

PyMca, Hyperspectral Data and HDF5

V.A. Solé - European Synchrotron Radiation Facility

HDF5 as hyperspectral data format workshop

PyMca?

PyMca is set of software tools mostly known in the field of XRF analysis

It is certainly a set of programs and widgets for XRF analysis:

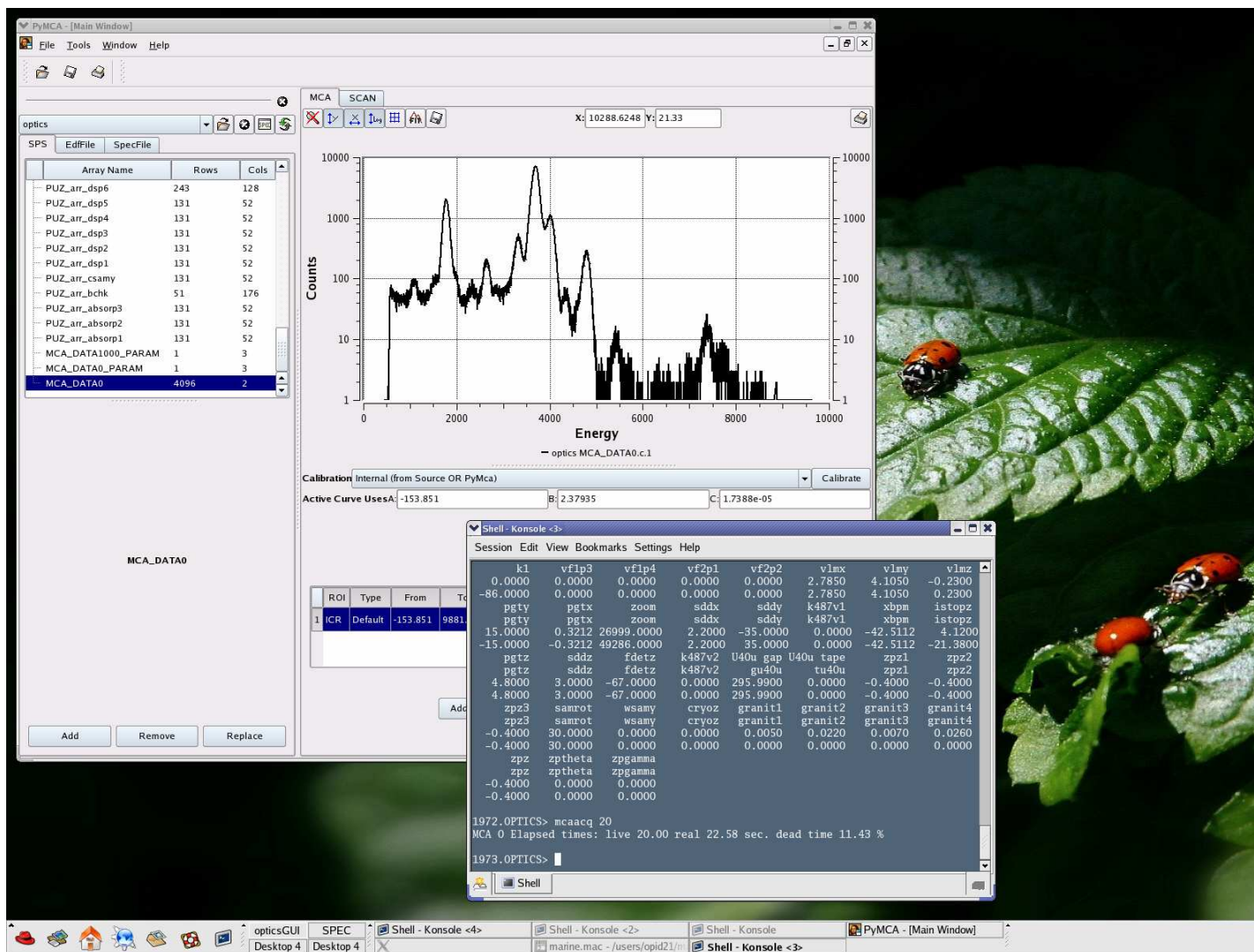
- Spectrum modeling
- Quantification
- ROI imaging
- Fit imaging via batch processing

But also a set of programs and generic python modules for:

- Data visualization
- Peak search
- Function fitting
- Imaging of stacked data

V.A. Solé, E. Papillon, M. Cotte, Ph. Walter, J. Susini, Spectrochimica Acta B 62 (2007) 63-68

Integration into our acquisition system

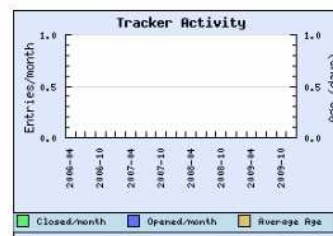
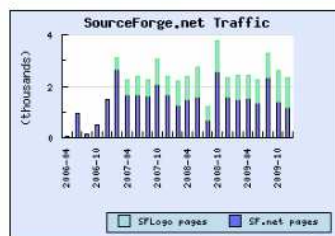
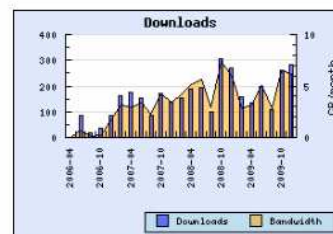
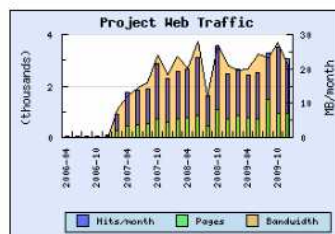


Free distribution

PyMCA

[Share](#)
[Summary](#) | [Files](#) | [Support](#) | [Develop](#) | [Tracker](#) | [Mailing Lists](#) | [Code](#)
[Downloads](#) | [Project Web Traffic](#) | [Tracker Activity](#) | [Forum Traffic](#) | [SourceForge.net Traffic](#) | [Subversion Activity](#)

Usage Statistics For PyMCA

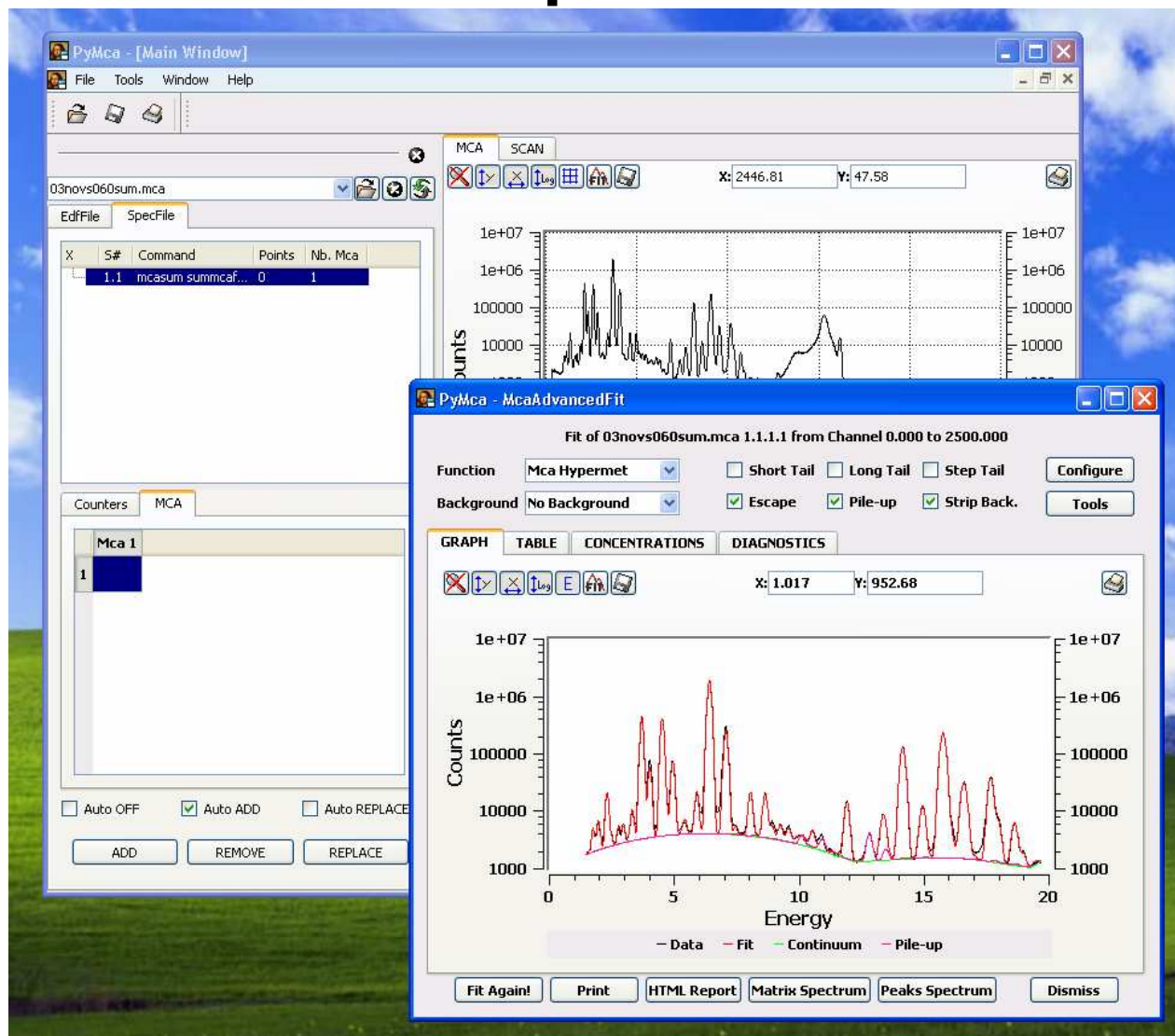


Click a graph for more data

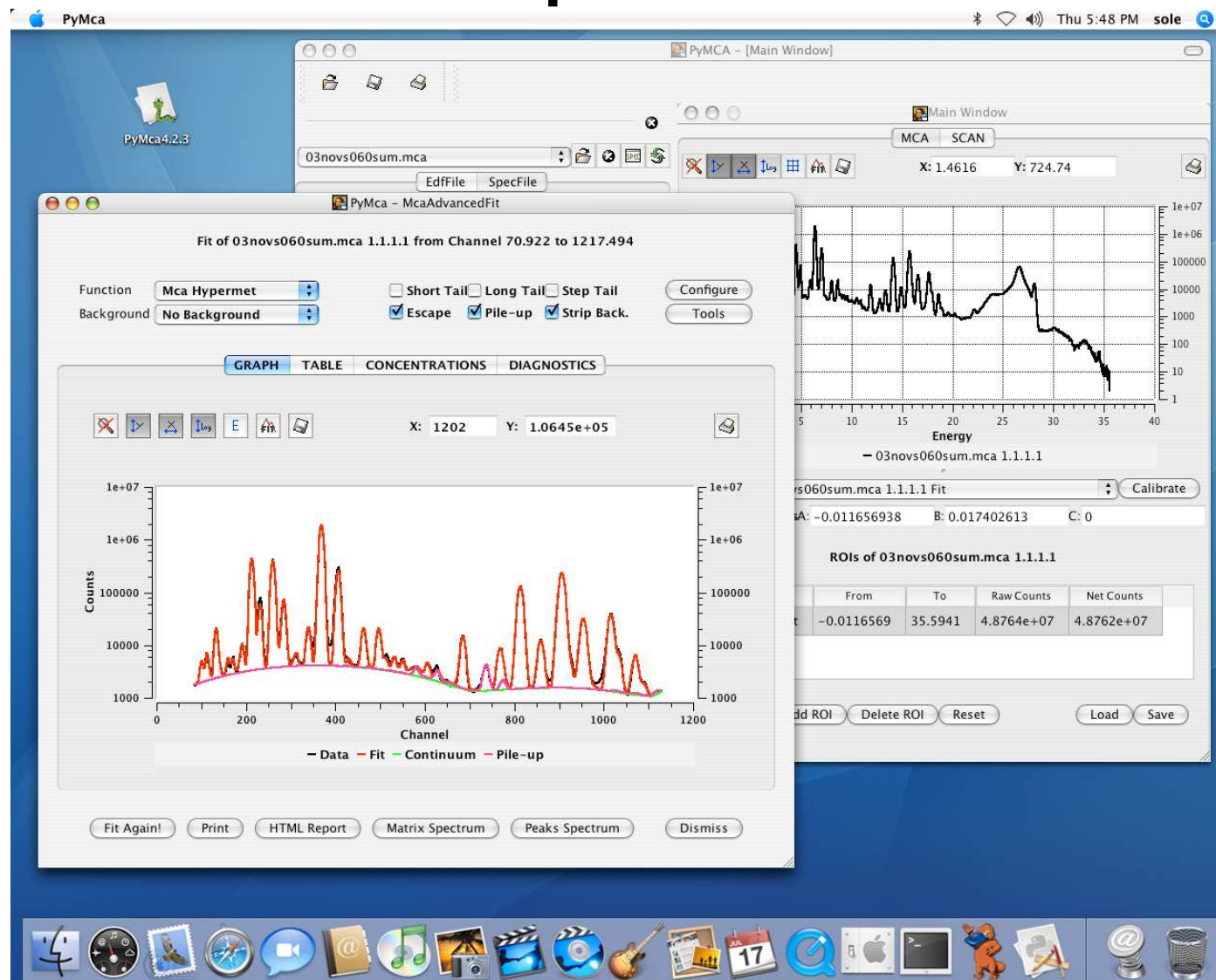
Statistics for All Time Change View

Date (UTC)	Rank	Total Pages	Downloads	Project Web Hits	Tracker opened (closed)	Forum Posts
Jan 2010	2,580	1,068	110	1,047	0 (0)	0
Dec 2009	7,181	2,324	282	3,060	0 (0)	0
Nov 2009	17,994	2,925	319	3,941	0 (0)	0
Oct 2009	16,647	2,624	262	3,526	0 (0)	0
Sep 2009	13,222	2,466	204	2,756	0 (0)	0
Aug 2009	10,046	3,273	106	3,298	0 (0)	0

