarithmic difference of the eigenvalues, IND, and singular values. As demonstrated in the examples below, these approaches can provide contrasted results. The limitations of some of these approaches are discussed in the Manceau *et al.* 2014 paper, entitled “*Estimating the number of pure chemical components in a mixture by X-ray absorption spectroscopy*” [14].

Among the four approaches, the IND method represents the least reliable method since it is particularly dependent on level of data noise and number of components in the data matrix [14].

The result obtained with IND can be much more erroneous compared to those obtained with the other methods, as demonstrated in Section 5.4.3.2.2 below.

The singular value method has been proposed in some algorithms, such as the MCR-ALS Toolbox of Jaumot *et al.* [3], as mentioned in the MCR-ALS chapter of this manual at Section 6.4.2. As a matter of fact, this approach can provide similar or even more accurate results than those obtained with other methods, as demonstrated in Section 5.4.3.2.3 below.

The total numbers of meaningful components inferred with different approaches may be similar but not exactly identical to each other. Similarly, more than one possible number of meaningful components can be sometimes inferred per method. In that case, all the possible numbers of principal components should be considered/tested in the data analyses post-PCA, including target transformation, MCR-ALS, or LCF, to determine whether any of them makes the most sense in terms of final results and considering prior knowledge of the system.

#### 5.4.3.1 Eigenvalue methods

The eigenvalues are equal to the square of elements in the diagonal of the [S] matrix obtained by PCA [7]. Each eigenvalue corresponds to the apparent variance contribution, relative to the total variance, of each Data Matrix component identified by PCA. The relative variance contribution of the eigenvalues decreases along the diagonal of the [S]. The number of principal components present in the system may be then inferred by plotting the eigenvalues following the methods described in the next three sections.

##### 5.4.3.1.1 The cumulated relative eigenvalues (default plot)

The eigenvalues are firstly normalized so that their sum is equal to 100, and their cumulated values are plotted starting from the last eigenvalues. Plotting the cumulated eigenvalues makes the component relative contribution more obvious to see compared to when the eigenvalues are plotted without cumulating them (method presented in next section), regardless of normalization to 100 %.

##### 5.4.3.1.2 The eigenvalues listed in the result table of the PCA window

In the example shown in Figure 60, the first eigenvalue listed in the table, corresponding to the first component, is equal to 13.2, thus representing 71.8% of the total variance. The second, and third eigenvalues are 3.6, and 0.7, representing about 20 and 4 % of the total variance, respectively. Therefore, the first three components represent most (i.e. 95 %) of the total variance, which can be seen in the column “Cum. Var.” (cumulated variance) in the result table.

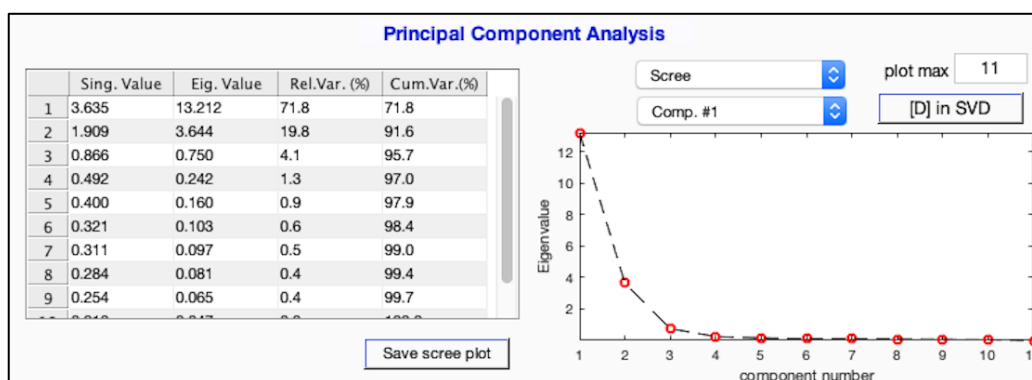


Figure 60 –PCA results after processing the Data Matrix previously shown

##### 5.4.3.1.3 Eigenvalue plotted against component number

Select “Scree (Eig.)” in the menu bar above the plot:

If the eigenvalue of each component is plotted against the component number in ascending order (“scree plot”), one can visually observe the relative contribution of each eigenvalue to the total variance. The scree plot typically features two distinct parts. The first one can be distinguished

by a sharp decrease in eigenvalues, and corresponds to the structural components in the Data Matrix. The second one is characterized by a lower and more linear decrease in eigenvalues compared to the first part, and mainly corresponds to the data noise in the Data Matrix. Hence, the number of principal components is equal to the number of data points featured in the first part of the scree plot. In the example shown in Figure 60, the first part corresponds to the first two data points of the scree plot. The third data point is not included in the first part since its corresponding contribution to the total variance is negligible (i.e. 2.5%) and it seems linearly in trend with the remaining data points based on a visual observation of the scree.

#### 5.4.3.1.4 Variance fraction plotted against component number

Select “Scree (Var. frac.)” in the menu bar above the plot:

Another type of scree plot commonly found in the literature consists in plotting the relative variance contribution of each component against the component number. The relative variance contribution of each component, in % of the total variance, is equal to each eigenvalue divided by the sum of all eigenvalues and multiplied by 100. The data trend in this type of scree plot is exactly the same as the one where eigenvalue is plotted against component number.

#### 5.4.3.1.5 Logarithmic first difference of the eigenvalues

Select “ $\Delta[\log(\text{eigenvalues})]$ ” in the menu bar above the plot:

A study demonstrated, using different data matrices consisting of micro-XANES spectra of real natural samples collected at a microprobe beamline, or artificial spectral mixtures of known principal components, that the scree approach employed to determine the number of principal components can provide inaccurate results [14]. It was also shown that plotting the logarithmic first difference of the eigenvalues ( $\Delta[\log(\text{eigenvalues})] = -[\log(\text{eigenvalue})_{j+1} - \log(\text{eigenvalue})_j]$  where  $j$  represents one specific position along the diagonal of  $[S]$ ) could represent a more accurate approach than the scree method to determine the number of principal components. In the  $\Delta[\log(\text{eigenvalues})]$  plot, the maximum observed in the last peak before all values reach a minimum close to zero indicates the number of principle components.

For example, Figure 61a shows the  $\Delta[\log(\text{eigenvalues})]$  plot using a data set from Manceau *et al.* 2014 [14]. The Data Matrix processed by PCA corresponds to an artificial spectral mixture of three principal components. A maximum is observed at component number 3 in the plot before all values reach a minimum close to zero. This suggests that there are three main components in the mixture, as it is truly the case. In contrast, the scree plot for the same data set erroneously suggests that there's only one principal components (Figure 61b).

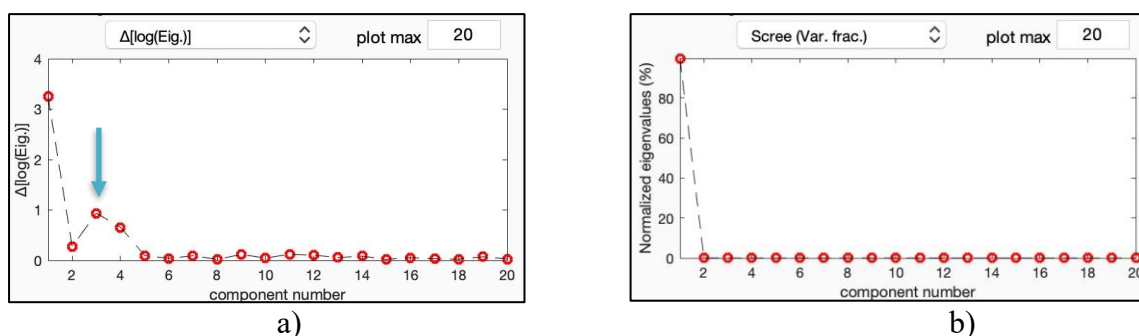


Figure 61 - Example of a)  $\Delta[\log(\text{eigenvalues})]$  plot using a data set from Manceau *et al.* 2014 [14]. The Data Matrix corresponds to an artificial XANES spectra mixture of three principal components; and b) scree plot for the same data set

#### 5.4.3.2 Other methods

##### 5.4.3.2.1 Eigenvalues, excluding the first one (XANES spectra only)

When a Data Matrix of XANES spectra is processed by PCA, the eigenvalue of the first component can be much higher than the rest of the eigenvalues, suggesting that there's only one principle components in the system. For example, this is the case with XANES data matrix of Manceau *et al.* 2014 whose corresponding scree plot is shown in Figure 61b [14]. This is due to

the fact that the edge-jump of all XANES spectra dominates the overall variance in the data set. The number of principal components can then be determined without considering the first eigenvalue by clicking on the box “Exclude 1<sup>st</sup> P.C.” below the result table (Figure 62). The cumulated variance fractions in the result table are then recalculated accordingly. Figure 62 shows the same scree plot reported in Figure 61b but with the 1<sup>st</sup> component excluded. This time, the eigenvalues and corresponding scree plot suggest that there are two principal components in the Data Matrix, which corresponds to an artificial XANES spectra mixture of three principal components. Therefore, removing the first eigenvalue does help elucidate the correct number of principal components for this data set.

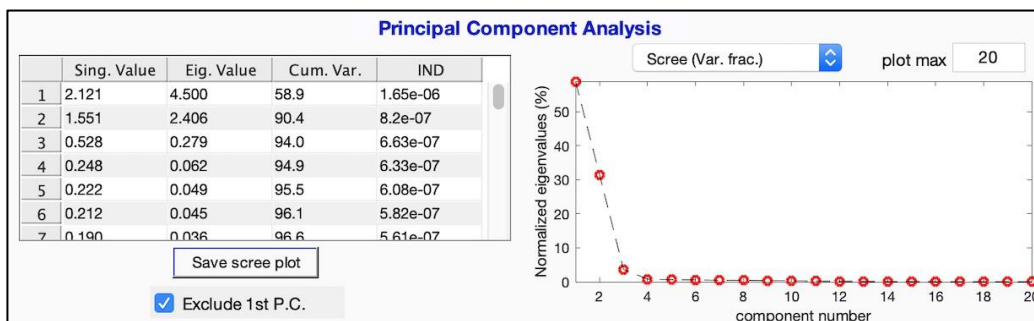


Figure 62 – PCA results for the same data used in Figure 61b, with the first component excluded

#### 5.4.3.2.2 IND

Select “IND” in the menu bar above the plot:

The indicator (IND) function was introduced in Malinowski (1977) [15] as an empirical function that can be employed to determine the number of principal components. For a specific component number  $N$  of a Data Matrix containing  $m$  spectra, the IND is equal to:

$$IND(N) = \frac{1}{(m - N)^2} \cdot \sqrt{\frac{\sum_{i=N+1}^m \lambda_i}{n(m - N)}}$$

Where  $\lambda_i$  corresponds to the eigenvalue at position number  $i$  in the diagonal of  $[S]$ ,  $m$  is the total number of spectra in the Data Matrix, and  $n$  is the total number of data points per spectrum. The IND is then successively calculated for all  $m$  components of the Data Matrix. The component number  $N$  providing the lowest IND value corresponds to the number of principal components in the sample mixture. The IND values reported in manuscripts can be normalized by the lowest value of IND, like in Manceau *et al.* (2014) [14]. Therefore, another plotting option enables to plot the normalized IND (select “norm. IND” in the menu bar). The data trends between IND and normalized IND plots are the same, thus the choice to report the results between these two options should not matter.

As demonstrated in Manceau *et al.* (2014) [14], the IND function is particularly dependent on the number of spectra and level of noise in the data. If the number of spectra is adequate (i.e. not too little, or large), the IND result can be consistent with those obtained with other approaches to determine the number of principal components. For example, Figure 63a shows the IND plot of the data set previously shown. The minimum of IND does suggest that there are two principal components, which is consistent with the scree plot (Figure 60). However, if there are too many samples in the Data Matrix, the estimation picks up too much noise, and the IND overestimates the total number of principal components. For instance, Figure 63b shows the IND plot of the 100 spectra Data Matrix from Manceau *et al.* (2014) [14] is also shown in Figure 61. The minimum of IND plot erroneously suggests that there are 28 number of principal components, although there are in reality only 3 in the artificial spectral mixture, as correctly suggested by the  $\Delta[\log(\text{eigenvalues})]$  plot (Figure 61a).

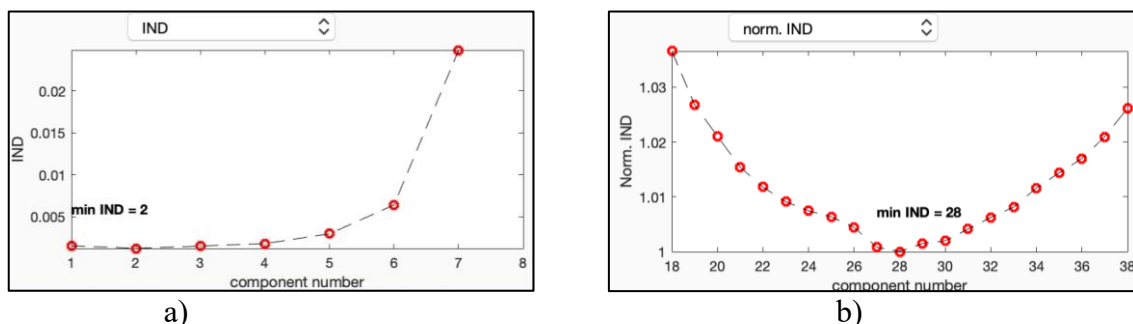


Figure 63 – a) IND plot of data set shown in Figure 60 ; and b) normalized IND plot of data set from Manceau *et al.* (2014) [14] shown in Figure 57

#### 5.4.3.2.3 Singular values

Select “Norm. Sing. Values” in the menu bar:

As previously mentioned, the eigenvalues are defined as the square of the elements in the diagonal of the [S] matrix obtained by PCA [7]. Some codes report eigenvalues following this definition, such as the LabView code presented in Manceau *et al.* (2014) [14]. In contrast, some other programs proposing PCA functionalities report eigenvalues using an alternative definition: the eigenvalues in these codes correspond to the elements in the diagonal of [S]. This corresponds to the definition of “singular values” [13]. As previously mentioned, the eigenvalues derived from XAFS data represent apparent fractions of the variance, not true fractions, since XAFS spectra in the Data Matrix are not centered [13]. One could then consider that a singular value also represents another apparent variance indicator as it is linked to eigenvalue by a square root relationship. As a matter of fact, plotting singular value against component number, and interpreting the plot like a scree to determine the number of principal components, can provide results more accurate than other approaches including the regular scree plot method previously described, as demonstrated in three examples below. Accordingly, the program proposes an option to plot the normalized singular value against the component number. The normalized singular value of each component, in %, is equal to each singular value listed in the result table divided by the sum of all singular values and multiplied by 100.

##### Example 1: Data shown in Figure 60 (Data matrix: EXAFS spectra)

In this first example, the data matrix corresponds to 8 EXAFS spectra. The scree plot corresponding to this data matrix, which was shown in Figure 60, indicated that there are two main components. The plot of singular value vs component number corresponding to this data is shown below (Figure 64). The plot indicates that the third component represents about 10% of the sum of all singular values. This could then suggest that 3 principal components are present in the mixture. Based on a prior knowledge of the system, having 3 PC would make more sense than 2 PC in this particular mixture.

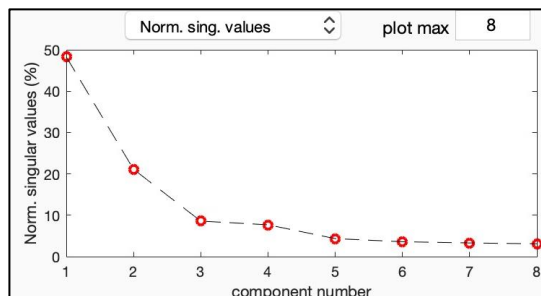


Figure 64 -Singular value plot corresponding to the data shown in Figure 45 & Figure 46

##### Example 2: Data shown in Figure 61, Figure 62, & Figure 63 (Data Matrix: 100 artificial XANES featuring 3 principal components, from Manceau *et al.* (2014) [14])

In this second example, it is known that 3 principal components are present in the sample mixture. However, the regular scree plot corresponding to this data

suggested the presence of only 2 components, when the 1st component was included (Figure 61b) or not (Figure 62). The plot of singular value vs component number corresponding to this data, when the 1st component is not considered to exclude the variance contribution of the edge-jump, is shown below (Figure 65). The third component only contributes to 6% of the sum of the singular values, however, this data point does not seem to be part of the obvious linear relationship shared by all data points starting from the fourth one to the last data point of the scree plot. Hence, this plot could suggest the presence of 3 main components in the system, as it is truly the case.

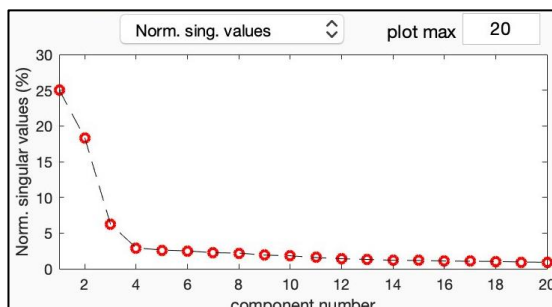
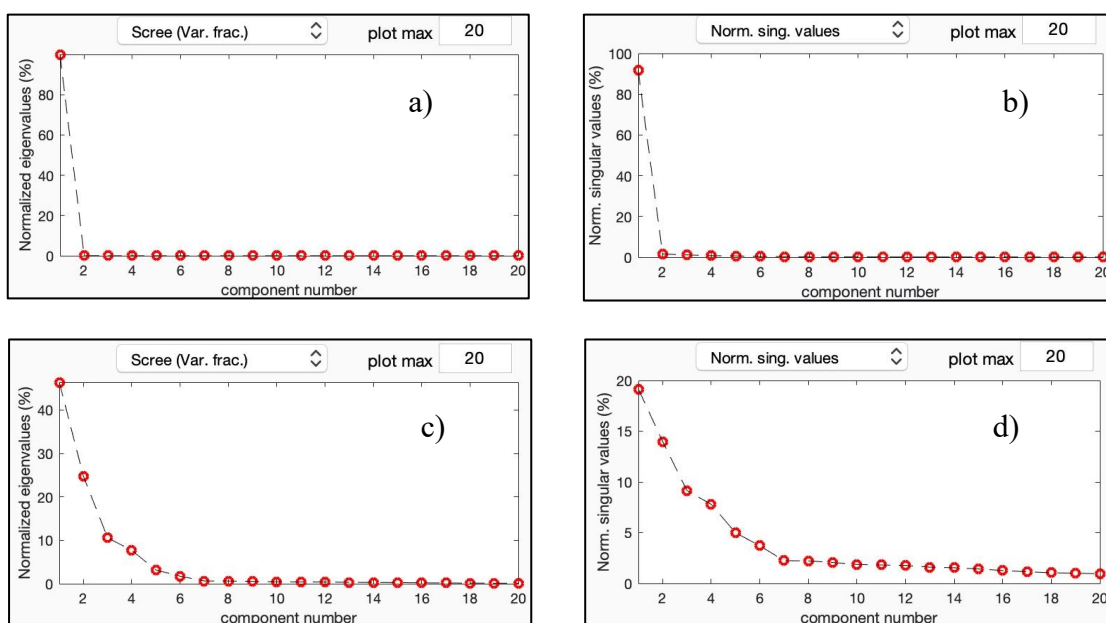


Figure 65 Singular value plot corresponding to the data shown in Figure 61, Figure 62, & Figure 63

### Example 3: Data Matrix: 100 artificial XANES featuring 7 principal components (from Manceau *et al.* (2014) [14])

In this third example, it is known that 7 principal components are present in the sample mixture. Both the regular scree plot (Figure 66a) and singular value plot (Figure 66b) suggest that the mixture only has one component as the variance is overwhelmed by the contribution of XANES edge-jump. When the first component is excluded, the regular scree plot suggests the presence of at least 4 principal components (Figure 66c), while the singular value plot suggests 6 or 7 (Figure 66d). The plot of logarithmic first difference of the eigenvalues (Figure 66c) features a maximum occurring at component number 7 before all values reach a minimum, suggesting 7 principal components in the mixture. Additionally, it was concluded in Manceau *et al.* (2014) [14] for this mixture that the number of principal components could be between 5 and 7 since another peak occurs at component number 5 in the  $\Delta[\log(\text{eigenvalues})]$  plot (Figure 66c). Lastly, the number of principal components suggested by the IND plot is very far from the truth since the minimum occurs at component number 29 (Figure 66d).



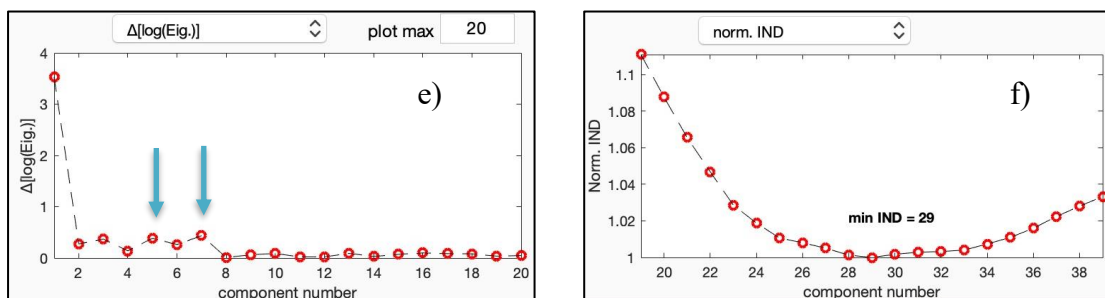


Figure 66 – Data matrix made of 100 artificial XANES featuring 7 principal components processed by PCA [14]: regular scree plot with (a) or without (c) the first component; singular value plot with (b) or without (d) the first component; plot of logarithmic first difference of the eigenvalues (e); plot of IND (f)

These examples suggest that the singular value plot (excluding the 1<sup>st</sup> component, if XANES spectra) and  $\Delta[\log(\text{eigenvalues})]$  plot are the most accurate methods among all proposed approaches to determine the number of principal components. The IND approach is the least accurate.

## 5.5 Target Transformation

### 5.5.1 Principle

- *First step: getting the R matrix*

The PCA results enabled to determine the number of principle components in  $[D]$ , which is referred as “r” in this paragraph. Based on the r value, a new expression of  $[D]$ , corresponding to its noise-filtered form,  $[D]_{\text{noise-filtered}}$ , is obtained:

$$(6) \quad [D] = [D]_{\text{noise-filtered}} = [U]_{\text{reduced}} \cdot [S]_{\text{reduced}} \cdot ([V]^t)_{\text{reduced}}$$

In (6),  $[D]_{\text{noise-filtered}}$  is a  $m \times n$  matrix,  $[U]_{\text{reduced}}$  is a  $m \times r$  matrix and essentially the  $[U]$  matrix cropped after its  $r^{\text{th}}$  column,  $[S]_{\text{reduced}}$  is a  $r \times r$  matrix and essentially the  $[S]$  matrix cropped after its  $r^{\text{th}}$  column and row, and  $([V]^t)_{\text{reduced}}$  is a  $r \times n$  matrix and essentially the  $[V]^t$  matrix cropped after its  $r^{\text{th}}$  row. In other words, only the elements of  $[U]$ ,  $[S]$ ,  $[V]$  corresponding to the principal components are kept in this new expression of  $[D]$  since they describe most of the data variance; the elements corresponding to the noise are removed. The expression in (6) is equivalent to:

$$(7) \quad [D] = [R] \cdot [C]$$

where  $[C] = ([V]^t)_{\text{reduced}}$ , and  $[R] = [U]_{\text{reduced}} \cdot [S]_{\text{reduced}}$ . The  $[R]$  matrix is used in the next step to calculate the SPOIL value.

- *Second step: calculating the SPOIL value*

The target transformation test consists in determining whether the real error associated to the target vector (RET), i.e. the reference spectrum being target-transformed, is proportional to the real error associated to the target vector projected in the space of the data matrix (REP) [7]. The REP is calculated using the eigenvalues of the data matrix components that do not describe the variance, i.e. all eigenvalues positioned after those associated to the principal components, starting from the left side along the diagonal of  $[S]$ , and a transformation vector, which is obtained using the target vector and  $[R]$  matrix. The RET is calculated using REP,  $[R]$  matrix, and target vector. Once REP and RET are known, the SPOIL value can be inferred since  $\text{SPOIL} = \text{RET}/\text{REP}$ . The higher the SPOIL value, the more the variance in the data matrix is degraded (“spoiled”) by the reference being tested, hence the less likely the target is part of the sample mixture.

### 5.5.2 Comparison with SPOIL values reported in Malinowski (1978)

The target transformation procedure was introduced in Malinowski (1978) [7]. Two data matrices constituted of mass spectroscopy spectra, from Ritter *et al.* (1976) [16], were employed in Malinowski (1978) [7] to demonstrate the usefulness of target transformation. The first and second data matrices represented a mixture of cyclohexane/hexane and cyclohexane/cyclohexene, respectively. When the reference spectrum corresponding to cyclohexane or hexane was target-transformed using the first data matrix, excellent SPOIL values were obtained (i.e.  $SPOIL < 1.5$ ), suggesting that the two tested species were part of the mixture. Similarly, when the reference spectrum corresponding to cyclohexane was target-transformed using the second data matrix, an excellent SPOIL value was obtained. Lastly, when the reference spectrum corresponding to hexane was target-transformed in the second mixture, a very poor SPOIL value was obtained (i.e.  $>> 6$ ), suggesting that hexane was not present in the mixture. Therefore, the SPOIL values systematically well predicted whether the tested species was a principal component of the data matrix.

Using the same mass spectroscopy data of Ritter (1976) [16], cyclohexane or hexane was target transformed in the cyclohexane/hexane or cyclohexane/cyclohexene data matrix using the code of Fastosh. A comparison between the Target Transformation results obtained with the Fastosh code and those reported in Malinowski (1978) is shown in the table below:

Mixture	Target	RET		REP		SPOIL	
		Malinowski	Fastosh	Malinowski	Fastosh	Malinowski	Fastosh
cyclohexane/hexane	cyclohexane	0.13	0.08	0.13	0.13	0.70	0.62
	hexane	0.09	0.07	0.14	0.14	0.56	0.52
cyclohexane/cyclohexene	cyclohexane	0.20	0.18	0.15	0.15	1.39	1.17
	hexane	3.68	3.54	0.05	0.05	77.76	69.88

The results between Malinowski (1978) [7] and Fastosh are similar to each other. Since all target transformation results depend on those obtained from PCA of the data matrix, the small result discrepancies could be due to a difference in singular value decomposition (svd) method between the one employed in Malinowski (1978) and the one used in Fastosh (a Matlab built-in function).

### 5.5.3 Procedure

#### 5.5.3.1 Selecting reference spectra to target-transform

To start the target transformation procedure, specify the number of principal components determined by PCA and the reference spectra to target transform in the PCA/TT window (purple fields in Figure 67). All selected spectra to target transform must have the same or higher energy range than the E range employed to process by PCA the data matrix. If not, unselect the short reference spectra or reprocess by PCA the data matrix with a shorter E range. When done, click OK.

#### 5.5.3.2 Results

The target transformation results are shown at the bottom of the window (Figure 68). The reference spectra that were target transformed and their SPOIL are listed with increasing SPOIL values. The plot shows the target vector and its predicted vector relative to the reference spectrum highlighted in the SPOIL result table.

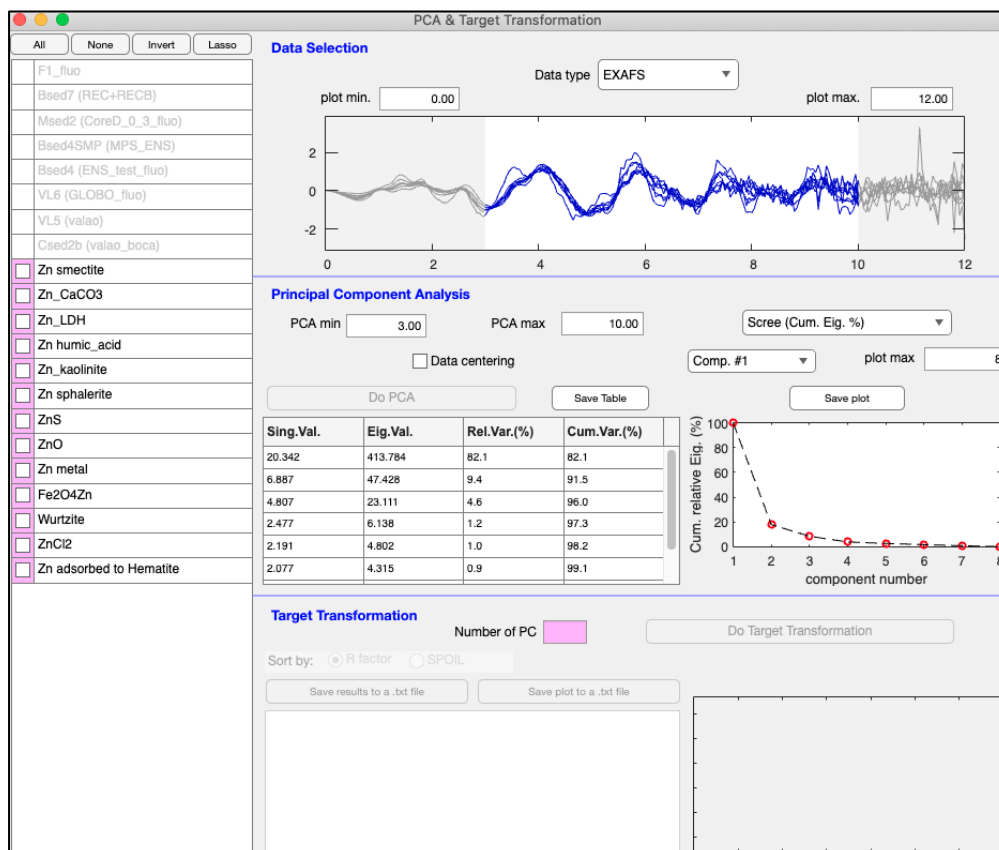


Figure 67 - Selection of the standards to target transform: at least one reference must be checked, and the number of PC must be specified



Figure 68 - Target Transformation results are shown at the bottom of the window

#### 5.5.3.2.1 SPOIL

According to Malinowski (1978) [7], the SPOIL values can be classified as:

0 to 1.5 - Excellent  
1.5 to 3. - Good  
3 to 4.5 - Fair  
4.5 to 6 - Poor  
> 6 - Not acceptable

Or :

0 to 3 - Acceptable  
3 to 6 - Moderately acceptable  
> 6 - Not acceptable

Therefore, some studies in the literature that applied the target transformation procedure on XAFS data considered that only the references providing excellent or good SPOIL values (i.e. SPOIL between 0 and 3) could possibly represent the principal components. Some other studies also considered the references providing fair SPOIL values (between 3 and 4.5) as possible principal components, since their SPOIL fall within the first part of the moderately acceptable range (3-6). Accordingly, the SPOIL value of the highlighted reference is reported in the window (Figure 68) with a specific color following a color code that reflects Malinowski's SPOIL classification:

SPOIL from 0 to 1.5: green to blue  
SPOIL from 1.5 to 3: blue to gray  
SPOIL from 3 to 4.5: gray to red  
SPOIL > 4.5: red

#### 5.5.3.2.2 Plot of target array vs projected vector

The plot at the bottom right corner of the window (Figure 68) shows the target and projected vectors relative to the reference highlighted in the result list. The target vector corresponds to the spectrum of the reference being target-transformed. The projected vector corresponds to the target vector projected in the space of the data matrix [7]. If the tested reference is indeed a component of the data matrix, the two spectra should be similar to each other. This can be visually assessed from the plot.

## 6 MCR-ALS

### 6.1 Usefulness

The Multivariate Curve Resolution – Alternating Least-Squares (MCR-ALS) is performed after Principal Component Analyses (PCA). PCA is firstly done to determine how many principal components are present in a sample mixture. The MCR-ALS approach then enables to determine the relative quantity of these principal components present in each sample of the sample mixture. Additionally, MCR-ALS allows to obtain the spectra corresponding to these principal components, but does not inform on the nature of these spectra. To help identify their nature, Fastosh proposes a post-MCR-ALS tool where the pure spectra extracted by MCR-ALS can be compared with those from a reference library.

### 6.2 Principle of MCR-ALS

The principal of PCA, and the three different PCA methods available in Fastosh, were described in Section 5.4.1. The Data Matrix is processed via a singular value decomposition approach to identify the number of principal components in the sample mixture. Once this number is known, a noise-filtered expression of the Data Matrix can be obtained:  $[D] = [D]_{\text{noise-filtered}} = [R].[C]$  (Section 5.5.1).

Then, MCR-ALS can be used to transform  $[R]$  and  $[C]$ , which are essentially two chemically meaningless matrixes obtained from PCA, into two chemically meaningful matrices, shown as  $[Pure\ Spectra]$  and  $[Pure\ Coefficients]$  in the expression below:

$$(8) [D] = [R].[C] = [Pure\ Spectra].[Pure\ Coefficients]$$

$[Pure\ Spectra]$  consists in all spectra corresponding to the principal components present in the sample mixture and  $[Pure\ Coefficients]$  corresponds to the relative quantity of these principal components needed to reconstruct each sample spectrum of the Data Matrix. The transformation of  $[R]$  and  $[C]$  into  $[Pure\ Spectra]$  and  $[Pure\ Coefficients]$  is done by MCR-ALS following these steps:

- Step 0: An initial estimation of  $[Pure\ Spectra]$  or  $[Pure\ Coefficients]$  is obtained using either the SIMPLISMA or EFA method (Section 6.4.3). In the following loop, the initial estimation was done on  $[Pure\ Spectra]$ , thus  $[Pure\ Coefficients]$  is obtained at step 1 below. If the initial estimation is done on  $[Pure\ Coefficients]$ , then  $[Pure\ Spectra]$  is obtained at step 1. Either way, both  $[Pure\ Coefficients]$  and  $[Pure\ Spectra]$  are alternatively refined in the loop:

#### ***MCR-ALS loop:***

- Step 1:  $[Pure\ Coefficients]$  is obtained:  
$$[D] / [Pure\ Spectra] = [Pure\ Coefficients]$$
- Step 2:  $[Pure\ Coefficients]$  is subjected to a set of user-defined constrains, detailed in Section 6.4.4, so that the solutions of this matrix can be physically meaningful.
- Step 3: A new expression of  $[Pure\ Spectra]$  is obtained:  
$$[D] / [Pure\ Coefficients] = [Pure\ Spectra]$$
- Step 4:  $[Pure\ Spectra]$  is subjected to a set of user-defined constrains, detailed in Section 6.4.4, so that the solutions of this matrix can be physically meaningful.
- Step 5: A fit residual is calculated to check whether convergence has been reached (Section 6.4.6.1.2). If so, the loop is ended and results are reported. Otherwise, the loop starts again a new iteration at Step 1.

The outcome of MCR-ALS is the two refined matrices [Pure Spectra] and [Pure Coefficients]. Each of them can be saved as a text file, as detailed in Section 6.4.6.2. Additionally, if the data are XANES or  $\mu$  (not EXAFS), all spectra of the principal components extracted by MCR-ALS can be transferred to the Main GUI (Section 6.4.6.2).

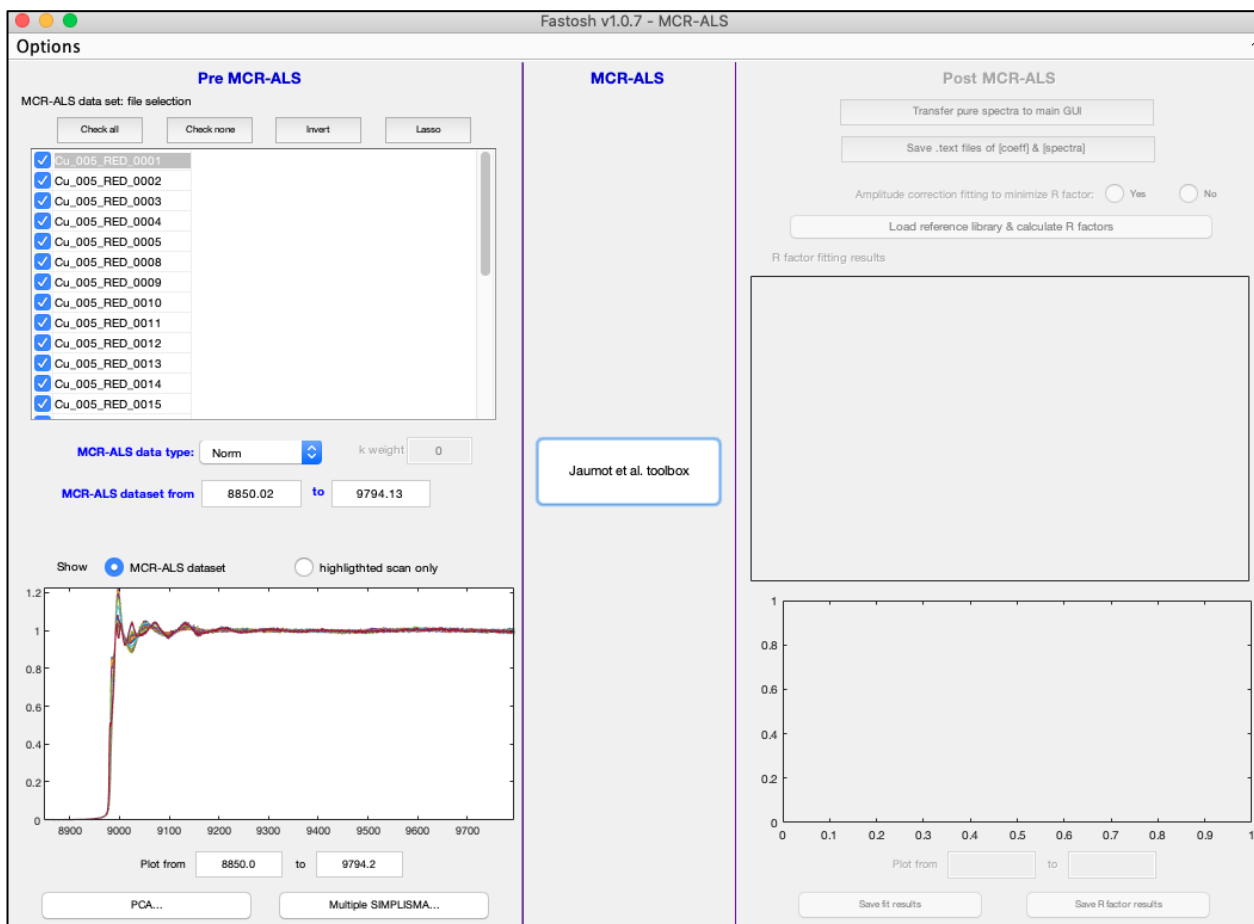


Figure 69- Pre MCR-ALS: selection of scans that are part of the Data Matrix

## 6.3 Pre MCR-ALS

### 6.3.1 Main steps

The data processed by MCR-ALS must be a group of normalized  $\mu$  or EXAFS experimental spectra, i.e. the “Data Matrix” mentioned in the previous paragraph (Section 6.1). Once MCR-ALS is completed, an option enables to compare the spectra of the pure components extracted by MCR-ALS with reference spectra from a standard library. Accordingly, before opening the MCR-ALS module, all data (i.e. the spectra of the Data Matrix to be processed by MCR-ALS, and optionally all reference spectra to utilize after MCR-ALS) must be imported and consistently preprocessed in the Main GUI, as mentioned in Section 4.