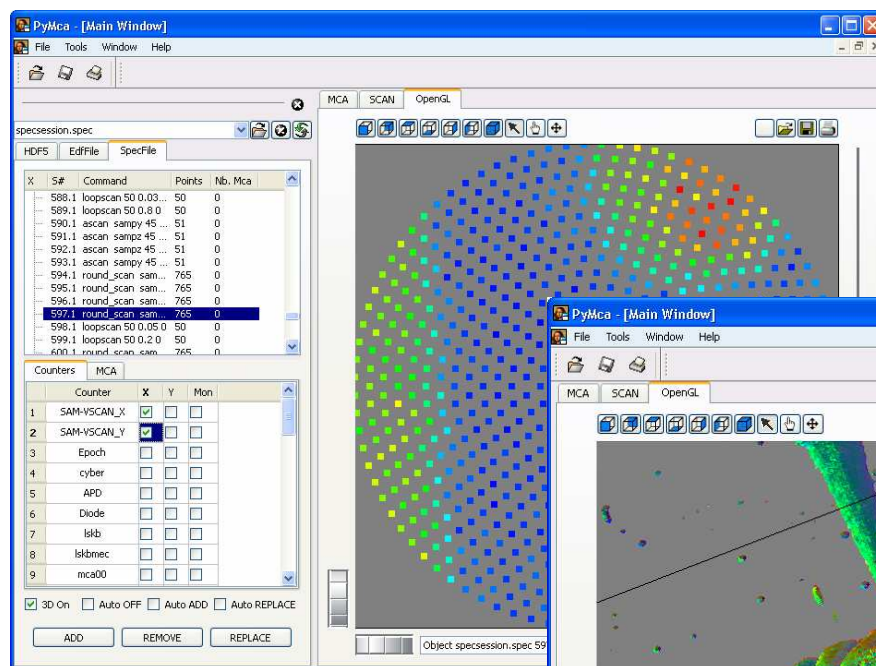
Multiplatform



Multiplatform

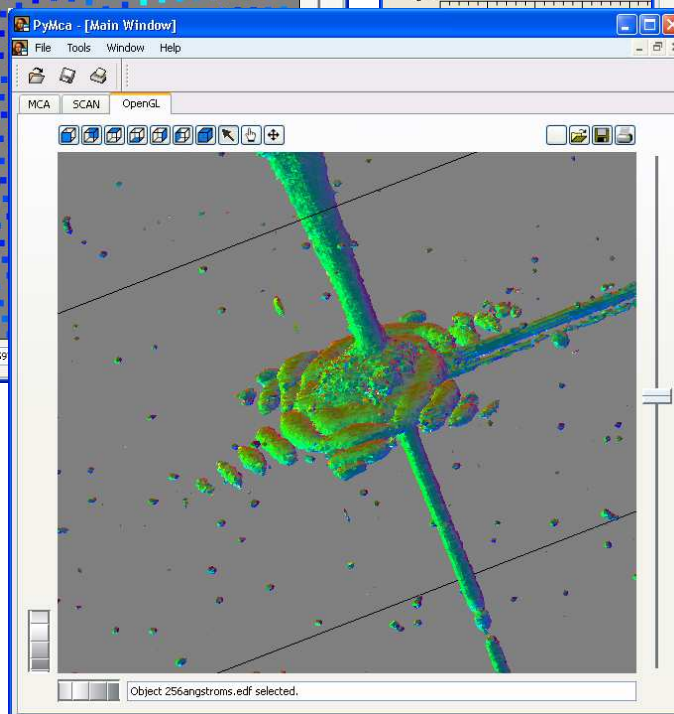


PyMca Visualization

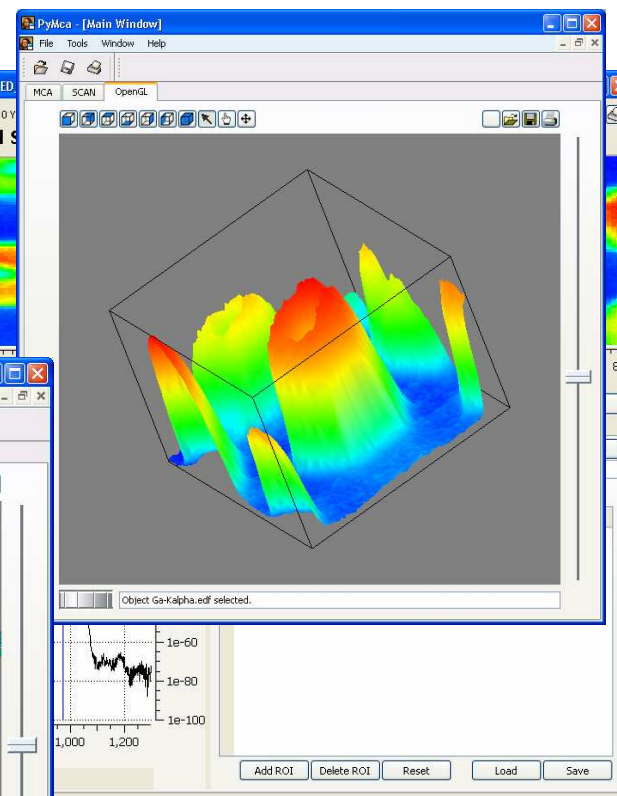


Data courtesy of P. Cloetens

PyMca Object3D Module
Up to 4D visualization

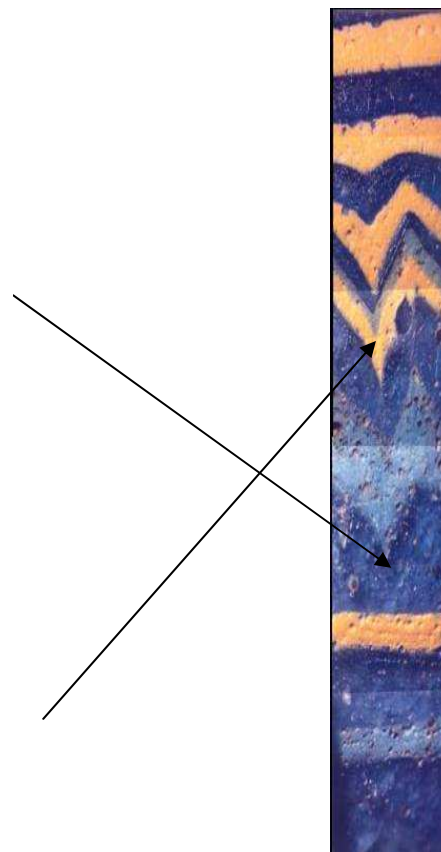
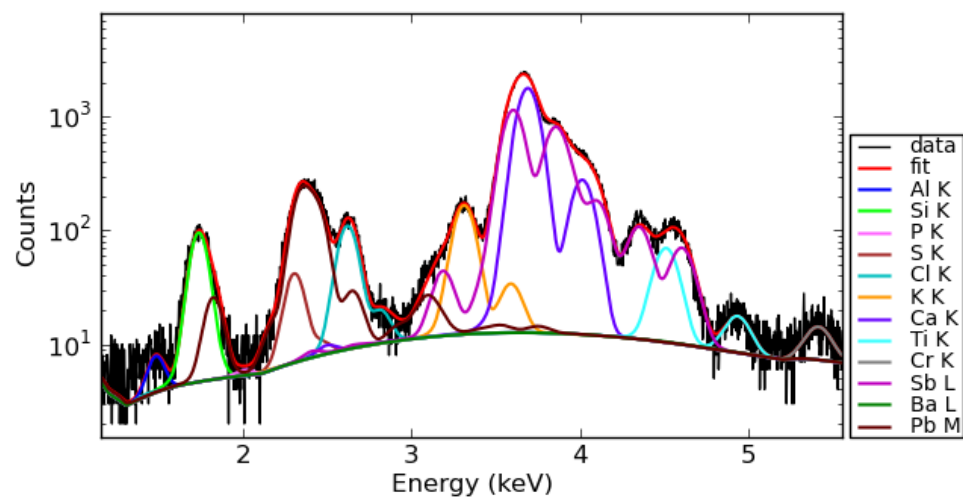
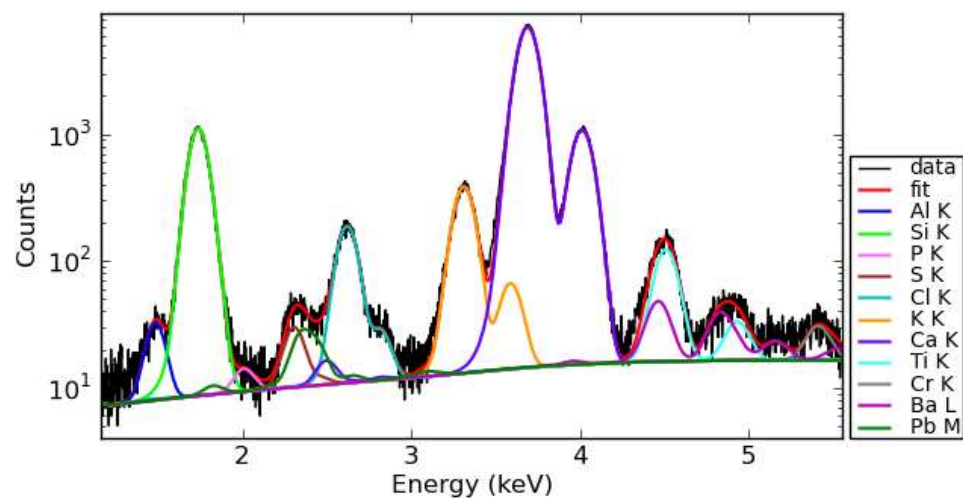


Data courtesy of A. Díaz



Data courtesy of J.A. Sans and G. Martínez

Complete M-shell support



Use of (sub-)shell mass attenuation coefficients

Direct photoelectric ionization of the i (sub-)shell:

$$\text{PyMca} \quad P_i \propto \left(\frac{\tau_i(E, i)}{\tau(E)} \right) \tau(E) \quad \text{versus} \quad P_i \propto \frac{J_i - 1}{J_i} \tau(E)$$

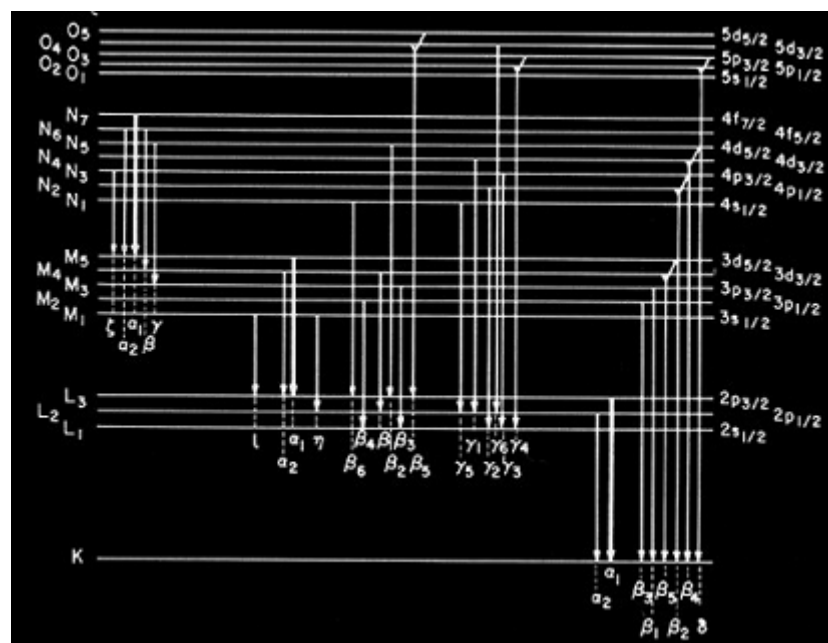
Term between parenthesis for Pb

Energy	16 keV	20 keV	25 keV	30 keV
L1	0.136	0.164	0.192	0.218
L2	0.248	0.247	0.244	0.240
L3	0.367	0.345	0.323	0.303

J.H. Scofield. Theor. Photo. Cross Sections from 1 to 1500 keV, LLNL Report UCRL-51326, Livermore, Ca 1973.

A. Brunetti, M. Sánchez del Río, B. Golosio, A. Simionovici, A. Somogyi, Spectrochimica Acta B 59 (2004) 1725-1731

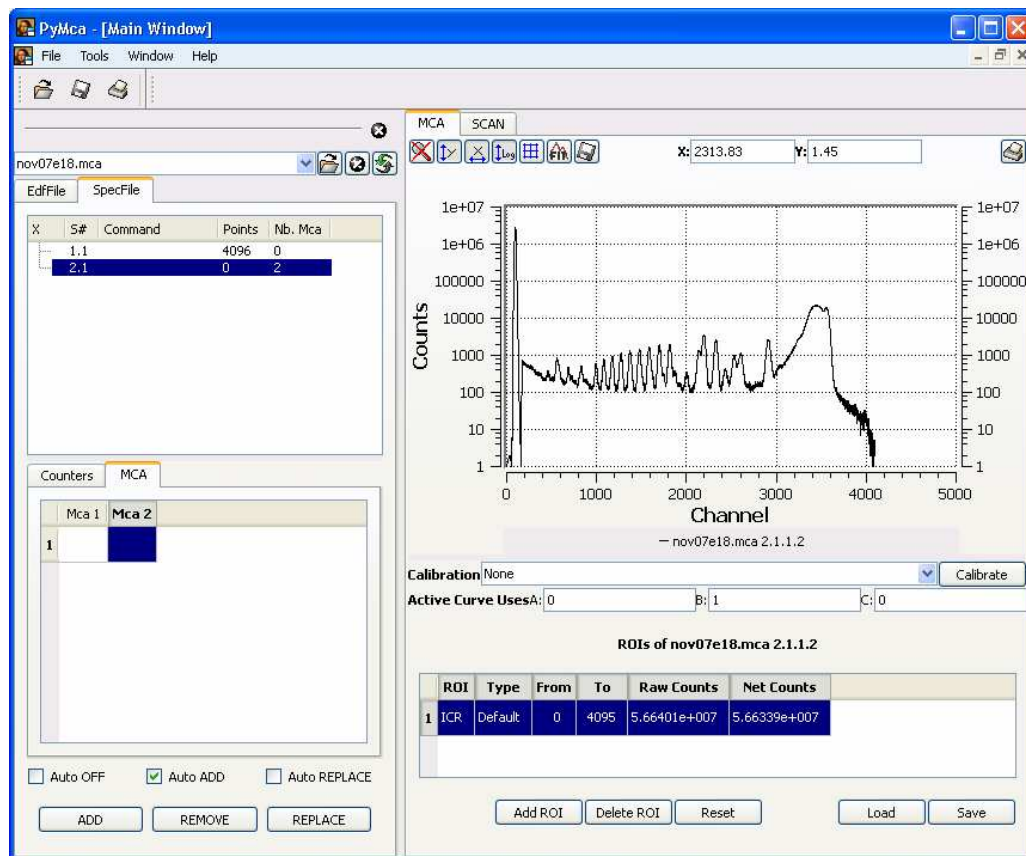
De-excitation cascade taken into account



PyMca considers the vacancies produced in an atomic shell or sub-shell by the de-excitation process of an inner shell or sub-shell.

Radiative and Coster-Kronig transitions correctly considered.
Auger transitions approximated.

XRF Spectrum Analysis



Typical procedure:

1. Calibration
2. Peak identification
3. Peak area extraction
Region of interest (ROI)
Deconvolution (FIT)
4. Quantification

Documentation at <http://pymca.sourceforge.net/documentation.html>

Fit configuration Dialog (I)

PyMca - MCA Fit Parameters

FIT DETECTOR BEAM PEAKS PEAK SHAPE ATTENUATORS MATRIX CONCENTRATIONS

Fit Function: Mca Hypermet

Continuum type: NO Continuum

Polynomial order: 1

Strip Background Width: 10

Strip Background Iterations: 6000

Strip Background Smoothing Width (Savitsky-Golay): 4

☐ Strip Background use Anchors: 3400 290 0 0

Statistical weighting of data: Poisson (1/Y)

Number of fit iterations: 10

Minimum χ^2 difference (%): 0.001

☐ Perform a Linear Fit Fixing non-linear Parameters to Initial Values

☒ Limit fitting region to : First channel: 290 Last channel: 3400

Include:

☒ Stripping ☒ Escape peaks ☐ Pile-up peaks ☐ Scattering peaks

☒ Short tail ☐ Long tail ☐ Step tail

Load From Fit Load Save Cancel OK

PyMca - MCA Fit Parameters

FIT DETECTOR BEAM PEAKS PEAK SHAPE ATTENUATORS MATRIX CONCENTRATIONS

Detector Composition: Si

Maximum Number of Escape energies: 4

	Fixed	Value	Delta
Spectrometer zero (keV)	<input type="checkbox"/>	-0.49276216102	+/- 0.1
Spectrometer gain (keV/ch)	<input type="checkbox"/>	0.00502845949602	+/- 0.001
Detector noise (keV)	<input type="checkbox"/>	0.1	+/- 0.05
Fano factor (Si ~ 0.12, Ge ~ 0.1)	<input type="checkbox"/>	0.15	+/- 0.114
Pile-up Factor	<input type="checkbox"/>	1e-008	+/- 1e-008

PyMca - MCA Fit Parameters

FIT DETECTOR BEAM PEAKS **PEAK SHAPE** ATTENUATORS MATRIX CONCENTRATIONS

	Fixed	Value	Error
Short Tail Area	<input type="checkbox"/>	0.04	+/- 0.03
Short Tail Slope	<input type="checkbox"/>	0.6	+/- 0.49
Long Tail Area	<input type="checkbox"/>	0.2	+/- 0.015
Long Tail Slope	<input type="checkbox"/>	10.0	+/- 7.0
Step Height	<input type="checkbox"/>	0.0005	+/- 5e-005
Pseudo-Voigt Eta	<input type="checkbox"/>	0.2	+/- 0.2

Fit Configuration Dialog (II)

PyMca - MCA Fit Parameters

FIT DETECTOR BEAM PEAKS PEAK SHAPE ATTENUATORS MATRIX CONCENTRATIONS

Excitation Energy (keV) 17.5000

H																	He				
Li	Be															B	C	N	O	F	Ne
Na	Mg															Al	Si	P	S	Cl	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr				
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe				
Cs	Ba	La	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn				
Fr	Ra	Ac	Rf	Db	Sg	Bh	Hs	Mt													
			Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu					
			Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr					

K Ka Kb L L1 L2 L3 M Reset Reset All

Load From Fit Load Save Cancel OK

Quantification (I)

Parallel beam approximation

PyMca - MCA Fit Parameters

FIT DETECTOR BEAM PEAKS PEAK SHAPE ATTENUATORS MATRIX CONCENTRATIONS

Attenuator	Name	Material	Density (g/cm ³)	Thickness (cm)	Funny Factor	
<input type="checkbox"/>	Filter 0	kapton	0.0	0.0	1.0	
<input type="checkbox"/>	Filter 1	atmosphere	0.0	0.0	1.0	
<input type="checkbox"/>	Filter 2	deadlayer	Si1	2.33	0.002	1.0
<input type="checkbox"/>	Filter 3	absorber	-	0.0	0.0	1.0
<input type="checkbox"/>	Filter 4	window	-	0.0	0.0	1.0
<input type="checkbox"/>	Filter 5	contact	Au1	19.37	1e-006	1.0
<input type="checkbox"/>	Filter 6	-	0.0	0.0	0.0	1.0
<input type="checkbox"/>	Filter 7	-	0.0	0.0	0.0	1.0
<input type="checkbox"/>	BeamFilter0	BeamFilter0	-	0.0	0.0	1.0
<input type="checkbox"/>	BeamFilter1	BeamFilter1	-	0.0	0.0	1.0
<input type="checkbox"/>	Detector	Detector	Si1	2.33	0.5	1.0
<input type="checkbox"/>	Matrix	Matrix	MULTILAYER	0.0	0.0	1.0

Plot T(filters) * (1 - T(detector)) Efficiency Term

Material Editor

Material: Air

Number of Compounds: 5

	Material	Mass Fraction
1	C1	0.000124
2	N1	0.75527
3	O1	0.23178
4	Ar1	0.012827
5	Kr1	3.2e-006

Default Density (g/cm³): 0.0012048

Default Thickness (cm): 1

Material Transmission

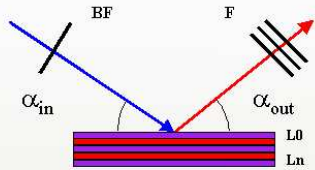
Mass Att. Coefficients

Material Name/Comment: Dry Air (Near sea level) density=0.001204790 g/cm³

Load From Fit Load Save Cancel OK

PyMca - MCA Fit Parameters

FIT DETECTOR BEAM PEAKS PEAK SHAPE ATTENUATORS MATRIX CONCENTRATIONS



Incoming Angle (deg.): 45

Outgoing Angle (deg.): 45

☐ Scattering Angle (deg.): 90

This matrix definition will only be considered if Matrix is selected and material is set to MULTILAYER in the ATTENUATORS tab.

Layer	Name	Material	Density (g/cm ³)	Thickness (cm)
<input type="checkbox"/>	Layer0	Layer0	-	0.0
<input type="checkbox"/>	Layer1	Layer1	-	0.0
<input type="checkbox"/>	Layer2	Layer2	-	0.0
<input type="checkbox"/>	Layer3	Layer3	-	0.0
<input type="checkbox"/>	Layer4	Layer4	-	0.0
<input type="checkbox"/>	Layer5	Layer5	-	0.0
<input type="checkbox"/>	Layer6	Layer6	-	0.0
<input type="checkbox"/>	Layer7	Layer7	-	0.0
<input type="checkbox"/>	Layer8	Layer8	-	0.0
<input type="checkbox"/>	Layer9	Layer9	-	0.0

Load From Fit Load Save Cancel OK

Quantification (II)

Nominal concentration 500 ppm

PyMca - McaAdvancedFit

Fit of nov07e18.mca 2.1.1.2 from Channel 0.000 to 5000.000

Function: Mca Hypermet ☒ Short Tail ☐ Long Tail ☐ Step Tail
Background: No Background ☒ Escape ☐ Pile-up ☒ Strip Back.

GRAPH TABLE CONCENTRATIONS DIAGNOSTICS

☐ From fundamental parameters

Flux (photons/s) 1e+010 x time(seconds) 1
Active Area (mm2) 10 distance (mm) 10

☒ From matrix composition
Matrix Reference Element: Co

☒ Consider attenuators in calculations
☐ Consider secondary excitation from deeper matrix layers (non intralayer nor above layers)
☐ Elemental mM concentrations (assuming 1 l of solution is 1000 * matrix_density grams)

Element	Group	Fit Area	Sigma Area	Mass fraction
4 Ca	K	9.505276e+003	1.56e+002	0.0005063
5 Ti	K	1.488192e+004	1.56e+002	0.0004811
6 V	K	1.896828e+004	1.78e+002	0.0004823
7 Cr	K	2.522468e+004	1.99e+002	0.0004938
8 Mn	K	3.174454e+004	2.17e+002	0.0005054
9 Fe	K	4.015540e+004	2.37e+002	0.0005087
10 Co	K	4.702938e+004	2.51e+002	0.0005
11 Ni	K	5.823319e+004	2.73e+002	0.0004983
12 Cu	K	6.540779e+004	2.85e+002	0.0004967
13 Cu	L	1.981973e+002	3.28e+002	0.0004981
14 Zn	K	7.570149e+004	3.01e+002	0.0004916
15 Zn	L	4.475429e+002	1.43e+002	0.0005415

PyMca - McaAdvancedFit

Fit of nov07e18.mca 2.1.1.2 from Channel 0.000 to 5000.000

Function: Mca Hypermet ☒ Short Tail ☐ Long Tail ☐ Step Tail
Background: No Background ☒ Escape ☐ Pile-up ☒ Strip Back.