After all data has been consistently preprocessed in the Main GUI, open the MCR-ALS module via the Main GUI menu:

Main GUI Menu > Chemometry > “MCR-ALS”

Once the main MCR-ALS window is open, there are essentially three main steps before launching the toolbox:

- **Step 1: Specify the Data Matrix**

Select all scans that are part of the Data Matrix from the sample list in the Pre MCR-ALS part of the main MCR-ALS window (Figure 69). At least 5 scans must be selected to activate the Jaumot *et al.* toolbox in the middle of the window.

- **Step 2: Select the data type**

Specify if the data is normalized XANES/mu (select "Norm") or chi/EXAFS (select "EXAFS"), using the menu in the middle of the Pre MCR-ALS part of the window. If the raw data imported to Fastosh is already normalized, select "Raw" instead of "Norm".

If the data corresponds to mu/XANES spectra, all spectra except the first one selected from the list are interpolated based on the energy array of the first spectrum of the list.

Additionally, if the data corresponds to XANES/mu spectra and the dataset is not centered, the 1<sup>st</sup> principal component can represent most of the variance in the data set, since it can mainly correspond to the XAFS edge-jump (in fact, the spectral mean). The option "Exclude 1<sup>st</sup> PC", available in the Pre MCR-ALS corner of the main MCR-ALS window (Figure 69) enables to remove the contribution of the 1<sup>st</sup> component from the uncentered [D]. This can help determine the number of principal components before starting MCR-ALS, as mentioned below in Section **Erreur ! Source du renvoi introuvable.. However, it is recommended to always include the 1<sup>st</sup> component to process the Data Matrix by MCR-ALS.** If there were N number of principal components identified based on PCA performed at the Pre MCR-ALS stage without including the 1<sup>st</sup> component, then N+1 number of principal components should be considered to perform MCR-ALS with the 1<sup>st</sup> component included. If the 1<sup>st</sup> component is excluded from the Data Matrix to perform MCR-ALS, it will be eventually added to the extracted pure phases once MCR-ALS is completed.

- **Step 3: Select fitting range**

Select the range of the Data Matrix to process in the field "MCR-ALS data set from: to:"

- Optional last step: PCA & Multiple SIMPLISMA.

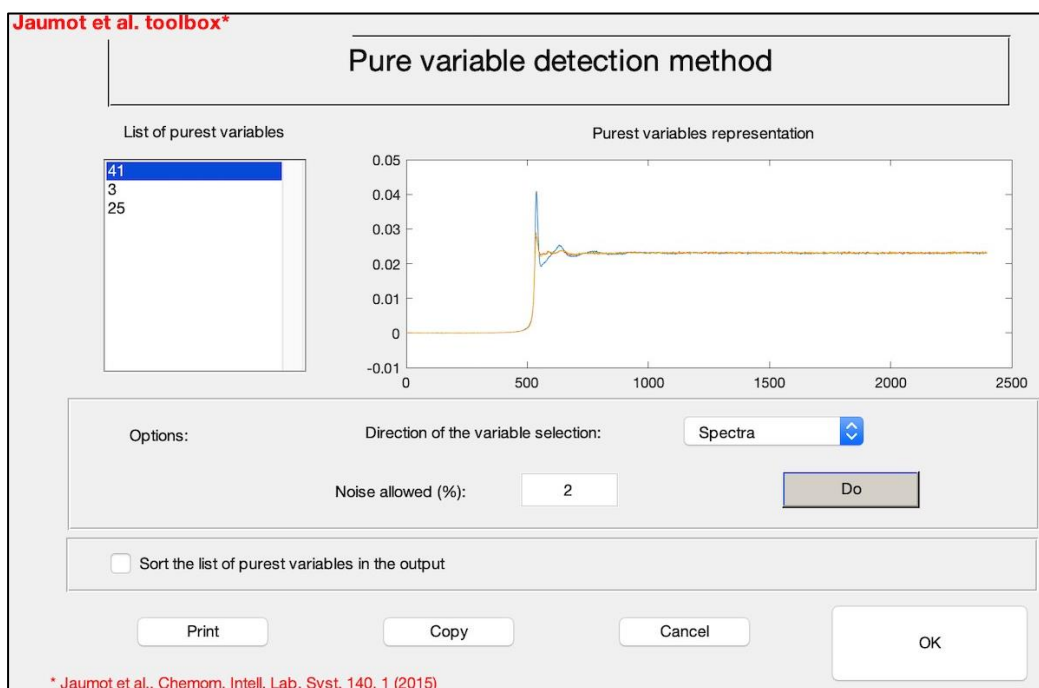
Two tools called "PCA" & "Multiple SIMPLISMA" are available at the bottom of the Pre MCR-ALS corner of the main MCR-ALS window. The first one can be used to perform PCA on the Data Matrix prior to MCR-ALS. It proposes more options to visualize PCA results than those available in the Jaumot *et al.* toolbox (Section 6.3.2).

Additionally, if the goal is to use the SIMPLISMA method to do an initial estimation of [Pure Spectra] or [Pure Coefficients] at the beginning of MCR-ALS (Section 6.4.3), the "Multiple SIMPLISMA" tool can be used prior to starting the MCR-ALS toolbox to rapidly test multiple SIMPLISMA thresholds as explained in the next Section (6.3.2).

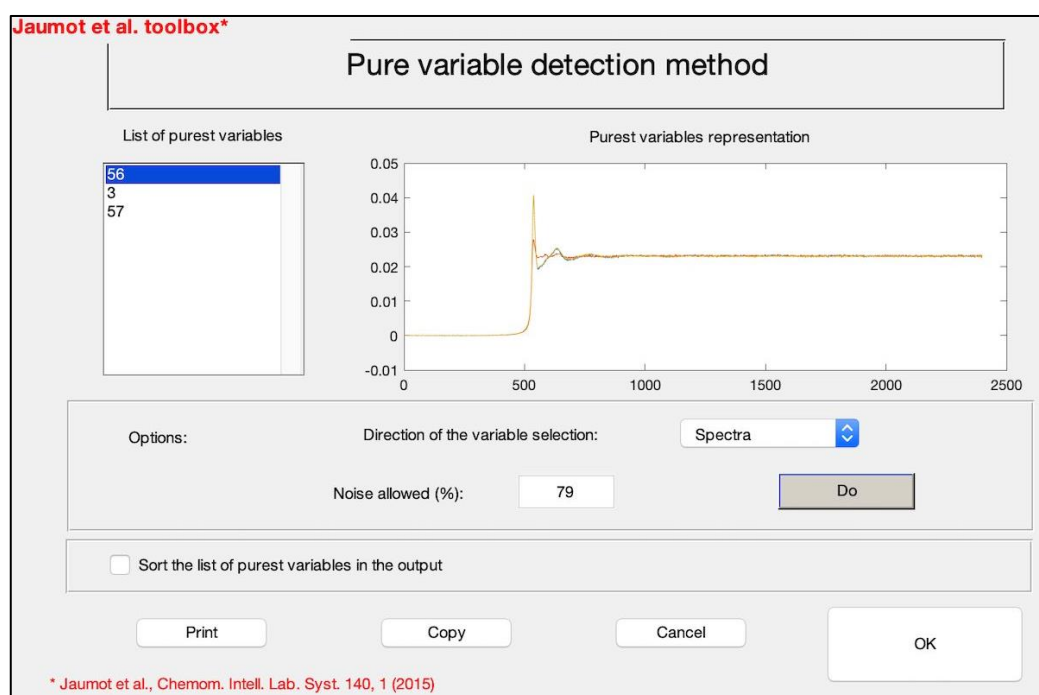
## 6.3.2 PCA & Multiple SIMPLISMA

### 6.3.2.1 PCA

The "PCA" window enables to perform PCA on the Data Matrix selected in the main MCR-ALS. Select "PCA..." at the bottom of the Pre MCR-ALS corner of the main MCR-ALS window. A new window opens up, where different options to perform SVD and visualize the PCA results are available: scree plot of eigenvalues, singular value plot, logarithmic first difference of the eigenvalues, & IND plot. All these options were described in the section dedicated to PCA analyses (Section 5.4.3). Note that the singular value plot in this window is the same as the scree plot reported in the Jaumot *et al.* toolbox (Section 6.4.2).



A)



B)

Figure 70 - Jaumot *et al.* toolbox: SIMPLISMA done on a Data Matrix with 3 principal components with  
A) a 2% noise threshold and B) a 79% noise threshold

## 6.3.2.2 Multiple SIMPLISMA

### 6.3.2.2.1 Usefulness

SIMPLISMA is one of the two methods that can be employed in the Jaumot *et al.* toolbox to do an initial estimation of [Pure Spectra] or [Pure Coefficients]. The Principle of SIMPLISMA will be detailed below in Section 6.4.3.1. To perform SIMPLISMA on a data matrix, the number of principal components must be known, thus PCA must be done prior to SIMPLISMA. Additionally, a noise threshold value, in %, must be provided to perform SIMPLISMA as explained later on (Section 6.4.3.1). If the initial estimation is done on [Pure Spectra], each row and column of the

data matrix correspond to a full experimental spectrum and a specific data point energy in all spectra, respectively. The SIMPLISMA approach finds what are the most dissimilar spectra (*“pure spectra”*) in the data matrix based on the chosen threshold value (more details on how SIMPLISMA finds these spectra in Section 6.4.3.1). It is assumed that each of these dissimilar spectra correspond to the spectrum of a specific principal component. For example, in the SIMPLISMA window of the Jaumot *et al.* toolbox shown in Figure 70a, the user-defined threshold value was set to 2 %. The row numbers in the data matrix corresponding to the most dissimilar spectra were then 3, 25, & 41 (Figure 70a). These three spectra are shown in the SIMPLISMA window (Figure 70a). When the user-defined threshold value was set at 79 %, the row numbers in the data matrix of the most dissimilar spectra were 3, 56, & 57 (Figure 70b).

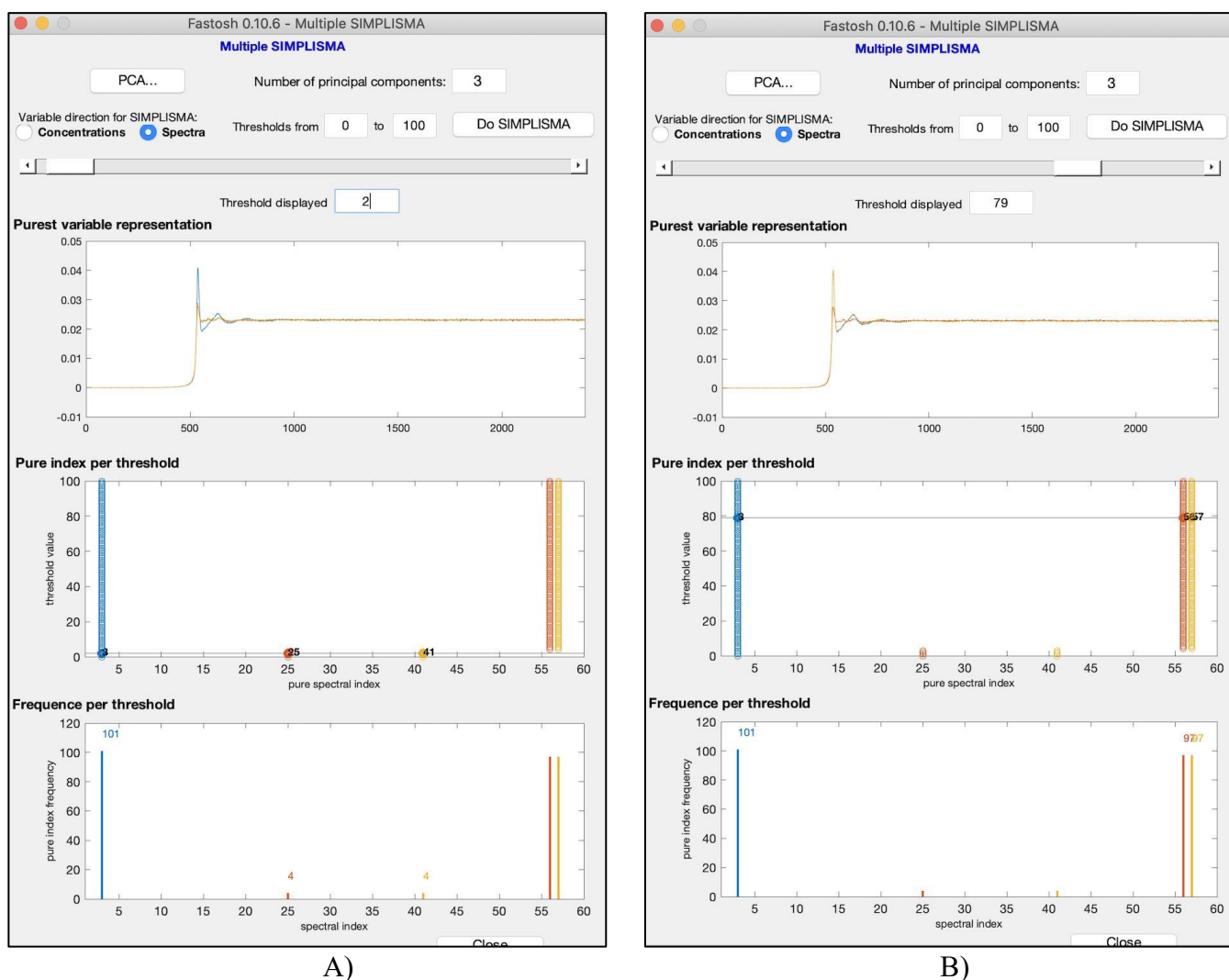


Figure 71 –Results for 101 SIMPLISMA operations in the Multiple SIMPLISMA window, considering 3 principal components in the sample mixture. In this window, the highlighted threshold value is equal to A) 2% or B) 79 %.

To test multiple threshold values in the SIMPLISMA window of Jaumot *et al.* toolbox, each value must be individually provided to the GUI as shown in Figure 70 a & b. This can be quite time consuming since that there are 101 possible threshold values that can be employed to perform SIMPLISMA (the threshold must be an integer between 0 and 100 %). **The “Multiple SIMPLISMA” functionality of Fastosh enables to test all 101 possible thresholds at once.** One can then identify at a glimpse how many thresholds provide the exact same results.

For example, the data employed in the Jaumot *et al.* toolbox’s SIMPLISMA window in Figure 70 was also processed with “Multiple SIMPLISMA” (Figure 71). Results indicate that the 101 threshold values can be categorized into two groups. The first group consists in 4 threshold values, between 0 and 3 %. Using any of these threshold values, the numbers/positions in the data

matrix of the most dissimilar spectra found by SIMPLISMA are 3, 25, & 41. The second group consists of 97 threshold values, between 4 and 100 %. Using any of these threshold values, the numbers/positions in the data matrix of the most dissimilar spectra found by SIMPLISMA are 3, 56, & 57. Therefore, in this example, there's finally only two possible choices of threshold values to perform SIMPLISMA in order to obtain the initial estimation of [Pure Spectra] in the MCR-ALS Jaumot *et al.* toolbox. If not sure about the choice in threshold value, SIMPLISMA & MCR-ALS could be successively done two times, using any threshold value from Group1, and then Group 2, and the final results could be eventually compared to each other. However, given that the pure spectra do not seem to be really dependent on threshold values based on their appearances in the SIMPLISMA window (Figure 70 and Figure 71), the choice of threshold value should not significantly matter in this particular example. Also, the data matrix is constituted of  $\mu$  spectra in this example, which implies that the threshold is actually not so relevant (see Section 6.4.3.1).

If the initial estimation is done on [Pure Spectra], with a data matrix constituted of EXAFS spectra, the threshold should be similar to the variance contribution of the noise, which corresponds to the base level of the scree plot obtained by PCA. Accordingly, if the base level of the scree plot is, for example, around 5-10%, one can perform multiple SIMPLISMA with a noise threshold between 5 and 10% (i.e. 6 SIMPLISMA done at once) to check whether different results can be obtained.

#### 6.3.2.2.2 Description of the Multiple SIMPLISMA window

Before performing multiple SIMPLISMA, the number of principal components must be provided on top of the Multiple SIMPLISMA window. If this number is not known, press "PCA..." to determine this number (Section 6.3.2.1).

Additionally, either [Pure Spectra] or [Pure Coefficients] must be chosen to do the initial estimation. In most cases, the initial estimation should be done on [Pure Spectra], as explained later in Section 6.4.3.1.2. Then choose what are the threshold range to perform multiple SIMPLISMA (default range is from 0 to 100%, thus 101 threshold values/SIMPLISMA operations). Finally, press "Do SIMPLISMA" to perform Multiple SIMPLISMA.

The results are shown in three figures featured in the Multiple SIMPLISMA window. In all of them, all data points corresponding to a specific principal component are displayed with a specific color. For example, in Figure 71, it was specified that 3 principal components were present in the sample mixture. Therefore, all data points corresponding to the 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> principal component are displayed in the figures with a blue, red, and yellow color, respectively.

- **"Purest variable representation"**

If the initial estimation is done on [Pure Spectra], the most dissimilar spectra found ("*pure spectra*") in the data matrix are shown in this plot.

If the initial estimation is done on [Pure Coefficients], the most different concentrations ("*pure concentrations*") in the data matrix are shown in this plot.

- **"Pure index per threshold"**

The numbers/positions of each pure spectrum/concentration in the data matrix is shown on the X axis for each principal component. The specific threshold value employed to perform SIMPLISMA is shown on the Y axis. For example, in Figure 71, the pure spectrum number corresponding to the first principal component was always 3, regardless of the threshold value. For the second and third principal components, one can observe that two sets of results were obtained: the first one for threshold values between 0 and 3 % (pure spectra 25 & 41), and the second one for threshold values between 4 and 100 % (pure spectra 56 & 57).

The horizontal bar highlights the results corresponding to the threshold value specified on top of the window. When the latter was set to 2 and 79 %, the pure spectra numbers were 3, 25, 41 (Figure 71a) and 3, 56, 57, (Figure 71b), respectively. These are the same SIMPLISMA results as those obtained in the Jaumot *et al.* MCR-ALS toolbox, when these two threshold values were employed (Figure 70 a & b).

- **"Frequency per threshold"**

This figure shows how many times the pure spectra/concentrations per principal component corresponding to the threshold value specified on top of the window was

found for all Multiple SIMPLISMA performed. For example, in the example shown in Figure 71a, the threshold value specified on top of the window was 2 %. The pure spectrum of the first principal component corresponding to the 2 % threshold value was found with 101 multiple SIMPLISMA/threshold values. The pure spectrum corresponding to the second or third principal component corresponding to the 2 % threshold value was found with 4 multiple SIMPLISMA/threshold values (Figure 71a).

#### 6.3.2.2.3 Choice of COO expression

As detailed below in Section 6.4.3.1, a “Correlation around the Origin” (COO) matrix is used by SIMPLISMA to find the pure variables except for the first one. The expression of COO employed in the Jaumot *et al.* toolbox differs from the one in Windig & Guilment (1991) [17], which introduced the SIMPLISMA method. The COO expression of Windig & Guilment (1991) method is then used by default in the Multiple SIMPLISMA window to find the pure variables. The choice of COO expression (either Windig & Guilment or Jaumot *et al.*) can be modified via the menu bar on top of the Multiple SIMPLISMA window:

#### Multiple SIMPLISMA Menu > COO Expression

The type of COO expression usually does not affect the final SIMPLISMA results when the initial estimation is done on [Pure Spectra], but may have an effect when [Pure Coefficients] is estimated. Since the initial estimation should be usually done on [Pure Spectra] when XAFS data is processed by SIMPLISMA as mentioned below in Section 6.4.3.1.2, this difference in COO expressions should not matter too much.

## 6.4 Jaumot *et al.* MCR-ALS Toolbox

This part of the manual does not represent a complete guide of the toolbox; please rather refer to the publications of Jaumot *et al.* for specific details [3, 4]. It does aim, however, to provide practical information on how to apply this toolbox specifically to XAFS data, so that any XAFS user who is unfamiliar with MCR-ALS can autonomously start using the toolbox. Firstly, a description of all steps to follow in the toolbox is provided.

### 6.4.1 Overall procedure

There are five steps:

- Step 1: (*First window*) Provide the number of principal components. If the number is already known, directly specify it (press “Manual”) or find it, using the PCA module of the toolbox (press “SVD”) -see Section 6.4.2 below.
- Step 2: (*Also, first window*) Do the initial matrix estimation by choosing either SIMPLISMA (press “PURE”) or EFA (press “EFA”) -see Section 6.4.3 below.
- Step 3: second window “Selection of the data set”: specify the number of individual datasets, if the data matrix is a group of datasets (“augmented” data matrix).
- Step 4: (*window “Constraints: row mode”*) Specify all constraints to apply to [Pure Coefficients] -see Section 6.4.4 below.
- Step 5: (*window “Constraints: column mode”*) Specify all constraints to apply to [Pure Spectra] -see Section 6.4.4 below.
- Step 6: (*last window before ALS optimization*) Specify the convergence criterion before starting the fitting procedure -see Section 6.4.6.1.2 below.

## 6.4.2 PCA, prior to MCR-ALS

The PCA window of the Jaumot *et al.* toolbox displays the result of the Singular Value Decomposition (SVD) of the Data Matrix (Figure 72). The “Eigenvalues” reported in this window correspond to the elements in the diagonal of [S] obtained from SVD. This Eigenvalue definition thus differs from the one employed in Malinowski (1978) [7], Manceau *et al.* (2014) [14], or in the PCA windows of Fastosh (Sections 5.4 & 6.3.2.1), where the Eigenvalues correspond to the square of the elements in the diagonal of [S]. As mentioned in Section 5.4.3.2.3, an Eigenvalue defined using any of these two definitions can be considered as an apparent variance indicator. Plotting singular value against component number just like in the Jaumot *et al.* toolbox (Figure 72a), and interpreting the plot like a scree to determine the number of principal components, may provide results more accurate than other approaches including the regular scree plot method, as demonstrated in Section 5.4.3.2.3. To test other approaches to determine the number of principal components in the data matrix, use the PCA window of the Pre MCR-ALS part (Figure 72b) before opening the toolbox, as it proposes more than one option to plot the PCA results (Section 6.3.2.1).

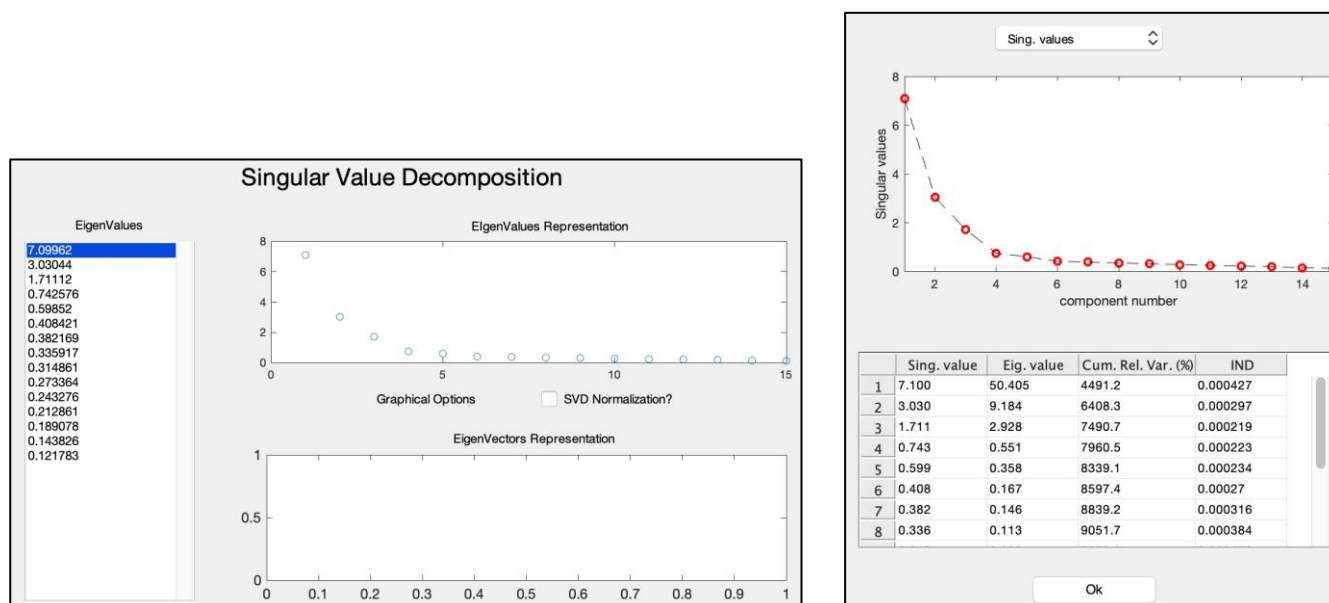


Figure 72 - PCA result window in A) Jaumot *et al.* toolbox and B) Fastosh's pre MCR-ALS, using the same data matrix

## 6.4.3 Initial estimation: SIMPLISMA or EFA

The Simple-to-use Interactive self-Modeling Mixture Analysis (SIMPLISMA) [17] and Evolving Factor Analysis (EFA) [18] are two methods available in the Jaumot *et al.* toolbox to do the initial estimation of [Pure Spectra] or [Pure Coefficients]. The SIMPLISMA is applied to a data matrix whose spectra are not temporally related to each other. For example, SIMPLISMA can be applied to spectra corresponding to different soil samples from a contaminated site. The EFA is applied to spectra that gradually evolve with time, such as spectra successively obtained during an operando experiment conducted at the beamline.

### 6.4.3.1 SIMPLISMA

#### 6.4.3.1.1 Principle

The SIMPLISMA approach is referred as “PURE” in the Jaumot *et al.* toolbox as it is a pure variable-based method [17]. A “pure variable” is a variable whose intensity is due to only one of the components in the mixture under consideration. The steps of the SIMPLISMA procedure are:

- Step 1: The form of the data matrix [D] is defined based on whether the initial estimation is done on [Pure Spectra] or [Pure Coefficients]. In the former case, each row and column

of the data matrix correspond to a full spectrum and the values at a specific incident x-ray energy in all spectra, respectively. In the latter case, [D] is transposed so that each row and column correspond to the values at a specific incident x-ray energy in all spectra and a full spectrum, respectively.

- Step 2: The first pure variable is identified. Firstly, “Possible Pure  $i_1$ ” is calculated for each row “i” of the data matrix:

$$(9) \text{ Possible Pure } i_1 = \frac{\sigma_i}{\mu_i + \alpha}$$

where “i” is the row number in the Data Matrix, “ $\sigma_i$ ” is the standard deviation of all column values at row “i”, “ $\mu_i$ ” is the mean of all column values at row “i”, and “ $\alpha$ ” is equal to:

$$\alpha = \frac{\text{threshold}}{100} \cdot \max(\mu)$$

where “threshold” is the user-defined threshold value (in %) and “ $\max(\mu)$ ” is the highest average among all column means calculated along each row of the data matrix. The “ $\alpha$ ” is employed in Expression (9) to avoid the case where “ $\mu_i$ ” is equal to zero (case impossible with XANES but possible with EXAFS spectra since the data oscillates around 0).

The row number in the data matrix corresponding to the first pure variable (Pure  $i_1$ ) is equal to the value of “i” for which the highest “Possible Pure  $i_1$ ” is obtained.

- Step 3: The remaining pure variable(s) is/are found. The value “Possible Pure  $i_j$ ” is calculated for each row “i” of the data matrix and each pure variable j, with  $j > 1$ :

$$(10) \text{ Possible Pure } i_j = \left( \frac{\sigma_i}{\mu_i + \alpha} \right) \cdot w_{i,j}$$

where  $w_{i,j}$  is a determinant-based function whose meaning is described below. The row number in the data matrix corresponding to the pure variable (Pure  $j$ ) is equal to the value of “i” for which the highest “Possible Pure  $i_j$ ” is obtained.

Any pure variable other than the very first one must be independent from the previous pure variable. Accordingly,  $w_{i,j}$  is used as a weight function in Expression (10) to determine the independence of the variable j. It is equal to:

$$(11) w_{i,j} = \begin{vmatrix} C_{i,i} & C_{i,p_1} & \dots & C_{i,p_{j-1}} \\ C_{i,p_1} & \dots & \dots & \dots \\ \vdots & \dots & \dots & \dots \\ C_{i,p_{j-1}} & \dots & \dots & C_{p_{j-1},p_{j-1}} \end{vmatrix}$$

where “i” is a row number in the Data Matrix [D],  $p_1$  is the row number in the Data Matrix corresponding to the first pure variable,  $p_{j-1}$  is the row number in the Data Matrix corresponding to the last pure variable found before the current pure variable being searched, and “C” is the “correlation around the origin” (COO) matrix. Its expression is:

$$(12) C = \frac{1}{s} \cdot \frac{[D]}{\sqrt{(\mu^2 + (\sigma + \alpha)^2)}} * \left( \frac{[D]}{\sqrt{(\mu^2 + (\sigma + \alpha)^2)}} \right)^T$$

**Note:** the above COO expression corresponds to the one introduced in Windig & Guilment (1991) [17]. It differs from the COO expression originally defined in the Jaumot *et al.* toolbox, which is:

$$(13) \mathbf{C} = \frac{1}{s} \cdot \frac{[\mathbf{D}]}{\sqrt{(\sigma^2 + (\mu + \alpha)^2)}} * \left( \frac{[\mathbf{D}]}{\sqrt{(\sigma^2 + (\mu + \alpha)^2)}} \right)^T$$

The SIMPLISMA procedure in the Jaumot *et al.* toolbox featured in Fastosh was then slightly modified so that the COO expression employed by default by SIMPLISMA is the one introduced in Windig & Guilment (1991) (Expression (12)). To use instead the expression originally employed in the toolbox (Expression (13)), uncheck the box “COO expression of Windig & Guilment 1991” in the SIMPLISMA window (Figure 70). The actual effect on the results due to this COO expression difference is discussed in the next section (Section 6.4.3.1.2).

#### 6.4.3.1.2 Procedure

Once the SIMPLISMA window of the toolbox is open, follow these four steps:

- Step 1: Specify if the initial estimation is done on [Pure Spectra] or [Pure Coefficients]. **With XAFS data, SIMPLISMA should be done on [Pure Spectra] in most cases.** Indeed, XAFS data matrix are usually composed of XAFS spectra where the number of data points per spectrum exceed the number of spectra in the data matrix. A typical XAFS data matrix can, for example, consist in 10 spectra, each of them featuring 1000 data points. If the initial estimation is done on [Pure Spectra], the standard deviation and mean calculated to find the pure variables (Expressions (9) & (10)) will reflect the data variability in all 1000 x-ray energies per spectrum. In contrast, if the initial estimation is done on [Pure Coefficients], the standard deviation and mean will reflect the data variability in all 10 spectra at a specific x-ray energy of the spectra. Although this variability can reflect structural variations between spectra, it can also, partly or mainly, reflect the variations only due to noise since XAFS spectra usually feature a non-negligible amount of random noise.
- Step 2: Specify whether the COO expression employed by SIMPLISMA to find the pure variables is the one introduced by Windig & Guilment (1991) [17] (default choice) or the one originally employed in the toolbox. The type of COO expression usually does not affect at all the final SIMPLISMA results when the initial estimation is done on [Pure Spectra], but may have an effect when [Pure Coefficients] is estimated. Since the initial estimation should be done in most cases on [Pure Spectra] as mentioned in the previous paragraph, the difference in COO expressions should not matter.
- Step 3: Specify a threshold, in %. If the initial estimation is done on [Pure Spectra], the threshold should be similar to the variance contribution of the noise, which corresponds to the base level of the scree plot obtained by PCA.
- Step 4: Press OK to do SIMPLISMA. The row numbers of all pure variables in the data matrix and their respective pure spectrum are shown in the window.

#### 6.4.3.2 EFA

##### 6.4.3.2.1 Principle

The EFA method is strictly applied to data matrices where spectra are temporally related to each other. Therefore, the ordering of the spectra in the data matrix must reflect this temporal relationship.

A PCA analysis is repeatedly performed on a data matrix where the total number of spectra is gradually increased, starting from the first spectrum collected to the last spectrum collected. The eigenvalues obtained at the end of all these PCA analyses are plotted in a figure against number of spectra in the data matrix. This procedure is then repeated, but starting from the last spectrum collected to the first spectrum collected. All obtained eigenvalues are again plotted in the same

figure as the previous one. The overlay of the forward and reverse analyses provides an estimation of the concentration profiles, i.e. [Pure Coefficients].

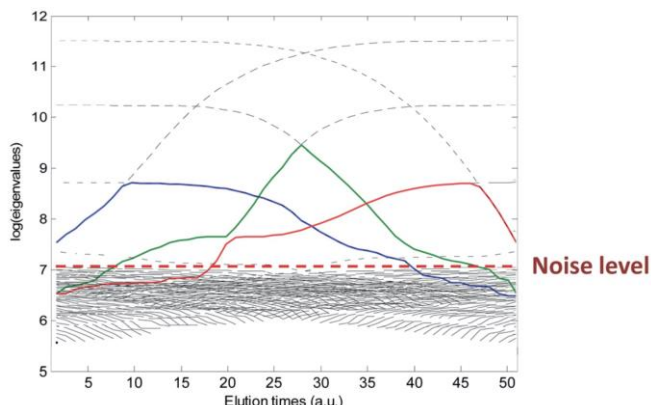


Figure 73 – Example of noise level determination when a data matrix of chromatographic spectra was processed by EFA [18].

From version 1.0.6, Fastosh proposes two different EFA approaches. The first one is employed when the data matrix, to be processed by MCR-ALS, consists of a single kinetics dataset. The second approach is employed when the data matrix, to be processed by MCR-ALS, is an “augmented” data matrix consisting of several *kinetics datasets* grouped together. In the first and second case, select “EFA on one matrix” and “EFA on multi matrices” in the first window of the Jaumot et al. Toolbox, respectively. The procedure related to each approach is detailed below.

#### 6.4.3.2.2 Procedure: case of an individual kinetics dataset

If the data matrix to be processed by MCR-ALS consists of a single kinetics dataset, select “EFA on one matrix” in the first window of the Jaumot et al. Once the EFA window of the toolbox is open, simply repeatedly press on “Continue” button in the window to successively:

- Do all forward PCA analyses
- Do all backward PCA analyses. Once completed, the forward and backward results are plotted in a same figure, as log(eigenvalue) per number of spectra (Figure 74a).
- Show a box in the EFA window where the noise level can be provided. This corresponds to where the concentration profiles and noise separate from each other on the Y axis, as shown in the example in Figure 73. Providing a value will only rescale the figure and mask the noise contribution so that the concentration profiles can be better visualized. Since this noise level value is actually not used in the EFA procedure, this step can be skipped. However, in any case a value must be provided to the box to activate again the button “Continue” in the window and move on to the next step.
- Display the final concentration profiles of the principal components plotted as eigenvalue per number of spectra in the data matrix (Figure 74c). The concentration profile of each principal component is determined using the forward and backward PCA results of the principal component. The concentration profile is equal to the forward profile as long as the eigenvalues of the forward profile are smaller than those of the backward profile. If they become higher, the concentration profile is then equal to the backward profile (Figure 74b).

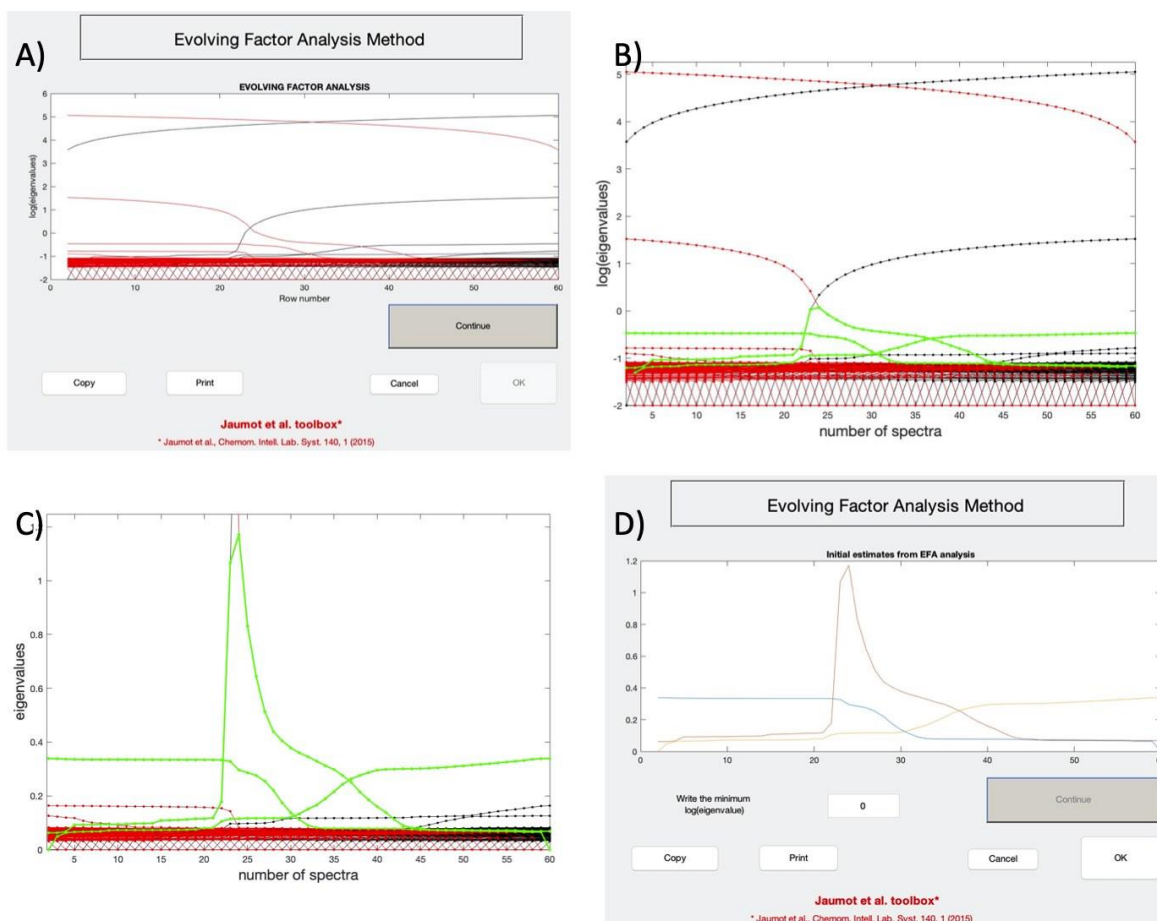


Figure 74 -EFA results in the Jaumot *et al.* toolbox: A) forward and backward analyses results reported as log(eigenvalue) per number of spectra/row number in the data matrix B) same plot as the previous one, but with the concentration profiles of all principal components highlighted in green color; C) same plot as the previous one, but zoomed in, and with results reported as eigenvalue per number of spectra; and D) last window of EFA in Jaumot *et al.*, where the concentration profile of each principal component is shown with a specific color, in eigenvalue per number of spectra.