GRAPH TABLE CONCENTRATIONS DIAGNOSTICS

☐ From fundamental parameters

Flux (photons/s) 1e+010 x time(seconds) 1
Active Area (mm2) 10 distance (mm) 10

☒ From matrix composition
Matrix Reference Element: Co

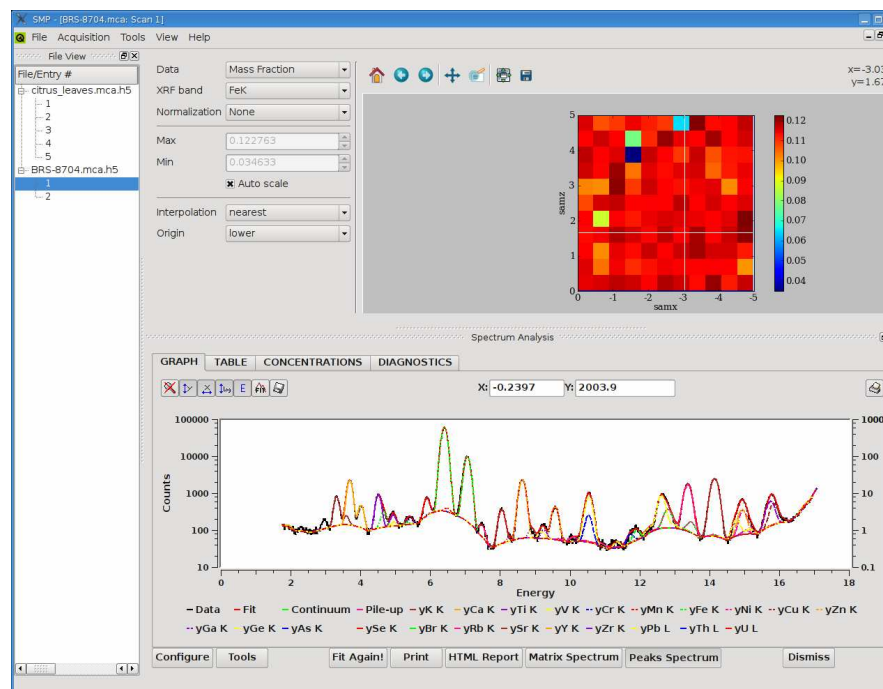
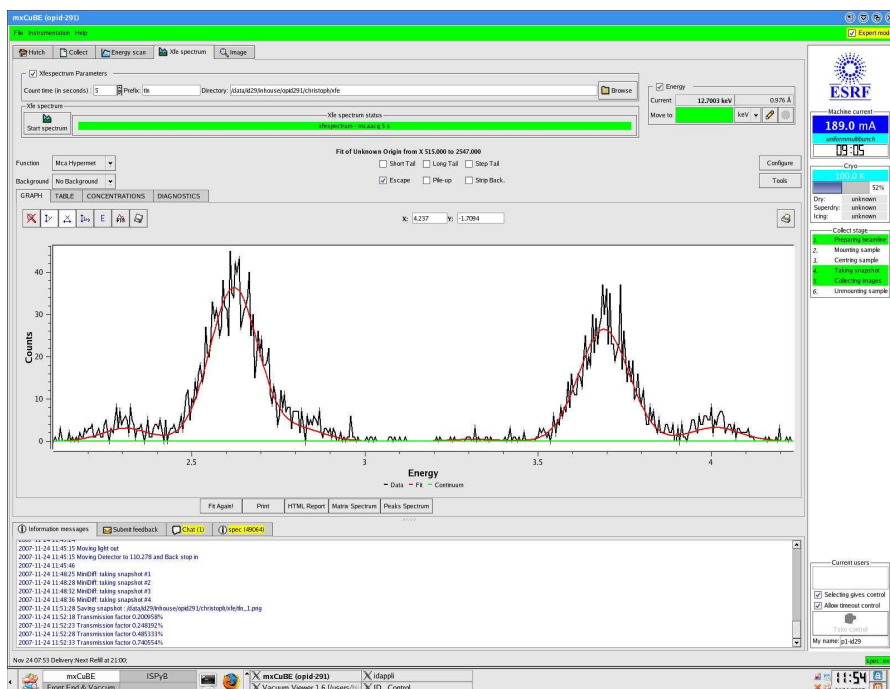
☒ Consider attenuators in calculations
☐ Consider secondary excitation from deeper matrix layers (non intralayer nor above layers)
☐ Elemental mM concentrations (assuming 1 l of solution is 1000 * matrix_density grams)

Element	Group	Fit Area	Sigma Area	Mass fraction
16 As	K	1.025779e+005	5.08e+002	0.0004861
17 As	L	4.211980e+002	1.10e+002	0.0001407
18 Se	K	1.092334e+005	3.70e+002	0.0004862
19 Se	L	9.883950e+002	1.10e+002	0.0002429
20 Sr	K	1.250538e+005	4.21e+002	0.0004871
21 Sr	L	2.772601e+003	1.25e+002	0.0002579
22 Ag	L	4.074779e+003	3.20e+002	0.0005242
23 Sb	L	6.267671e+003	1.88e+002	0.0004673
24 Tl	L	9.733588e+004	4.16e+002	0.0004989
25 Tl	M	9.156083e+003	6.19e+002	0.0002697
26 Pb	L	1.004333e+005	6.03e+002	0.0004878
27 Pb	M	1.461444e+004	2.88e+002	0.0004031

XRF Analysis Integration in other Applications

Integration in mxCuBE (ESRF)

Integration elsewhere



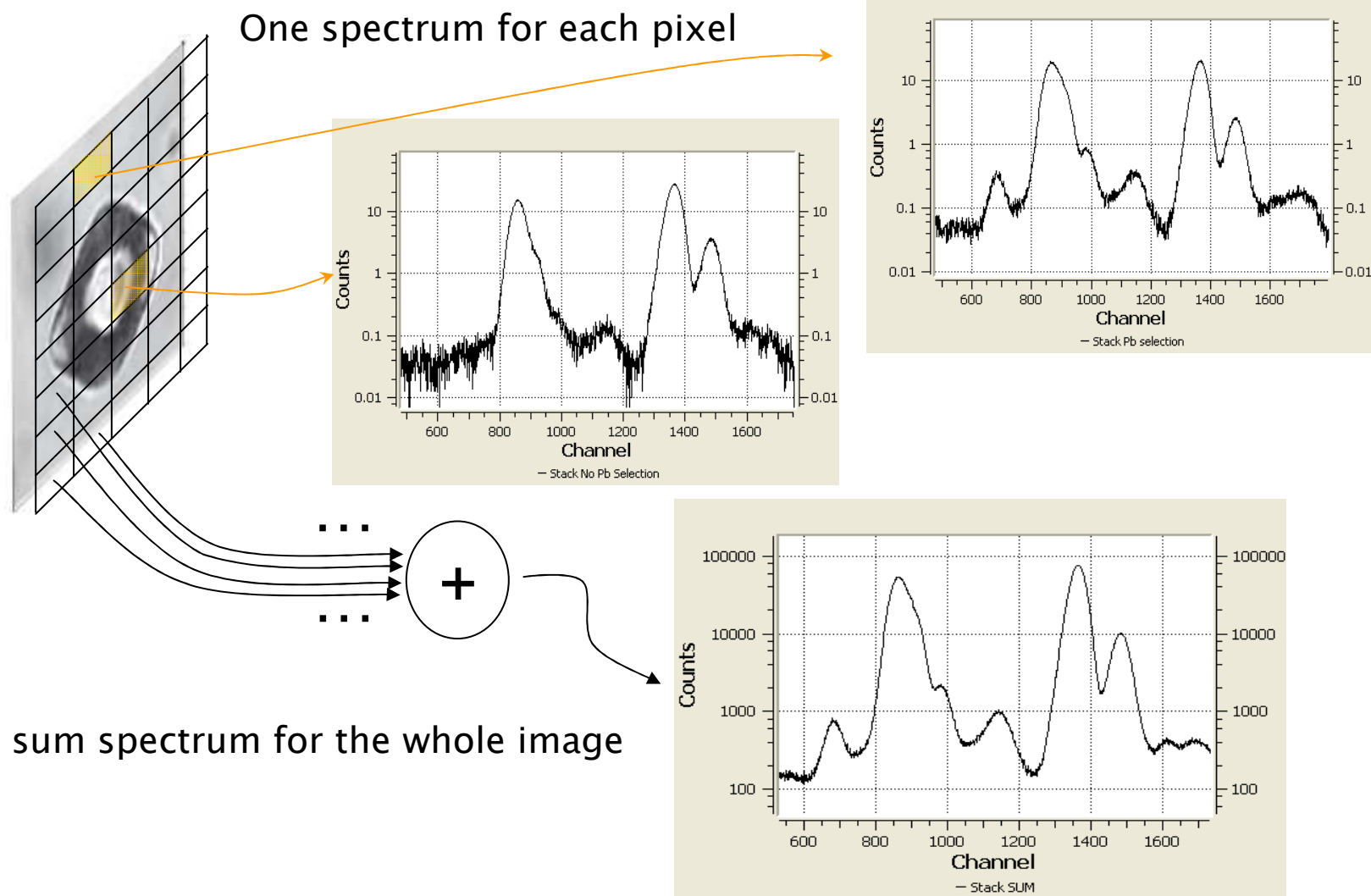
Is it easy to embed?

For the previous examples, basically one just needs 4 lines of code:

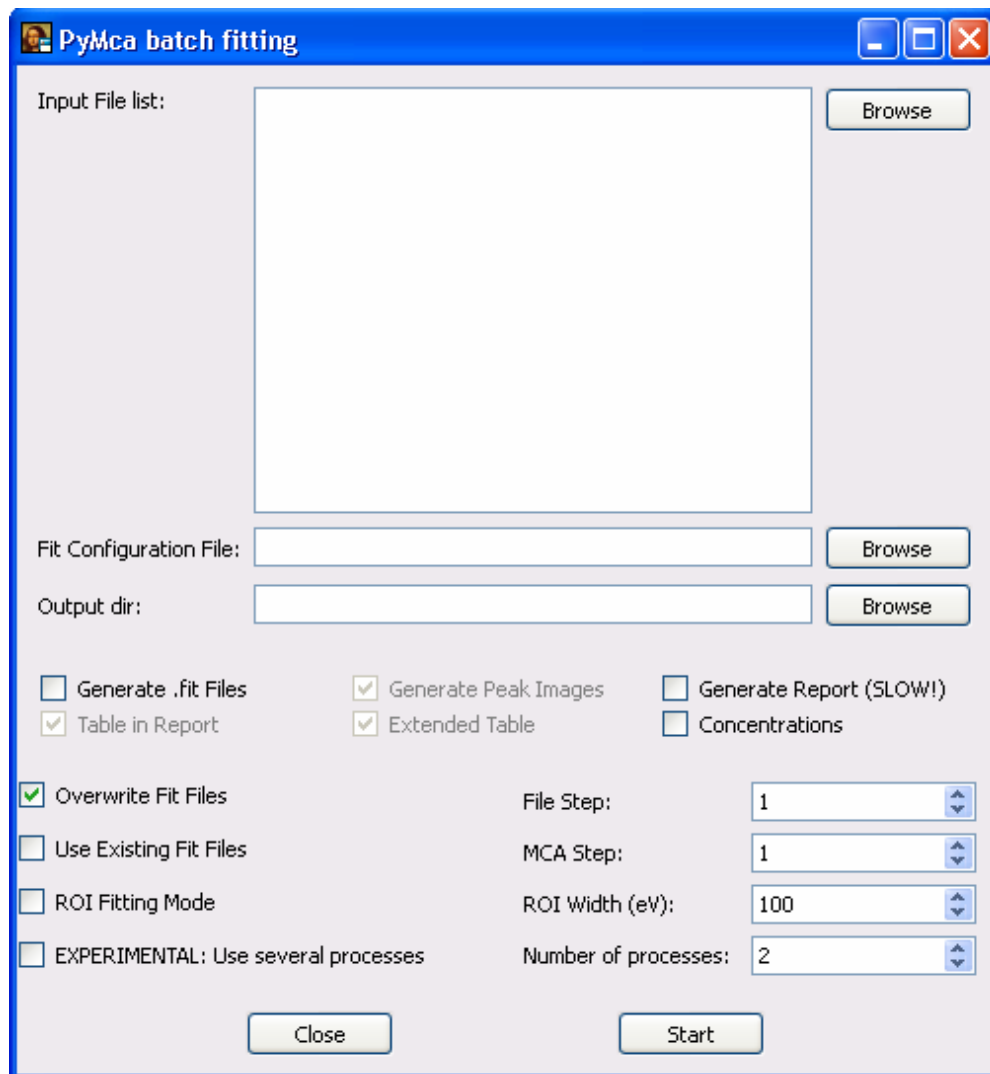
```
from PyMca import McaAdvancedFit  
  
fitWindow = McaAdvancedFit.McaAdvancedFit()  
fitWindow.setData(x, y)  
fitWindow.show()
```

It can be used interactively from ipython just starting it as “ipython -q4thread”

XRF Imaging



Advanced Fit Batch processing



The image shows a Windows-style dialog box titled "PyMca batch fitting". It contains several input fields and checkboxes for configuring a batch fit process.

- Input File list:** A large empty text area for listing input files, with a "Browse" button to its right.
- Fit Configuration File:** A text input field with a "Browse" button to its right.
- Output dir:** A text input field with a "Browse" button to its right.
- Checkboxes:**
 - ☐ Generate .fit Files
 - ☒ Table in Report
 - ☒ Generate Peak Images
 - ☒ Extended Table
 - ☐ Generate Report (SLOW!)
 - ☐ Concentrations
 - ☒ Overwrite Fit Files
 - ☐ Use Existing Fit Files
 - ☐ ROI Fitting Mode
 - ☐ EXPERIMENTAL: Use several processes
- Spinners:**
 - File Step:** A spinner box set to 1.
 - MCA Step:** A spinner box set to 1.
 - ROI Width (eV):** A spinner box set to 100.
 - Number of processes:** A spinner box set to 2.
- Buttons:** "Close" and "Start" buttons are located at the bottom of the dialog.