#### 6.4.3.2.3 Procedure: case of an “augmented” data matrix (multiple kinetics datasets)

A limitation of the original Jaumot *et al.* toolbox is that the EFA cannot be performed on an “augmented” data matrix consisting of several *kinetics datasets* grouped together. Indeed, in this particular case, using the EFA method is impossible as it can only be applied to a matrix corresponding to a single kinetic experiment. Similarly, using SIMPLISMA is not ideal either, as the purest species identified with this method may not necessarily correspond to the principal components in the different kinetics datasets that are defined, in the ISP matrix, as relevant for the MCR-ALS calculation. From version 1.0.6, Fastosh proposes a tool to perform the EFA method on each individual dataset, and then concatenate the initial [coefficients] associated to each dataset into one [coefficients] corresponding to the augmented data matrix. In the first window of the toolbox, select “EFA on multi matrices” instead of “EFA on one matrix”. Then, in a new window, specify the number of individual kinetics datasets, and the first and last scans belonging to each of these individual kinetics datasets.

Note: the first and last scans belonging to the last individual kinetics dataset is automatically inferred, based on the selection relative to all previous individual kinetics datasets. The results are plotted in a similar way as the EFA module originally included in the toolbox.

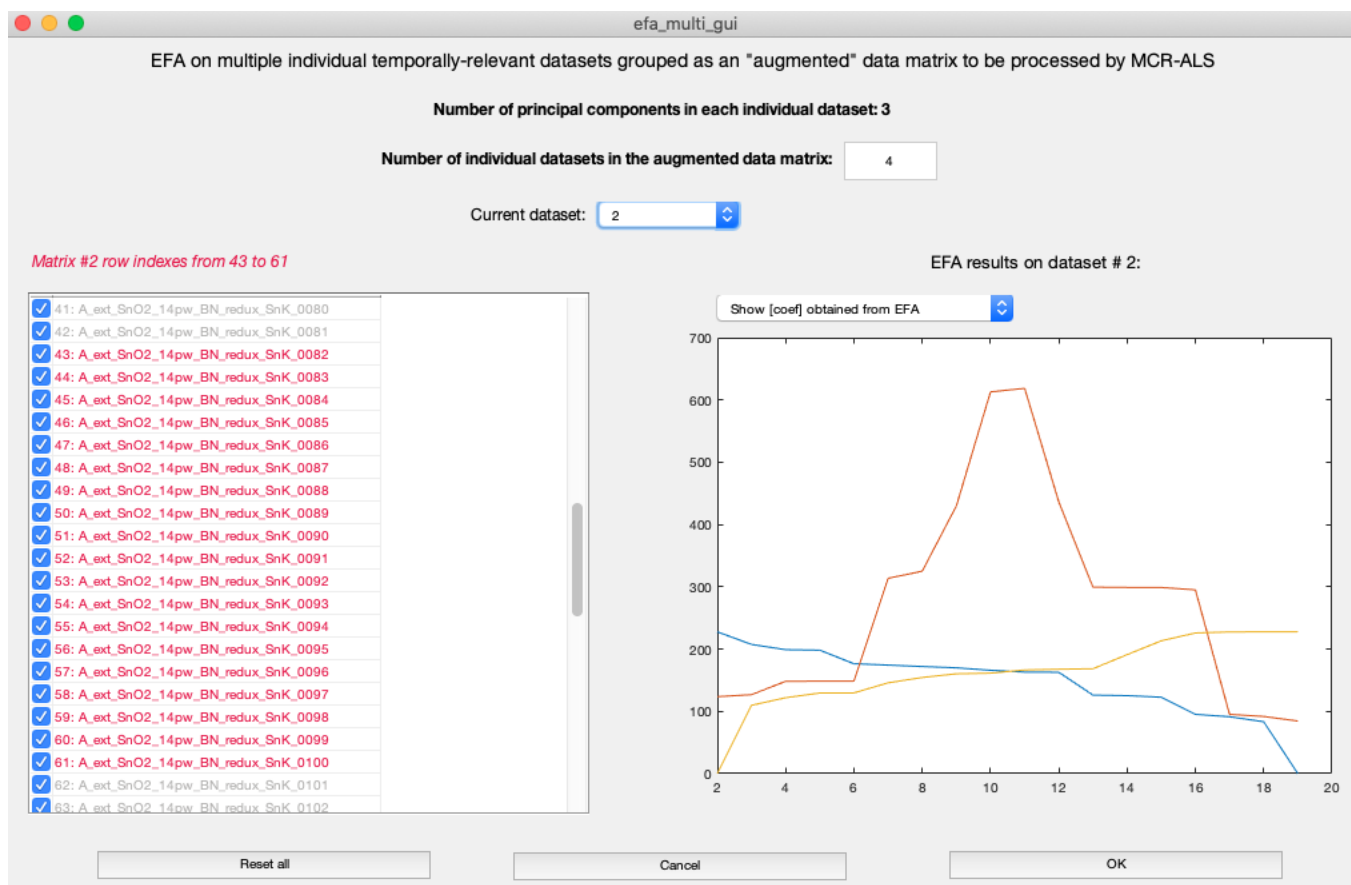


Figure 75 -Tool to perform EFA on an augmented data matrix consisting of more than one temporally-relevant datasets

## 6.4.4 Constraints

Selecting the appropriate constraints and knowing how to apply them is the most crucial point to ensure obtaining meaningful and reliable MCR-ALS solutions [19]. This section describes the MCR-ALS constraints that can be applied to XAFS data: non-negativity, unimodality, closure, and equality. The constraints are firstly applied to [Pure Coefficients] in the window “Constraints: row mode”, and then to [Pure Spectra] in the window “Constraints: column mode”.

### 6.4.4.1 Automatic setting of constraints

From version 1.0.6, Fastosh features a function that automatically sets the basic constraints applied to XAFS data. If MCR-ALS is to be performed on...

**...Norm data:** Non-negativity and closure are automatically set to [Pure Coefficients], and non-negativity is automatically set to [Pure Spectra]

**...EXAFS data:** Non-negativity and closure is automatically set to [Pure Coefficients] only.

**...non-XAFS:** No constraint is automatically applied.

To deactivate this function, unchecked in the menu of the Fastosh’s MCR-ALS window:

Menu of Fastosh’s MCR-ALS window > Options > “Typical MCR-ALS constraints already selected in the toolbox”

#### 6.4.4.2 Non-negativity

This applies a non-negativity constraint on all values of the matrix. **This constraint should be systematically imposed to [Pure Coefficients] whether the spectra are XANES or EXAFS**, since the coefficients (i.e. relative contribution of the spectra corresponding to all principal components to reconstruct any experimental spectrum of the data matrix -the coefficients are referred as “concentrations” in the toolbox) should always be positive. **Additionally, a non-negativity constraint should be always imposed to [Pure Spectra] if the spectra are XANES**, since they are essentially normalized  $\mu$  spectra and thus cannot be negative. However, this constraint should not be imposed to [Pure Spectra] if the spectra are EXAFS as the data oscillates around 0.

There are three types of non-negativity constraints available, which are described below.

##### 6.4.4.2.1 Force to zero

This constraint simply replaces negative values with zeros. Choose this option only when not all profiles in [Pure Spectra] or [Pure Coefficients] are constrained or when statistically sounder non-negative least square methods (e.g. fnnls described below) fail or take too long to complete [3]. Therefore, “Force to zero” should not be chosen in most cases. If this option is chosen, specify also the number of profiles to constrain (i.e. all profiles, most likely).

##### 6.4.4.2.2 nnls

The non-negative least-squares (nnls) constraint is an old least square-based approach that may imply long and unnecessary computational steps. Use instead the fnnls constraint as it is an improved version of nnls.

##### 6.4.4.2.3 fnnls

The fast non-negative least-squares (fnnls) improves upon nnls by avoiding redundant computations [20]. This should be the non-negativity method to choose by default. If it is chosen, specify that all principal components are constrained, which is the only choice available in the menu “Nr. Of species with non-negative profiles?”. If this is not done, the toolbox may return an error later on.

#### 6.4.4.3 Unimodality

The unimodality constraint forces that only one maximum is allowed per profile. The unimodality can be either vertical, horizontal, or average (Figure 76).

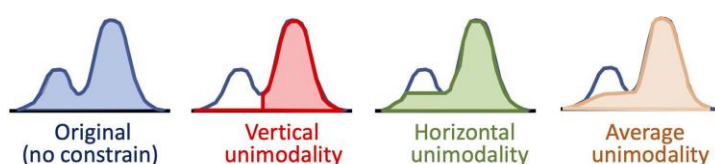


Figure 76 -Schematic representation of the three different unimodality types

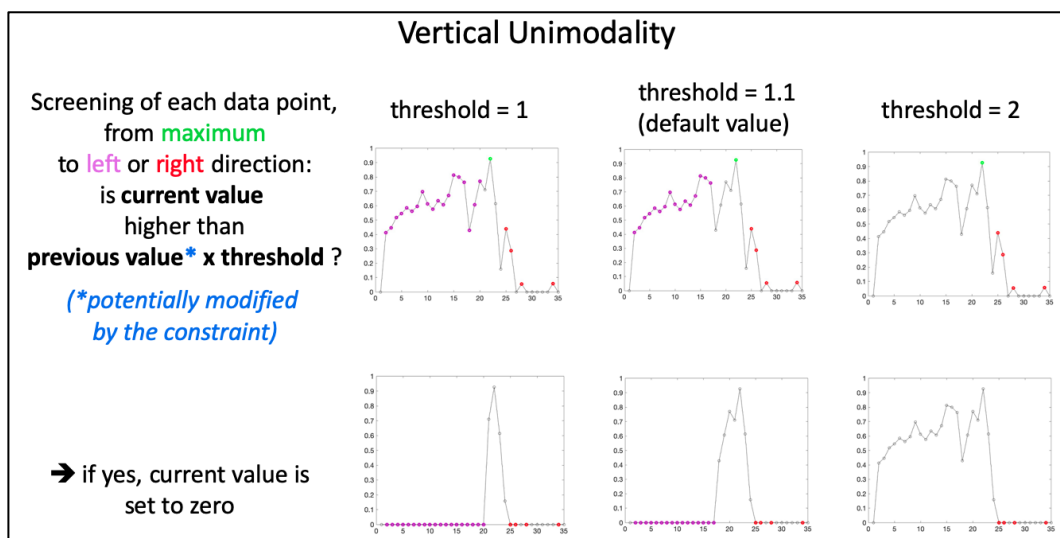
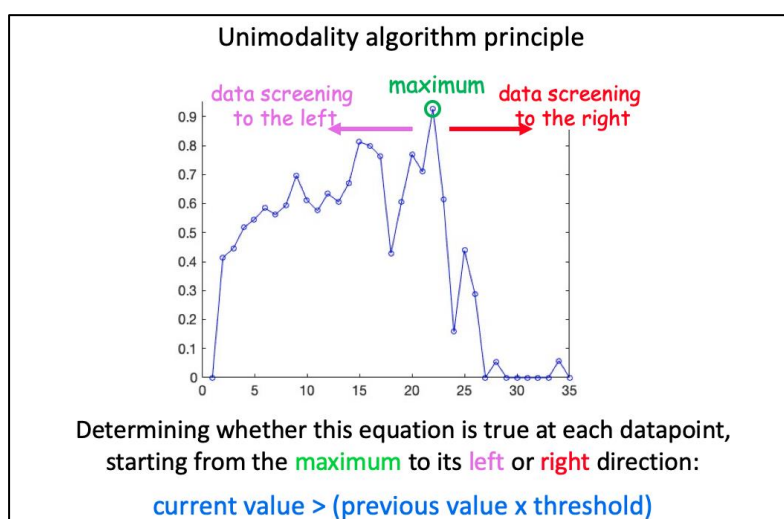
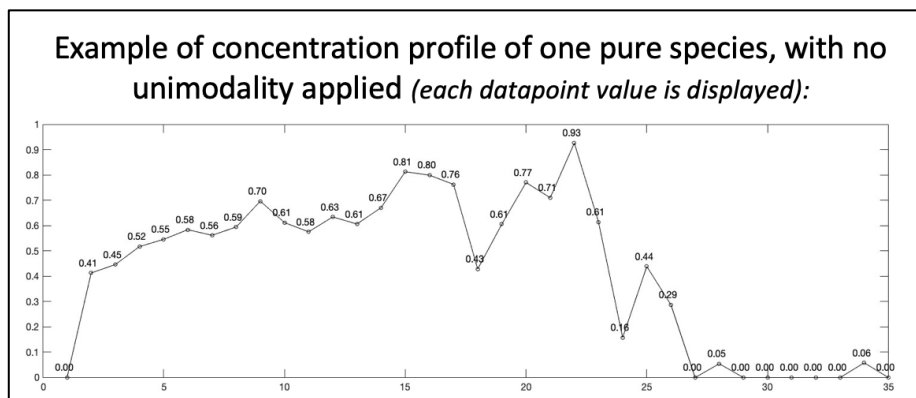
The vertical unimodality forces all non-unimodal parts of the profile to zero. The horizontal unimodality forces all non-unimodal parts to be equal to the last unimodal value. The average unimodality (the smoothest one) forces all non-unimodal parts to be an average between the two extreme values being still unimodal using a regression procedure.

A constraint tolerance (a.k.a. “constraint threshold”) can be selected to allow for some local departures of the unimodality condition. For instance, 1.5 means that 50% of local departure of the unimodal condition is allowed, i.e. that in the decreasing slopes of the main peak, a particular point can increase a maximum of 50% of the previous value before the unimodality constraint is applied. Values between 1.0 (no departures from the unimodal condition allowed) and 1.1 are usual in systems with low to medium noise levels [3]. More details on the effects of the threshold value applied to the vertical, horizontal, or average constraints is shown in Figure 77. Based on the results shown in this figure, the vertical approach is the most radical method. It may yield to uncorrect results when the threshold is too low (e.g. 1.1) and data is as noisy as the one shown in

Figure 77 as a significant portion of the data is set to zero. The average method is the most conservative relative to the original data.

Once the unimodality type and constraint tolerance have been set, specify how many principal components are constrained by unimodality using the menu “Nr. Of species with unimodality profiles?”.

If the data matrix consists of XAFS spectra that are temporally linked to each other, e.g. a set of spectra that were collected during a kinetic experiment followed *in-situ* at the beamline, a unimodality may be applied to [Pure Coefficients], if a non-unimodal behavior is not expected to occur based on prior-knowledge of the system. However, if the data matrix consists of XAFS spectra that are temporally independent from each other, e.g. XAFS spectra corresponding to different polluted soil samples, a unimodality constraint should not be applied to [Pure Coefficients].



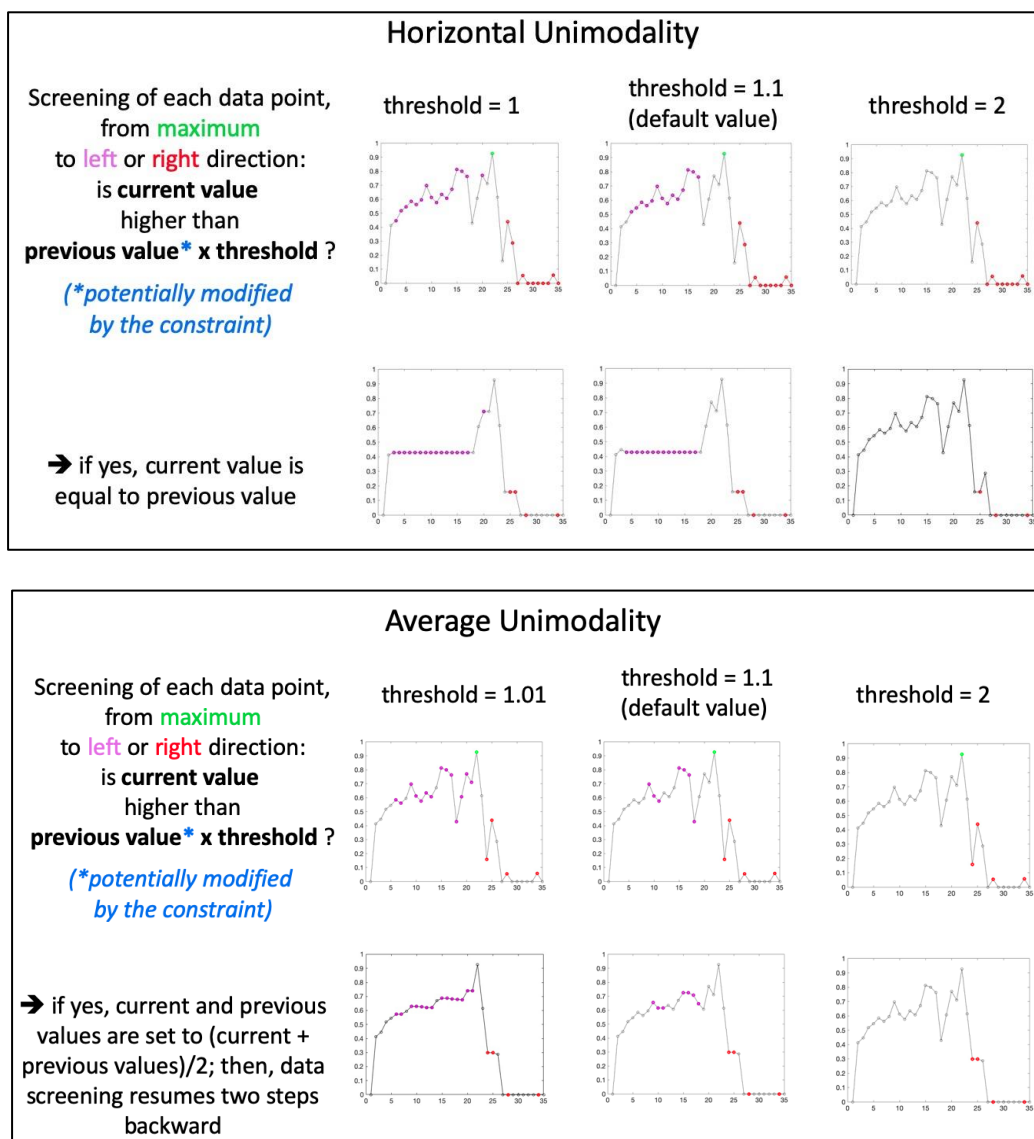


Figure 77 - Threshold effect when a vertical, horizontal, or average constraint is applied

#### 6.4.4.4 Closure

##### 6.4.4.4.1 Principle

*Excerpt from Jaumot et al. (2015) [3]:*

The closure constraint permits to fulfill the mass balance in the system (Figure 78). The total concentration of the system (closure constant) can be fixed to a single value or to a variable (changing) value. If the variation of the closure constant along the experiment is known (e.g., titration experiments with known dilutions), the name of a vector variable that contains the total concentration values at each point of the process should be introduced in the suitable box. The program also allows for the introduction of two closure conditions (two mass balance equations); however, the application of this option is not recommended when common species are shared by both mass balances. Finally, closure can be implemented as an equality constraint (the closure constant is exactly equal to some preselected value) or as a smoother inequality constraint. In the latter case, the mass balance should equal or be lower than the preselected value for the closure constant.

#### 6.4.4.4.2 Application of closure constraint to XAFS data

As previously mentioned, the term “concentration” referred by Jaumot *et al.* corresponds to the relative amounts of the principal components in each sample of the data matrix. The experimental spectrum of a sample belonging to the data matrix can be reconstructed by summing all spectra of the principal components multiplied by their respective relative amount (“coefficients”). For example, if two principal components of [D] are present in a sample belonging to [D] in equal proportions (in mol % of the element of interest), their respective coefficients are 0.5 for this sample. Consequently, the experimental XAFS spectrum of this sample can be reconstructed by summing the XAFS spectra of the two principal components, each of them multiplied by 0.5. **Therefore, a closure on [Pure Coefficients] should be applied to XANES or EXAFS data, to force the sum of all principal component coefficients to 1 for each spectrum of the data matrix.** This can be done in the “Constraints: row mode” window by:

- Applying the “closure” constraint
- In the menu “Nr. of closure constraints to be included?”, Select “1”, so that only one closure constraint is applied.
- Next to “Which species are in 1<sup>st</sup> closure”, check the box “All”, so that the constraint is applied to all principal components of the data matrix.
- In the menu “Closure conditions” menu, select “Equal to”
- In the field “First closure constraint equal to”, type “1”
- (Do not check the box “Closure Variable”)

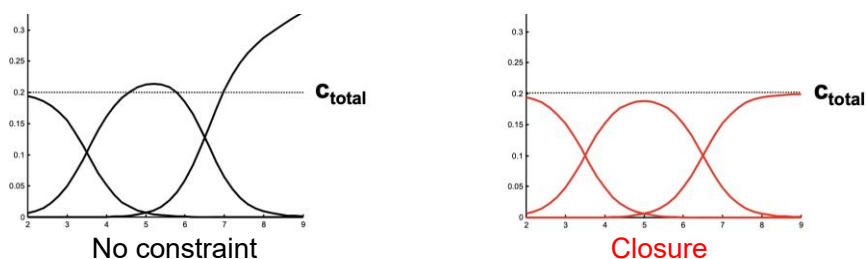


Figure 78 -Schematic representation of the closure constraint (From R. Tauler: <https://core.ac.uk/download/pdf/36047261.pdf>)

#### 6.4.4.5 Equality

##### 6.4.4.5.1 Principle

Since Fastosh v1.0.6, the coefficient(s) or spectrum corresponding to a specific principal component can be fixed instead of being floated during the ALS step. This option can be notably applicable to a dataset corresponding to an *in-situ* experiment conducted at the beamline where the first and last scans of the dataset entirely correspond to the reactant & product of the reaction, respectively. In that case, the spectra of the first and last principal components can be fixed during the ALS step and equal to the first and last spectra of the dataset. Moreover, the coefficients of the first and last principal components can be fixed to 1-0 and 0-1, for the first and last spectrum of the dataset, respectively.

If the Equality Constraint has to be applied to [concentrations], in the window dedicated to setting the constraints to [concentrations], the Jaumot *et al.* toolbox must be feeded with a “concentration csel matrix”. The size of “concentration csel matrix” is equal to the size of [Pure concentrations] refined during the ALS step. This matrix is created automatically (from Fastosh v1.0.6), and can be easily edited (Figure 79 A). In the original toolbox, it has to be manually created using Matlab. The “concentration csel matrix” contains either real numbers (i.e. fixed values during the ALS step) or “NaN”, i.e. “Not A Number” values (i.e. floated values during the ALS step).

Similarly, if the Equality Constraint is selected in the window dedicated to setting the constraints to [Pure spectra], the Jaumot *et al.* toolbox must be feeded with a “spectra csel matrix”.

This matrix is also created automatically (from Fastosh v1.0.6), and can be easily edited (Figure 79 B). In the original toolbox, it has to be manually created using Matlab. The size of “spectra csel matrix” is equal to the size of [Pure spectra] refined during the ALS step. The “spectra csel matrix” contains either real numbers (i.e. fixed values during the ALS step) or “NaN” values (i.e. floated values during the ALS step).

#### 6.4.4.5.2 Application of Equality Constraint to [Pure concentrations]

Let's consider a dataset corresponding to a reaction followed *in-situ* at the beamline and featuring three principal components. The first and last scans of the dataset entirely correspond to the reactant (first principal component) & product (third principal component) of the reaction, respectively. The second principal component corresponds to an intermediate species that temporally forms during the *in-situ* reaction.

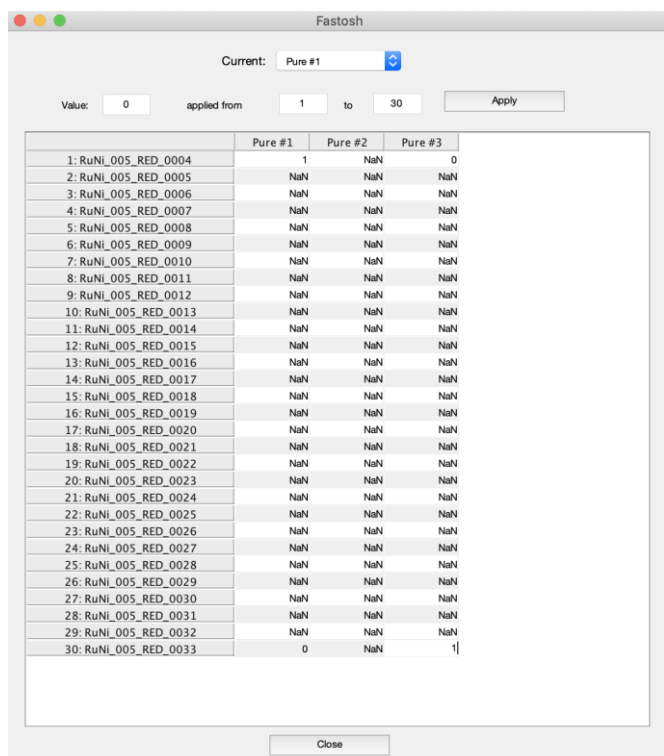
The coefficients of the first and last principal components can be then fixed to 1-0 and 0-1, for the first and last spectrum of the dataset, respectively. To do that, follow these steps:

- In the window dedicated to setting the constraints to [Pure concentrations], select “Equality Constraint”.
- Still in this window, press on “Modify concentration csel matrix”. A window appears (Figure 79 A).
- The coefficients of the first and third principal components present in the first sample of the dataset (“RuNi\_005\_RED\_004”) are modified, from NaN & NaN to 1 & 0, respectively. To do that, directly edit the values in the table. Similarly, the coefficients of the first and third principal components present in the last sample of the dataset (“RuNi\_005\_RED\_033”) are modified, from NaN & NaN to 0 & 1, respectively (Figure 79 A).  
Once done, close the window.
- In the field “Select csel matrix:”, scroll down the menu “select a variable from the WS” and choose “conc\_csel\_matrix”
- In the field “Constraints are:”, scroll down the menu “select...” and choose “equal than”

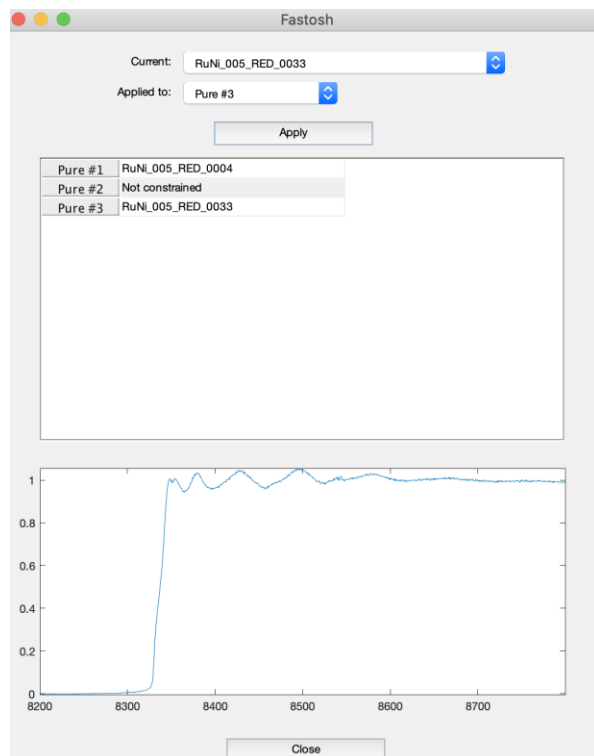
#### 6.4.4.5.3 Application of Equality Constraint to [Pure spectra]

Let's consider the same dataset described in the previous section. The spectra of the first and third principal components can be fixed during the ALS step and equal to the first and last spectra of the dataset. To do that, follow these steps:

- In the window dedicated to setting the constraints to [Pure spectra], select “Equality Constraint”.
- Still in this window, press on “Modify spectra csel matrix”. A window appears (Figure 79 B).
- The first spectrum of the dataset (“RuNi\_005\_RED\_004”) is selected and applied to “Pure #1”. The last spectrum of the dataset (“RuNi\_005\_RED\_033”) is selected and applied to “Pure #3” (Figure 79 B).  
Once done, close the window.
- In the field “Select csel matrix:”, scroll down the menu “select a variable from the WS” and choose “spectra\_csel\_matrix”
- In the field “Constraints are:”, scroll down the menu “select...” and choose “equal than”



A)



B)

Figure 79- Constraining the A) concentration csel matrix or B) spectra csel matrix

## 6.4.5 Advanced toolbox fonctionnalités

From the version 1.0.6 of Fastosh, the advanced fonctionnalités of the toolbox, e.g. “Correlation”, “Multiway”, and “Kinetic HM”, as detailed in the Jaumot et al. 2015 manuscript, are unlocked and usable in the program. The correlation module has been slightly customized: in this module (Figure 80, left), press the red button “Edit concentration matrix” to open a window (Figure 80, right) where the concentration matrix has already been created and can be easily edited - in the original toolbox, this matrix has to be manually created using Matlab). In this window, all values are set by default to “NaN” (= concentrations not constrained).

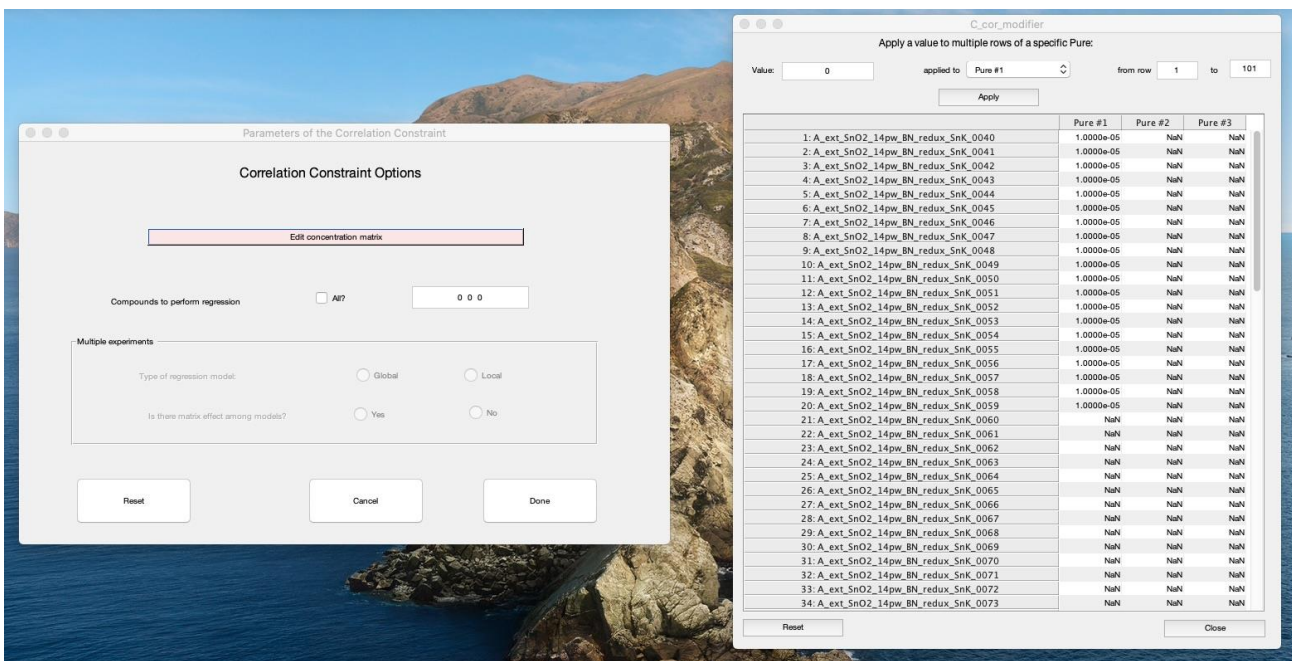


Figure 80 - Customized “Correlation matrix” tool of Jaumot et al. Toolbox.

## 6.4.6 Fitting

### 6.4.6.1 ALS optimization: general parameters & window

Before starting the ALS optimization, the number of iterations and convergence criterion of the fit procedure can be modified from their default values, as mentioned below.

#### 6.4.6.1.1 Number of iterations

The maximum number of iterations allowed during the ALS optimization is set to 50. To ensure that the fitting procedure does not stop only because the number of iterations exceeded this maximum, this parameter can be systematically set to a very high value, e.g. 500.

#### 6.4.6.1.2 Convergence Criterion

At the end of each MCR-ALS fitting loop, the residual matrix is calculated:

$$[\text{Residual}] = [D] - ([\text{Pure Spectra}] * [\text{Pure Coefficients}])$$

where [D] is the original data matrix, and [Pure Coefficients] & [Pure Spectra] are the new matrices refined during each individual loop. The standard deviation “ $\sigma$ ” of [Residual] is then obtained:

$$\sigma = \sqrt{\frac{\sum_{ij} \text{Residual}_{ij}^2}{n * m}}$$

where “Residual<sub>ij</sub>” is the value in [Residual] at row position “i” and column position “j”, and n & m are the total numbers of rows & columns in the data matrix, respectively. Convergence is achieved when in two consecutive iterative cycles, the relative difference between the  $\sigma$  values obtained at the end of the two loops is less than the convergence criterion, in %. Once the actual ALS fitting procedure has started,  $\sigma$  obtained at the end of each iteration is shown as “Std dev. of residuals vs. exp data” in the ALS optimization window.

The default value of the divergence criterion is set to 0.1 %. This value may be modified depending on the stage of the optimization. For instance, a higher value can be used (e.g. 1 %) at the first attempt for exploratory purposes. Once a good model has been found, lower values can be attempted to check whether there is any appreciable improvement in the results [4]. In contrast, the value of the convergence criterion may be increased if the fitting procedure fails to reach convergence after 20 iterations with no result improvements between each loop (in that case the ALS Optimization window returns the message: “Fit not improving for 20 times consecutively (divergence?), stop! ”).

#### 6.4.6.1.3 ALS optimization window

Once the fitting procedure has begun, the data of [Pure Spectra] and [Pure Coefficients], being progressively & alternatively refined at each fit iteration, is shown in two figures in the upper part of the ALS optimization window. This window also shows at each iteration the explained data variance ( $R^2$ ) and the lack of data fit to assess the dissimilarity among the experimental data matrix [D] and the data modeled by MCR-ALS. The expressions of  $R^2$  and lack of fit are:

$$R^2 = 100 \sqrt{\frac{\sum_{ij} D_{ij}^2 - \sum_{ij} \text{Residual}_{ij}^2}{\sum_{ij} D_{ij}^2}} \quad \text{Lack of fit (\%)} = 100 \sqrt{\frac{\sum_{ij} \text{Residual}_{ij}^2}{\sum_{ij} D_{ij}^2}}$$

where “D<sub>ij</sub>” is the value in [D] at row position “i” and column position “j”, and “Residual<sub>ij</sub>” is the value in [Residual] at row position “i” and column position “j”.

Two different lack of fit values are reported at each iteration in the ALS optimization window. They are calculated using two different types of data matrix [D]: “lack of fit (PCA)” is calculated using the PCA-reproduced data matrix using the same number of components as in the MCR-ALS

model, and “lack of fit (exp)” is calculated using the raw experimental data matrix. These values can be employed to assess whether experimental data were well fitted and also to evaluate whether the ALS optimization fit results are similar to those obtained by PCA [4].

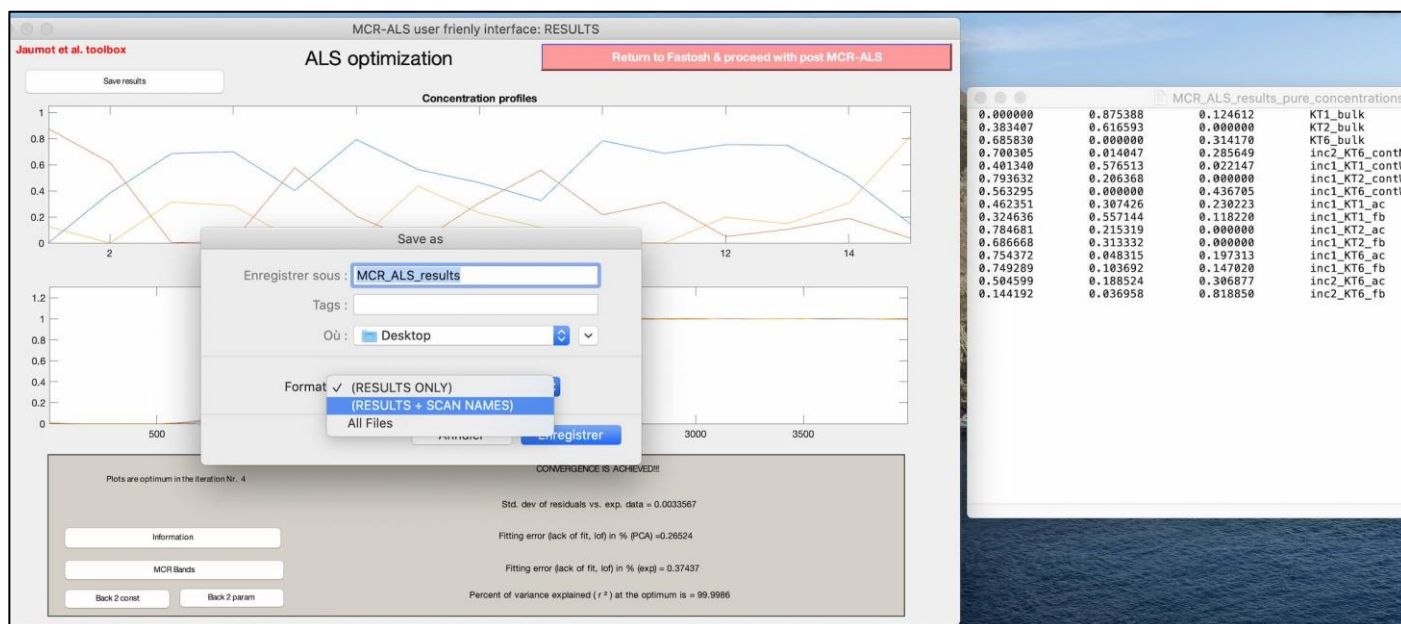


Figure 81 - ALS optimization window at the end of the fitting procedure (Jaumot *et al.* toolbox)

## 6.4.6.2 Handling MCR-ALS results

### 6.4.6.2.1 Saving as a text file

Once the fitting procedure is completed, return to Fastosh main window by pressing “Return to Fastosh & proceed with post MCR-ALS”. This button closes the toolbox and makes the main MCR-ALS of Fastosh reappears on the screen (Figure 81). Once back in the MCR-ALS window, the “Save text files of [Spectra] and [Coef]” allows to save the MCR-ALS results in separate text files.

The first text file “Pure\_Spectra.txt” features, in the first column, the energy (XANES) or  $k$  (EXAFS) values of the all pure spectra. All remaining columns of this text file correspond to the absorption data of the pure spectra.

The coefficients of the pure spectra in each experimental spectrum of the data matrix are saved in the second text file “Pure\_Concentration.txt”. The columns and rows correspond to the pure spectra and experimental spectra of the data matrix, respectively. By default, the text files only feature the values of [Pure Spectra] or [Pure Coefficients]. To add to “Pure\_Concentration.txt” the name of the experimental spectrum corresponding to each row as shown in Figure 81, scroll down the menu “Format” in the window that appears after pressing the button “Save results” and select the option: “Results + scan names” (Figure 81).

### 6.4.6.2.2 Transferring pure spectra to main GUI

The pure XANES/ $\mu$  or  $\chi$  spectra obtained by MCR-ALS can be transferred to the main GUI via “Transfer pure spectra to Main GUI” in the post MCR-ALS part of the main MCR-ALS window of Fastosh.

The pure spectra obtained by MCR-ALS are actually normalized  $\mu$  spectra given that the XANES employed in MCR-ALS are normalized. At this point, the transferred pure spectra can be plotted or saved from the Main GUI

## 6.4.7 Displaying maps of pure species, for datasets acquired in 2D

### 6.4.7.1 Displaying the GUI

If the dataset processed in the Jaumot et al. toolbox was acquired in 2D, a window called “Image Distribution Maps and Spectra” built-in the Jaumot-et-al toolbox, allows to display the map of each pure species, once convergence is achieved. To display this window:

- In the “ALS optimization” window, once convergence is achieved, click on the “Information” button (see 1 in Figure 82). This opens the “optimization information” window.
- In the “Optimization Information” window, click on “Image”. (see 2 in Figure 82). This opens the “Image scores refolding” window
- In the “Image scores refolding” window, one must provide the correct number of pixels in the vertical and horizontal directions of the map, and the number of maps processed. Then, click on “Plot”. (see 3 in Figure 82). This opens up the “Image Distribution Maps and Spectra”.

### 6.4.7.2 Added features for displaying 2D maps, non available in the original toolbox

Some fonctionnalités were added to the window “Image Distribution Maps and Spectra”, notably the possibility of deactivating the “homing” on each axis, and the possibility to create composite map of all pure species.

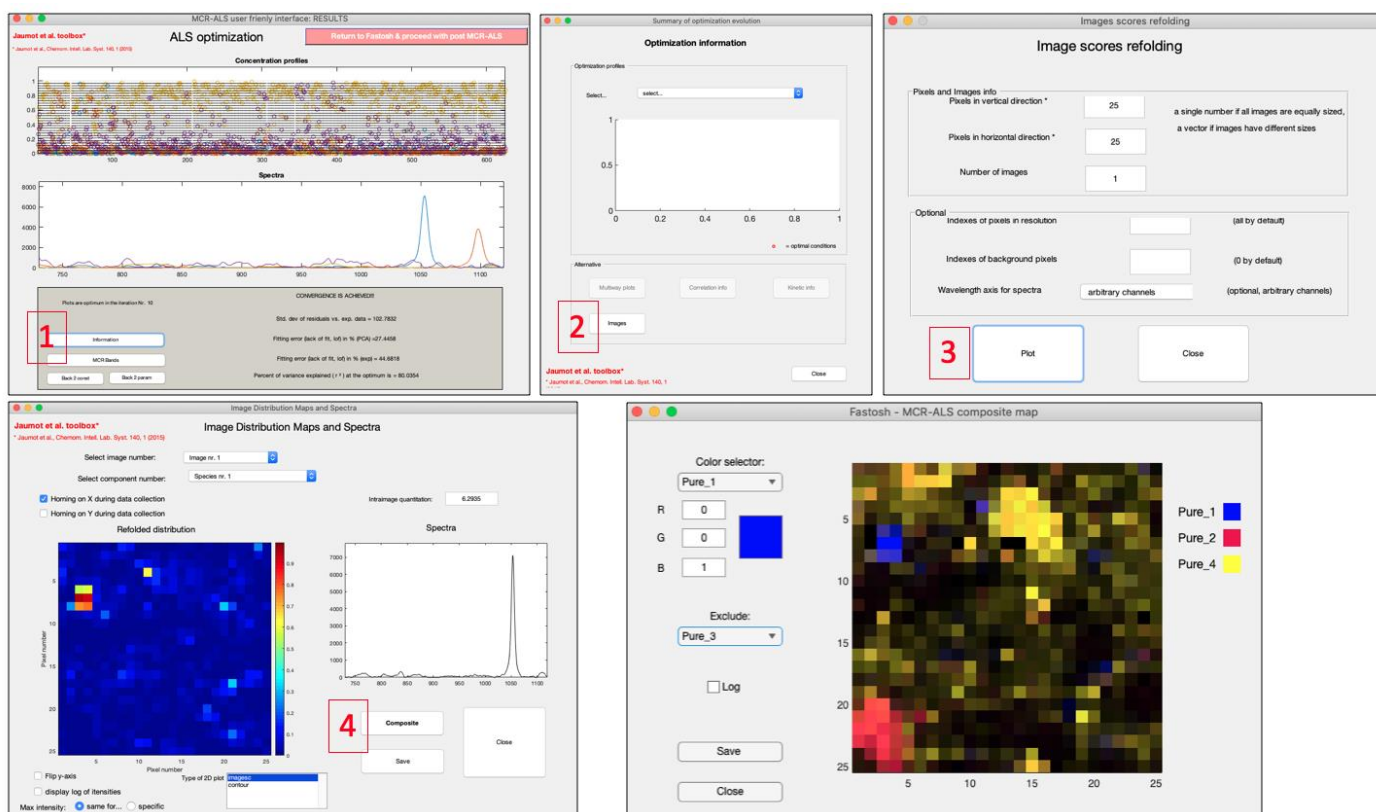


Figure 82 – Displaying the composite map of all pure species present in a dataset acquired in 2D

#### 6.4.7.2.1 Homing on each axis during data acquisition

Suppose that the “Homing on Y during data collection” is checked. This means that when the recording of given row of the image was completed, the data acquisition system systematically

returned to the first column of the image to start recording a new row, similarly to a typewriter homing motion. The original toolbox strictly expected this data acquisition behavior. In contrast, if homing is not checked, it is assumed that the recording of a new row of the image alternatively started at the first or last column of the map, similarly to an Alpine Skier going down the hill in zig-zag motion. This is typically the behavior that a fly-scan acquisition mode employs.

#### 6.4.7.2.2 Composite map of all pure species

The original Jaumot et al. toolbox could only display the map of one specific pure species at a time. An added function allows to display the composite map of all pure species. To display it, click on the button “Composite” available in the window “Image Distribution Maps and Spectra” (see 4 in Figure 82). The composite map appears in a new window, where it is possible to modify the color of each pure species. It is also possible to exclude one pure species from the composite map (typically the one corresponding to the background), as shown in Figure 82 (species 3 was excluded from the composite map).

## 6.5 Post MCR-ALS

### 6.5.1 Generalities

#### 6.5.1.1 Principle

The MCR-ALS enables to obtain the spectra corresponding to the sample mixture's principal components, but does not inform on the nature of these spectra. To help identify their nature, they are used, along with all spectra belonging to a user's reference library, to calculate multiple R factors. For each possible combo of pure spectrum and reference spectrum, an R factor is calculated to estimate the mismatch between the two spectra. Given the R expression, the lower the R value, the more similar the two spectra are to each other:

$$R \text{ factor} = \frac{\sum_{p=1}^t [\text{pure spectrum}(p) - \text{reference spectrum}(p)]^2}{\sum_{i=1}^m [\text{pure spectrum}(p)]^2}$$

where “t” is the total number of points in each spectrum, and “p” is a specific data point in each spectrum. Therefore, the reference spectrum that gives the lowest R factor value is, among all reference spectra considered, the one matching the most the pure spectrum.

#### 6.5.1.2 Procedure

Simply press the button “Load reference library & calculate R factor” in the post MCR-ALS window: a new window appears featuring a list where all wanted references can be selected. This list corresponds to the same sample list as the one featured in the pre MCR-ALS part but without the samples that were processed by MCR-ALS. The spectra of the selected references are then cropped to the same energy range as the one employed to perform MCR-ALS. Additionally, if the spectra are XANES/mu, all references are interpolated to the same energy array as the one of the data matrix.

### 6.5.2 Amplitude correction factor (EXAFS only)

Suppose that the studied element is entirely present in a sample as one chemical form. The XAFS spectra corresponding to the sample and the reference exactly representing this chemical form should be identical, thus the R factor obtained with these two spectra should be equal to 0. This is only true, however, if the two spectra were collected using the same experimental conditions (e.g. temperature, beamline optics, scan parameters, etc...) and there is also no artefact or systematic error in the whole data set. For example, if the data are XANES/mu spectra, incorrect normalization prior to MCR-ALS can represent a source of systematic error. Additionally, if the data

was acquired in fluorescence mode, over-absorption occurring during data acquisition can be another source of systematic error in XANES or EXAFS spectra.

To compensate for any artificial amplitude difference between two EXAFS spectra used to calculate R, an option enables to iteratively minimize this difference via a least-square fitting approach. The amplitude of one of the two spectra (the reference spectrum) is floated in the fit, i.e. an “amplitude correction factor”, applied to the reference spectrum, is varied. The floated amplitude correction factor is not allowed to be negative so that the EXAFS of the reference cannot be flipped upside down in case the two spectra do not match at all; the lowest value allowed is zero. In this case, the amplitude-modified EXAFS of the reference is all flat and the R factor is equal to 1. This functionality is not available for XANES spectra, and by default not activated for EXAFS spectra: the option “Amplitude correction fitting to minimize R factor” in the post MCR-ALS part of the main window is set to “No” (Figure 83).

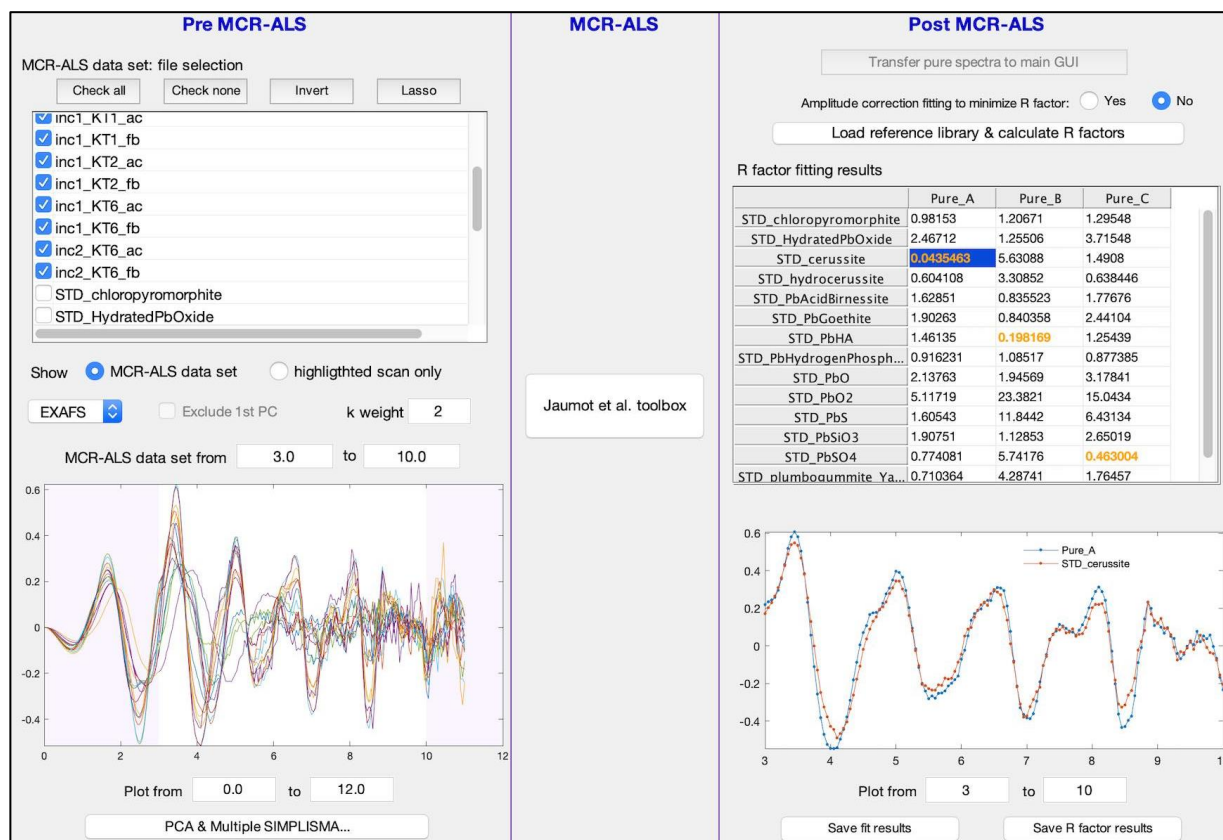


Figure 83 -Post MCR-ALS results

### 6.5.3 Result table

Once all R values have been calculated, the results are shown in a table in the post MCR-ALS part of the main MCR-ALS window (Figure 83), where the columns and rows correspond to the pure spectra and references, respectively. The lowest R factor value obtained for each column/pure spectrum is highlighted in orange color. For example, in Figure 83, the spectrum of reference “cerussite” is, among all references, the most similar to the spectrum of pure A, based on the R factor value. This table is interactive: clicking on any value in this table will plot the data corresponding to the pure and reference spectra employed to calculate the selected R factor value.

To save in a text file these two spectra shown in the figure, click on “save results” button at the bottom of the Post MCR-ALS part of the window. To save in a text file all R factor values shown in the table, click on “Save R factor results” also at the bottom of the window (Figure 83).

## 7 Linear Combination Fitting

The LCF method is applied to normalized  $\mu$  (XANES),  $\chi$  (EXAFS), or normalized  $\mu$  1<sup>st</sup> derivative of experimental and reference XAFS spectra. These spectra must be already imported to the Main GUI and consistently pre-processed before opening the LCF module, as mentioned in Section 4. Once all data involved in LCF has been well preprocessed, open the LCF module via the menu of the Main GUI:

Main GUI Menu > Operations > “Linear Combination Fitting”

### 7.1 Usefulness

Suppose a sample mixture where the chemical forms of the element of interest are the same in each sample of the mixture, but in specific quantities per sample. If the number of these chemical species, their nature, and relative proportions in each sample are unknown, they can be potentially determined by PCA (Section 5.4), Target Transformation (Section 5.5) or MCR-ALS (Section 6), and Linear Combination Fitting (LCF) or MCR-ALS (Section 6), respectively.

If all samples of the mixture contain the same principal components, the spectrum of any sample belonging to the mixture can be reconstructed using the spectra of the principal components. The reconstructed spectrum is equal to the sum of the spectra of the principal components multiplied by the relative proportions of the principal components in the sample. Determining these relative proportions is the purpose of LCF: the fit consists in minimizing, via a least-square approach, the difference between the experimental spectrum and the fit spectrum, which is equal to the sum of all reference spectra multiplied by their spectral fraction. These fractions, or “coefficients”, are the floated parameters in the fit. When the fit reaches convergence, the final coefficient obtained for each reference is proportional to the relative quantity in the sample of the principal component represented by the reference, in mol % of the element of interest present in the sample.

### 7.2 LCF procedure

To perform LCF on an experimental spectrum, firstly select:

- **The experimental spectrum to fit, and reference spectra used to construct the fit spectrum**

The spectrum highlighted in green color in the list (Figure 84) is the experimental spectrum fitted by LCF. All selected spectra in the sample list are the reference spectra used in the fit. However, when opening the LCF module window, no spectrum is selected in the sample list. To select the reference spectra, either manually select them by checking each spectrum, use the “check all” or “lasso” buttons (2.4.1.1), or press “import selection” button (Figure 84). The latter allows to select the same spectra as those selected in the sample list of the Main GUI.

- **Type of data fitted**

The fit is done on the normalized  $\mu$  (XANES),  $\chi$  (EXAFS), or normalized  $\mu$  1<sup>st</sup> derivative of the XAFS spectrum. Choose the spectrum type on top of the window. Make sure that the data has been well preprocessed in the Main GUI before opening the LCF module as mentioned in Section 4.

- **Fitting range**

The fitting range, whose minimum and maximum values are shown below the spectrum type on top of the LCF window, should be also specified.

- To fit a XANES or 1<sup>st</sup> derivative spectrum, the fitting range can extend much beyond the XANES region of the spectrum. It should at least contain the XANES part, a few dozens of e.V. below and above E<sub>0</sub> value to include the main oscillations around the white line of the spectrum, where the principal spectral differences are likely to occur between spectra.
- To fit an EXAFS spectrum, the fitting range minimum should not be too low (e.g. < 2 /Å) notably since the lower part of the  $\chi$  is dominated by multiple scatterings that are not

representative of the EXAFS signal [9]. The fitting range maximum should allow to include as much signal as possible. However, if the last part of the experimental EXAFS spectrum is mostly noise, it should not be included in the fit.

- **Select fitting options**

The fitting options are detailed in the next section (Section 7.3).

- **Start fit**

To start the fit, press the “fit” button in the middle of the LCF window (Figure 84). If the data are XANES or 1st derivative spectra, all reference spectra used in the fit are interpolated based on the energy grid of the experimental spectrum, before starting the fit. The fit minimization is done using a Levenberg-Marquardt least-square algorithm with function and optimality tolerances both set to  $10^{-4}$ .

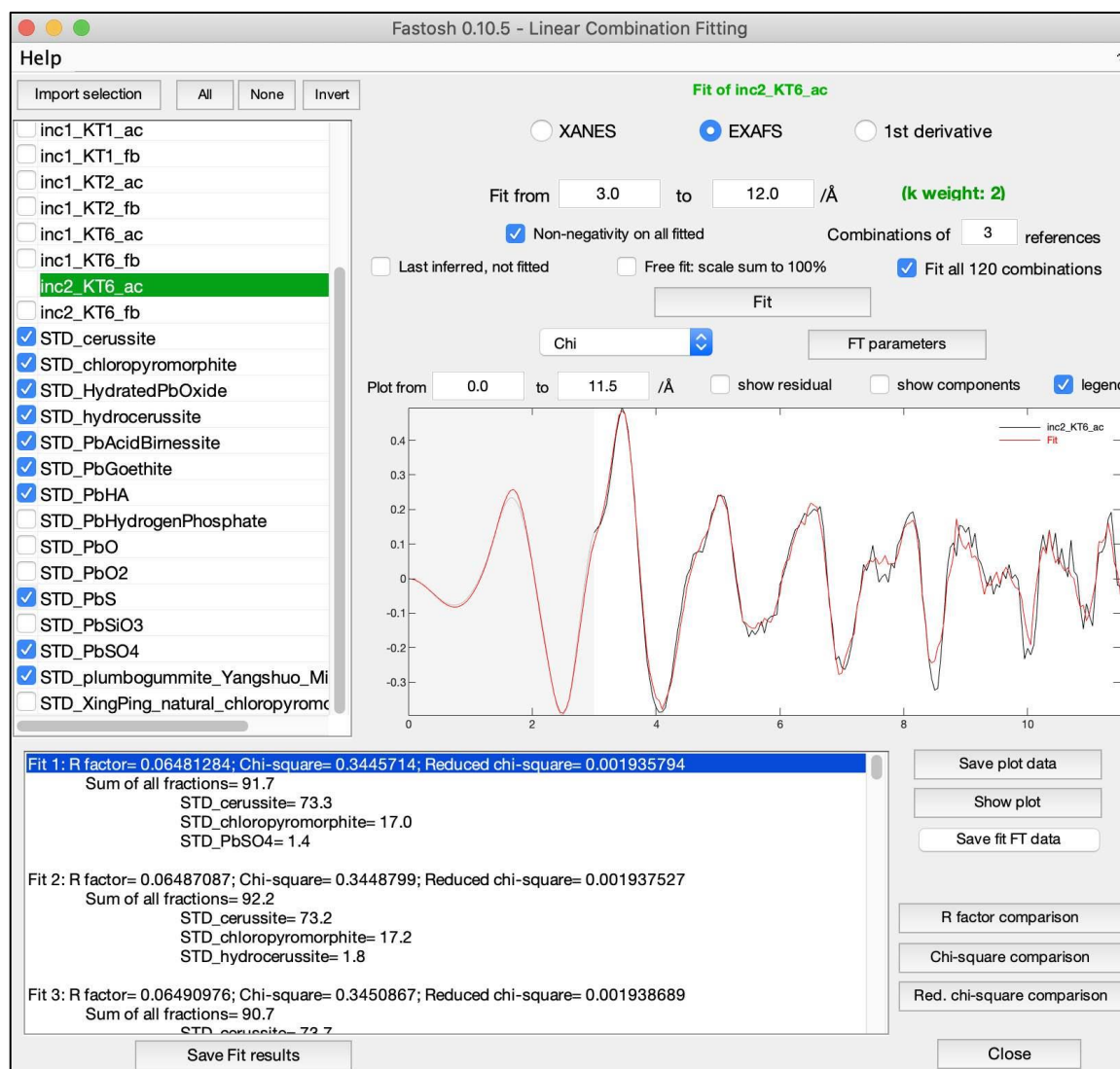


Figure 84 - Linear Combination Fitting window

## 7.3 Fitting options

### 7.3.1 Fit constrains

#### 7.3.1.1 Non-negativity on all floated

This option forces the reference spectra coefficients floated in the fit to be always positive. As explicitly stated, this constraint only applies to *all floated* coefficients. Therefore, if the option

“Last inferred, not floated” (Section 7.3.1.2) is activated, the coefficient of the last reference spectrum may be negative even if the Non-negativity option is activated, since it is not floated during the fit.

It is reasonable to have this option always activated.

#### 7.3.1.2 Last inferred, not floated

This constraint forces the sum of all coefficients to be equal to 100%. To do that, the last coefficient, corresponding to the last reference spectrum involved in the fit, is not floated but inferred using the other coefficients: it is equal to 100 minus the sum of all floated coefficients. This implies that if the sum of all floated coefficients is above 100% (case where the reference spectra employed in the fit do not perfectly reproduce the experimental spectrum), the last coefficient is negative, even if the “Non-negativity on all floated” option (Section 7.3.1.1) is activated.

Not activating this option allows maximum degree of freedom in the fit, which is recommended.

#### 7.3.2 Free fit: scale sum to 100%

This option can be activated only if the “Last inferred, not floated” option is not activated, thus only when all coefficients are floated during the fit (free fit). If all coefficients are floated, their sum is likely to be different than 100% at the end of the fit. Accordingly, if this option is activated, the values of all coefficients obtained at the end of the fit are scaled after the fit is completed so that their sum is equal to 100%.

Given that the uncertainty associated with the LCF method is about 10% [9], this option should be activated only if the sum of all freely fitted coefficients is between 90 and 110%. Therefore, a free fit without activating this option should always be done as a first step to make sure that the scaling is applicable.

#### 7.3.3 Multiple fits

This option enables to specify the number of reference spectra employed in the fit in the field: “*Combination of ... references*”. If this number is lower than the number of selected reference spectra in the list, multiple fits will be performed. Each fit will employ a unique combination of reference spectra.

Suppose a sample that is part of a mixture, where the main chemical forms of the element of interest in the mixture represent the mixture’s principal components (i.e. these species are present in each sample of the mixture). The XAFS spectrum corresponding to the sample may be reconstructed by LCF using the reference spectra corresponding to the principal components. As previously mentioned, PCA and Target Transformation can be employed before performing LCF to determine the number and nature of principal components of the mixture, respectively. Therefore, the number of spectra employed in the LCF fit can be set to the number of principal components inferred from PCA. Similarly, the reference spectra to use in the LCF fit can be selected based on the Target Transformation results.

For example, PCA analyses suggest that there are 3 principal components present in the mixture, and Target Transformation results indicate that 7 reference spectra could potentially represent one of these 3 principal components. Therefore, all these 7 references can be selected in the sample list of the LCF window, and the number of reference spectra employed in the fit can be set to 3. The program will then perform 35 fits. Each fit will consist in a unique combination of 3 references among the 7 references selected in the list.

### 7.4 Results

#### 7.4.1 Goodness of fit: R Factor, chi-square, & red. chi-square

##### 7.4.1.1 Definition

The R factor estimates the discrepancy between the experimental and reconstructed spectra once the fit is completed:

$$(14) \quad R \text{ factor} = \frac{\sum_{p=1}^t [\text{data}(p) - \text{fit}(p)]^2}{\sum_{i=1}^m [\text{data}(p)]^2}$$

where “data” is the experimental spectrum, “fit” is the spectrum reconstructed by LCF, “t” is the total number of points in each spectrum, and “p” is a specific data point in each spectrum.

Two additional goodness of fit indicators are also reported,  $\chi^2$ , i.e. “chi square” (not to be confused with the “chi” function that contains the EXAFS), and reduced  $\chi^2$ . Their theoretical expressions are:

$$\chi^2 = \frac{t_{\text{IND}}}{t} \cdot \frac{\sum_{i=1}^{t_{\text{IND}}} [\text{data}(i) - \text{fit}(i)]^2}{\varepsilon(i)^2}$$

$$\text{Reduced } \chi^2 = \frac{1}{d} \cdot \chi^2$$

where “ $t_{\text{IND}}$ ” represents the total number of independent measurements associated with the experimental spectrum, “i” is an independent data point corresponding to a specific independent measurement, “ $\varepsilon_i$ ” is the measured uncertainty at the independent data point “i”, and “d” is the degree of freedom of the fit, which is equal to  $d = t_{\text{IND}} - \text{number of fitted parameters}$ . The number of fitted parameters corresponds to the number of floated coefficients in the fit (i.e. the number of reference spectra employed in LCF). Given the difficulty to determine the number of true independent points and  $\varepsilon$ , it is approximated in this LCF module that  $t_{\text{IND}} = t$  and  $\varepsilon(i)=1$ . Therefore, the reported  $\chi^2$  and reduced  $\chi^2$  are approximated to:

$$\chi^2 = \sum_{i=1}^t (\text{data}(i) - \text{fit}(i))^2$$

$$(15) \quad \text{Reduced } \chi^2 = \frac{1}{(t-S)} \sum_{i=1}^t (\text{data}(i) - \text{fit}(i))^2$$

where “S” is the number of reference spectra employed in the fit.

#### 7.4.1.2 Interpretation of R, $\chi^2$ , and reduced $\chi^2$

While the values of R,  $\chi^2$ , and reduced  $\chi^2$  can be reported among the LCF results, none of them represent an absolute criterion of fit quality. However, the value of reduced  $\chi^2$  can be employed as a criterion to compare the relative quality of two fits that utilize the same fitting range but different types of reference spectra to reconstruct the data. If the reduced  $\chi^2$  of one fit is at least two times smaller than the reduced  $\chi^2$  of the second fit, then the first fit can be considered as significantly better than the second fit -in other words, the latter is significantly worse than the first fit. Otherwise, the qualities of the two fits can be considered as equivalent [9].

#### 7.4.2 Result list

When a fit is completed, the R,  $\chi^2$ , and reduced  $\chi^2$ , the value of each coefficient (in mol % of the element of interest) associated to each reference spectrum, and the sum of all coefficients, are reported in the result list at the bottom of the LCF window (Figure 84). The experimental spectrum and the reconstructed spectrum obtained from the fit are displayed in the middle of the LCF window in black and red colors, respectively (Figure 84).

If multiple fits are done, all fit results are listed in increasing order of R factor values in the result list. The reconstructed fit spectrum corresponding to the first fit reported in the result list, i.e. the one having the lowest R factor value, is displayed by default in the LCF window, in red color (the experimental spectrum is displayed in black color). **The result list is interactive: clicking in the list on the results of any fit displays in the plot the reconstructed spectrum of the corresponding fit.**

### 7.4.3 Red chi square comparison

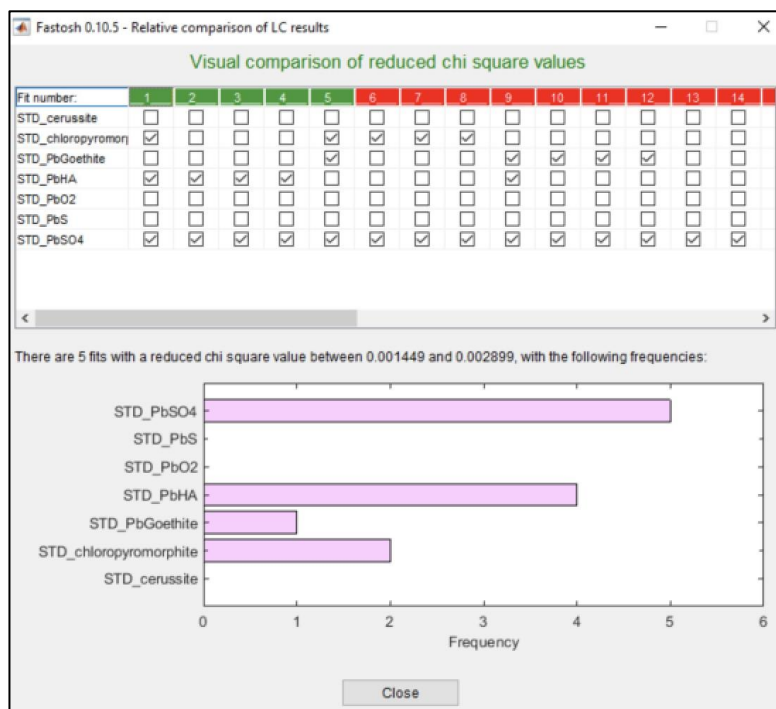


Figure 85 - Result comparator window

A tool enables to quickly visualize how many equivalent best fits were obtained by LCF based on the reduced chi-square results, and what are the most frequent references that contributed to these fits. To open this tool, press “Red. Chi-square comparison” on the right side of the LCF window (Figure 84) once all multiple fits are completed.

In the table on top of the comparator window (Figure 85), each column corresponds to a fit, with increasing reduced chi-square value from left to right. The first column on the left of the table thus corresponds to the fit with the lowest reduced chi square value. The number of this fit is 1, and it is shown in green color in the column header. The numbers of all fits whose reduced chi square values are lower than 2 times the lowest reduced chi square value are also highlighted in green color in the column header. In contrast, the numbers of all fits whose reduced chi square values are at least 2 times higher than the lowest reduced chi square value are highlighted in red color in the column header. Therefore, the quality of the first fit with the lowest reduced chi-square is equivalent to the quality of all other fits marked in green and significantly better than the quality of all fits marked in red.

Each row of the table corresponds to the name of a reference spectrum used in the fit. Any reference spectrum, whose fit contribution is higher than the threshold (set to 10 % by default, but it can be modified on top of the comparator window), is checked in the table, for each fit.

The horizontal bar plot at the bottom of the comparator window (Figure 85) shows the total number of fit contributions above the threshold of each reference spectrum in all fits marked in green. For example, in the example shown in Figure 85, there are 5 fits marked in green. The reference “STD\_PbSO4” gave a contribution above 10% in all these five fits. The reference “STD\_PbHA” contributed to four fits, the reference “STD\_chloropyromorphite” contributed to two fits, “STD\_PbGoethite” contributed to one fit, and all remaining references did not give a contribution above 10 % in any of the five fits.

### 7.4.4 Show residual and/or components

Select “show residual” on top of the plot to display the difference between the experimental and reconstructed spectra. The Residual thus corresponds to the parts of the experimental spectrum that were not reproduced by the fit. A perfect fit should have a residual equal to 0 throughout the fitted range.

Select “components” on top of the plot to display the contribution of each reference spectrum to the reconstructed spectrum. This contribution corresponds to the reference spectrum multiplied by its associated fit coefficient (Section 7.2).

#### 7.4.5 EXAFS fit: view Fourier transform results

If the fitted spectrum is an EXAFS, the Fourier Transform data of the fit can be displayed. From the menu tab in the middle of the LCF window, choose to plot the magnitude, imaginary part, real part, imaginary part + magnitude, or real part + magnitude of the experimental spectrum and fit spectrum shown in black and red colors, respectively (Figure 86). The Fourier Transform parameters can be modified by pressing the “FT parameters” button.

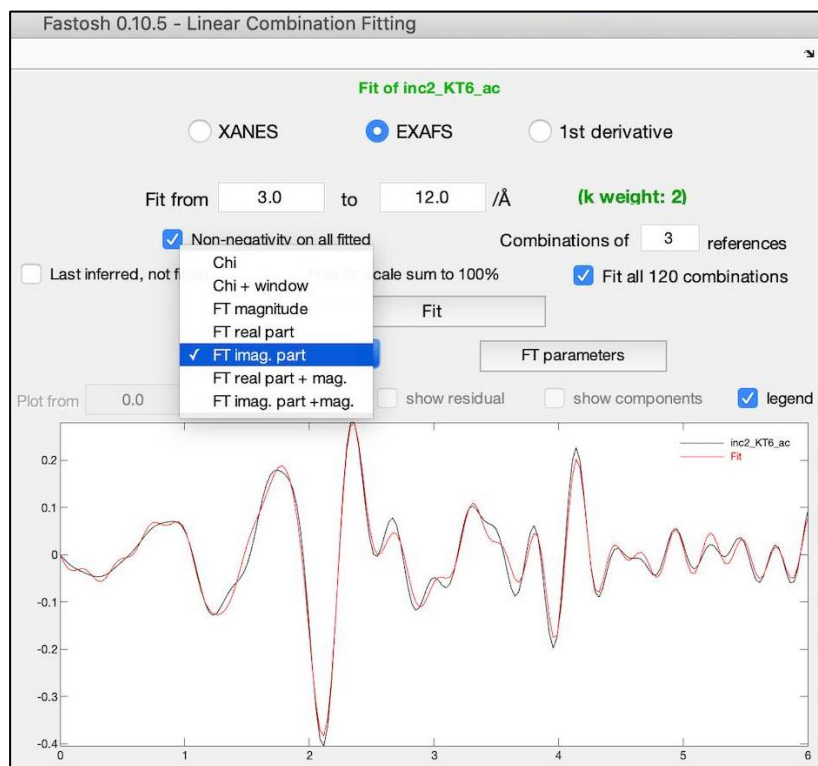


Figure 86 - LCF of EXAFS: options to plot the Fourier Transform data of the fit

#### 7.4.6 Saving results

Different options are available to save the LCF results:

- Press “Save fit results” at the bottom of the Result List in the LCF window to save as a text file the entire content of the Result List.
- Press “Save plot data” on the right side of the LCF window to save as a text file the plot data corresponding to the current fit displayed. This includes the experimental spectrum and its reconstructed spectrum for the current fit, the residual, and reference contributions (i.e. reference spectra used in the fit multiplied by their respective coefficients).
- Press “Show plot” to open up a window showing the plot of the current fit. From there, a picture of the plot can be saved as .PNG, .JPG, .TIF, or .PDF (compatible notably with Adobe or Illustrator) file. Before saving, the plot grid or legend can be added/removed from the plot.

### 7.4.7 Transferring normalized mu or chi to main GUI

The reconstructed normalized mu or chi spectrum can be transferred to the main GUI via “Transfer fit to Main GUI” at the bottom of the LCF window. This can be useful notably to create a figure where multiple spectra are vertically stacked with the functionality “stack plot” available from the Main GUI (Section 2.4.1.2.4).

## 7.5 LCF on multiple samples using a specific set of references

### 7.5.1 Guidelines

Prior to Fastosh v 1.0.6, a specific sample could be fitted using multiple sets of references, all at once. Since Fastosh v 1.0.6, multiple samples can be fitted using a specific set of references, all at once. The results can be visualize using different options: R factor or red. Chi.2 of all fitted samples can be plotted in 2D; residual spectra of all fitted samples can be plotted in 3D.

- Firstly select in the menu >LCF mode> “Fit multiple samples with a single combination of references”
- As usual, specify the nature of the data to fit (norm mu, chi, 1<sup>st</sup> or 2<sup>nd</sup> derivative), and fitting range
- Select in the list the first sample of the dataset, and the last one: all samples are then colored in green color. Note that all samples of the sample dataset to fit simultaneously must be listed next to each other in the list.
- Then, select at least one reference that will be used to reconstruct by LCF each sample of the dataset. *Note that a sample can also be a reference (example: a kinetic dataset can be fitted with its first and last samples, as in the example shown below)*

### 7.5.2 Visualization of results

The results relative to all samples can be notably displayed as:

- R or reduced chi square: these parameters in the LCF module of Fastosh were defined earlier in this chapter. If the reduced chi squares of all samples are shown, a red color will be assigned to all values that are at least two times higher than the lowest reduced chi square obtained in the dataset, as shown in Figure 87 (in this example, all values are in red color, except the last one). Additionally, the name of the sample with the highest/poorest reduced chi square is displayed in the plot as well.
- Residuals, in 2 or 3D: Since Residual corresponds to the difference between the experimental and reconstructed spectra, one can visualize the parts of the experimental spectrum that were not reproduced by the fit. For example, Figure 88 shows the residuals obtained from the dataset displayed in Figure 87. The misfits seem to correspond mostly to the XANES region of the spectra collected in middle of the reaction.