Select the input files

Select the fit configuration

Select the output directory

Select the output options

Start

Output

Images in ASCII and ESRF format

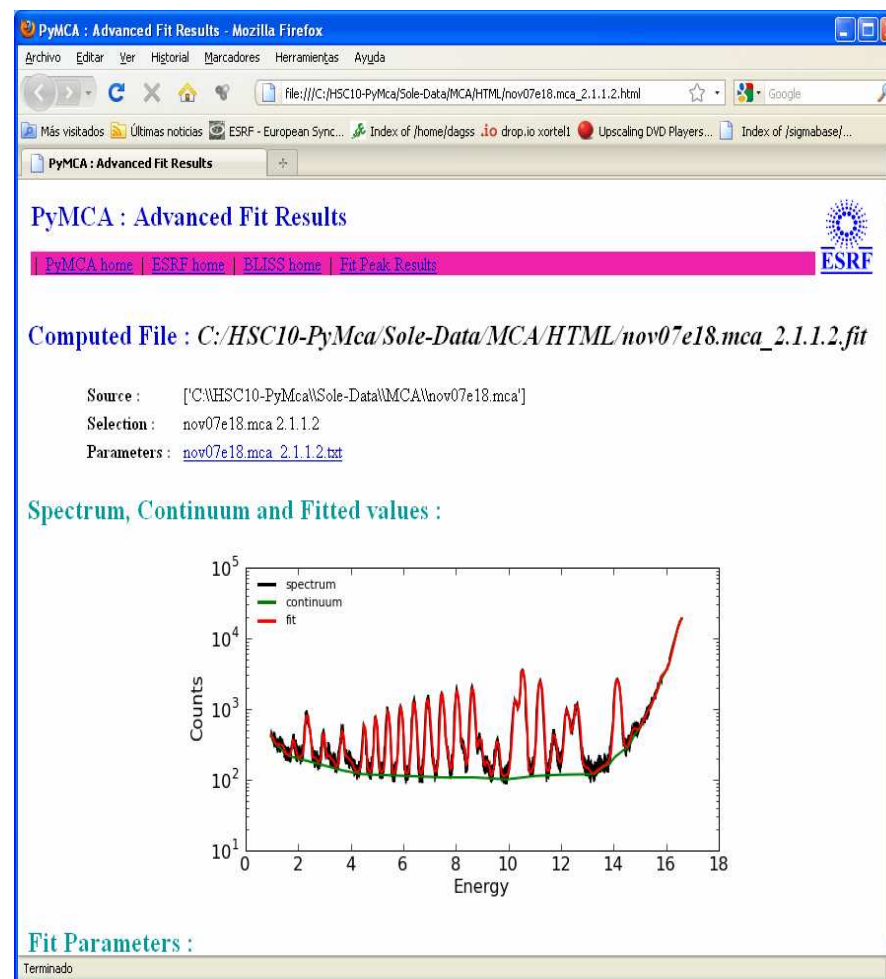
- Easy to import in other programs

Individual peak contributions in ASCII

- Use your own plotting program

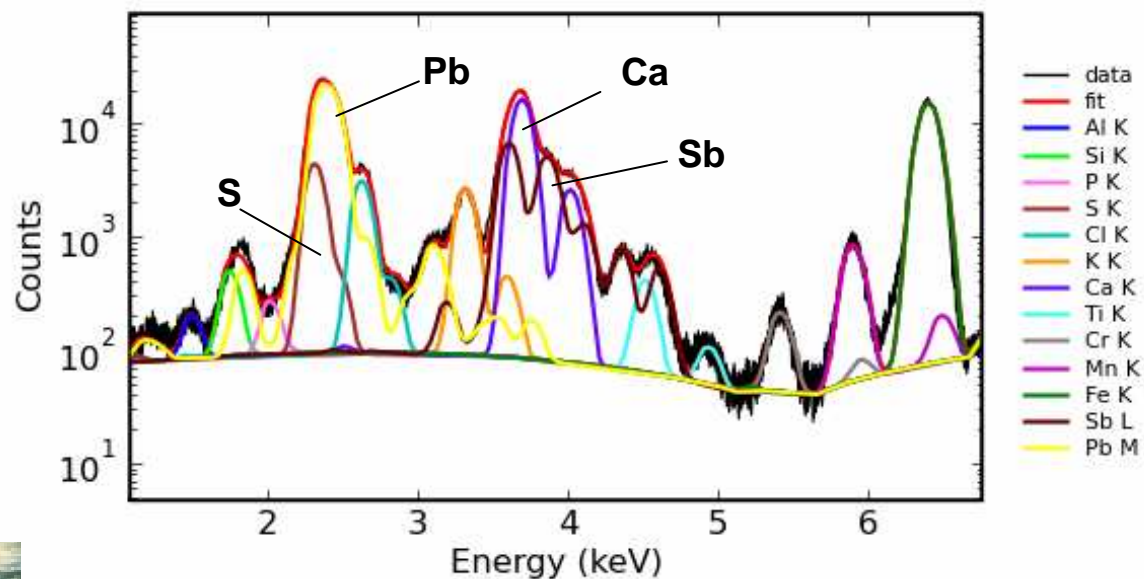
Fully automated HTML report

- Browse your results!



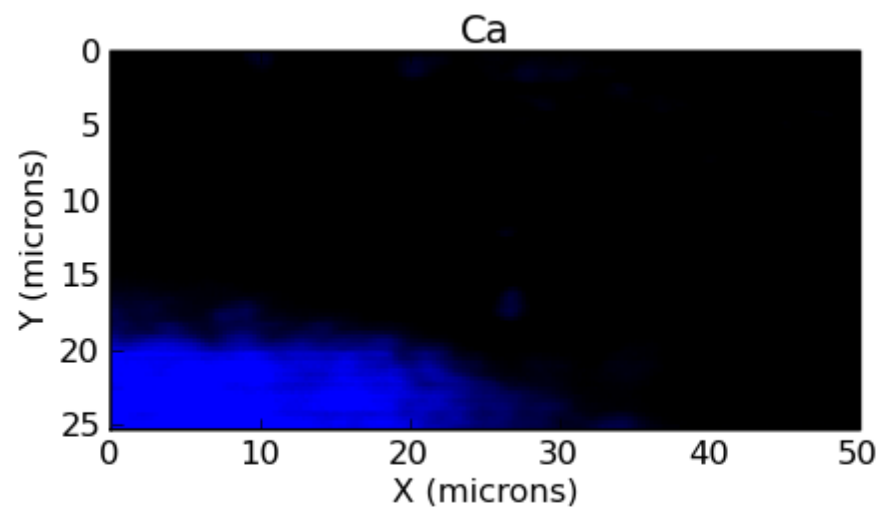
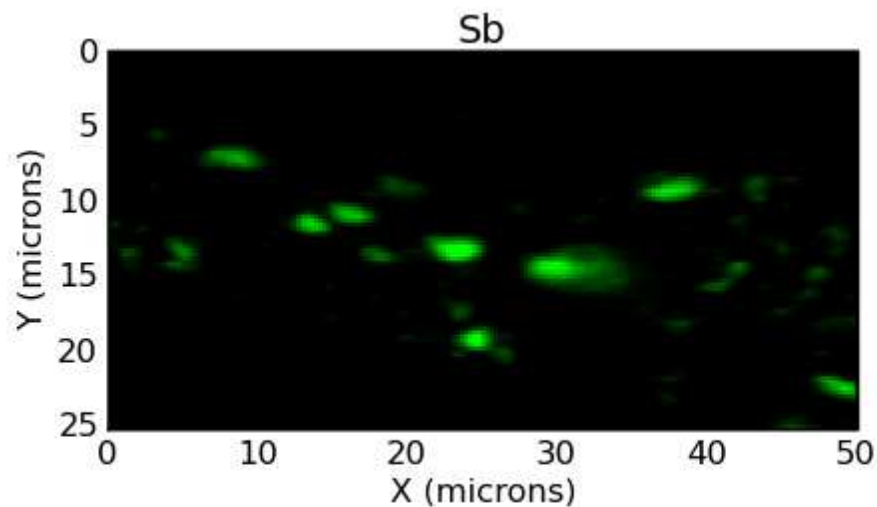
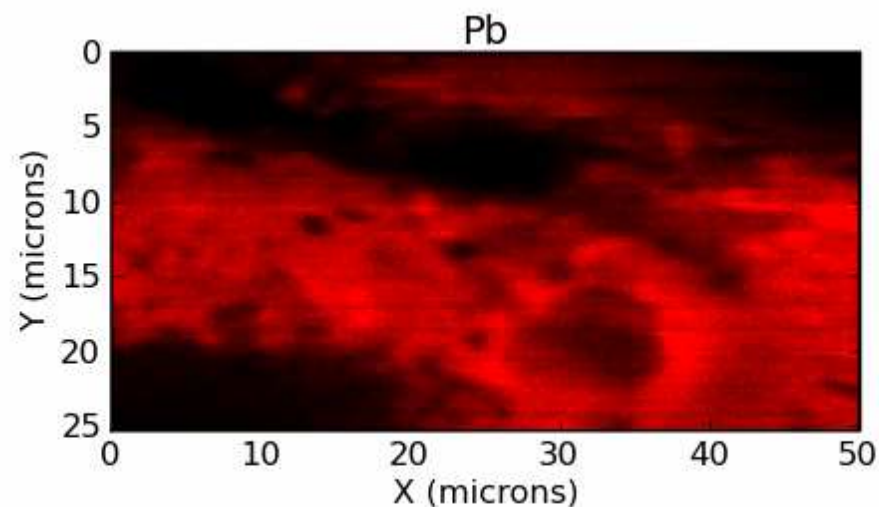
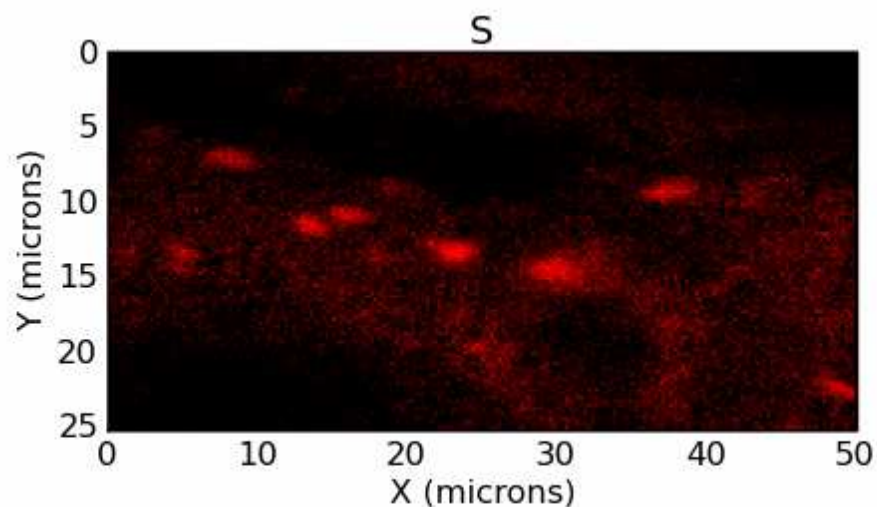


Copyright C2RMF

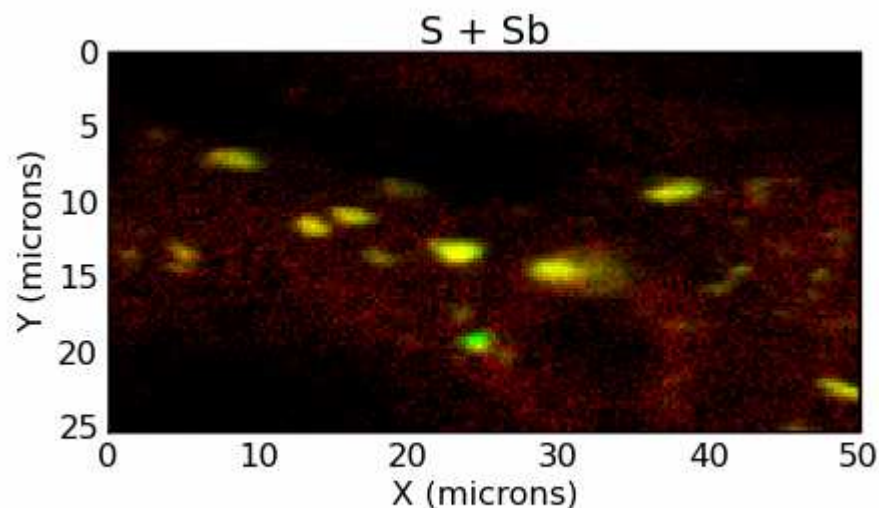


We wanted to know what pigments were used in this section of the painting

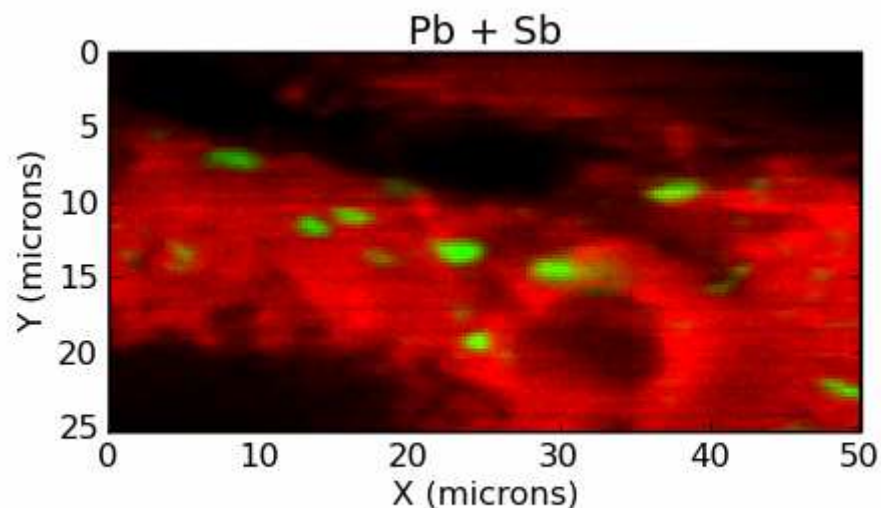
Based on the batch generated element distribution maps ...