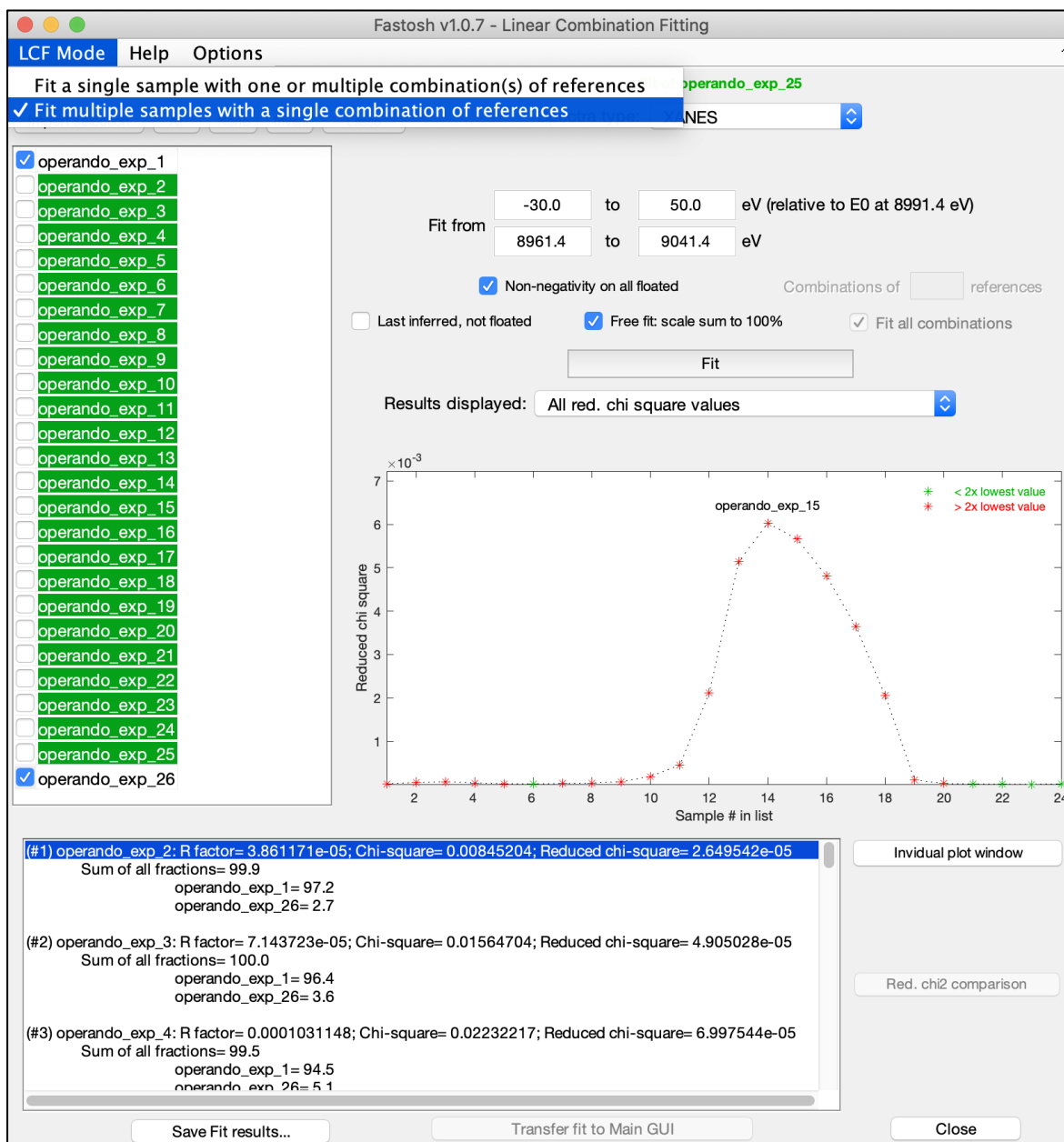


Figure 87 – LCF of multiple samples belonging to a kinetic dataset using a specific set of references : the first and last samples of the data set

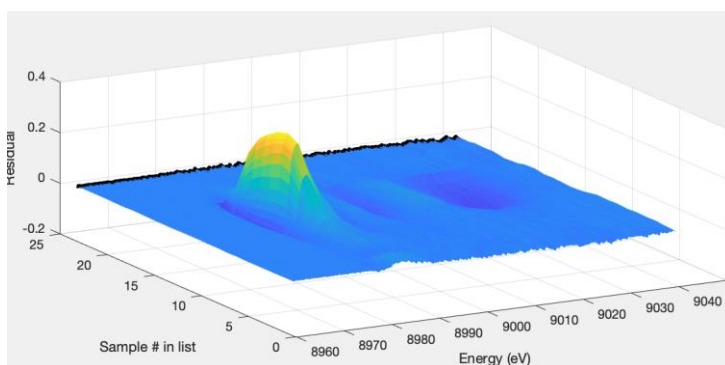


Figure 88 - Results of residuals, shown in 3D (same multiple LCF results as the previous figure)

### 7.5.3 Usefulness of LCF on multiple samples belonging to a kinetic dataset

In the example shown in Figure 87 & Figure 88, a drastic increase in reduced chi square, R factor, and residuals in the middle of the dataset, which corresponds to an experiment that was carried out at the beamline, suggests that the first sample (reactant of the reaction) and last sample (product of the reaction), which were both used as the LCF references, cannot, by themselves, reproduce entirely the dataset, especially in the middle of the reaction. This suggests the presence in significant amounts of an intermediate species: there are more than two principal components present in the dataset. Therefore, performing LCF on multiple samples belonging to a temporally-relevant dataset may be notably useful as a complementary approach to PCA to determine the number of principal components present in a sample mixture corresponding to a kinetic reaction prior to performing Target Transformation or MCR-ALS.

## 8 Quick EXAFS fitting using single scattering paths modelled by FEFF

### 8.1 Presentation

#### 8.1.1 Purpose

This module enables to model an EXAFS spectrum using single scattering paths created by FEFF8L, which is included in Fastosh. This module is accessible via the menu of Fastosh Main GUI:

Main GUI Menu > Operations > “Quick EXAFS fitting using single scattering FEFF models”

This represents a rapid and simple method to model the EXAFS, meant notably to help interpret rapidly a region of interest of a WT map. For example, it may be used to reconstruct a region of a WT map that corresponds to two shells with different neighbor types but similar distances, or to rapidly obtain the real distance of the first shell. It is not meant to reproduce a full EXAFS spectrum, which may involve multiple scattering paths. A future Fastosh module will enable to fully model the EXAFS, including its possible multiple scattering contributions.

#### 8.1.2 Main functionalities

While it already exists a plethora of FEFF-based tools to model an EXAFS spectrum, this Fastosh module proposes a unique approach to perform such task. Its main functionalities are:

- Theoretical EXAFS spectra representing single scattering path models can be created using FEFF8L, and fitted to the experimental EXAFS spectrum, using a GUI very easy and rapid to use
- The minimization of the EXAFS fitting procedure can be done either from the Fourier Transform or Wavelet Transform. Regardless of this choice of fitting minimization approach, any created FEFF model can be plotted either in FT or WT space, in the Fastosh plot window
- Fitting can be done on the sample currently highlighted in the sample list featured in the main GUI, or all samples checked in the list (multiple fits done at once)
- The fitting results are reported in different ways: as a fitting report (traditional way), as a table, summarizing the model parameters, similarly to the common way fitting parameters are reported in peer-reviewed manuscripts, or as a bar plots to visually compare the fitting results obtained for different samples

### 8.2 Single scattering path creation using FEFF8L

#### 8.2.1 Example of a path creation

In the “Quick EXAFS Modelling & Fitting” window, one can rapidly create using FEFF a new theoretical chi corresponding to a specific single scattering path model. Simply specify the atom type of absorber, the atom type of scatterer, the absorber-scatterer distance, and the absorption edge corresponding to the desired single scattering path model: basically fill out the four empty fields colored in red and marked with a “required field” star in the window (Figure 89) . Once this information is given, press the button “Create model shell”. Fastosh then automatically creates a feff input file with the supplied information and run FEFF8L. After completion, the model data is added on top of the sample data when the FT or WT is displayed in the Fastosh plot window.

For example, it was demonstrated in Muñoz *et al.* [1] that the first shell of thorite consists of 8 oxygen around thorium at 2.4 Å. This theoretical shell was then created in the module (Figure

90a). The resulting model data is then displayed along with the sample data, corresponding to the experimental spectrum of Thorite, in the Fastosh plot window when the FT or WT plot option is selected (Figure 90b).

Fastosh - Quick EXAFS Modelling & Fitting

Options

**New shell modelled by FEFF:**

\*required fields

Atom type of absorber \*

Atom type of scatterer \*

Absorber-scatterer distance \* Å

Edge \* Select an edge type...

$\sigma^2$  0.003

SO2 x N 1

$\Delta E0$  0

Create model shell

Delete last shell Delete all

**EXAFS fitting:**

Fitting minimization done from ☒ F.T. of EXAFS ☐ W.T. of EXAFS

Fit current

Fit all marked, with current's model & parameters applied to them

Recall previous Reset

Take this fit as the 1st fit to perform an Hamilton Test

Close

Figure 89 - To create a single scattering model, simply fill out the four fields colored in red in the window

Fastosh - Quick EXAFS Modelling & Fitting

Options

**New shell modelled by FEFF:**

\*required fields

Atom type of absorber Th

Atom type of scatterer \*

Absorber-scatterer distance \* Å

Edge L3

$\sigma^2$  0.003

SO2 x N 1

$\Delta E0$  0

Create model shell

Delete last shell Delete all

**EXAFS fitting:**

Fitting minimization done from ☒ F.T. of EXAFS ☐ W.T. of EXAFS

FT fit range: k space 3.00 - 17.00 /Å, dk=4.00 (kaiser), & R space 1.50 - 2.49 Å

=> Nb of independent points: 9.8 & Degree of freedom: 5.8 (with 4 fit variables)

Fit current: "thorite"

Fit all marked, with current's model & parameters applied to them

Recall previous Reset

Take this fit as the 1st fit to perform an Hamilton Test

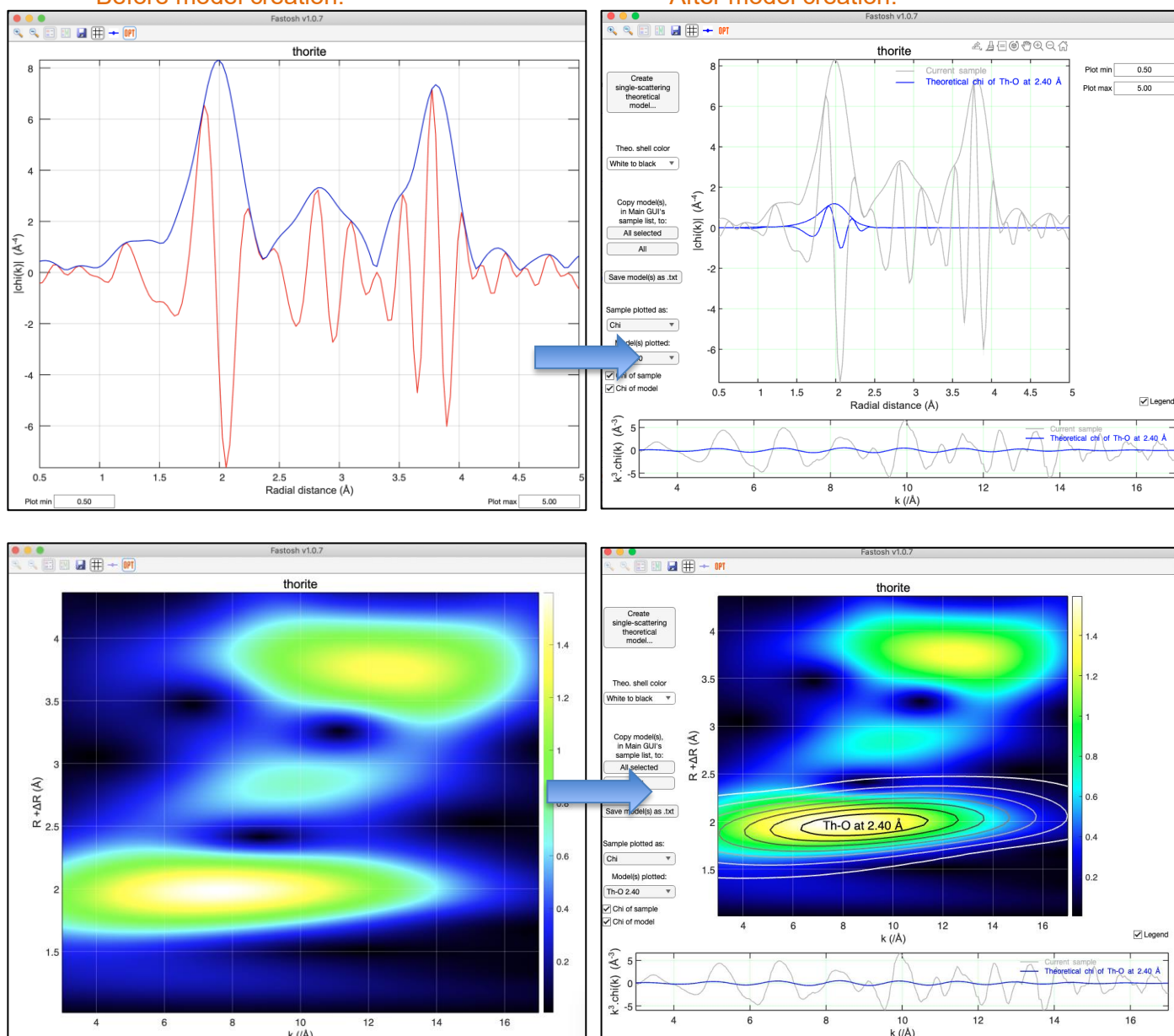
Close

	Abs.	Scat.	Edge	R theo	R + $\Delta R$	$\sigma^2$	$\Delta e0$	SO2 x N
	Float	Float	Float	Float	Float	Float	Float	Float
	Th	O	L3	8.00	8.00	0.0030	0.00	1.000

(a)

Before model creation:

After model creation:



(b)

Figure 90 -Quick EXAFS Modelling & Fitting window after creating a theoretical model representing the first shell of thorite (a) and the Fastosh plot window displaying the FT or WT of the sample thorite before and after this model creation (b)

## 8.2.2 Parameters required to build the theoretical chi

The previous section detailed how a theoretical model representing a single scattering path could be created in Fastosh. This section describes more in details what Fastosh does to obtain the theoretical chi spectrum corresponding to the model.

To build a model chi spectrum, one can rely on the theoretical expression of the EXAFS:

$$(16) \quad \chi(k) = S_0^2 \sum_i N_i \frac{f_i(k)}{k R_i^2} e^{\frac{-2R_i}{\lambda(k)}} e^{-2k^2 \sigma_i^2} \sin(2kR_i + \delta_i(k))$$

This equation features multiple physical parameters, which are described in details in the literature (e.g. [9]). The values of these parameters are either obtained by FEFF or set to a user-defined or default value, as described in the three next sections.

#### 8.2.2.1 Parameters whose values are obtained by FEFF8L

Fastosh employs FEFF8L to obtain three of the parameters involved in the EXAFS equation in order to build the theoretical chi, i.e. the probability of elastic scattering ( $f_i(k)$ ), the mean free path ( $\lambda(k)$ ), and the phase shift correction factor ( $\delta_i(k)$ ):

$$\chi(k) = S_0^2 \sum_i N_i \frac{f_i(k)}{kR_i^2} e^{\frac{-2R_i}{\lambda(k)}} e^{-2k^2\sigma_i^2} \sin(2kR_i + \delta_i(k))$$

The values of these three parameters are specific to the atoms type of the absorber and scatterer, the distance between them, as well as the energy of absorption. Thus, specifying this information is a prerequisite to run FEFF8L.

For the energy of absorption, not only an absorption edge must be provided by the user prior to running FEFF, but also also an additional term is taken into account,  $\Delta E0$ , which is not featured in the EXAFS equation above. This term accounts for the discrepancy between the theoretical and experimental energies corresponding to the photoelectric phenomenon. The value of  $\Delta E0$  is equal to zero by default when a path is created in Fastosh. If multiple paths are created, the same  $\Delta E0$  value is applied for all paths.

#### 8.2.2.2 Parameters set to specific values, other than the amplitude term

The  $R_i$  parameter featured in the EXAFS equation corresponds to the absorber-scatterer distance of the path  $i$ . In Fastosh, the value of this parameter for a path  $i$ , used to build the theoretical chi, is obtained using two additional parameters, namely  $R_{\text{theoretical}}$  and  $\Delta R$ , following the equation:

$$R = R_{\text{theoretical}} + \Delta R$$

The  $R_{\text{theoretical}}$  parameter represents the theoretical distance of the path  $i$  used by FEFF, and its value is user-defined. The  $\Delta R$  accounts for the discrepancy between the theoretical and actual distance (**not to be mistaken with another  $\Delta R$  term that appears for example in the Fastosh plot window when the Fourier Transform is displayed, which accounts for the phase shift in the distance!**). The value of  $\Delta R$  is equal to zero by default when a path is created in Fastosh. Similarly, for the mean square relative displacement term  $\sigma_i^2$  (a.k.a. Debye-Waller factor) featured in the EXAFS equation, Fastosh assigns by default a value equal to 0.003 when creating a path.

#### 8.2.2.3 The amplitude term, User-defined as $S_0^2 \times N$ , $S_0^2$ , or $N$

Not only the “Quick EXAFS Modelling & Fitting” Fastosh module enables to create theoretical single-scattering path models, but it also allows to refine their associated parameters via a fitting approach to minimize as much as possible the difference between theoretical and experimental data (fitting in Section 8.4). During a fit, the values of some EXAFS equation parameters are floated, which may include the Reduction Amplitude Factor ( $S_0^2$ ) and the degeneracy  $N_i$  (i.e. the number of neighboring atoms in an atomic shell  $i$ ). Since these two parameters, which both affect mainly the amplitude of the EXAFS signal, are directly correlated with each other in the EXAFS equation, a single value must be floated for the combination  $S_0^2 \times N$ , referred as the amplitude term, when none of these two parameters are fixed at a given value during the fit. Since this is the case by default when opening for the first time the “Quick EXAFS Modelling & Fitting” window, the model parameter table shows, by default, one column named  $S_0^2 \times N$ , once a path is created (Figure 90a).

Optionally, one can fix N to a user-defined value, so that the amplitude term in the model parameter table (and during fitting, see Section 8.4 ) is only equal to  $S_0^2$ , and vice-versa. To fix any of these two parameters, go to:

Quick EXAFS Modelling & Fitting Menu > Options > “Define amplitude parameter”

A window appears (Figure 91). To define the amplitude term as  $S_0^2$  only, supply a value for N. Conversely, to define the amplitude term as N only, supply a value for  $S_0^2$ . Typically, a reference compound of known structure is firstly employed to obtain the  $S_0^2$  value (case where “SO<sub>2</sub>, with N fixed to:” is selected, with N value typically constrained to the degeneracy of the 1st shell of the reference compound). Then, this  $S_0^2$  value is used afterwards for all remaining samples of a dataset acquired at the same absorption edge [9], so that the amplitude term is only equal to  $N_i$  (case where “N, with SO<sub>2</sub> fixed to:” is selected, with SO<sub>2</sub> value previously constrained using the reference compound).

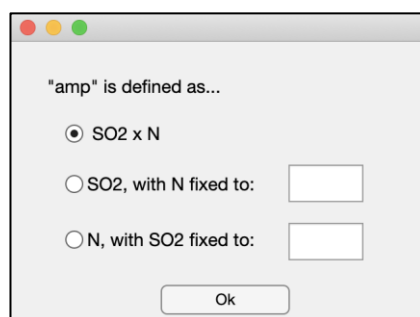


Figure 91 - GUI to define the amplitude term

### 8.3 Manual modification of a model parameter value

Once the model has been created, one can compare the sample and model data using the Fastosh plot window. For example, the WT map of thorite, displayed as a surface in Figure 90b, seems to match well with the WT map corresponding to theoretical Th-O path at 2.4 Å, displayed as contours in Figure 90b. However, one should also compare the chi and FT data (magnitude and real/imaginary part data), not only the WT data. Indeed, when comparing the FT data corresponding to thorite and the theoretical Th-O path at 2.4 Å, one can see that the magnitude of FT spectrum's amplitude of the model shell is much lower compared to the one of thorite, and the frequency is slightly off based on the FT real part. Therefore, while the model parameters values employed to create the Th-O path allowed to obtain a model close to the experimental data, they seem to be not entirely accurate, and should be refined to minimize as much as possible the difference between the model and experimental data.

To modify the model parameters, one method consists in changing manually the value of each model parameter directly from the interactive model parameter table featured in the “Quick EXAFS Modelling & Fitting” (Figure 92). For instance, with the previous Th-O path example, increasing manually from 1 to 8 the amplitude term (i.e. SO<sub>2</sub> x N) in the table will definitely provide a model more faithful to the experimental data (Figure 92). However, with this example, the frequency of the FT real part spectra seems still slightly off between the experimental and model data. Therefore, more parameter modifications should be done to improve this model. Given that more than one model parameter can affect the frequency and amplitude of the EXAFS signal, manually modifying the model parameters one at a time represents a very tedious approach. A more efficient way to refine the model parameters all together is to employ a fitting procedure via least-square minimization, as detailed in the next section.

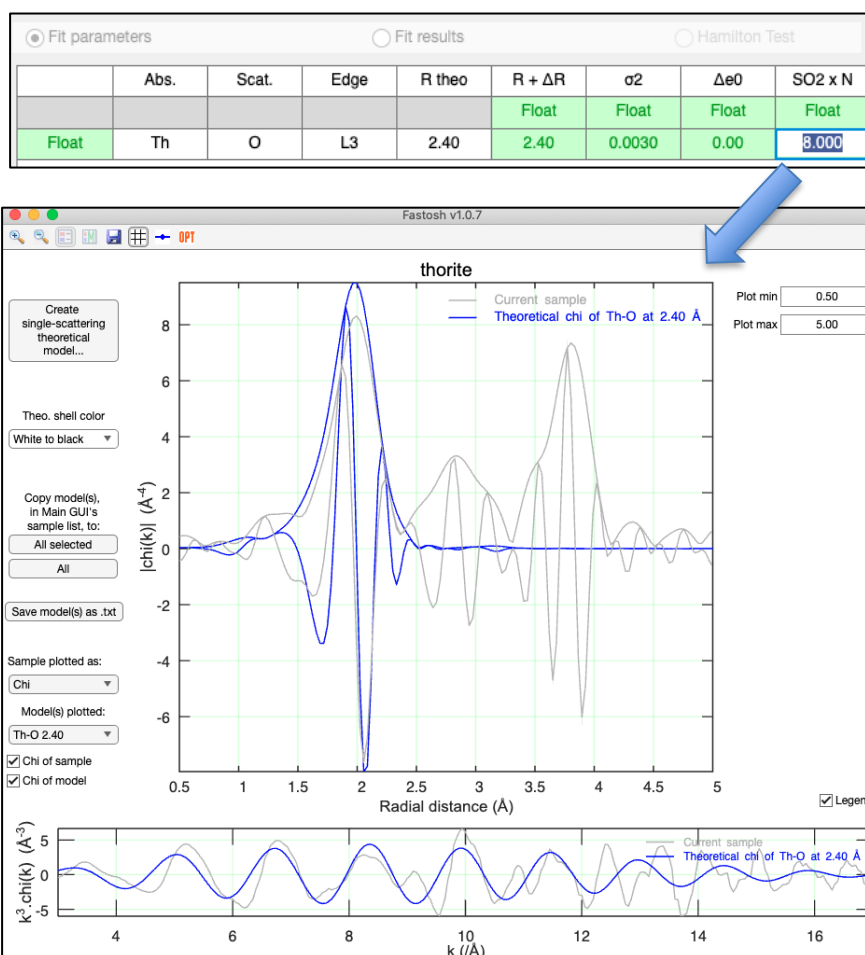


Figure 92 -The model parameter table is interactive: if any of its values is manually modified, the Fastosh plot window will be refreshed accordingly.

## 8.4 Fitting a model EXAFS to the experimental EXAFS

### 8.4.1 Principle

As mentioned in the previous paragraph, employing a fitting procedure via least-square minimization to refine the model parameters represent a better approach than manually modifying each model parameters, one at a time.

Firstly, a model chi spectrum is built based on the EXAFS equation using the theoretical parameters mentioned previously in Section 8.2.2. Then, during the fit, the values of the parameters that are allowed to float are modified at each fit iteration in order to minimize, using a non linear least-square approach (trust-region-reflective algorithm), the difference between the experimental and theoretical data. The nature of the data used during minimization is detailed in the next section.

### 8.4.2 Choice of minimization approach: Fourier Transform (FT) or Wavelet Transform (WT)

The minimization of the fit can be done either using the Fourier transform or the Wavelet Transform data. Select either “FT of EXAFS” or “WT of EXAFS” in the field “Fitting minimization done from” featured in the “Quick EXAFS Modelling & Fitting” window (Figure 90a). For example, if FT is selected, the fit will try to minimize the difference between the FT of the experimental EXAFS and the FT of the theoretical EXAFS spectrum. More precisely, for both the experimental & theoretical data, it is an array consisting of the real part of the FT concatenated with the imaginary part of the FT that is used during minimization. The same approach is used when the minimization is done using the WT maps of the model and experimental EXAFS: for both the experimental &

theoretical data, it is a 2D array consisting of the real part of the WT map concatenated with the imaginary part of the WT map that is used during minimization.

As mentioned in the next section, when the FT minimization is employed, the k range (kmin and kmax) and apodization window (type and dk value) used in the fit, for both the theoretical and experimental spectra, correspond to the values specified for the current sample in the FT section of the Main GUI, while the R range (Rmin and Rmax) used in the fit is specified in the “Quick EXAFS Modelling & Fitting” window.

In contrast, when the WT minimization is employed, the k range (kmin and kmax), R range (Rmin and Rmax), and apodization window (type and dk value) used in the fit, for both the theoretical and experimental spectra, correspond to the values specified for the current sample in the WT section of the Main GUI.

Because the size of a FT spectrum is much smaller than the size of a WT map using equivalent fitting ranges, the FT minimization is much faster than the WT minimization, that is why the FT minimization is the default minimization choice in Fastosh. However, it may provide slightly less accurate results, especially when the WT R resolution is optimized, as the 2D map carries more information than a 1D spectrum. Hence, the WT minimization might be preferred to perform the final fit, while the FT minimization might be preferred before performing the final fit.

### 8.4.3 Fit preparation: checking all parameters involved in the fit

Before starting a fit, or between two consecutive fits, all parameters involved in the fit should be checked or modified if needed to optimize the fitting results. These parameters are reviewed in the next three sections.

#### 8.4.3.1 Parameters exclusive to the experimental spectrum that affect the fit results

As previously mentioned, the experimental EXAFS spectrum is employed during the fitting procedure, along with the theoretical EXAFS spectrum. Hence, the experimental EXAFS spectrum must be well processed to optimize the fitting results. The parameters relative to only the experimental EXAFS spectrum affecting the fitting results are basically  $e_0$ , normalization, and chi extraction parameters relative to the current sample highlighted in the sample list of the Main GUI and whose values are all listed in the Main GUI (Figure 93). If necessary, these values can be modified directly from the Main GUI between two fits. For instance, if the  $\Delta E_0$  was floated during a fit and its value is a bit too high once the fit is completed, the  $e_0$  value of the experimental spectrum may be slightly increased from the Main GUI by a few electron Volts as an attempt to reduce the  $\Delta E_0$  value in the next fit.

**In Fastosh Main GUI:**

**Normalized mu**

EO: 29208.68 **e0 (i.e. beginning of EXAFS)**

☒ Norm ☐ 1st deriv. ☐ 2nd deriv. Post edge Poly. func. deg: 2

Pre edge function from: -208.5 to -70.0 eV

Post-edge function from: 150.0 to 789.9 eV

**Chi**

Rbkg: 1.0 Spline k weight: 2

Spline k range from: 0.00 to 14.40 /Å

Spline E range from: 0.00 to 789.90 eV

Low clamp: 0 Hi clamp: 0

☐ background

Plotting k weight: ☐ 0 ☐ 1 ☒ 2 ☐ 3

Figure 93 – Parameters exclusive to the experimental EXAFS spectrum that affect the fit results

### 8.4.3.2 Parameters applied during a fit to both the experimental and theoretical spectra

#### 8.4.3.2.1 Generalities

The fit parameters applied to both the experimental and theoretical spectra are essentially the fit  $k$  range (fit  $k_{\min}$  and  $k_{\max}$ ), fit  $R$  range, (fit  $R_{\min}$  &  $R_{\max}$ ), chi apodization window (type and  $dk$ ), and optionally wavelet transform type & wavelet parameters. **Their values, and where to find them in Fastosh, depend on the current choice of fitting minimization approach (described previously in Section 8.4.2), as detailed in the next two sections.** Anyway, regardless of the minimization approach chosen (FT or WT minimization), it is always important that:

- The **fit  $k$  range** is maximized as far as the signal of the chi goes, in order to maximize the number of independent points of the fit and consequently the degree of freedom of the fit (more on them in Section 8.4.4). Hence  $k_{\max}$  should be set towards the end of the chi where the EXAFS signal dies off. A value of  $3/\text{\AA}$  for  $k_{\min}$  generally represents a good conservative value to start the EXAFS and the fitting range, too.
- The **fit  $R$  range** must only cover the portion in  $R$  space that is modelled by the theoretical path(s), not below or above it. Since the  $R$  range goes by default from 0 to 6  $\text{\AA}$ , it should be reduced when the current model covers a region in  $R$  space shorter than 0-6  $\text{\AA}$ . For example, if one wants to fit the Th-O path at 2.4  $\text{\AA}$  of thorite created in Figure 90, the  $R$  range should be reduced from 0-6  $\text{\AA}$  (Figure 94a) to 1.4-2.4  $\text{\AA}$  before starting the fit (Figure 94b), in order to obtain the best possible fit results (Figure 94c).

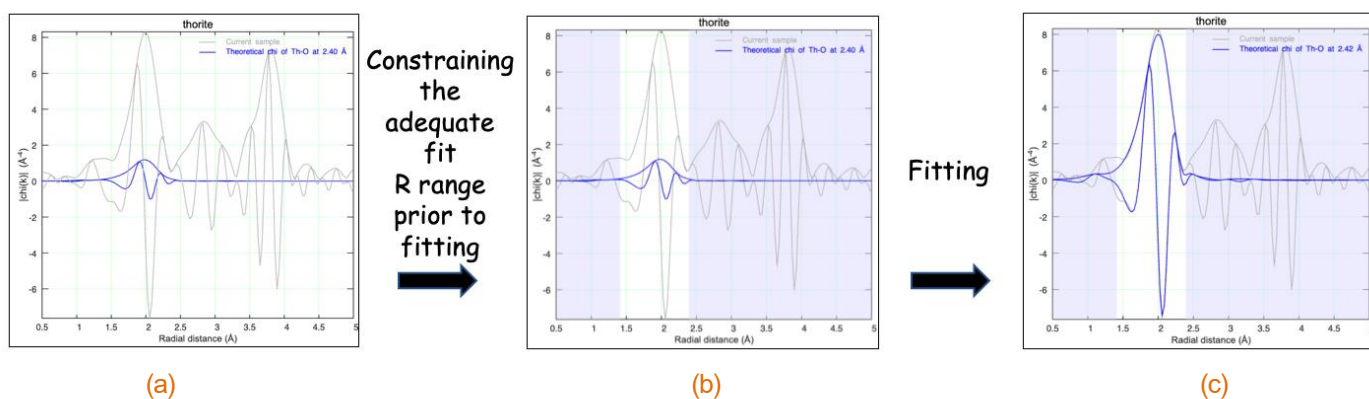


Figure 94 -Optimization of fit  $R$  range prior to fitting the model representing the Th-O path at 2.4  $\text{\AA}$  (theoretical spectrum) over thorite (experimental spectrum) (a & b), and fitting results (c)

#### 8.4.3.2.2 Case where fitting minimization is done using the FT

When the fitting minimization is done using the FT, the fit  $k$  range ( $k_{\min}$  and  $k_{\max}$ ) and apodization window (type and  $dk$  value) applied to both the experimental and theo. spectra during the fit correspond to the values specified in the FT section of the Main GUI, while the fit  $R$  range ( $R_{\min}$  and  $R_{\max}$ ) is specified in the “Quick EXAFS Modelling & Fitting” window (Figure 95).

#### 8.4.3.2.3 Case where fitting minimization is done using the WT

When the WT minimization is employed, the  $k$  range ( $k_{\min}$  and  $k_{\max}$ ),  $R$  range ( $R_{\min}$  and  $R_{\max}$ ), and apodization window (type and  $dk$  value) applied to both the experimental and theo. spectra during the fit correspond to the values specified in the WT section of the Main GUI (Figure 96).

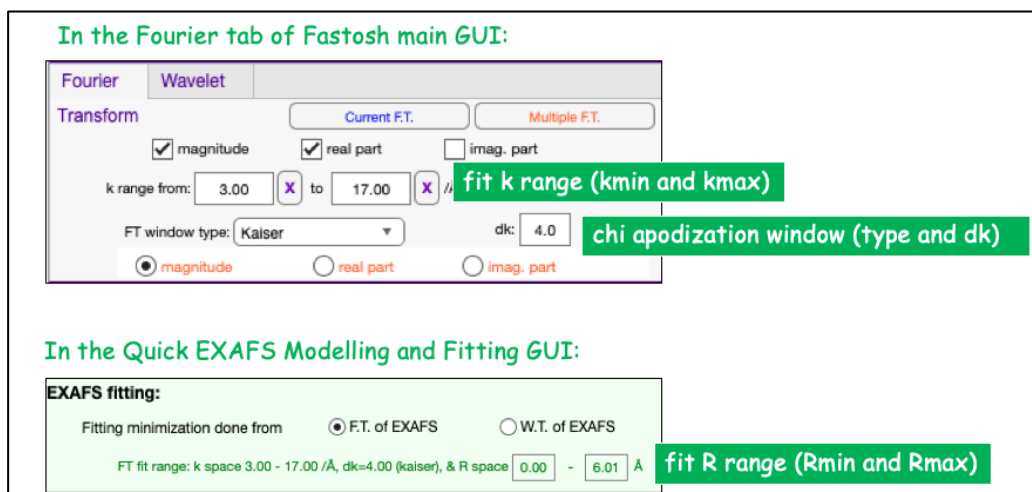


Figure 95 -Fit parameters applied to the experimental & theoretical EXAFS spectra, when FT minimization method is selected

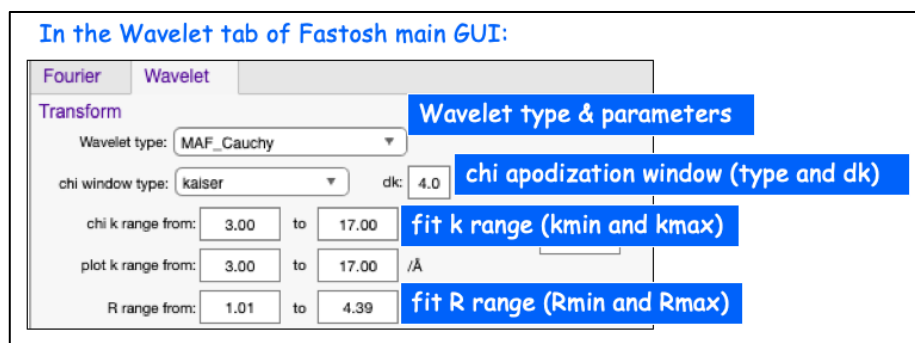


Figure 96 -Fit parameters applied to the experimental & theoretical EXAFS spectra, when WT minimization method is selected

#### 8.4.3.3 Parameters relative to the theoretical spectrum only

The values of the parameters relative to the theoretical spectrum used in the fit were originally all either used-defined or calculated by FEFF at the model creation step. Before starting a fit, one can modify their values from the interactive model parameter table of the “Quick EXAFS Modelling & Fitting” window. One can also specify whether these parameters are floated, fixed, or unused during a fit. In the interactive model parameter table, click on a specific column or row header to modify the fitting status of a specific model parameter or path, respectively. For example, if a column header of the table is set to “Fix”, all values in this column will be set to Fix. Therefore, the specific model parameter corresponding to this column is fixed during a fit. A specific model parameter can be either floated or fixed (Figure 97a).

Similarly, if a row header of the table is set to “Unused”, all values in this row will be set to Unused. In that case, the specific model path corresponding to this row is skipped during a fit. A specific model path can be floated, fixed, or unused (Figure 97b).

	Abs.	Scat.	Edge	R theo	R + ΔR	σ <sup>2</sup>	Δe0	SO2 x N
					Float	Fix	Float	Float
Float	Th	O	L3	2.40	2.41	0.0062	3.17	Fix
Float	Th	Si	L3	3.20	3.20	0.0025	3.17	Float
Float	Th	Th	L3	4.00	3.93	0.0031	3.17	4.790
Float	Th	Si	L3	4.00	3.98	0.0030	3.17	3.411

(a)

	Abs.	Scat.	Edge	R theo	R + ΔR	σ <sup>2</sup>	Δe0	SO2 x N
					Float	Float	Float	Float
Fix	Th	O	L3	2.40	2.41	0.0062	3.17	10.347
Fix	Th	Si	L3	3.20	3.20	0.0025	3.17	2.758
Float	Th	Th	L3	4.00	3.93	0.0031	3.17	4.790
Jnused	Th	Si	L3	4.00	3.98	0.0030	3.17	3.411
Fix								
Float								
Unused								

(b)

Figure 97 - The fitting status of a specific model parameter or path can be modified using the column (a) or row (b) headers of the table, respectively

#### 8.4.4 Condition to allow the fitting functionality

To being able to fit a given experimental EXAFS spectrum, the degree of freedom of the fit must be positive. If the degree of freedom was negative, there would be not enough data point available to perform the fit. The degree of freedom is equal to the number of independent points minus the number of fit variables. The latter corresponds to the total number of model parameters employed in the fit. The number of independent points is expressed as [9]:

$$N_{independent\ points} = \frac{2}{\pi}(\text{fit } k_{\max} - \text{fit } k_{\min})(\text{fit } R_{\max} - \text{fit } R_{\min}) + 1$$

Therefore, the number of independent points depend on both the fit k and R ranges. Once a path is created, Fastosh will calculate the number of independent points, using the current fit k and R ranges (mentioned previously in Section 8.4.3.2), the number of variables, and degree of freedom. Their values are returned in green color in the EXAFS fitting box (Figure 98). If the degree of freedom is positive, the fit buttons will be activated in the EXAFS fitting window (Figure 98a). If it is negative, the fit buttons will be deactivated in the EXAFS fitting window (Figure 98b).

**EXAFS fitting:**

Fitting minimization done from ☒ F.T. of EXAFS ☐ W.T. of EXAFS

FT fit range: k space 3.00 - 17.00 /Å, dk=4.00 (kaiser), & R space 1.01 - 4.51 Å

=> Nb of independent points: 32.2 & Degree of freedom: 28.2 (with 4 fit variables)

Fit current: "kw3\_thorite"

Fit all marked, with current's model & parameters applied to them

(a)

**EXAFS fitting:**

Fitting minimization done from ☒ F.T. of EXAFS ☐ W.T. of EXAFS

FT fit range: k space 3.00 - 9.00 /Å, dk=4.00 (kaiser), & R space 1.26 - 1.99 Å

=> Nb of independent points: 3.8 & Degree of freedom: -0.2 (with 4 fit variables)

Fit current: "kw3\_thorite"

Fit all marked, with current's model & parameters applied to them

(b)

Figure 98 - Case where fitting is allowed (a) and case where fitting is not allowed (b)

## 8.4.5 Fitting report

Once the fit is completed, the “Quick EXAFS Modelling & Fitting” window automatically displays the fit report. It features multiple information related to the fit, including, of course, the fit results, but also notably all the fit parameters employed, date of fit, and time to complete the fit. The first report parts feature the results and also inform on the quality of the fit, i.e. the goodness of fit, the uncertainties associated with each fitted parameter, and fit correlations. These parts are described in details in the next sections. Lastly, a color code, expressing the physical likelihood of the model parameters (described below in Section 8.4.6), is applied to all fit parameter results returned in the fitting report.

### 8.4.5.1 Goodness of fit

The fit report returns the R factor and Reduced Chi Square as indicators of goodness of fit. Their expression were provided in Equations (14) & (15), respectively. As mentioned later in Section 8.4.7, the Reduced Chi Square is mostly employed to compare two fits with each other, and more precisely, two fits having the same fit k range but different fit R ranges. In contrast, the R factor may be employed to assess the goodness of fit of a single fit, as it simply reflects on the spectral misfit between the theoretical spectrum, refined by the fit, and the experimental spectrum. Fastosh applies a color code when reporting the R factor, following the R factor guidelines proposed in the book of Scott Calvin [21]:

R factor	Guidelines	Color code
0 - 0.02	Good enough	green
0.02 - 0.05	Consistent with a broadly correct model, with some details wrong or low quality data	green to blue
0.05 - 0.1	Serious flaws in model or very low data quality	blue to red
>0.10	Model may be fundamentally incorrect	red

### 8.4.5.2 Uncertainty

The error bar associated with each fitted parameter expresses the uncertainty in the fitting results, as standard deviation. It corresponds to the square root of each element featured in the diagonal of the covariance matrix (not returned in the fitting report), which is calculated by Fastosh once the least-square fitting procedure is completed. For now, in the current version of Fastosh (v1.0.7), no scaling is applied to this value.