... and their correlations as shown by the program



Sulfur and antimony correlated

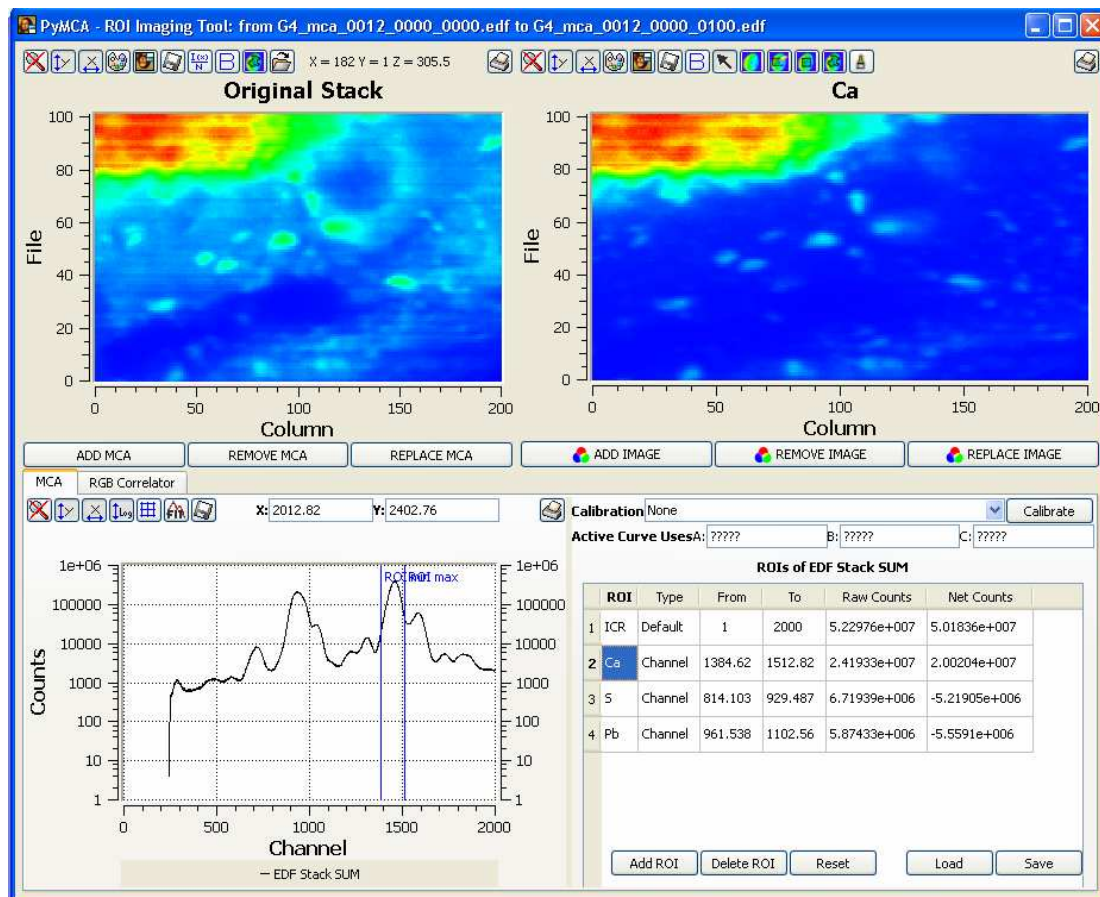


Lead and antimony not correlated

... we were able to determine the possible presence of stibnite grains (Sb_2S_3) embedded in a lead containing matrix.

M. Cotte, E. Welcomme, V.A. Solé, M. Salomé, M. Menu, Ph. Walter, J. Susini, Anal. Chem. 79 (2007) 6988-6994

Stack ROI Imaging



In this example:

Stack = 101x200x2000 numpy array

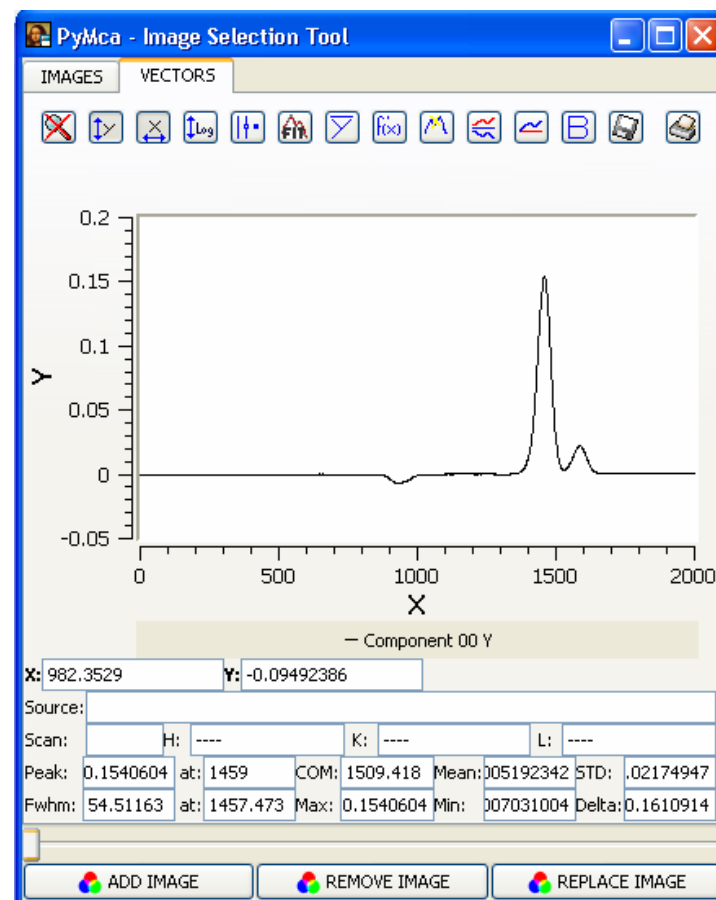
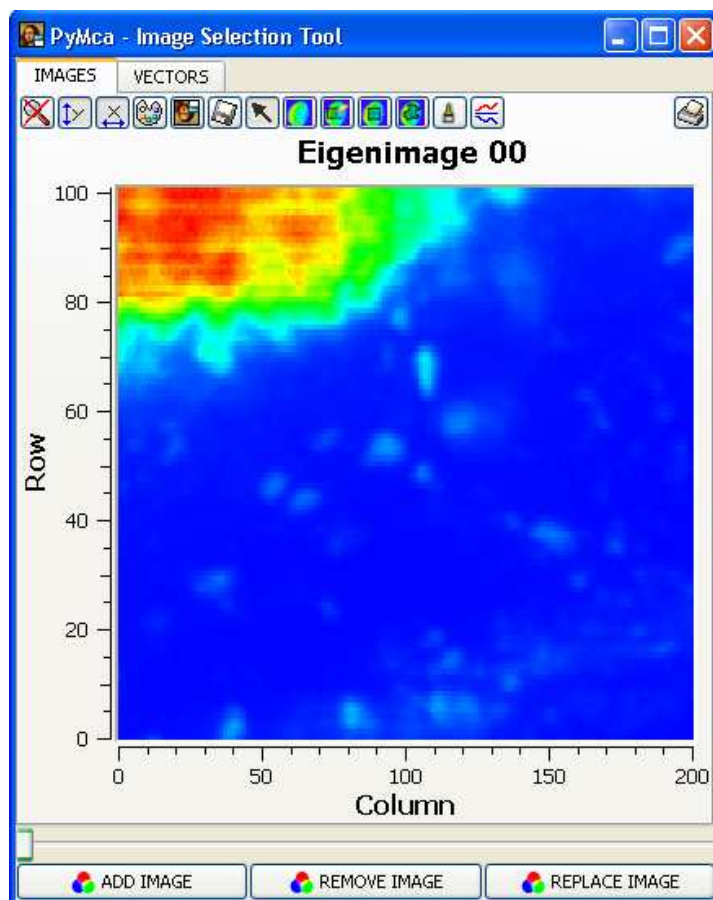
20200 spectra of 2000 channels

$\text{Pixel}[i, j] = \text{numpy.sum}(\text{Stack}[i, j, :])$

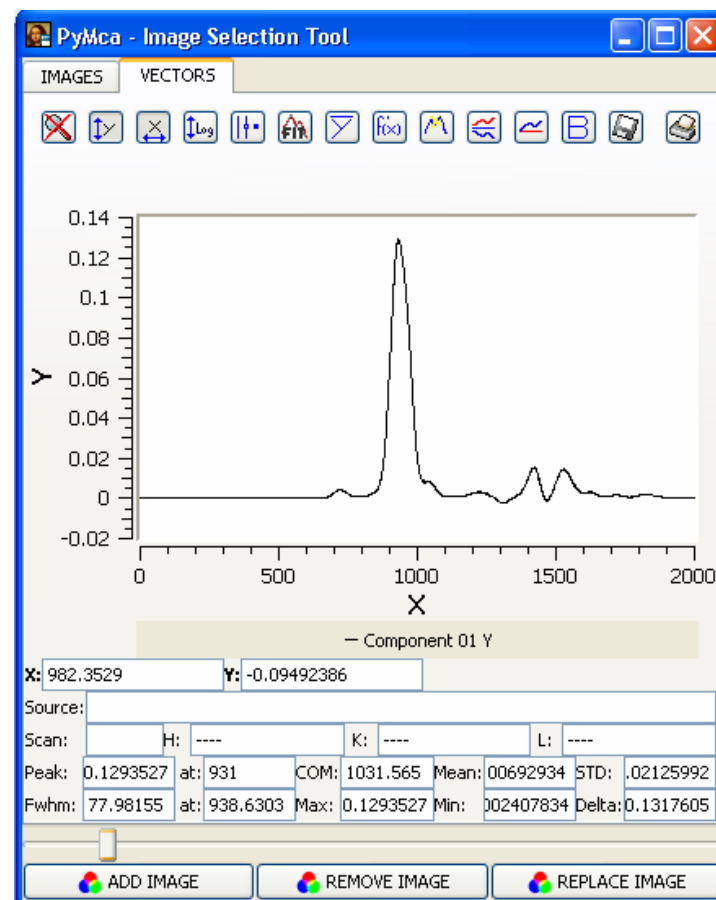
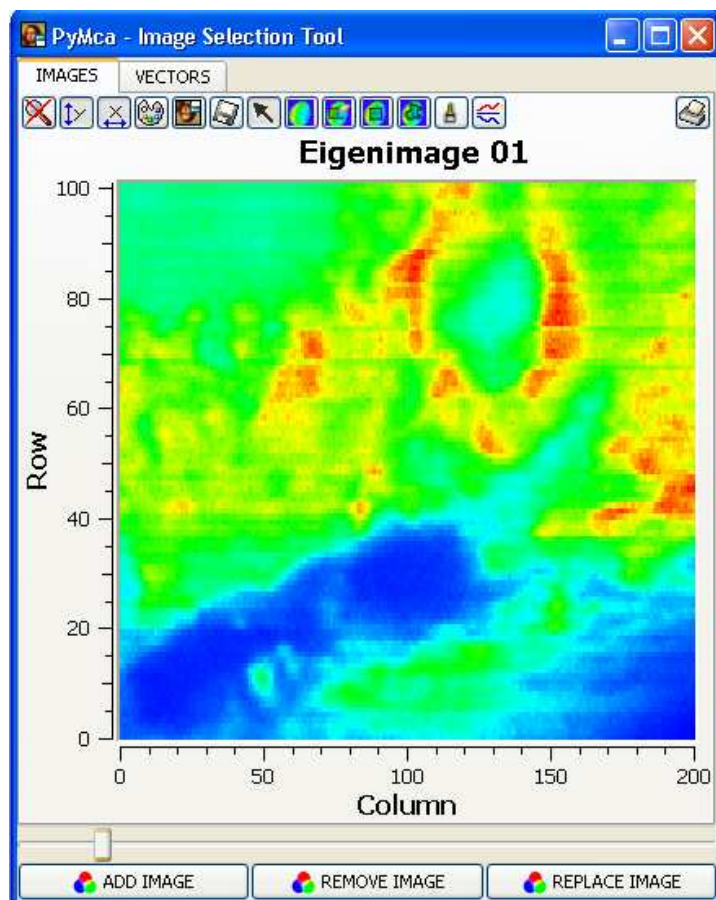
$\text{Pixel}[i, j] = \text{numpy.sum}(\text{Stack}[i, j, \text{ch0}:\text{ch1}])$

We can generate new images by moving the cursors or defining new ROIs in the table

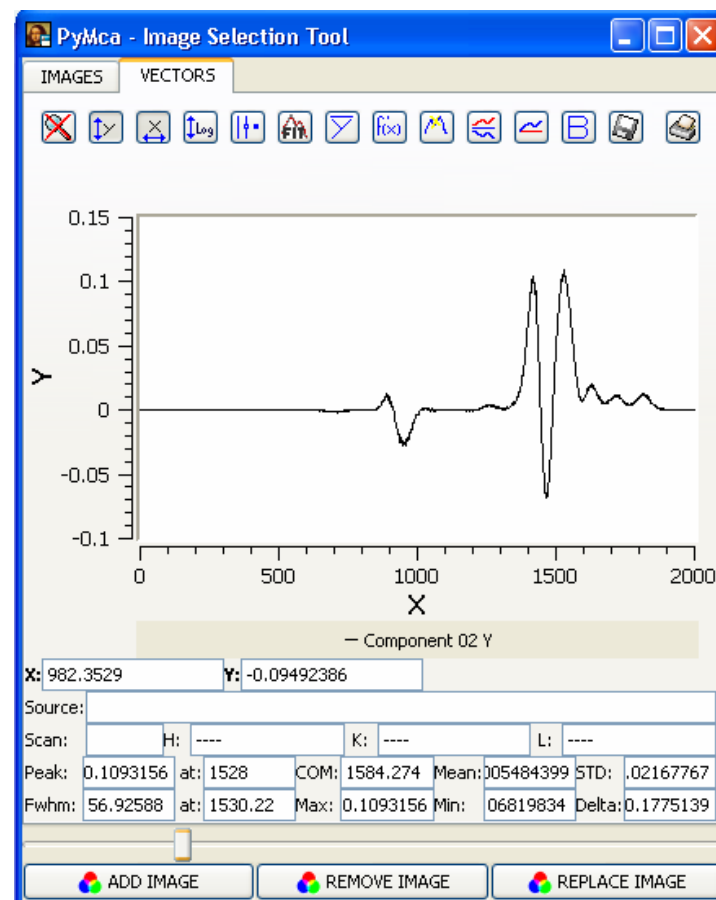
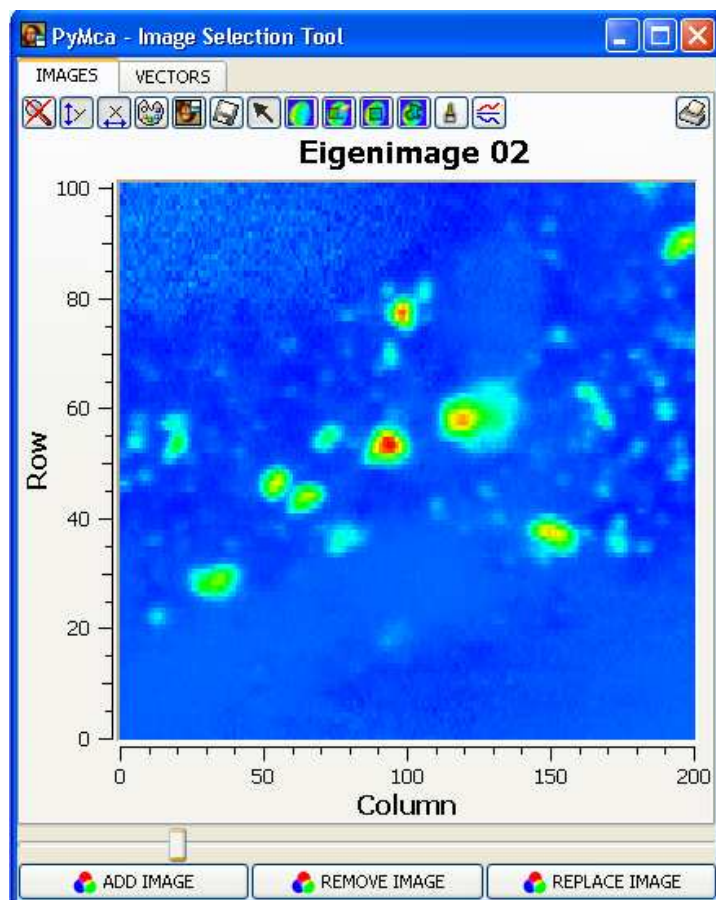
Eigenimages and Eigenvectors



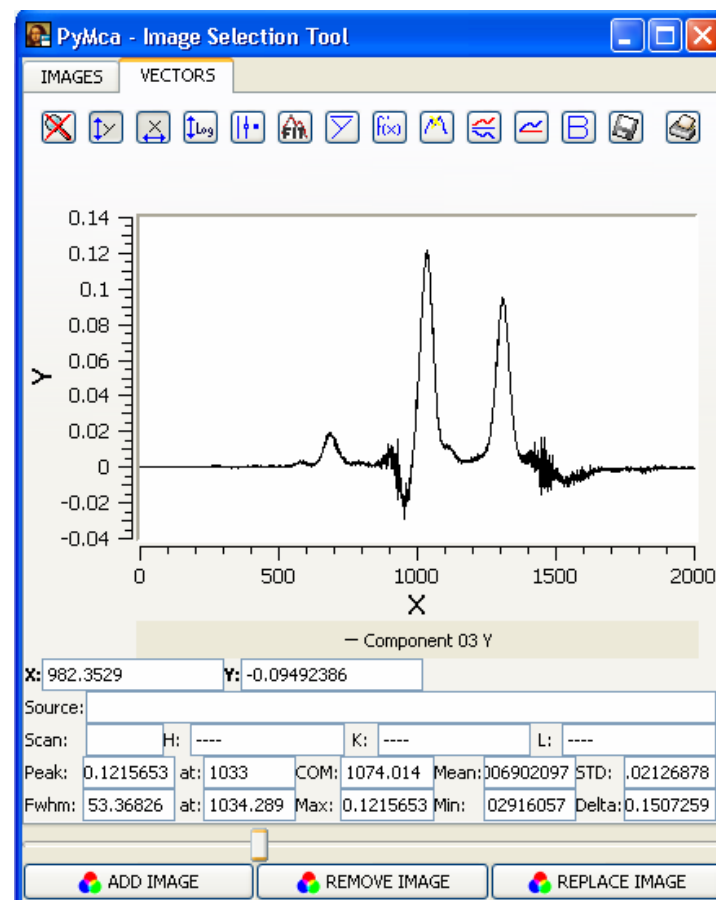
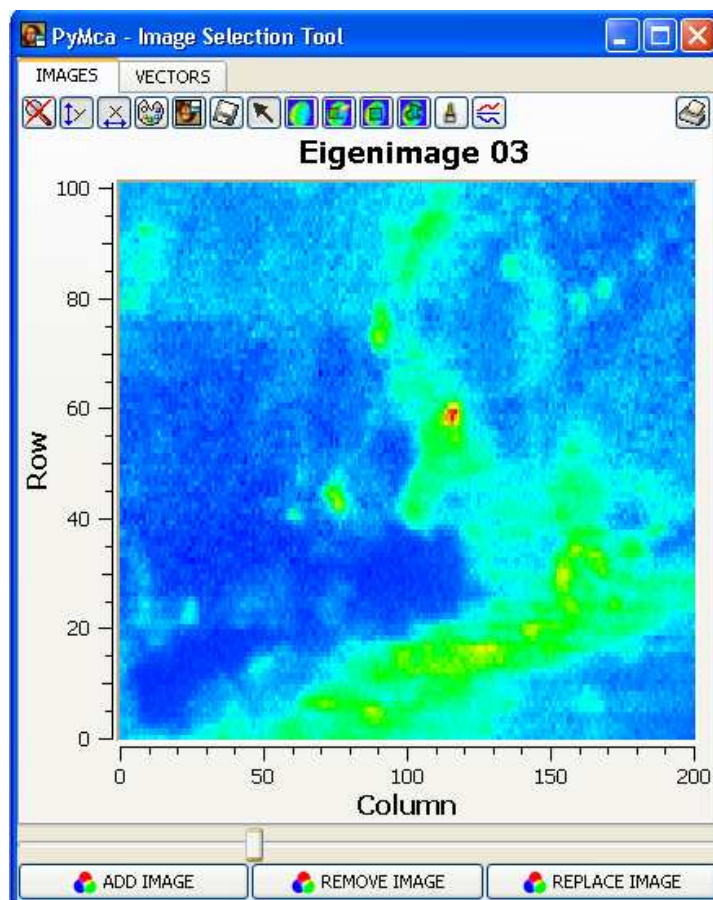
Eigenimages and Eigenvectors



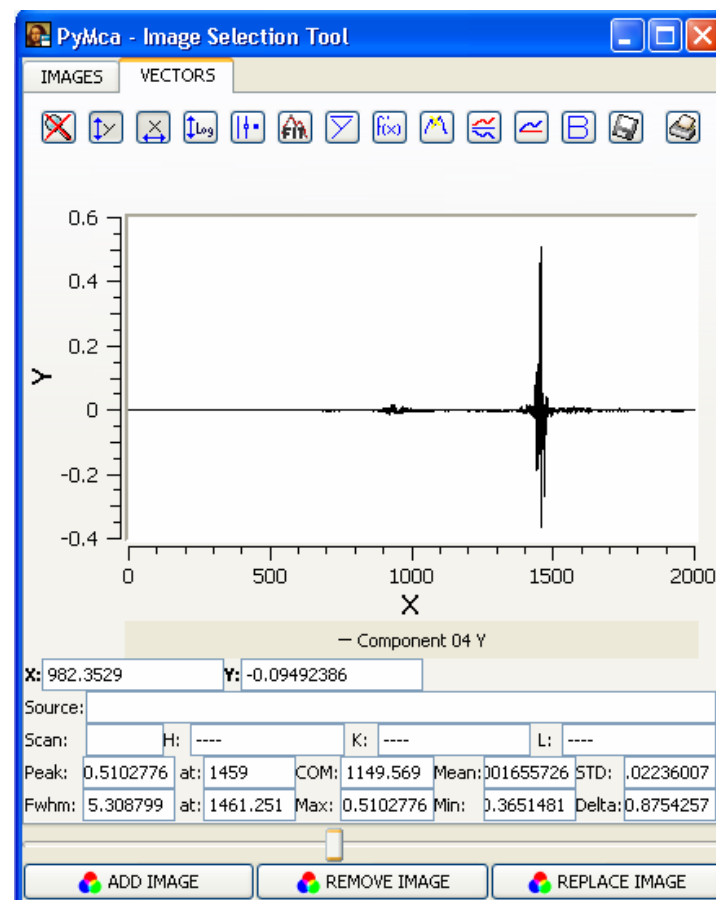
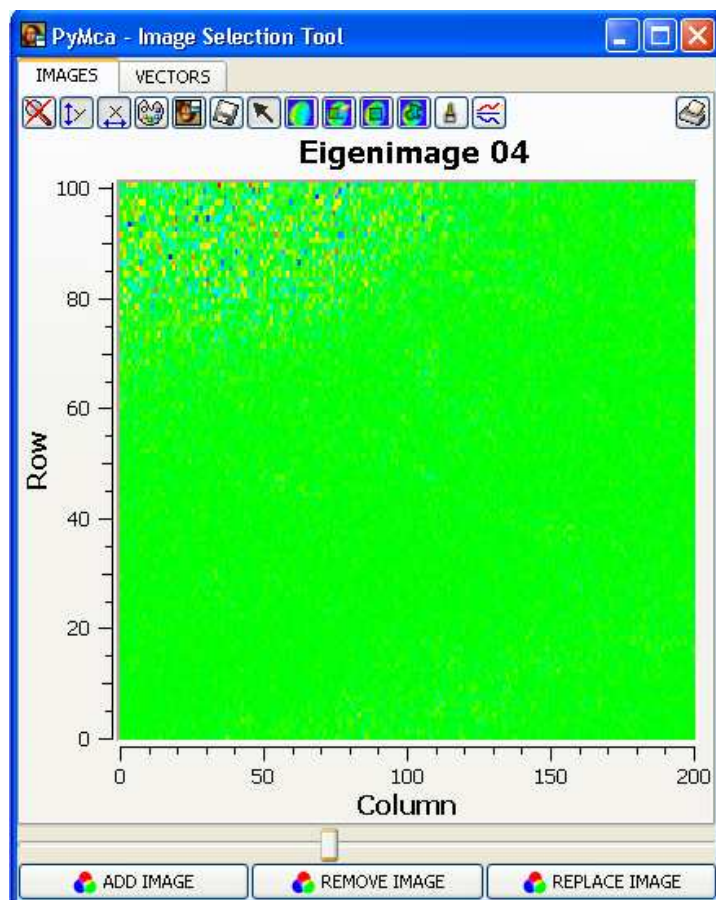
Eigenimages and Eigenvectors