### 8.4.5.3 Correlation

This expresses how much two fitted parameters are correlated with each other. Its value ranges from 0 (absolutely no correlation between the two parameters) to 1 (the two parameters are perfectly correlated). Ideally, the parameter correlations in a fit should not be too high, otherwise it may indicate that the fit features too many variables for the number of independent points available. Fastosh applies a color code when reporting the correlation values, ranging from green (correlation= 0) to red (correlation = 1). The expression of the correlation (Cor) between the parameters A and B is:

$$\text{Cor}(A, B) = \frac{\text{Covar}(A, B)}{\sqrt{\text{diagCovar}(A) \times \text{diagCovar}(B)}}$$

where Covar and diagCovar correspond to the covariance matrix and its diagonal array, respectively.

## 8.4.6 Color code reflecting the physical likelihood of model parameters

For all results relative to the fitted parameters and their associated error bars, reported in the Fitting Report (8.4.5), Summary Table (8.5.2.18.4.5), or Bar plot (8.5.2.2), a color code is applied to express whether the results makes sense from a physical standpoint. This is done considering these limits for each model parameter, which are used as general guidelines:

Fit parameter	Minimum	Maximum
amplitude	>0	
$\sigma^2$	>0	0.02
$\Delta E0$	-10	10
$\Delta R$	-0.1	0.1

Given these limits, the color code follows these criteria:

Fit parameter value + error margin < parameter limits

→ the fit parameter value & associated error bar are colored in green

Fit parameter value < parameter limits

AND

Fit parameter value + error margin > parameter limits

→ the fit parameter value & associated error bar are colored in orange

Fit parameter value > parameter limits

→ the fit parameter value & associated error bar are colored in red

## 8.4.7 Determining what is the best fit between two fits

### 8.4.7.1 Three possible cases

The nature of the indicator employed to identify the best fit among two fits depends on how fitting was performed [21]. There are three possible cases :

- **If the two fits were performed using the same fit k range, fit R range, but the first fit had more fit variables than the second fit, use the Hamilton Test** (see Section 8.4.7.2). The test will determine whether the first fit is significantly better than the second one. This is employed typically to determine whether adding another model shell in the same fitting region improved significantly the fit. It can be then particularly useful in interpreting a Wavelet Transform map, as demonstrated below in Section , as it allows to determine whether a secondary shell of weak contribution is trully present at a given distance.
- **If the two fits were performed using the same fit k range, but different fit R ranges, use the reduced  $\chi^2$ .** If the first fit has a reduced chi square at least two times lower than the second fit, the first fit can be considered better than the second one [9].
- **If the two fits were performed using different fit k ranges, use the R Factor.** As previously mentioned in Section 8.4.5.1, the R factor can be employed to assess the goodness of fit for a single fit. Therefore, if this indicator is employed to assess the quality of two fits or more, it actually does not provide a relative comparison of all fits but more an individual goodness-of-fit assessment done on all fits.

### 8.4.7.2 Hamilton Test

This test is employed to determine whether one fit is significantly better than another one, with the prerequisite that the two fits have the same fit k range, fit R range, and the number of variables in the first fit must be higher than the one in the second fit. The test consists in calculating the Regularized Lower Incomplete Beta Function (RLIBF), expressed as %, representing a lower bound on the probability of one fit being better than another one purely due to randomness [21].

The RLIBF is calculated following these six steps:

- 1)  $x$  is obtained with  $x = \frac{\text{R factor of 1st fit}}{\text{R factor of 2nd fit}}$
- 2)  $a$  is obtained with  $a = \frac{\text{degree of freedom in 1st fit}}{2}$
- 3)  $b$  is obtained with  $b = \frac{(\text{number of variables in 1st fit} - \text{number of variables in 2nd fit})}{2}$
- 4) Beta function (BF) is calculated following this equation:  
$$\text{BF}(a,b) = \int_0^1 t^{a-1} (1-t)^{b-1} dt$$
- 5) Lower incomplete beta function (LIBF) is calculated following this equation:  
$$\text{LIBF}(x;a,b) = \int_0^x t^{a-1} (1-t)^{b-1} dt$$
- 6) The Regularized lower incomplete beta function (RLIBF) is obtained given:  
$$\text{RLIBF} = \frac{\text{LIBF}(x;a,b)}{\text{BF}(a,b)}$$

### 8.4.8 Example: fitting of thorite EXAFS and Hamilton Test

The EXAFS spectrum of thorite featured in the study of Muñoz *et al.* [1] and an equivalent one, employed multiple times in this User Manual, were shown in Figure 18. According to Muñoz *et al.* [1], the first four shells around thorium in the structure of thorite are:

- Th-O shell at 2.42 Å, with 8 O atoms
- Th-Si shell at 3.16 Å, with 2 Si atoms
- Th-Si and Th-Th shells both at 3.9 Å, with 4 Si and 4 Th atoms, respectively

The goal of this example is to demonstrate that the structure above, including the presence of both Si and Th around 3.9 Å, can be verified via a fitting procedure using the experimental EXAFS spectrum of thorite. Firstly, it is important to mention that the experimental EXAFS of thorite may be modelled in R space from 0 to 4 Å using only single-scattering paths in the “Quick EXAFS Modelling & Fitting” tool. Indeed, according to FEFF8L, single scattering paths mostly contribute to the signal from 0 to 4 Å, except for two multiple scattering paths that contribute to about 21-22 % of the amplitude at about 3.98 Å (Figure 99). The FEFF calculations do confirm the presence of the four first shells around the thorium atom in the thorite structure as mentioned in Muñoz *et al.* [1]. Indeed, the FEFF results indicate that there are a Th-O shell at around 2.48 Å, with 8 O atoms, one Th-Si shell at 3.16 Å, with 2 Si atoms, and a shell with 4 Th / 4 Si at about 3.9 Å (Figure 99).

Path #	Path type	Description	Rel. amp	nlegs	Deg.	Distance
1	Single scattering	Core->O-> Core	100.0	2	4	2.4631
2	Single scattering	Core->O-> Core	92.4	2	4	2.4951
3	Single scattering	Core->Si-> Core	21.7	2	2	3.1635
4	Multiple scattering	Core->O->Si-> Core	4.7	3	8	3.5946
5	Single scattering	Core->Si-> Core	10.7	2	4	3.9056
6	Single scattering	Core->Th-> Core	70.6	2	4	3.9056
7	Multiple scattering	Core->O->Si-> Core	21.7	3	8	3.9497
8	Multiple scattering	Core->O->Si->O-> Core	21.1	4	4	3.9937
9	Single scattering	Core->O-> Core	3.5	2	4	4.2983
10	Single scattering	Core->O-> Core	6.0	2	8	4.4022
11	Single scattering	Core->O-> Core	5.0	2	8	4.5452

Figure 99 -FEFF8L results (calculation done independently of Fastosh v1.0.7) using as input the atomic environment of thorium inside the structure of thorite at the L3 absorption edge.

Therefore, these four theoretical shells were created using the “Quick EXAFS Modelling & Fitting” tool of Fastosh (Figure 100).

	Abs.	Scat.	Edge	R theo	R + ΔR	σ <sup>2</sup>	Δe0	N
					Float	Float	Float	Float
Float	Th	O	L3	2.48	2.48	0.0030	0.00	8.000
Float	Th	Si	L3	3.16	3.16	0.0030	0.00	2.000
Float	Th	Th	L3	3.90	3.90	0.0030	0.00	4.000
Float	Th	Si	L3	3.90	3.90	0.0030	0.00	4.000

Figure 100 – Interactive model parameter table after the four path creations

Then the model was fitted using the EXAFS of thorite as the experimental spectrum. The first fit trial provided satisfactory results, except the  $\Delta R$  of the Th-Si model path at 3.9 Å was a bit too large, i.e.  $\Delta R = 0.09 \text{ Å} \pm 0.002$ . Consequently, prior to the next fit trial, the value of  $R + \Delta R$  for this path was interactively increased to 3.95 Å from the model parameter table (Figure 101). Note that when any  $R + \Delta R$  value is modified in the table, FEFF is recalculating the path parameters using this new distance. Therefore, it takes a little bit more time compared to when the value of another model parameter is modified in the table.

	Abs.	Scat.	Edge	R theo	R + ΔR	σ <sup>2</sup>	Δe0	SO <sub>2</sub> x N
					Float	Float	Float	Float
Float	Th	O	L3	2.48	2.41	0.0069	5.76	11.523
Float	Th	Si	L3	3.16	3.20	0.0027	5.76	2.875
Float	Th	Th	L3	3.90	3.93	0.0033	5.76	5.580
Float	Th	Si	L3	3.90	3.95	0.0017	5.76	3.336

Figure 101 -Modifying the  $R + \Delta R$  value from the Interactive model parameter table

After modifying the  $R + \Delta R$  value, a new fit was performed, which provided satisfactory results. The fitting was then again repeated multiple times until the model parameters reached a local minimum. The final fit results are shown in Figure 102.

As shown in Figure 102, the fit results suggest that there are notably a Th-Th path at 3.92 Å, and near it, a Th-Si path at 4 Å. However, since the Th-Th path at 3.92 Å seems to contribute significantly to the region around 3.5 - 4 Å in both the FT or WT corresponding to thorite (Figure 102), one could challenge whether the Th-Si path is truly present at 4 Å.

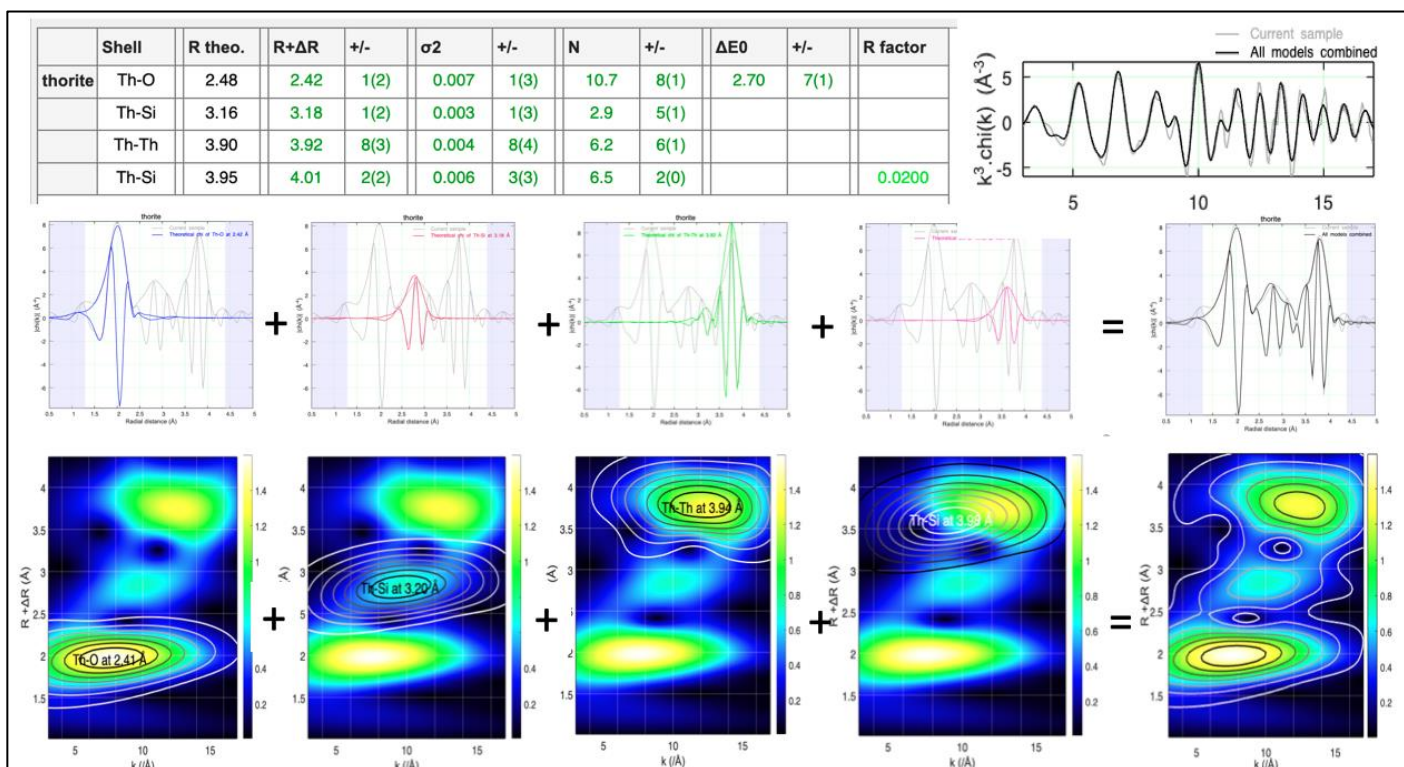


Figure 102 – Fit results of a model involving four single-scattering paths to reproduce the local atomic environment from 0 to 4 Å around thorium in the thorite structure

Moreover, performing an additional fit, using the same model but this time without the Th-Si path at 4 Å (this path is set to “Unused” in the interactive model parameter table featured in the “Quick EXAFS Modelling & Fitting” window) provide results that overall seem to be consistent with a plausible structure (Figure 103).

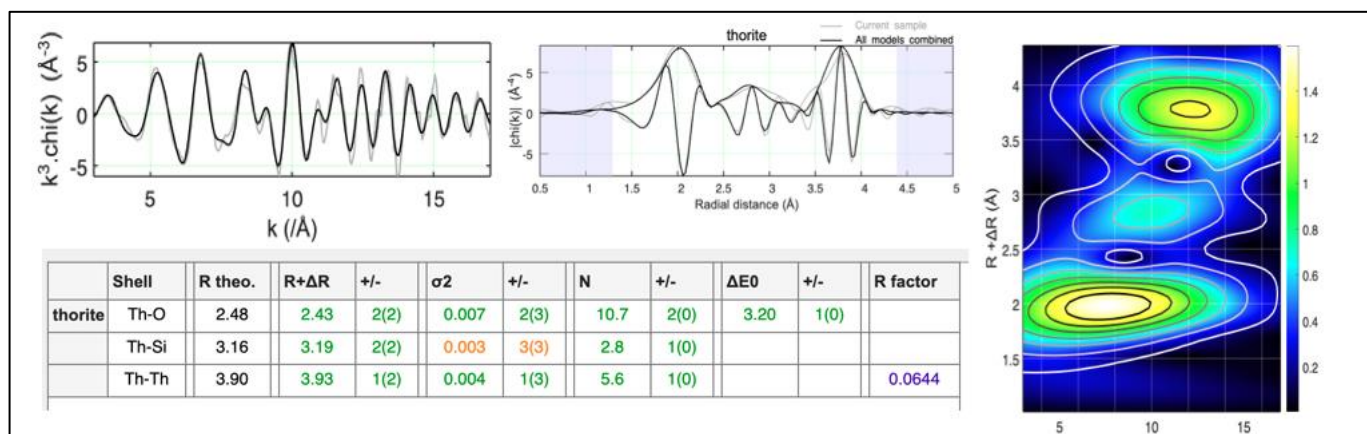


Figure 103 - Fit results of a model identical to the previous one, but without the Th-Si path at 4 Å

Therefore, to determine whether the Th-Si shell at 4 Å is truly present in the structure, one can perform an Hamilton Test. To perform this test in the “Quick EXAFS Modelling & Fitting” Fastosh window, using the above example:

- 1- Press the button “Take this fit as the 1st fit to perform an Hamilton Test”, at the bottom left corner of the “Quick EXAFS Modelling & Fitting” window, once the fit corresponding to the 4-path model is completed
- 2- Set the path whose presence in the structure is challenged to “Unused” in the interactive parameter table, and redo the fitting several times if needed until reaching a local minimum

- 3- Press the button "Take this fit as the 2<sup>nd</sup> fit to perform an Hamilton Test" , at the bottom left corner of the "Quick EXAFS Modelling & Fitting" window, once the fit corresponding to the 3-path model is completed.

The result report of the test is then displayed automatically in the "Quick EXAFS Modelling & Fitting" window. With the example above, the Hamilton test results indicate that the 1<sup>st</sup> fit is significantly better than the 2<sup>nd</sup> fit (Figure 104). This implies that the Th-Si at 4 Å is indeed trully present in the structure, at a similar distance to a Th-Th shell.

**Result summary of the Hamilton Test:**  
The Regularized Lower Incomplete Beta Function is 0.03 %, thus the Hamilton Test indicates that the 1st fit represents an improvement over the 2nd fit at the 99.97 % confidence level.  
Therefore, considering a minimum acceptable confidence level at 95%,  
**=> the first fit is significantly better than the second fit.**

**Result details of the Hamilton test:**  
var\_1: R factor from sample WT map fitting, 1st fit: 0.020015  
var\_2: degree of freedom in 1st fit: 15.629298  
var\_3: number of variable in 1st fit: 13  
var\_4: R factor from sample WT map fitting, 2nd fit: 0.064374  
var\_5: number of variable in 2nd fit: 10  
"x" used in the functions below: var\_1 / var\_4 : 0.310908  
"a" used in the functions below: var\_2 / 2 : 7.814649  
"b" used in the functions below: (var\_3 - var\_5) / 2 : 1.500000  
Beta Function: 0.038743  
Lower Incomplete Beta Function: 0.000012  
Regularized Lower Incomplete Beta Function: 0.000305 (or 0.03 %)

Figure 104 -Result of the Hamilton test corresponding to the example described in the text

## 8.5 Multiple fits at once

### 8.5.1 Principle

This functionality enables to fit all samples that are marked in the sample list of the Main GUI. Once the button "Fit all marked" is pressed, the marked samples are automatically assigned some parameters relative to the current sample before the fitting begins on all marked, as described below.

#### Each fitted sample will be assigned:

- The current sample's theoretical model paths
- The current sample's fit parameters that are applied to both experimental and theoretical spectra during the fit. Those parameters were defined at Section 8.4.3.2. They are essentially the fit k range (kmin and kmax), fit R range, (Rmin & Rmax), apodization window choices (type and dk), and optionnally wavelet parameters when the WT minimization approach is used.

#### Each fitted sample will not be assigned:

- The current sample's fit parameters that are exclusive to the experimental spectrum. Those parameters were defined in Section 8.4.3.1. These are essentially the e0 value, normalization and chi extraction parameters. Indeed, a given set of parameters is not always applicable to multiple samples even when the latter are supposed to be very similar to each other --that's notably the case when the data is noisy. **Nevertheless, the resulting normalization and EXAFS extraction must be consistent when employing the fitting approach on multiple spectra.**

## 8.5.2 Fitting results

### 8.5.2.1 Table summarizing the fitting results

Once the fitting is completed, click on the tab “Table” to display a summary table of the results. The results summarized in the table shown as an example in Figure 105 correspond to ten spectra. They belong to the same dataset and were collected successively during an *operando* experiment conducted at the beamline. One can conveniently notice from the table that a model corresponding to a Cu-O path at 1.96 Å does reproduce well the first shell of all spectra at the beginning of the reaction, but progressively this model is less accurate as the distance of the first shell tends to increase during the reaction (Figure 105). In this table:

- The each error results are systematically displayed with only two numbers (the second one being in parenthesis): the first and second numbers correspond to the first non-zero number of the error bar value and its decimal place, respectively. For instance, 4(3) stands for 0.004.
- All fitted model parameters are displayed following the color code described previously at section 8.4.6.
- The R factor is displayed following the color code described previously at Section 8.4.5.1.

	Shell	R theo.	R+ΔR	+/-	σ <sup>2</sup>	+/-	SO <sub>2</sub> xN	+/-	ΔE0	+/-	R factor
experiment_1	Cu-O	1.96	1.96	3(2)	0.003	3(3)	1.9	3(1)	-0.89	1(0)	0.0392
experiment_2	Cu-O	1.96	1.96	2(2)	0.005	3(3)	1.8	2(1)	0.14	1(0)	0.0224
experiment_3	Cu-O	1.96	1.96	2(2)	0.004	2(3)	1.8	2(1)	-0.66	1(0)	0.0204
experiment_4	Cu-O	1.96	1.96	2(2)	0.006	3(3)	1.8	3(1)	-0.74	2(0)	0.0388
experiment_5	Cu-O	1.96	1.97	2(2)	0.004	3(3)	1.3	2(1)	0.82	1(0)	0.0270
experiment_6	Cu-O	1.96	1.96	3(2)	0.003	3(3)	0.9	2(1)	0.72	2(0)	0.0425
experiment_7	Cu-O	1.96	2.03	5(2)	0.006	8(3)	1.3	6(1)	5.87	3(0)	0.1833
experiment_8	Cu-O	1.96	2.16	3(2)	0.001	4(3)	1.8	5(1)	18.59	3(0)	0.0655
experiment_9	Cu-O	1.96	2.20	2(2)	0.004	3(3)	3.1	6(1)	22.12	2(0)	0.0311
experiment_10	Cu-O	1.96	2.21	2(2)	0.007	4(3)	4.4	9(1)	22.93	2(0)	0.0538

Figure 105 - Fitting result summary table

### 8.5.2.2 Bar plot relative to each model parameter or R factor

Once the fitting is completed, the results relative to a specific fitted model parameter or the R factor can be displayed as a bar plot, using the appropriate tab. This can be useful to visually observe a trend in the fit results between different samples. For example, to take the same example as the one employed in the previous section, one can clearly observe that the the distance in R increases at some point during the reaction (from experiment 7) , and the Cu-O path at 1.96 Å seems no longer valid. The bar plot display the data in color, following a color code:

- All fitted model parameters are displayed following the color code described previously at section 8.4.6.

- The R factor is displayed following the color code described previously at Section 8.4.5.1.

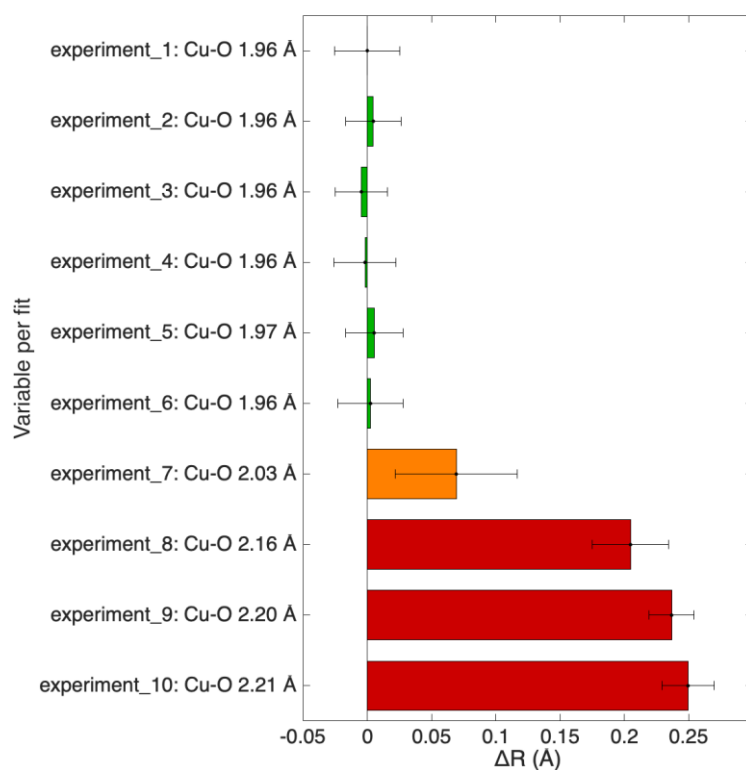


Figure 106 -Bar plot corresponding to the  $\Delta R$  fitting results of 10 spectra belonging to the same dataset collected at the beamline in operando conditions, and fitted simultaneously

## 9 Modules to exploit SAMBA files (SAMBA data only)

### 9.1 HDF Processor (SAMBA data only)

This module of Fastosh can be open via the menu of the Main GUI:

Main GUI Menu > Operations > For SAMBA's files only... > "HDF Processor"

#### 9.1.1 Usefulness

This module has three main purposes:

- To rapidly glimpse at all collected XAFS scans saved in the data folder without importing any data to the program. This can be notably helpful in identifying the iterations that are not exploitable.
- To merge HDF files corresponding to successive iterations. The average file can be open in "MCA Viewer" module of Fastosh for further data analysis.
- To convert an HDF file to a text file. For example, if 60 scan iterations were collected for a given sample, 60 individual HDF files would be created at the beamline. All 60 HDF files can be converted to 60 text files. For each scan iteration, a new fluorescence spectrum can be extracted from a specific set of pixel MCA data recorded by SAMBA's multi pixel fluo detector (i.e. Ge 36 pix or SDD 13 pix detector). For example, specific pixels that were affected by diffraction phenomena during acquisition can be deactivated in order to extract a new, undistorted fluorescence spectrum. Additionnaly, a new ROI can be employed to extract the fluorescence spectrum.

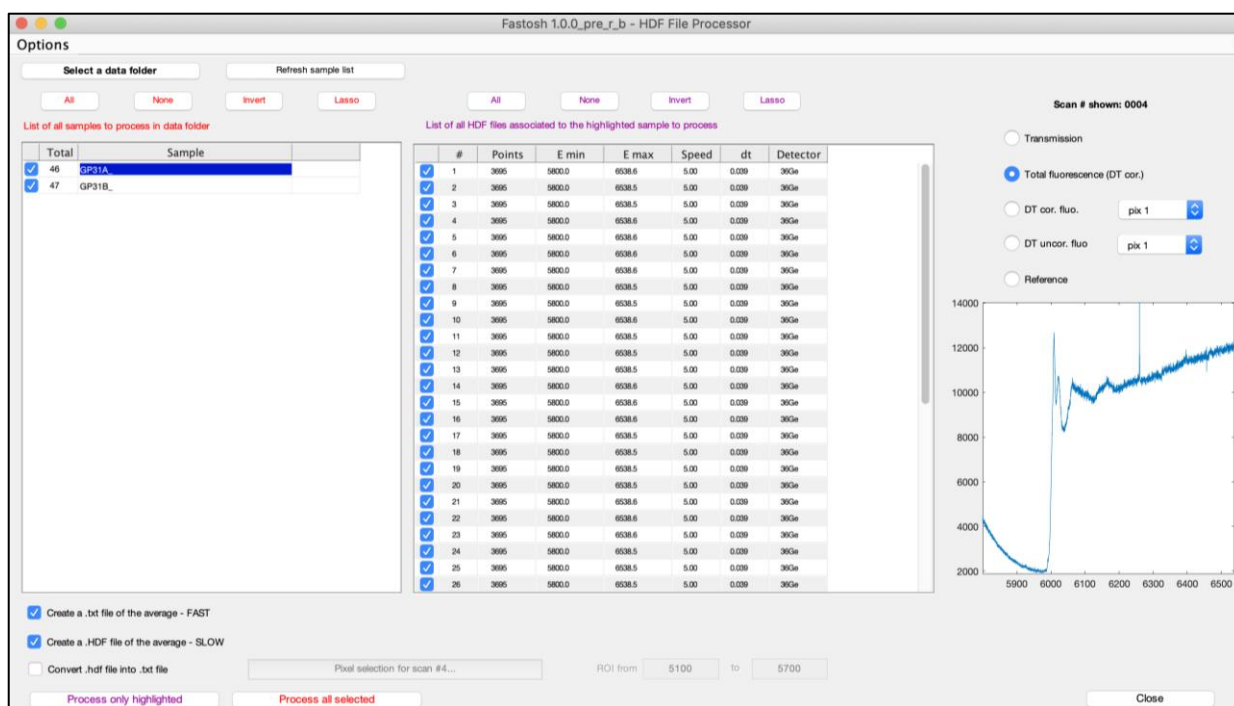


Figure 107 - HDF Processor window

## 9.1.2 How to use this module

### 9.1.2.1 Specify repertory of data folder

Firstly, make sure that all SAMBA HDF files to be exploited are located in the same folder. The repertory of this folder must be specified after opening the HDF Processor window. This can be done by pressing the button “Select a data folder” at the top left corner of the HDF Processor window. The names of all HDF files present in the data folder are then automatically listed on the left side of the HDF Processor window (Figure 107). The total number of iterations per HDF file is specified on the left side of each name. One name is highlighted in blue color in the sample list (the first sample in the list, by default). The details of each iteration associated to the highlighted sample are shown in the middle of the window (Figure 107). These details are iteration number, total number of data points, energy range ( $E_{\min}$  and  $E_{\max}$ ), monochromator velocity (“Speed”) in eV/sec, acquisition integration time in seconds/data point (“dt”) per scan, and type of detector employed:

36Ge = 36 pixel germanium fluorescence detector, Canberra  
13SDD =13 pixel Si fluorescence detector, Mirion

### 9.1.2.2 Data visualization

By default, the data corresponding to the last iteration of the highlighted sample is displayed on the right side of the HDF Processor window. **One can interactively click on any other iteration listed in the iteration table to display its corresponding data.** Also, by default, the data shown corresponds to the transmission (“Transmission” =  $\log(I_0/I_1)$ ) of the current iteration. One can alternatively display the data corresponding to the deadtime-corrected total fluorescence (“Total fluorescence (DT cor.)” =  $\sum(IF(i) \times (\text{input count rates}(i) / \text{output count rates}(i)) / I_0)$ , with  $i$  being a given fluo detector pixel), the deadtime-corrected data collected at a specific fluo detector pixel (“DT cor. fluo” =  $IF(i) \times (\text{input count rates}(i) / \text{output count rates}(i)) / I_0$ ), the deadtime-uncorrected data collected at a specific fluo detector pixel (“DT uncor. fluo” =  $IF(i)/I_0$ ), or the reference (“Reference” =  $\log(I_1/I_2)$ ) (Figure 107). This functionality thus allows to quickly inspect the collected spectra without importing them to the Main GUI of the program.

#	Points	E min	E max	Speed	dt	Detector
1	1000	26500.0	27300.0	10.00	0.08	36Ge
2	750	26500.0	27100.0	10.00	0.08	36Ge
3	750	26500.0	27100.0	10.00	0.08	36Ge
4	750	26500.0	27100.0	10.00	0.08	36Ge
5	750	26500.0	27100.0	10.00	0.08	36Ge
6	750	26500.0	27100.0	10.00	0.08	36Ge
7	750	26500.0	27100.0	10.00	0.08	36Ge
8	750	26500.0	27100.0	10.00	0.08	36Ge
9	750	26500.0	27100.0	10.00	0.08	36Ge
10	750	26500.0	27100.0	10.00	0.08	36Ge
11	750	26500.0	27100.0	10.00	0.08	36Ge
12	750	26500.0	27100.0	10.00	0.08	36Ge
13	750	26500.0	27100.0	10.00	0.08	36Ge
14	750	26500.0	27100.0	10.00	0.08	36Ge
15	750	26500.0	27100.0	10.00	0.08	36Ge
16	750	26500.0	27100.0	10.00	0.08	36Ge
17	750	26500.0	27100.0	10.00	0.08	36Ge
18	750	26500.0	27100.0	10.00	0.08	36Ge
19	750	26500.0	27100.0	10.00	0.08	36Ge
20	750	26500.0	27100.0	10.00	0.08	36Ge
21	750	26500.0	27100.0	10.00	0.08	36Ge
22	750	26500.0	27100.0	10.00	0.08	36Ge
23	750	26500.0	27100.0	10.00	0.08	36Ge
24	750	26500.0	27100.0	10.00	0.08	36Ge
25	750	26500.0	27100.0	10.00	0.08	36Ge

Figure 108- Example where not all scan iterations are selected by default because of differences in scan parameters between scan iterations

### 9.1.2.3 Data averaging

#### 9.1.2.3.1 Best practice to select scan iterations to average

If the goal is to process the HDF files of multiple samples, whose names are listed in the table located in the left side of the HDF Processor window, it is recommended before processing the data to inspect all iterations collected for each sample, by highlighting one by one each sample name. This allow to identify and unselect all problematic iterations before processing the data.

If scan parameters were modified at SAMBA beamline between different scan iterations of a given sample analyzed, all scans acquired for this sample may not have exactly the same energy range or number of data points. In that case, the program will not merge together all these iterations. **Only scan iterations that have the same total number of data points,  $E_{\min}$  ( $\pm 2$  eV), and  $E_{\max}$  ( $\pm 2$  eV) as those corresponding to the 1<sup>st</sup> iteration are selected by default.** For instance, in the example shown in Figure 108, the  $E_{\max}$  (27300 eV) of first iteration is different than the  $E_{\max}$  of all remaining iterations (27100 eV). Consequently, only the first iteration is selected by default in the table, and not the remaining ones (Figure 108). In this example, the button “invert”, located right above the table, should be pressed to invert the iteration selection so that all scans of the highlighted sample except the first scan can be selected and subsequently processed together.

#### 9.1.2.3.2 Creation of TXT or HDF merge file

Once the data of all relevant samples have been primary checked following the method discussed in the previous section (9.1.2.3.1), the data can be processed.

- Firstly, select at the bottom left of the window the desired processing method(s) (choosing the two methods is possible):

- **“Create a .txt file of the average -FAST”**  
This merges all transmission and fluorescence data for all selected iterations of the sample(s) to process. The output is a text file containing 5 columns: Energy, normalized transmission ( $\log(I_0/I_1)$ ), normalized fluorescence ( $I_F/I_0$ ), normalized reference ( $I_1/I_2$ ), and  $I_0$  channel.
- **“Create a .HDF file of the average -SLOW”**  
This merges all data in the SAMBA HDF files corresponding to all selected iterations of the sample(s) to process. The output is an HDF file, which can be open in MCA Viewer module (Section 9.2). Merging HDF file is a slow process. The reason why is explained in details below (Section 0).

- Then, starting the data averaging. The data processing can be done on either:

- The current sample highlighted in blue color in the sample list:  
⇒ Press **“Process only highlighted”** button at the bottom of the HDF File Processor window.
- All samples whose names are checked in the sample list:  
⇒ Press **“Process all selected”** button at the bottom of the HDF File Processor window.

After choosing any of these two options, one must specify the folder where the created output files will be stored.

#### 9.1.2.3.3 Operations done during merging

Since the monochromator movement during a scan is not perfectly reproducible, multiple XAFS scans may not have exactly the same energy arrays, even when the same scan parameters are employed to collect the data. This implies that the data corresponding to multiple XAFS scans must be interpolated, so that they share the same energy array, prior to merging them together.

The number and type of interpolations performed vary depending on whether a merge .txt or .HDF file is created. This is described below.

#### - Creation of merge text file (Fast method)

Suppose there are 10 XAFS scans to merge together. The data arrays corresponding to I0, I1, I2, and IF<sub>total</sub> (total fluorescence collected at all detector pixels, deadtime-corrected) of scans 2 to 10 are then interpolated. This is done using the energy array of the first scan as the interpolation query. Hence, all interpolated data eventually share the same energy array, i.e. the energy array of the first scan. Finally, each type of interpolated data (I0, I1, I2, and IF<sub>total</sub>) is merged.

#### - Creation of merge HDF file (Slow method)

The MCA patterns collected at each pixel of the fluorescence detector represent in size most of the data saved in SAMBA HDF file. Indeed, when an XAFS scan is collected at SAMBA beamline using a continuous acquisition mode, an MCA pattern containing 4096 data points is recorded by each fluorescence detector pixel, at every energy step of the XAFS scan. For example, if an XAFS scan features 3000 energy steps, and the fluorescence detector has 36 pixels, there will be  $3000 \times 36 = 108000$  MCA patterns collected and saved in the HDF file of the scan. Now suppose that 10 iterations of such XAFS scan are successively collected at the beamline, and their corresponding HDF files should be merge together. As previously mentioned, these 10 XAFS scans do not exactly have the same energy arrays even though they were collected using the same scan parameters. Consequently, the MCA patterns of all 10 XAFS scans were actually not recorded at the exact same energy value at each step of the XAFS scan. The MCA data of scans 2 to 9 are then interpolated using the energy array of the first scan (containing 3000 data points) as the interpolation query. The program then must perform a very large number of interpolations, which explains why merging SAMBA HDF file is a slow process.

For instance, with the example discussed above, the total number of interpolations to perform is:

- 4096 points/MCA collected at each energy step of the XAFS scan
- 36 detector pixels, each of them collected a specific MCA pattern
- 9 HDF files with MCA data to interpolate

→  $4096 \times 36 \times 9 = \text{more than 1.3 million interpolations of arrays containing 3000 data points!}$  It is then recommended to merge all HDF files overnight, for example during the night before exploiting the merge HDF files.

Additionally, when a merge HDF file is created, the data arrays corresponding to I0, I1, I2, and IF<sub>total</sub> (total fluorescence collected at all detector pixels, deadtime-corrected), and fluorescence deadtime array are interpolated. Finally, each type of interpolated data is merged.

#### 9.1.2.4 Data conversion from HDF to TEXT file while redefining ROI and pixel selection

To convert HDF to TEXT files while redefining ROI and pixel selection, firstly check on “Convert .hdf file into .txt file” at the bottom left corner of the HDF processor window (Figure 109). Note that when selecting this option, the button “Pixel Selection for scan #...” and the fields to modify the ROI limits become activated (Figure 109). These are not activated when the option “Convert .hdf file into .txt file” is not checked. In the example shown in Figure 109, the current iteration was iteration #4. One can then redefine the pixel selection for iteration #4 by pressing on the button “Pixel Selection for scan # 4” (Figure 109). This opens up a new window where all pixel data corresponding to the current iteration, as shown in Figure 110.

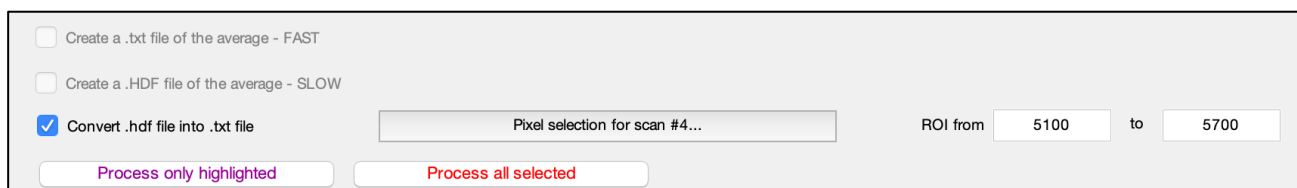


Figure 109 – Bottom left corner of the HFD Processor window

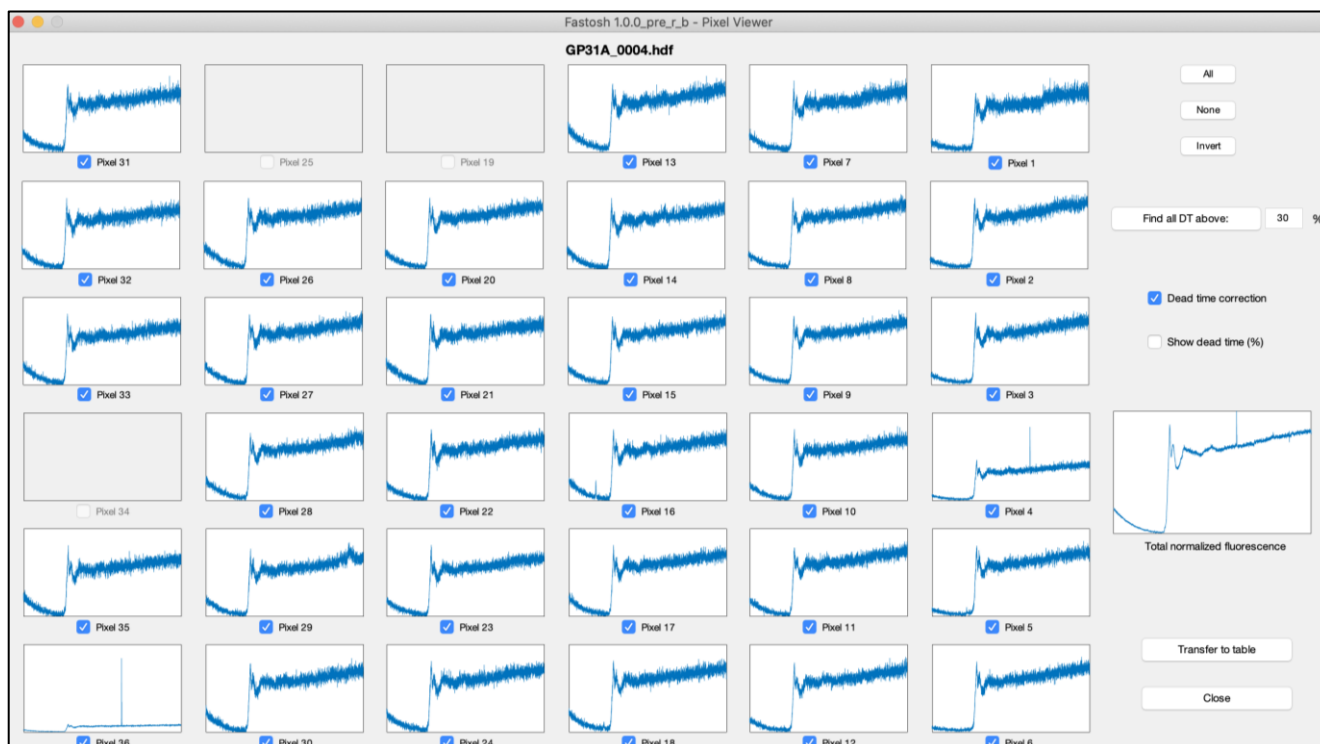


Figure 110 – All pixel data corresponding to the current iteration that was interactively selected in the HDF Processor window

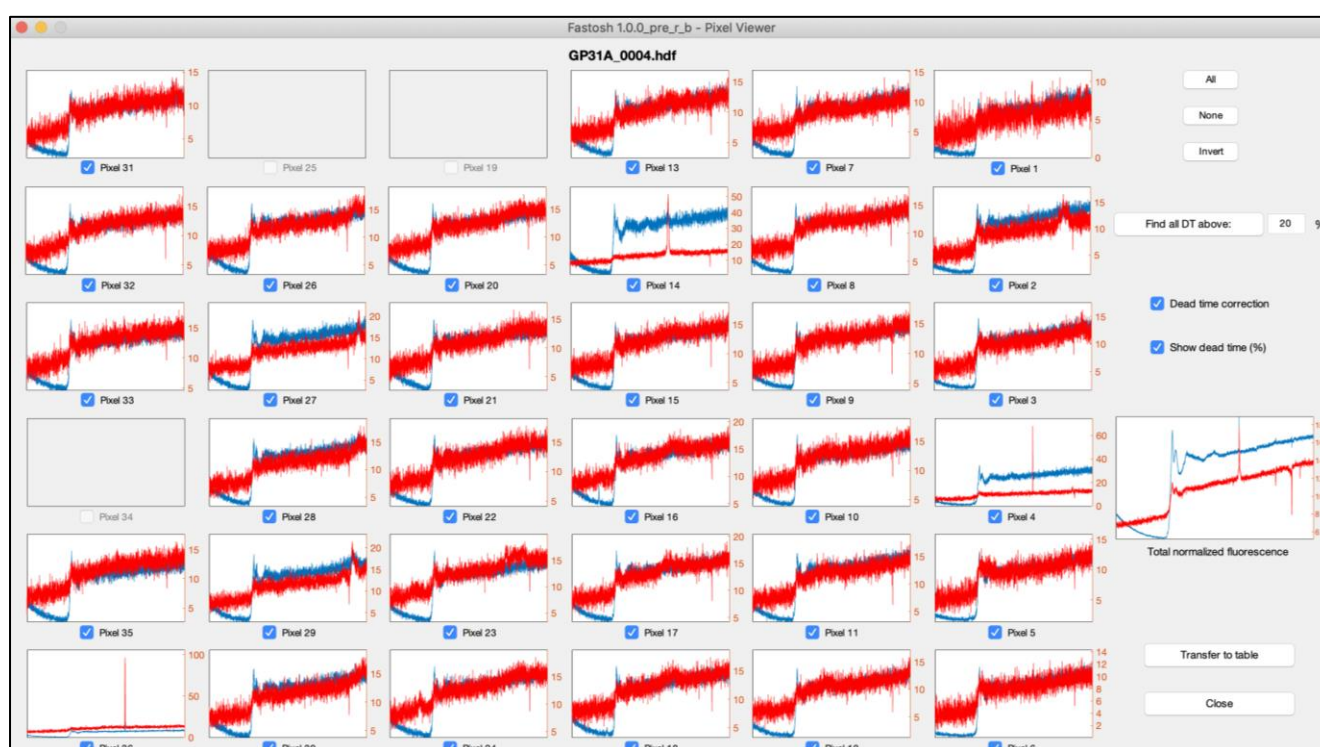


Figure 111 – After checking on “Show Dead Time (%)” option on the right side of the Pixel Viewer window, deadtime is displayed in red color for each pixel, and the total spectrum

Note that the Pixel Viewer window shown in Figure 110 can be automatically refreshed when interactively selecting a new scan iteration from the iteration table featured in the HDF Processor window. For instance, using the example shown in Figure 110 where the data of iteration #4 is displayed in the Pixel Viewer window, if one clicks on iteration #5 in the table featured in the middle of the HDF Processor window, the Pixel Viewer window automatically updates and eventually shows the data corresponding to iteration #5.

The deadtime-corrected fluorescence data is displayed by default when opening the Pixel Viewer window. To display the deadtime-uncorrected fluorescence data, check on the “Dead Time Correction” option on the right side of the window.

Additionally, the deadtime-corrected fluorescence data is not displayed by default when opening the Pixel Viewer window. To display it, check on the “Show dead time (%)” option on the right side of the window. The deadtime is then displayed in red color, as shown in Figure 111. The deadtime scale, in %, is shown on the Y axis on the right side of each spectrum plot. In the example shown in Figure 111, one can observe, from the deadtime plots, that some pixels have been significantly affected by diffraction phenomena. Therefore, displaying the deadtime can help identify the pixels to unselect. Methods to unselect pixels are described in the next two sections.

#### 9.1.2.4.1 Preliminary step: manual or automatic pixel selection on a single iteration

Once the Pixel Viewer window is open (Figure 110), one can manually unselect some pixels by unchecking the box below each plot. Alternatively, one can automatically remove all pixels whose deadtime is above a user-defined threshold in at least one data point of the XAFS spectrum, by pressing “Find all DT above”, on the right side of the Pixel Viewer window (Figure 111). The value of this threshold, which is 30% by default, can be modified on the right side of the Pixel Viewer window (Figure 111). For example, using the example shown in Figure 110 where the data of iteration #4 is displayed in the Pixel Viewer window, and after setting the deadtime threshold to 20 %, 5 pixels were automatically unselected. Note that the plot of total fluorescence, on the right side of the Pixel Viewer window, is automatically updated after unselecting a pixel.

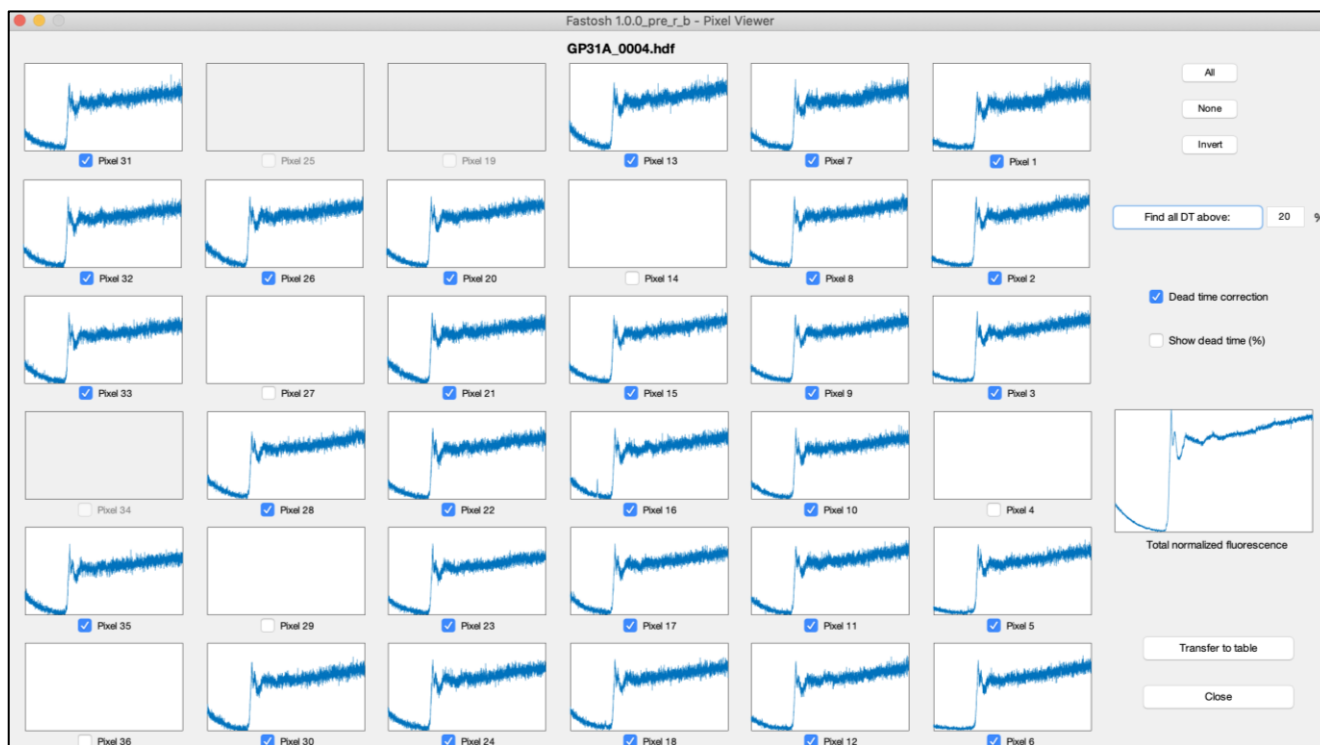


Figure 112 – Same as previous figure, but after pressing on the button “Find all DT above 20 %”. This automatically unselected all pixels whose deadtime was above 20% in at least one data point of the spectrum.

Once done with pixel selection, press the button “Transfer to table”. The numbers of the pixels unselected in the Pixel Viewer window are now written in the last column of the iteration table featured in the HDF processor, entitled “Unwanted pixels” (Figure 113). In Figure 113, one can see that 5 pixels were unselected for iteration #4. One can apply this specific pixel selection to all or only a specific set of iterations corresponding to the same sample.

After transferring a pixel selection to the table of the HDF Processor window, this pixel selection can be applied to all iterations of a sample using the following menu option available on top of the HDF Processor Window:

## HDF Processor > Options > “Apply current pixel selection to all scans”

Alternatively, a pixel selection can be applied to a specific set of iterations, by following this procedure (Figure 114):

- Firstly select all iterations to apply the same pixel selection, e.g. iterations #4, 5, 6, & 7.
- Do the pixel selection on one iteration, for example, #4, and transfer this pixel selection to the table.
- This pixel selection can be applied to pixels #5, 6, & 7 using the following menu option available on top of the HDF Processor Window:

## HDF Processor > Options > “Apply current pixel selection to selected scans”



List of all HDF files associated to the highlighted sample to process

	#	Points	E min	E max	Speed	dt	Detector	Unwanted pixels
<input checked="" type="checkbox"/>	1	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	2	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	3	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	4	3695	5800.0	6538.6	5.00	0.039	36Ge	4, 14, 27, 29, 36
<input checked="" type="checkbox"/>	5	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	6	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	7	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	8	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	9	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	10	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	11	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	12	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	13	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	14	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	15	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	16	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	17	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	18	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	19	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	20	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	21	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	22	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	23	3695	5800.0	6538.6	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	24	3695	5800.0	6538.5	5.00	0.039	36Ge	None
<input checked="" type="checkbox"/>	25	3695	5800.0	6538.5	5.00	0.039	36Ge	None

Figure 113 – Iteration table is refreshed after transfer of a pixel selection : pixels 4, 14, 27, 29, and 36 are now listed as unselected. Therefore, the data from these pixels will not be used to extract a new fluorescence spectrum, once the HDF file will be converted to a text file

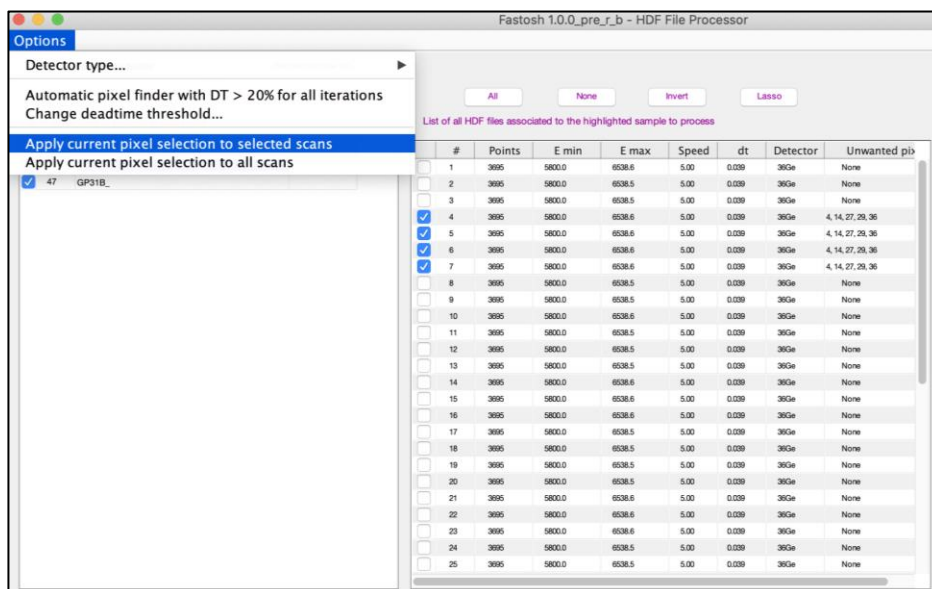


Figure 114 – Iteration table now shows that pixel selection was applied to only selected iterations

#### 9.1.2.4.2 Preliminary step: Automatic pixel selection on all iterations of a sample

The HDF Processor features a functionality to automatically perform, using a single command, pixel selections on all iterations of a sample. This function unselect any pixel whose deadtime is above a user-defined threshold in at least one data point of the XAFS spectrum. To use this function, follow this procedure:

- Specify the value of the threshold, which is 30% by default, using the following menu option available on top of the HDF Processor Window:

HDF Processor > Options > “Change deadtime threshold...”

- Start the automatic processing, using the following menu option available on top of the HDF Processor Window (highlighted in blue in Figure 115)

HDF Processor > Options > “Automatic pixel finder with DT > 30% for all iterations”

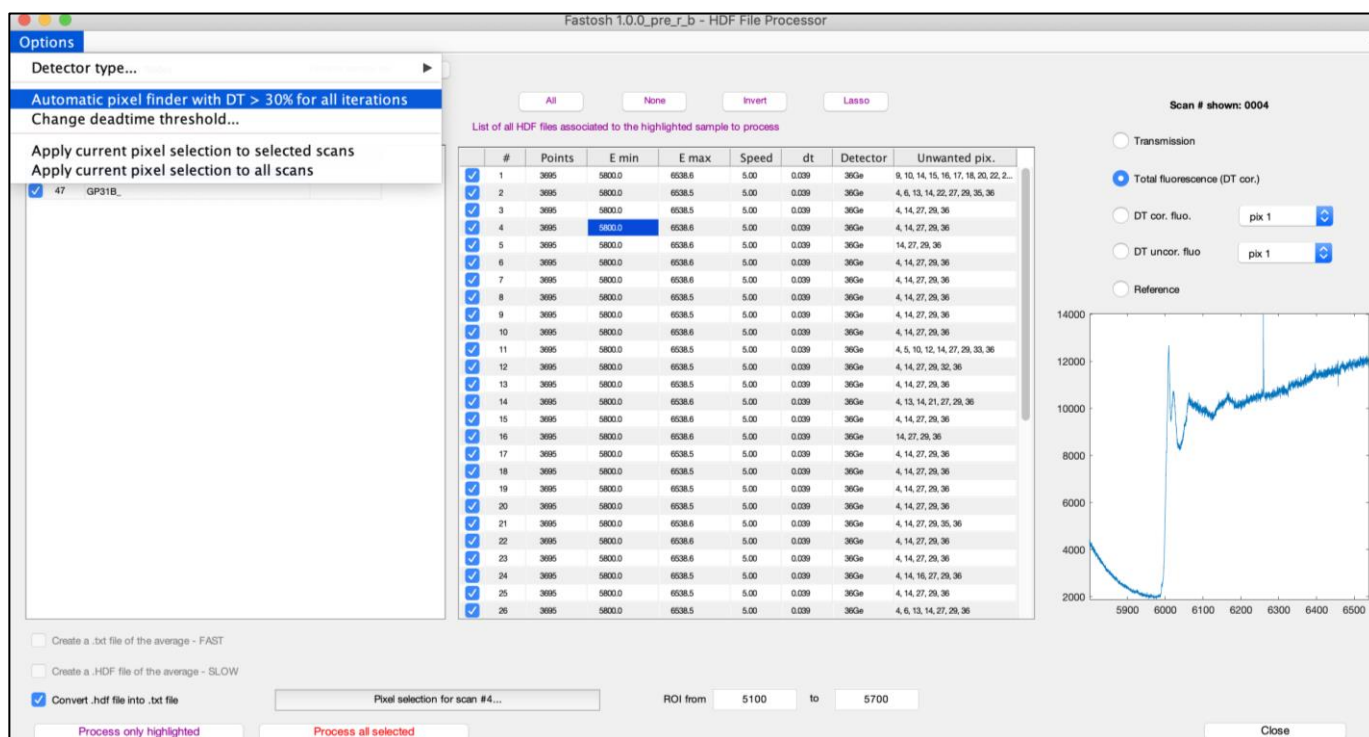


Figure 115 – Automatic pixel processing of all iterations corresponding to current sample. Results are shown in the column “Unwanted pixels” in the table featured in the middle of the HDF Processor window

#### 9.1.2.4.3 Preliminary step: Modifying ROI

While a specific pixel selection can be defined for each iteration of a sample as shown in the previous sections, only one Region of Interest (“ROI”, i.e. the specific region of the MCA pattern where the fluorescence is extracted) can be used for all iterations of a sample. Specify it in the fields “ROI from” at the bottom of the HDF Processor window (Figure 115). The ROI originally used at SAMBA for the first iteration of the current sample is displayed by default.

#### 9.1.2.4.4 Second step: creating TXT files

Once all optionnal preliminary steps have been completed (i.e. pixel selection and/or ROI modification), the HDF files can be converted to text files. Press on the button “Process Only highlighted” on the bottom left corner of the HDF Processor window. This will do the HDF/text file conversion for all selected iterations corresponding to the highlighted sample. Alternatively, if the button “Process all selected” is pressed, the HDF/text file conversion is done on all selected iterations corresponding to all samples selected in the sample list, on the left side of the HDF Processor window.

## 9.2 MCA Viewer (SAMBA data only)

This module of Fastosh can be open via the menu of the Main GUI:

Main GUI Menu > Operations > For SAMBA’s files only... > “MCA Viewer”

### 9.2.1 Usefulness

A single SAMBA HDF file generated at SAMBA beamline, corresponding to a single XAFS spectrum, or a merge HDF file created with the “HDF Processor” module of Fastosh (see Section 9.1), corresponding to an average XAFS spectrum, can be open in this module. From there, one can:

- Display all MCA patterns saved in the HDF file.

- Display the XAFS spectra collected at all pixels of the fluorescence detector and their associated deadtime. Problematic pixels can be unselected and a new XAFS spectrum can be extracted from MCA without them.
- Redefine, specifically for each pixel, the original ROI employed to collect the fluorescence data at SAMBA. A new XAFS spectrum can be extracted from MCA using the new ROI values.
- Subtract the baseline from the counts above it in the ROI of the MCA patterns and then extract a new XAFS spectrum.
- Determine the fraction of MCA outside of the ROI that is affected by spectral artefacts (e.g. diffraction peaks), and use this fraction to correct the XAFS spectrum extracted from the ROI.

### 9.2.2 Importing an HDF file

A single HDF file, generated at SAMBA beamline, or a merge HDF file, created using the HDF Processor module of Fastosh (see Section 9.1), can be imported to MCA Viewer via the menu of the module:

MCA Viewer Menu > File > “Open an HDF file”

When an HDF file is imported, all fluorescence detector data saved in the HDF file are loaded and processed. This represents a more convenient approach than reopening and reprocessing the data every time an operation is requested by the user in the MCA Viewer module. The file importation thus requires a few seconds to complete, and proceeds in two steps. A message is displayed on top of the MCA Viewer at each of these two steps:

- “*First step: loading & processing spectra of pixel #*”: the fluorescence spectra and associated deadtime arrays are loaded at this step, and deadtime correction is applied to each fluorescence spectrum, for each detector pixel.
- “*Second step: loading & processing MCA pattern of pixel #*”: all MCA patterns and associated deadtime arrays for each detector pixel are loaded at this step, and deadtime correction is applied to each MCA pattern. If all loaded MCA patterns corrected or uncorrected for deadtime were stored in the computer’s flash memory, the computer performances could be severely diminished. They are then stored in the temporary folder of the computer. All data is eventually deleted automatically from the temporary folder. This process is transparent for the user.

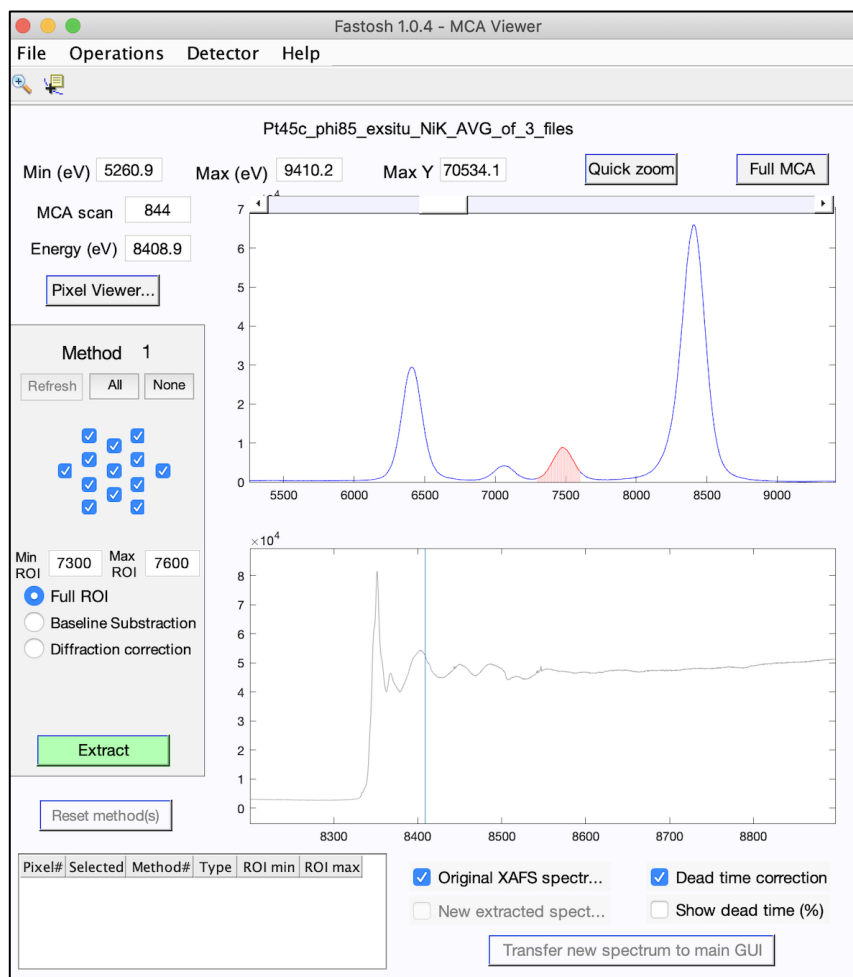


Figure 116 - Main window of MCA Viewer

## 9.2.3 Main Window

### 9.2.3.1 Presentation

The MCA Viewer window (Figure 116) features:

- At its top, the total MCA at a specific energy of X-rays. It corresponds to the sum of MCA patterns at all selected pixels and at a specific energy of incoming X-rays hitting the sample. The selection of pixels is shown on left side of the MCA Viewer window (checked boxes are all selected pixels). All pixels should be selected/checked by default right after opening an HDF file. If some of them are not checked and displayed with a grey color after opening a file, these pixels are permanently turned off in the detector. If some pixels are manually unchecked, the button “refresh” on the left side of the window (Figure 116) must be pushed to recalculate the sum of MCA and display it on top of the MCA Viewer window. The specific energy of X-rays probing the sample of the current MCA is shown in the field “Energy” as well as its corresponding tile number in the field “MCA” in the top left corner of the MCA Viewer window (Figure 116). For example, if an XAFS spectrum features 3000 data points, there are 3000 MCA tiles, from 1 to 3000.
- At its bottom, the original XAFS spectrum collected in fluorescence mode at SAMBA, shown in gray color (Figure 116). This spectrum corresponds to the total fluorescence collected in the ROI at all selected pixels ( $IF_{total}$ ) divided by  $I_0$  channel. By default, it is shown with deadtime correction applied to it. Optionally, deadtime correction can be unselected by unchecking the box “deadtime correction” at the bottom right of the MCA Viewer (Figure 116).
- On the left side of the MCA Viewer window, a box shows the current pixel selection and the method number. See Section below for further details on “Method”.

### 9.2.3.2 Browsing MCA tiles

There are four methods to browse the MCA tiles:

- Use the horizontal scroll bar on top of the displayed MCA pattern in the MCA Viewer window
- Click anywhere in the XAFS spectrum plot **except directly in the XAFS spectrum itself**. If one clicks directly on the XAFS spectrum, instead of anywhere else above or below it in the plot, the energy selection won't be considered. A vertical bar in the XAFS spectrum plot shows the position of the current energy corresponding to the MCA displayed in the window.
- Specify the energy in the field "Energy" on the top left corner of the MCA Viewer window
- Specify the MCA tile number in the field "MCA" on the top left corner of the MCA Viewer window

All these functionalities are linked to each other. For example, if one clicks in the XAFS plot to interactively select an energy, the corresponding position of the scroll bar, energy value, and tile number will be automatically refreshed in the MCA Viewer window.

## 9.2.4 Operations

### 9.2.4.1 Viewing and removing pixels

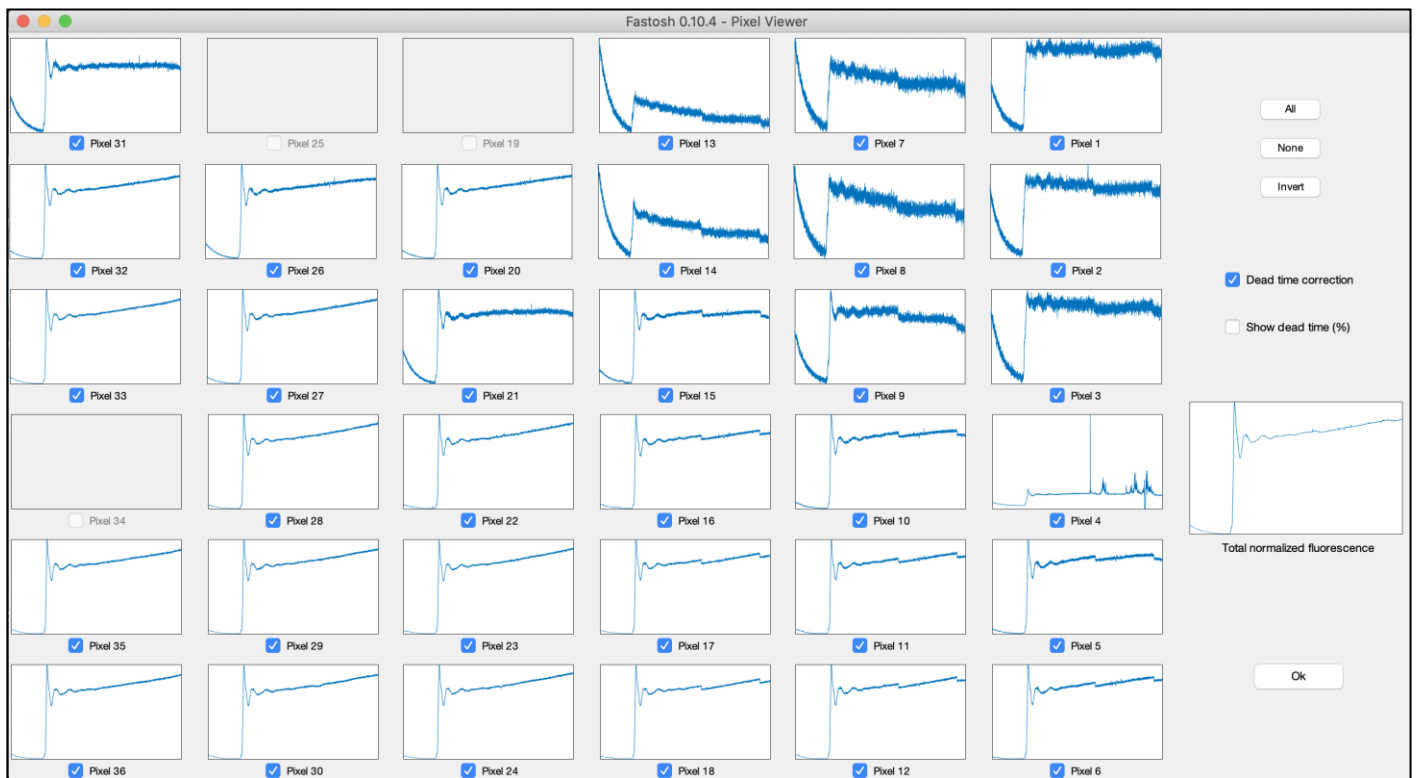
The XAFS spectrum collected in fluorescence mode at SAMBA beamline may sometimes be distorted due to acquisitions artefacts, such as diffraction phenomena. One example of such problematic XAFS spectrum is displayed at the bottom of the MCA Viewer window in Figure 116. If the XAFS spectrum is locally distorted, one should determine whether these distortions occur at all pixels of the fluorescence detector or only at specific pixels. In the latter case, the problematic pixels can be excluded and a new XAFS spectrum can be extracted without using the fluorescence data collected at these pixels.

To view all pixel data collected by the fluorescence detector, click on the button "Pixel Viewer..." on the left side of the MCA Viewer window (Figure 116). A new window appears displaying the normalized XAFS fluorescence spectra collected at each pixel ( $IF_{\text{pixel}}/I_0$ ), and the total normalized fluorescence spectrum ( $IF_{\text{total selected pixels}}/I_0$ ) on the right side of the window (Figure 117 A). **The pixel distribution on this window corresponds to the real pixel distribution on the detector**: for example, pixels #1, 6 31, and 36 at the four corners of the Pixel Viewer window (Figure 117) are indeed located at the four corners of the monolithic Ge crystal of SAMBA's 36 pix Canberra detector. The deadtime for each pixel can be also displayed, by clicking on the button "Show deadtime" on the right side of the window. The deadtime scale appears on the right side of each pixel plot (Figure 117 B) In the example shown in Figure 117, the problematic pixels are obviously all those located on the right side of the detector (pixels #1 to 18). Problematic pixels can be unselected by unchecking the boxes below the pixel plots. The total normalized fluorescence spectrum is then automatically updated on the right side of the window (Figure 117 C). When pixel selection is done, click on the "Ok" button at the bottom right corner of the window, to close the window.

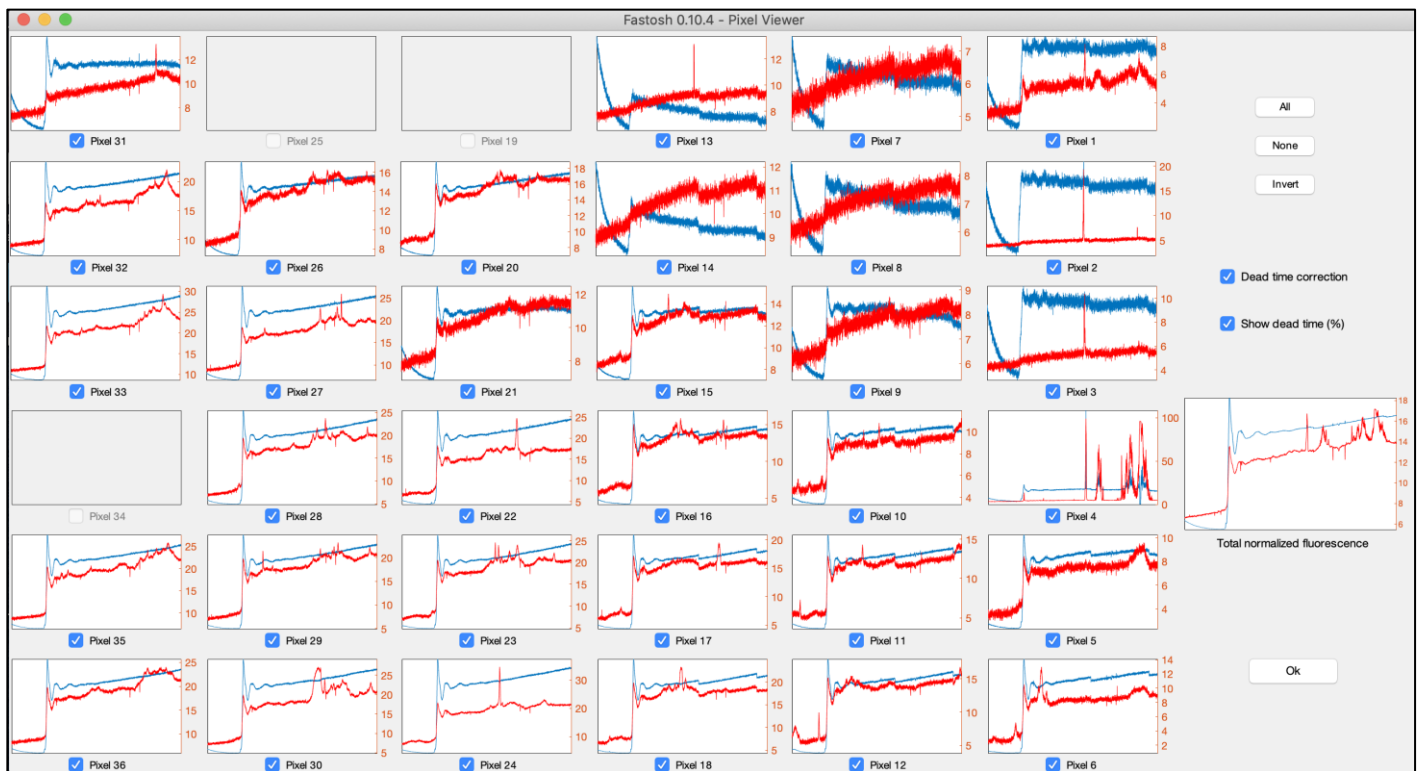
After closing the Pixel Viewer window, the pixel selection shown on the left side of the MCA Viewer window is automatically updated to match with the pixel selection done in the Pixel Viewer window. Also, the total MCA pattern displayed at the top part of the MCA Viewer window is automatically updated to only display the data relative to the selected pixels.

Click on the button "Extract" to obtain the new XAFS spectrum extracted using the current pixel selection. The new XAFS spectrum is shown at the bottom of the MCA viewer window with a blue color (Figure 118). In the above example, one can notice that unselecting half of the pixels of the detector (Figure 117) enabled to get rid of all distortions that were originally present in the XAFS spectrum collected at SAMBA (Figure 118).

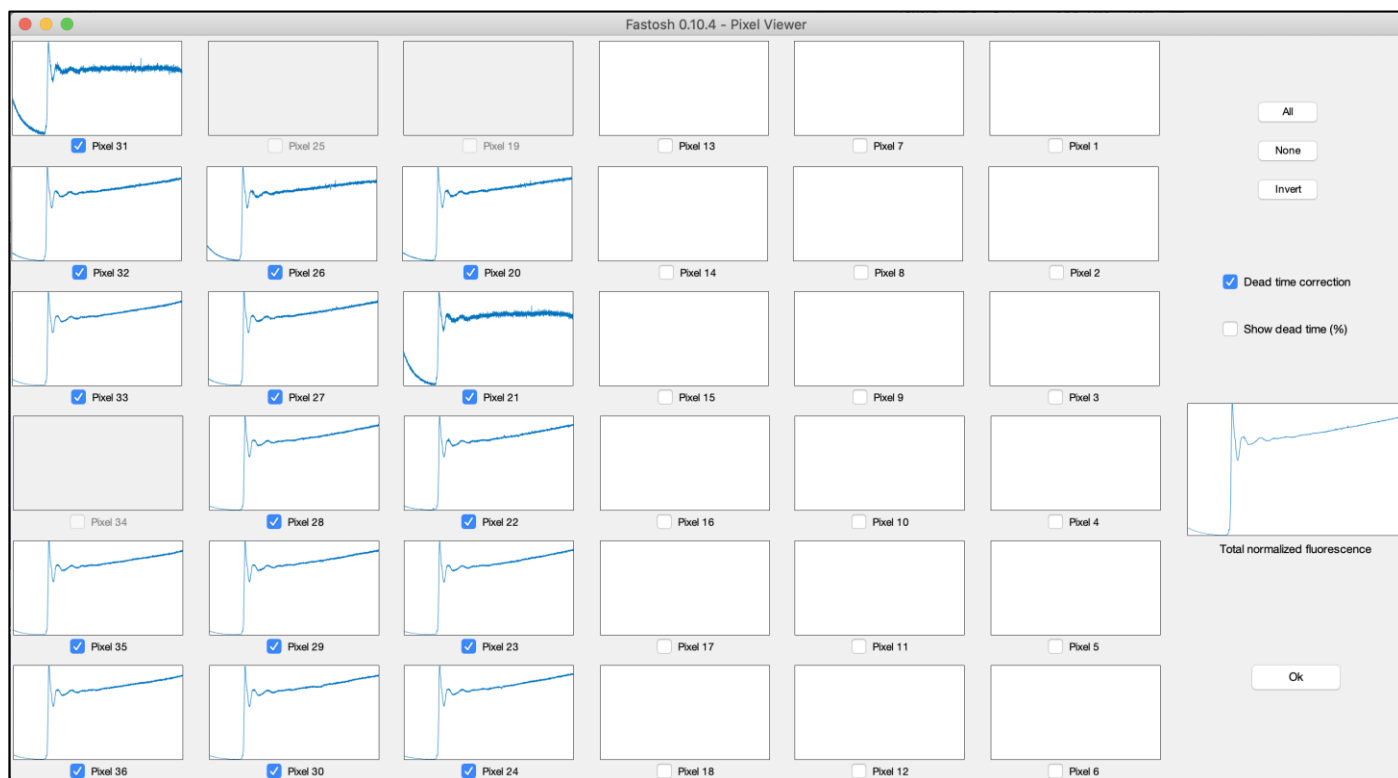
Finally, press the button "Transfer new spectrum to main GUI" at the bottom of the MCA Viewer window (Figure 118) so that the new spectrum can be saved in the Main GUI.



A)



B)



C)

Figure 117- Pixel Viewer window: data of XAFS spectrum shown in the previous figure, with A) all pixels ; B) all pixels + deadtime; C) after unchecking problematic pixels

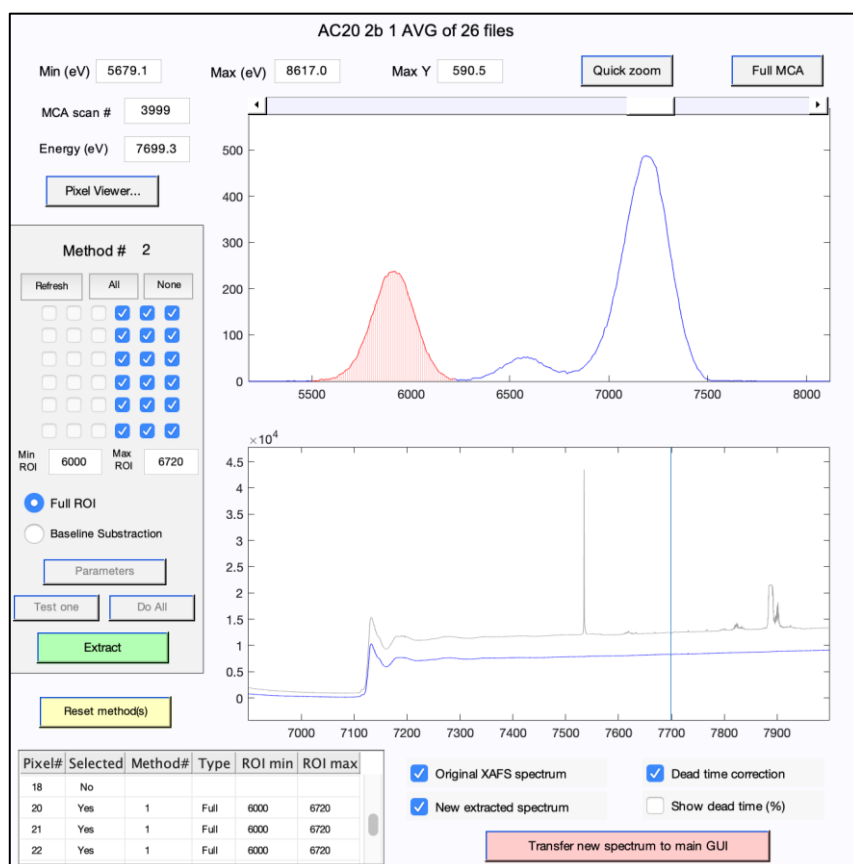


Figure 118 – New, artefact-free XAFS spectrum (shown in blue color) after deselecting problematic pixels and pressing the button “Extract”. Notice the difference with the original spectrum collected at SAMBA beamline (shown in gray color).

#### 9.2.4.2 Modifying ROI min & max

If an XAFS spectrum collected in fluorescence mode at SAMBA is locally distorted, one possible method to suppress the distortions is to modify post-acquisition the original ROI used at the beamline to collect the fluorescence data. Below are two examples of problematic XAFS collected at SAMBA. The cause of the distortions was specific to each example. These spectra could be fixed post-beamtime by modifying the ROI.

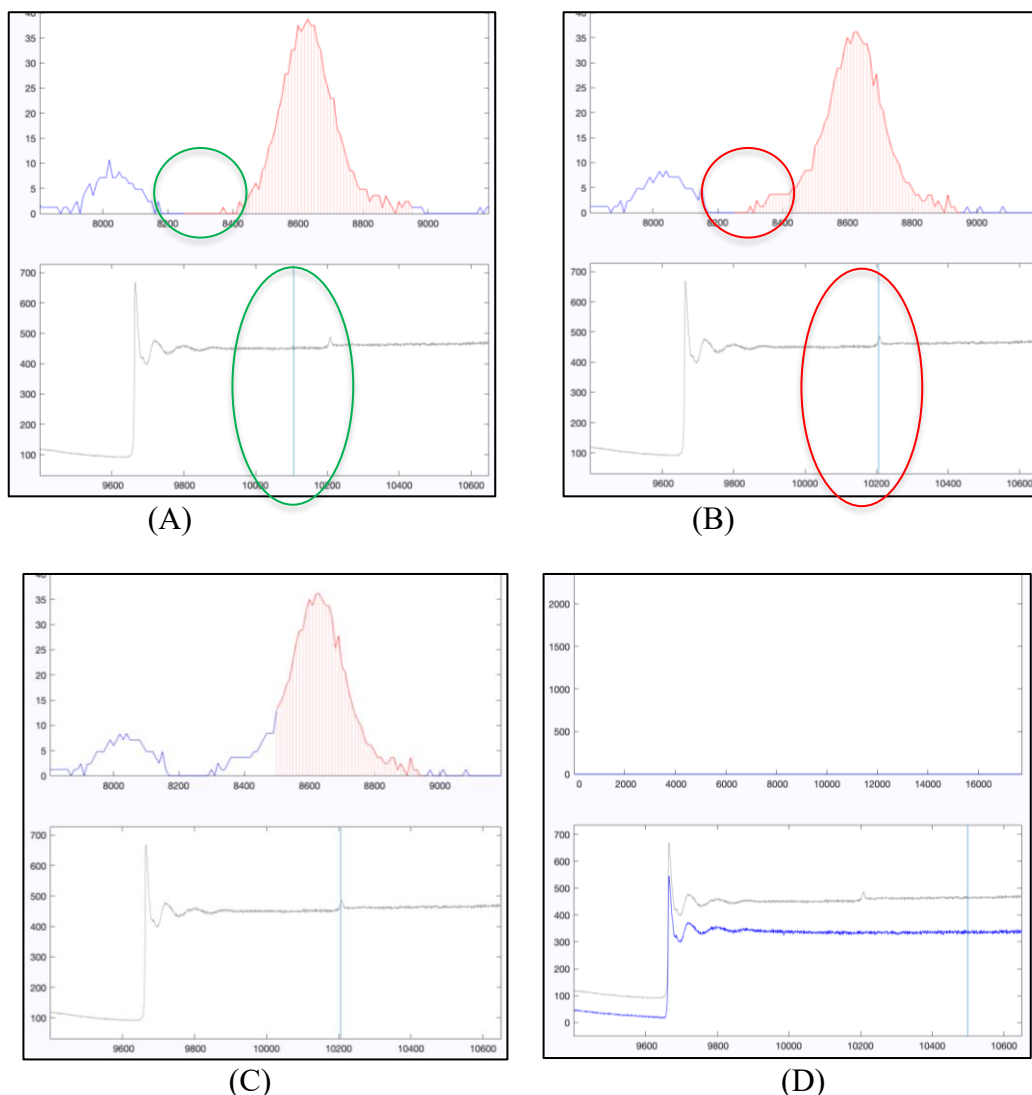


Figure 119 - Example of an Zn K-edge XAFS spectrum distorted due to fluorescence of another element leaking in the ROI: A) before reaching the W L3-edge ; B) at the W L3-edge ; C) MCA after increasing the ROI E<sub>min</sub> to 8500 eV ; and D) new XAFS spectrum obtained after increasing the ROI E<sub>min</sub> to 8500 eV and pressing the button "Extract"

##### Example 1: Fluorescence of another element leaking into the ROI

An XAFS spectrum of a Zn-polluted sediment core sample was collected at SAMBA at the Zn K edge (9659 eV). A small secondary edge-jump can be observed at about 10205 eV, which corresponds to the energy of the W L3 edge (Figure 119). This W contribution could come from the tungsten-made Auger Drill that was used to collect the sediment core sample. The E<sub>min</sub> of ROI used at SAMBA (8300 eV) was too low as the W L $\alpha$  emission (8398 eV) "leaked" inside the ROI when the energy of incoming X-rays hitting the sample reached the energy of W L3 edge at 10205 eV (see small shoulder at ~8400 eV in the MCA of Figure 119 B, which is not present in the MCA of Figure 119 A). Increasing the E<sub>min</sub> of the ROI from 8300 to 8500 eV (Figure 119

C) enabled to shun the W contribution and thus extract a clean XAFS spectrum (Figure 119 D).

### Example 2: Diffraction contribution leaking into the ROI

Diffraction phenomena emitted by crystalline samples and hitting the fluorescence detector can have multiple effects on the MCA patterns. Modifying the ROI range may result in different extracted XAFS spectra, as demonstrated in Figure 120.

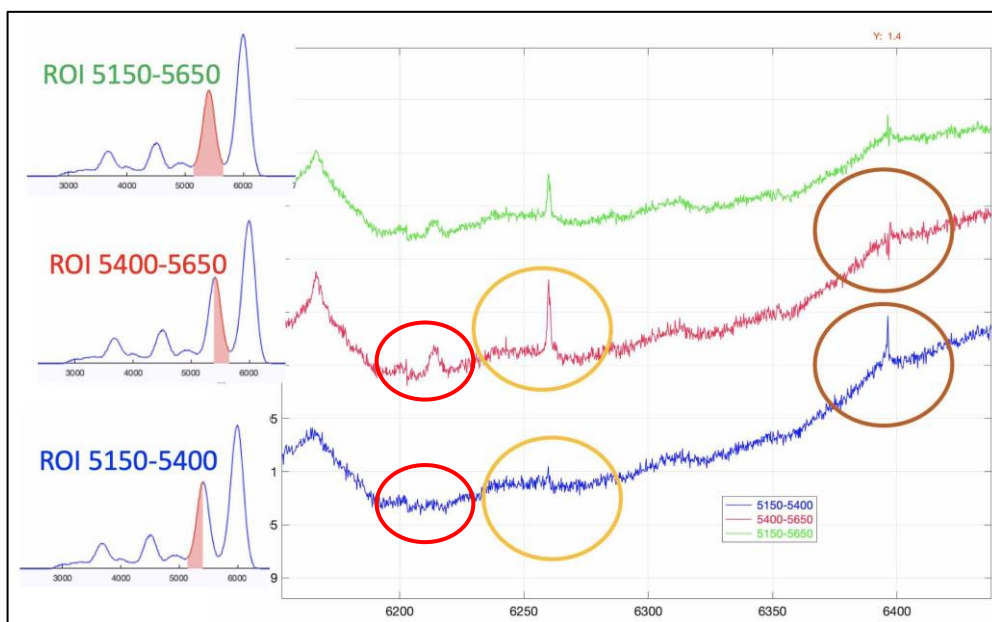


Figure 120 - Post-edge regions of XAFS spectra extracted using three different ROI ranges (shown on the left side of the figure) using the same MCA data, which corresponds to a soil sample analyzed at the Cr K edge.

#### 9.2.4.3 MCA background subtraction

As previously mentioned, diffraction phenomena emitted by crystalline samples and hitting the fluorescence detector can have multiple effects on the MCA patterns. In rare cases, the diffraction results in an elevation of the MCA background and does not significantly affect the rest of the MCA pattern above the background. The background can then be subtracted from the MCA pattern to minimize artefacts in the extracted XAFS spectrum. This is done by fitting a baseline in the MCA pattern using the codes of Mazet *et al.* (2005) [12], as explained below. However, in most cases, diffraction emission does not affect only the background but also the rest of the MCA pattern, including the peak corresponding to the element of interest. Therefore, performing background subtraction is often not effective in minimizing diffraction artefacts in the XAFS spectrum extracted from MCA, and mostly results in a decreased signal-to-noise ratio.

#### To do background subtraction:

- Select "Background subtraction" on the left side of the MCA viewer window

- Press "Test one" button on the left side of the MCA Viewer window: this will fit the baseline only on the current MCA tile displayed in the MCA Viewer window (i.e. this MCA corresponds to the sum of all MCA patterns collected at each pixel for a specific energy of incoming X-rays). Visually estimate on the plot whether the baseline was well fitted. If a poor baseline was obtained, modify the default baseline parameters after clicking on the button "Parameters".

- The baseline fitting method of Mazet *et al.* (2005) [12] has three parameters: function type, threshold, and polynomial order. The "asymmetric truncated quadratic" (default

choice) is often the most adequate function type. Also, a polynomial order equal to 3 (default choice) is often adequate. The threshold value can be sometimes modified to optimize the baseline.

- Once the parameters are suitable to fit correctly the baseline on the current MCA tile, test once again the baseline fitting on another MCA tile. It is recommended to test the baseline fitting at least on one MCA tile before the edge jump of the XAFS, and one after the edge-jump.

- Once convinced that the baseline parameters are effective after testing them on a few MCA tiles, click on "Do All" button to do baseline fitting on all MCA tiles. This will take a few moments: if the XAFS spectrum has 5000 data points, there are 5000 associated MCA tiles, thus 5000 background fitting routines to perform.

- Press "Extract" button to do the actual background subtraction to obtain a new XAFS spectrum. The final fluorescence calculated on each MCA tile (i.e. each data point of the extracted XAFS spectrum) is equal to the integral of the full MCA between ROI  $E_{min}$  to  $E_{max}$  minus the part of the MCA below the baseline between ROI  $E_{min}$  to  $E_{max}$ . Finally, press "Transfer spectrum to the Main GUI" to save the new extracted XAFS spectrum in the Main GUI.

#### 9.2.4.4 Diffraction corrector

##### 9.2.4.4.1 Principle

When the energy of the incident X-ray beam increases during acquisition of an XAFS spectrum, some parts of the MCA pattern collected by the fluorescence detector can considerably evolve in shape, amplitude, or energy position. This includes, of course, the fluorescence line in the ROI, the Compton peak (inelastic scattering of the incident X-ray beam), and the Rayleigh peak (elastic scattering of the incident X-ray beam) above the ROI. In contrast, some parts of the MCA is not supposed to significantly vary when the energy of the incident X-ray beam increases. This is the case of the fluorescence emission lines of all the sample's chemical elements whose absorption edges are below the one of the element studied by XAFS. The amplitude of these features should only decreases slowly as the energy of the incident X-ray beam increases. Therefore, if their amplitude change dramatically during an XAFS acquisition, they must be modified by artefacts. The diffraction corrector tool enables to quantify the fraction of the MCA that is artificially modified in a region that is not supposed to significantly vary. This fraction is then used to correct the new XAFS spectrum that is extracted from the ROI.

To calculate this factor, the integral of the MCA, in a region of the MCA whose features are not supposed to significantly evolve when the energy of the incident X-ray beam increases (i.e. somewhere outside of the ROI in the MCA), is calculated at each energy of the XAFS spectrum. The integral of the MCA region outside ROI vs energy of the XAFS spectrum, and its corresponding baseline obtained using the codes of Mazet *et al.* (2005), are displayed in the top part of the Diffraction corrector window (Figure 121). The former is divided by the latter to obtain the MCA distortion factor at each energy of the XAFS spectrum, which is displayed in the middle part of the Diffraction corrector window (Figure 121). Finally, the original and corrected XAFS spectra are shown in the bottom part of the Diffraction corrector window, in red and black color, respectively (Figure 121). The latter is obtained by dividing the MCA distortion factor from the former.

##### 9.2.4.4.2 Parameters

- "MCA used"

The specific region of MCA used by the Diffraction corrector tool to calculate the correction factor. This region should feature some fluorescence lines that do not significantly change as the energy of the incident X-ray beam increases. By default, this region is from 10eV to an energy of 10eV below the minimum of the ROI. These default values are often not ideal to obtain the correction factor and require to be modified, as shown in the example provided below. If the

minimum and maximum values are respectively set below and above the ROI, then the ROI part will be excluded. For example, if the MCA limits are set to 6000-8000 eV, and the ROI is 7300-

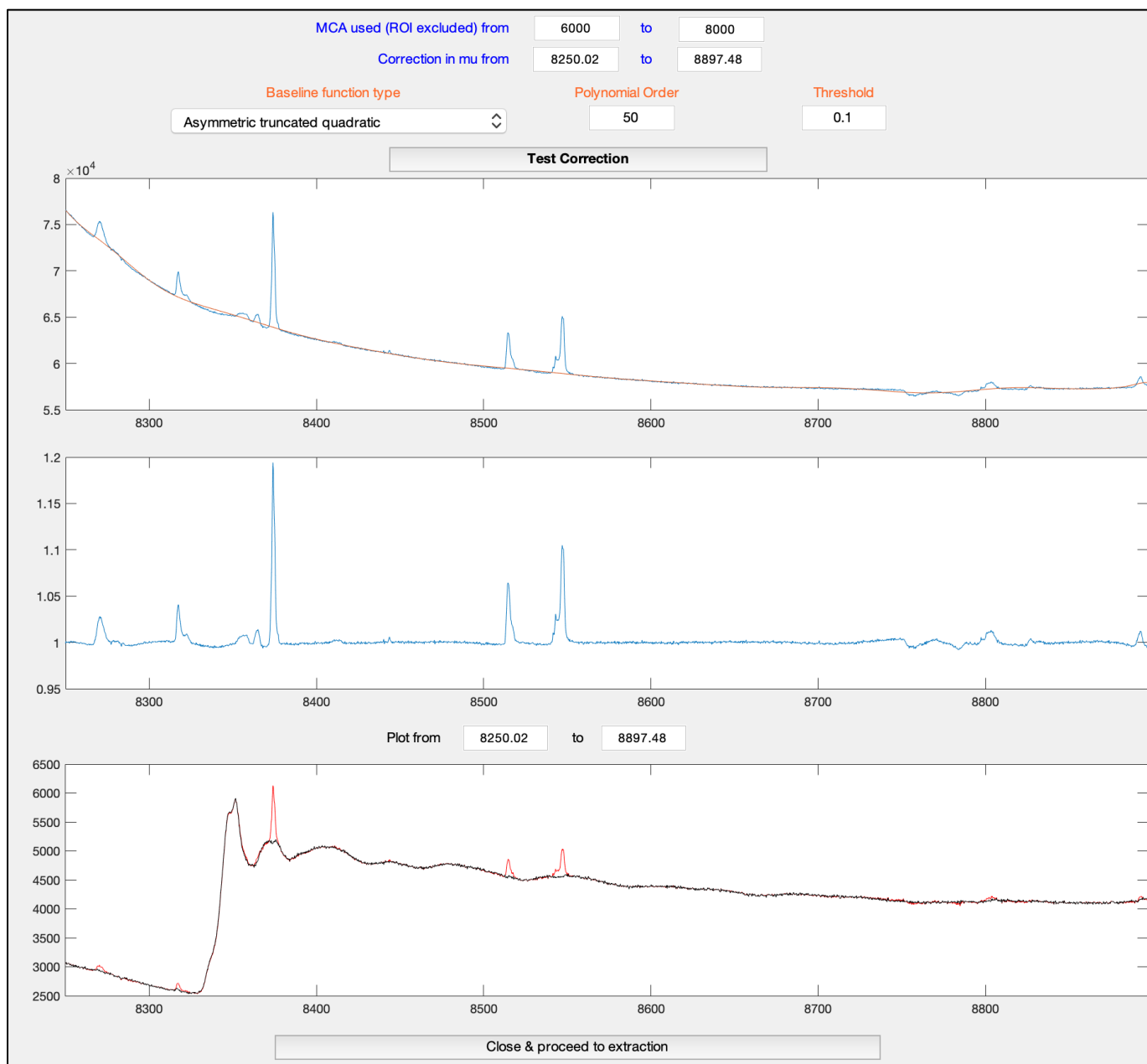


Figure 121 - Diffraction corrector window

7600, then the actual MCA part used by the Diffraction corrector tool will be 6000-7290 & 7610 to 8000 eV.

- “Correction in mu”

The part of the XAFS spectrum that is corrected by the Diffraction corrector tool. By default, this region corresponds to the whole energy range of the XAFS spectrum. In some cases, this region may be shortened. For example, if the white line of the XAFS spectrum is very intense, and the resolution of the fluorescence detector is not high, the fluorescence of the studied element may “leak” (“tailing effect”) inside the MCA region used by the Diffraction corrector window at energies around the white line. Real parts of the XAFS signal around the energy of the white line may then be considered as artefacts and removed by the Diffraction corrector tool! In that case, to avoid cutting the real data, the region of the XAFS spectrum to correct should be shortened, for example from an energy after the white line to energy corresponding to the end of the XAFS spectrum.

- Baseline parameters: function type, polynomial order, threshold

The codes of Mazet *et al.* (2005) are used to calculate the baseline. The “asymmetric truncated quadratic” (default function type choice) should be adequate in most cases. A polynomial order equal to 3 (default choice) is suitable for a baseline that is mostly flat. If the baseline does not match a curvy data trend, increase its polynomial order to an adequate value. Decreasing the threshold value decreases the level of the baseline below the data trend, hence more features in the MCA pattern can be removed by the diffraction corrector tool.

#### 9.2.4.4.3 Example

An example of an XAFS spectrum acquired at the Ni K edge and processed in the diffraction corrector tool is shown in Figure 121. When the diffraction tool window was firstly opened, the default limit values of the MCA region were 10-7290 eV, knowing that the ROI was 7300-7600 eV. It was empirically found that these limit values were not suitable and changed to 6000-8000 eV. The actual MCA region used by the tool was then 6000-7290 & 7610 to 8000 eV since the ROI is always excluded. This MCA region included the K alpha and beta emission of Fe (Figure 122). A high polynomial order was used so that the baseline could well fit the data. The default type of baseline function and threshold values were employed. All obvious distortions were successfully removed from the XAFS spectrum, as shown at the bottom of Figure 121.

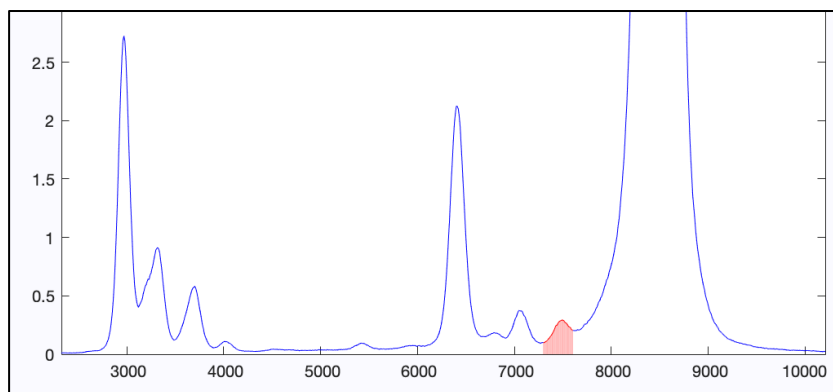


Figure 122 – Same sample as the example shown in Figure 121 : MCA at an energy above the Ni K edge

#### 9.2.4.5 Method number

The “Method number (#)” on the left side of the MCA viewer window enables to consecutively apply specific data treatment procedures to different group of pixels for the same HDF file. Since the total fluorescence spectrum collected by a multi pixel fluorescence detector is the sum of all fluorescence collected at each pixel, the multiple fluorescence spectra obtained for different groups of pixels, using various data treatment methods, can be eventually summed up together.

For example, the data of a sample analyzed in fluorescence using the 36 pixel Ge detector is loaded to the MCA Viewer. Suppose that everything went wrong with this sample during the acquisition: diffraction peaks significantly affected the MCA background for pixels 1 to 10 and an electronic issue decalibrated the MCA of pixels 20 to 25. The data of all remaining pixels are OK. In that case, a new fluorescence spectrum can be extracted in three consecutive steps, using a specific method for each step. In the first step (“Method # 1” is shown in the MCA Viewer window): pixels 1 and 10 are selected, all remaining pixels are not selected. A background subtraction method is applied to pixels 1 to 10. Then, the button “Apply & Extract” is pressed: the fluorescence spectrum for this pixel section is extracted. Once the extraction is completed, the MCA Viewer now displays “Method # 2”, pixels 1 and 10 are now disabled (since they have been used to extract data in the previous method), and all remaining pixels are now selected by default. Firstly, for this new

method, all pixels are unchecked except those corresponding to pixels 20 to 25. The ROI is then modified for pixels 20 to 25 to compensate for the MCA decalibration. Then, the button “Apply & Extract” is pressed: the fluorescence spectrum relative to pixels 20 to 25 is extracted, and added to the fluorescence spectrum obtained with Method # 1. Once the extraction of Method # 2 is completed, the MCA viewer now displays “Method # 3”, pixels 1 and 10 and pixels 20 to 25 are disabled, all remaining pixels are now selected. Since all these remaining pixels are OK, they are left selected, and the button “Apply & Extract” is pressed (i.e. no ROI modification or background subtraction is done on the remaining pixels): the fluorescence spectrum obtained for this last pixel group is extracted and added to the total fluorescence spectrum obtained with Method # 1 & 2. The final fluorescence spectrum can be saved and/or transferred to the main GUI.

### 9.3 Information Plot (SAMBA data only)

This module enables to plot against each other the values of two types of contextual information recorded in a set of HDF files corresponding to a data set. The contextual information can be time or the value of a motor/sensor employed in the SAMBA setup, at the beginning or end of a scan iteration.

Firstly, select all scan iterations in the Main GUI that correspond to the data set. Make sure that the data was imported to the Main GUI as HDF files. Then, open the “Information Plot” module from the menu bar of the Main GUI:

Main GUI Menu > Operations >For SAMBA’s files only...> “Information Plot”

For example, the temperature in a SAMBA oven at the beginning and end of 60 successive scan iterations is plotted against time in Figure 123. The iteration number of each spectrum, from 1 to 60, is also displayed in the plot. The temperature in the oven was saved in each of the sixty HDF files as “forno\_set”.

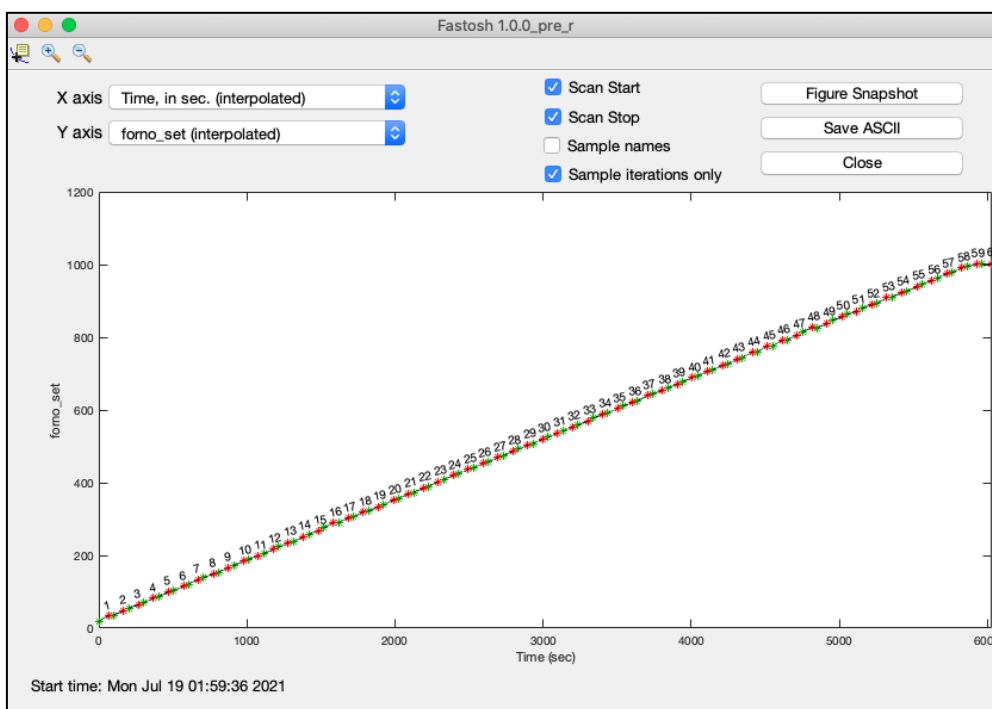


Figure 123 -Temperature in a SAMBA oven plotted against time for a data set of 60 scan iterations. The time & date at the beginning of the first iteration is shown at the bottom left corner of the window.

An interactive tool available on the top left corner of the window allows to display the X and Y values corresponding to a specific data point. For example, when clicking with this tool on the data point collected at the end of the 60<sup>th</sup> iteration in the plot shown in Figure 123, one can see that the temperature in the oven was 1002°C, and 6024 seconds went by since the beginning of the first scan iteration (Figure 124).

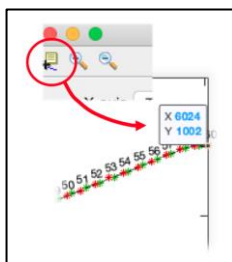


Figure 124 -An interactive tool is available on the right left corner of the window to display the exact X and Y values corresponding to a specific data point in the scatter plot

Lastly, a snapshot of the plot can be taken, and the data can be saved as an ASCII file. When generating an ASCII file, the data between the start and end of each scan iteration is interpolated. For example, if a scan has 2500 data points, the time values corresponding to all 2500 data points will be saved in the ascii file: they will be inferred by performing a linear interpolation, using the time at the beginning and end of the scan and total number of points (i.e. 2500).

## 9.4 ascan, dscan reader (SAMBA data only)

This function enables to open all dscan and ascan done during a beamtime at samba. Go to:

Main GUI Menu > Operations >For SAMBA's files only...> “Information Plot”

Once the window is open, specify the data folder that contains all dscans and ascans. This should list in green color the name of all ascans, and in purple color the name of all dscans, in chronological order, starting from the latest on top of the list (Figure 125). If more ascans or dscans have been measured, press “refresh folder” to refresh the list. The plot window is interactive; a pointer enables to provide the specific X and Y values in the plot.

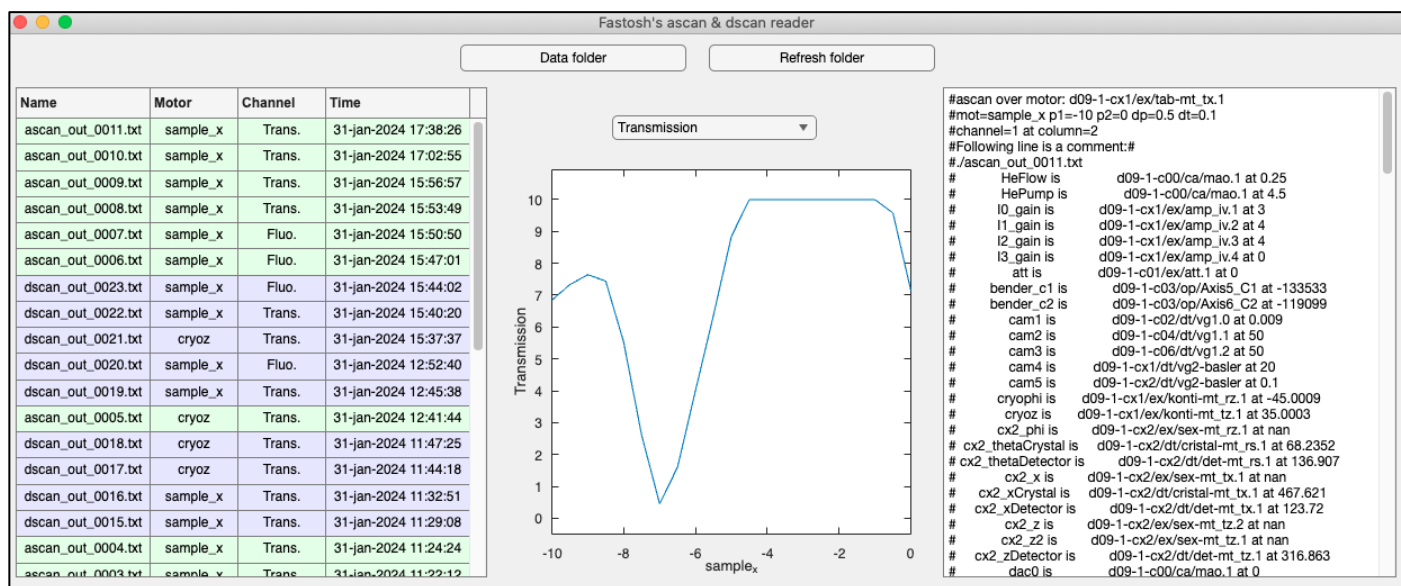


Figure 125 – ascan & dscan reader window

## 10 Acknowledgements

A number of SAMBA Users are acknowledged for their constructive feedbacks on multiples versions of Fastosh, which helped improve the code. Several functionalities were added to the program based on their comments and suggestions.

## 11 Troubleshooting

### Live Viewer:

**Problem:** The spectra automatically displayed in the Live Viewer window look terrible although the data currently being collected should look great! Or the window does not automatically update and keeps on displaying the data of previous acquisitions.

**Solution:** This could be due to a change made at the beamline in acquisition mode or x-ray energy range after opening the Live Viewer window. The Live Viewer strictly displays all data whose nature (i.e. acquisition mode & energy range) matches the one of the spectrum that was imported at opening. For example, if a spectrum in transmission mode was imported, the window will always display transmission data, unless the data type is redefined. The data type can be redefined at any time via the Menu of the Live Viewer window: "Select a data type...".

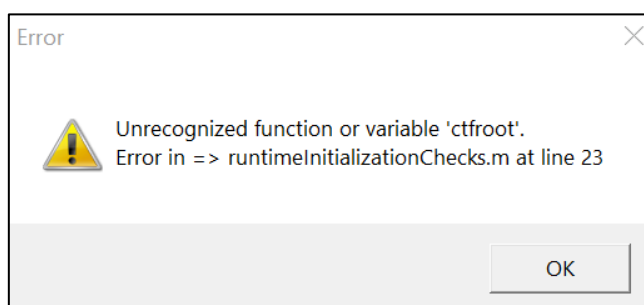
### Jaumot *et al.* MCR-ALS toolbox:

**Problem:** The final window of the toolbox, where the data is fitted by ALS, does not appear. Instead, the toolbox closes, and the main MCR-ALS window of Fastosh reappears.

**Solution:** Most likely, some information was not provided in the toolbox windows where the constraints are set. For example, the non-negativity constraint has been selected, but the number of principal components for which this constraint is applied has not been specified. Reopen the toolbox and make sure that all constraint information are specified following the instructions provided in this manual.

### Fastosh won't start and an error message appears:

**Problem:** On Windows systems, Fastosh won't start. Instead, the following message appears:



**Solution:** This is a rare, known bug from Matlab. Its cause is unclear; it might occur after an aborted installation. To fix this problem, follow this 3-step procedure:

1. Go to the C:/ Users/ <user name>/AppData/Local/Temp/<user name>/
2. Delete the folder "mcrCache9.8"
3. Start again Fastosh; if it still does not work, reinstall it.

## 12 References

1. Muñoz, M., P. Argoul, and F. Farges, *Continuous Cauchy wavelet transform analyses of EXAFS spectra: A qualitative approach*. American Mineralogist, 2003. **88**(4): p. 694-700.
2. Funke, H., A.C. Scheinost, and M. Chukalina, *Wavelet analysis of extended x-ray absorption fine structure data*. Physical Review B, 2005. **71**(9): p. 094110.
3. Jaumot, J., A. de Juan, and R. Tauler, *MCR-ALS GUI 2.0: New features and applications*. Chemometrics and Intelligent Laboratory Systems, 2015. **140**(Supplement C): p. 1-12.
4. Jaumot, J., et al., *A graphical user-friendly interface for MCR-ALS: A new tool for multivariate curve resolution in MATLAB*. Chemometrics and Intelligent Laboratory Systems, 2005. **76**: p. 101-110.
5. Newville, M., *Larch: An Analysis Package for XAFS and Related Spectroscopies*. Journal of Physics: Conference Series, 2013. **430**: p. 012007.
6. Briois, V., et al., *SAMBA: The 4–40 keV X-ray absorption spectroscopy beamline at SOLEIL*. UVX 2010 - 10e Colloque sur les Sources Cohérentes et Incohérentes UV, VUV et X ; Applications et Développements Récents, 2011: p. 41-47.
7. Malinowski, E., *Theory of error for target factor analysis with applications to mass spectrometry and nuclear magnetic resonance spectrometry*. Analytica Chimica Acta, 1978. **103**: p. 339-354.
8. Solé, V.A., et al., *A multiplatform code for the analysis of energy-dispersive X-ray fluorescence spectra*. Spectrochimica Acta Part B: Atomic Spectroscopy, 2007. **62**(1): p. 63-68.
9. Kelly, S., D. Hesterberg, and B. Ravel. *Analysis of Soils and Minerals Using x ray Absorption Spectroscopy*. 2015.
10. Timoshenko, J. and A. Kuzmin, *Wavelet data analysis of EXAFS spectra*. Computer Physics Communications, 2009. **180**(6): p. 920-925.
11. Ye, Z., *wtEXAFS-021*. 2022.
12. Mazet, V., et al., *Background removal from spectra by designing and minimising a non-quadratic cost function*. Chemometrics and Intelligent Laboratory Systems, 2005. **76**(2): p. 121-133.
13. Martini, A. and E. Borfecchia, *Spectral Decomposition of X-ray Absorption Spectroscopy Datasets: Methods and Applications*. Crystals, 2020. **10**.
14. Manceau, A., M. Marcus, and T. Lenoir, *Estimating the number of pure chemical components in a mixture by X-ray absorption spectroscopy*. J Synchrotron Radiat, 2014. **21**(Pt 5): p. 1140-7.
15. Malinowski, E.R., *Determination of the number of factors and the experimental error in a data matrix*. Analytical Chemistry, 1977. **49**(4): p. 612-617.
16. Ritter, G.L., et al., *Factor analysis of the mass spectra of mixtures*. Analytical Chemistry, 1976. **48**(3): p. 591-595.
17. Windig, W. and J. Guilment, *Interactive self-modeling mixture analysis*. Analytical Chemistry, 1991. **63**(14): p. 1425-1432.

18. Maeder, M., *Evolving factor analysis for the resolution of overlapping chromatographic peaks*. Analytical Chemistry, 1987. **59**(3): p. 527-530.
19. de Juan, A., J. Jaumot, and R. Tauler, *Multivariate Curve Resolution (MCR). Solving the mixture analysis problem*. Analytical Methods, 2014. **6**(14): p. 4964-4976.
20. Myre, J.M., et al., *TNT-NN: A Fast Active Set Method for Solving Large Non-Negative Least Squares Problems*. Procedia Computer Science, 2017. **108**: p. 755-764.
21. Calvin, S., *XAFS for Everyone*. 2013.