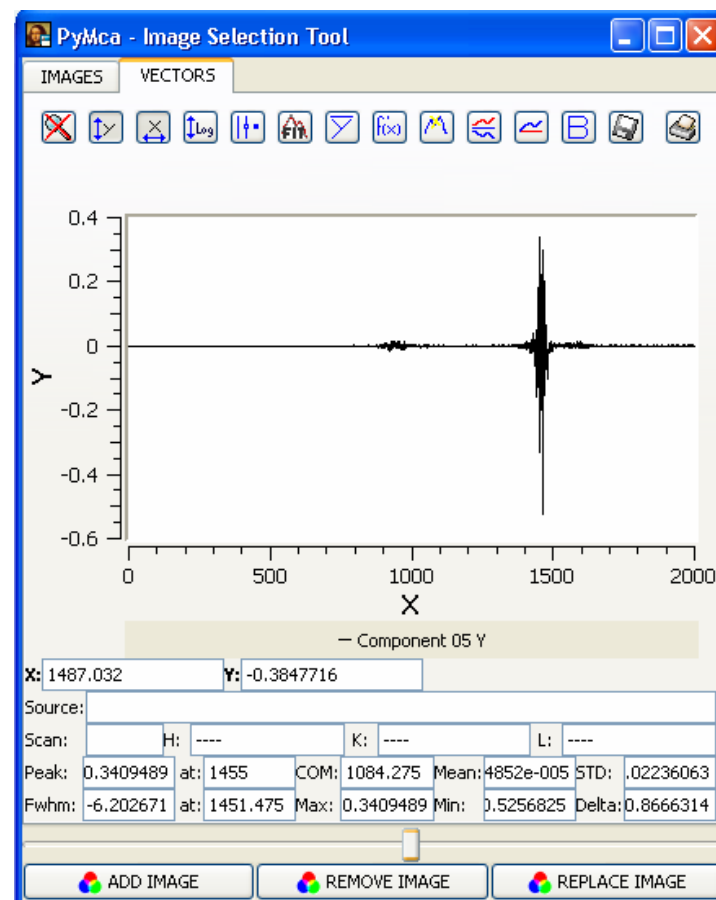
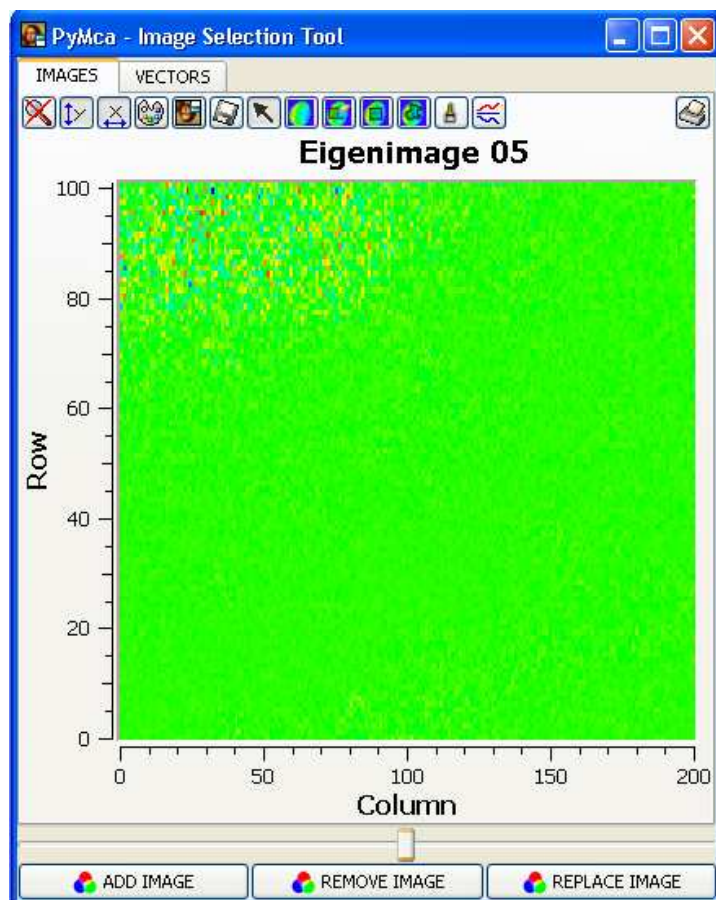
Eigenimages and Eigenvectors



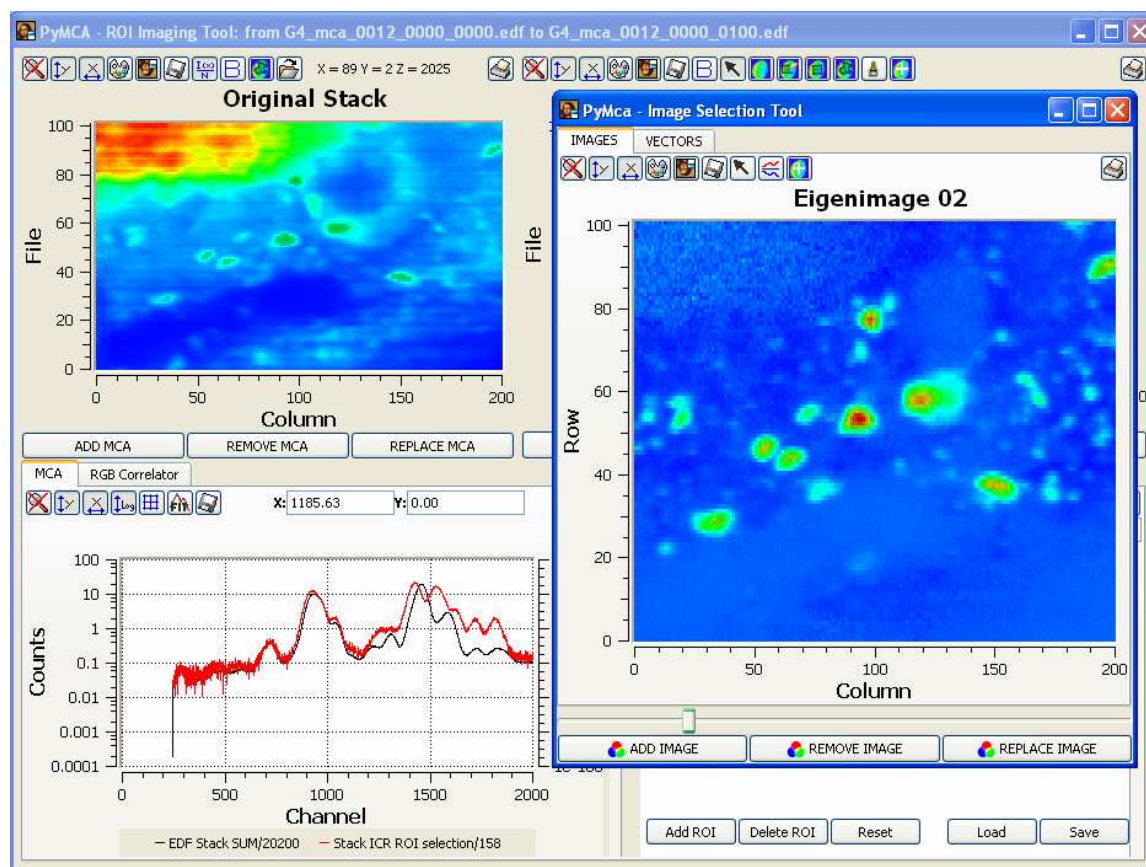
Eigenimages and Eigenvectors



Eigenimages and Eigenvectors

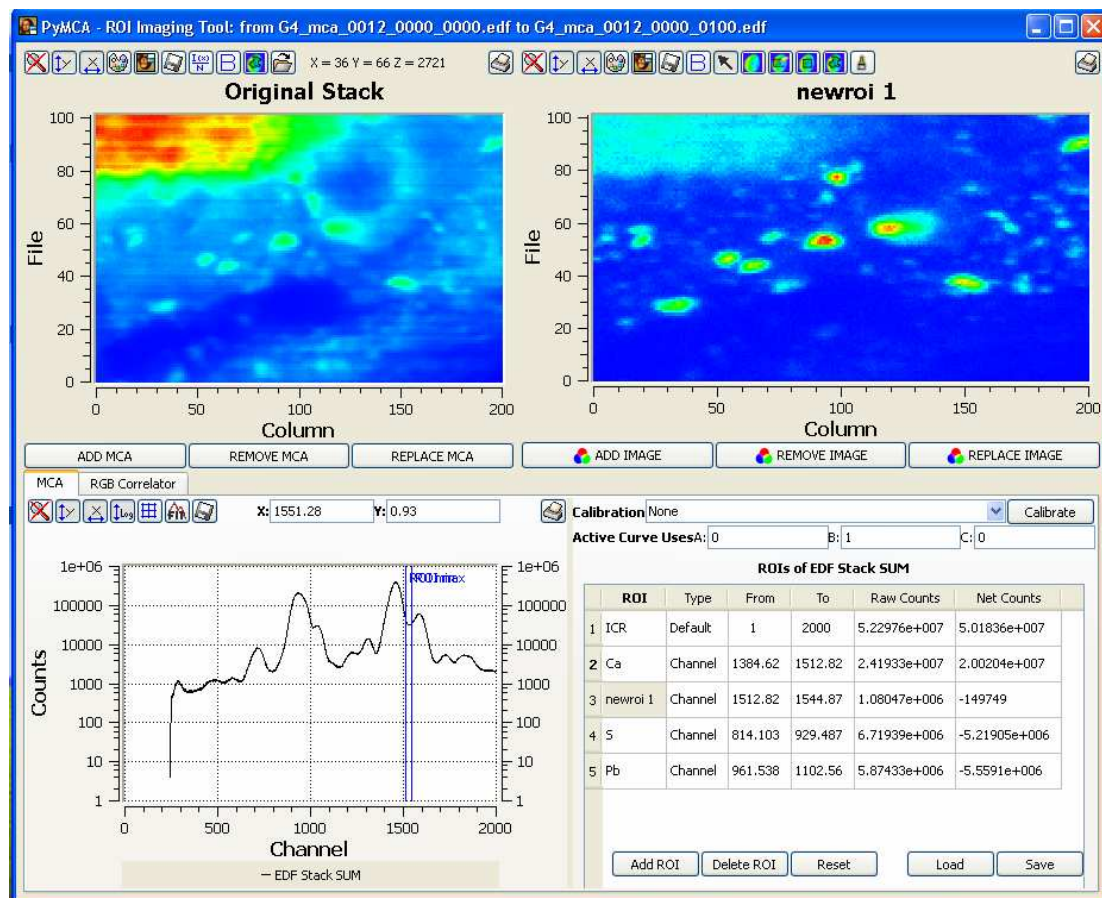


Getting the actual information



We can select a set of pixels on any of the displayed images and display the cumulative spectrum associated to those pixels.

Here we can see the average spectrum associated to the hotter pixels of the Eigenimage 02 (in red) compared to the average spectrum of the map (in black).



We could have easily missed the presence of one element if we would have just analyzed the sum spectrum via ROIs.

What have we done?

We have used multivariate analysis (PCA in this case) to know what sample regions were worth to take a closer look.

Not bad when you have a lot of data ...

This data treatment is totally generic and applicable to other techniques

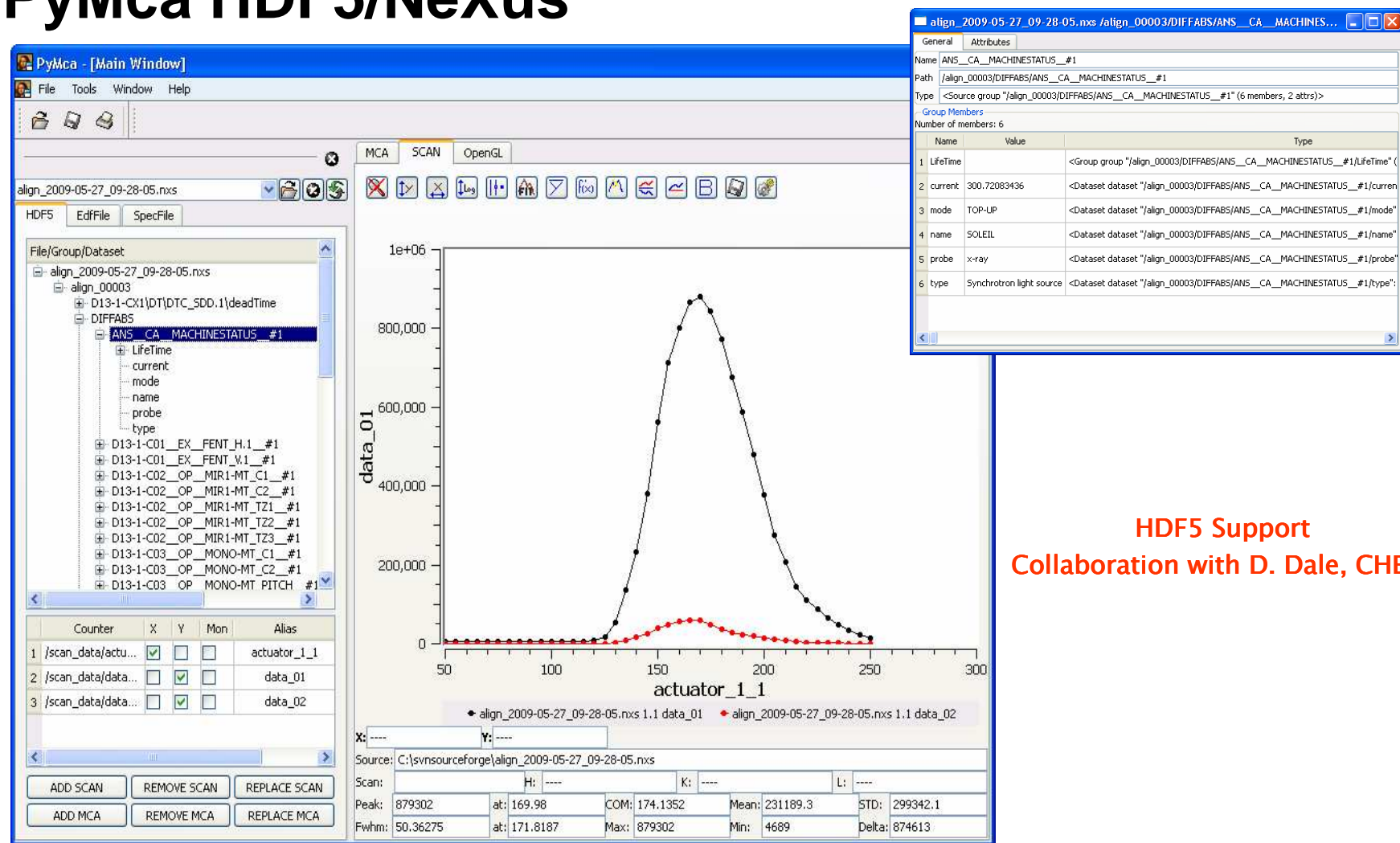
Multivariate Analysis Capabilities

PyMca makes use of PCA, ICA (via MDP toolkit) and NNMA (via py_nnma)

For the time being, multivariate analysis is used just to identify sample regions with different properties. The associated physical spectrum gives the actual information.

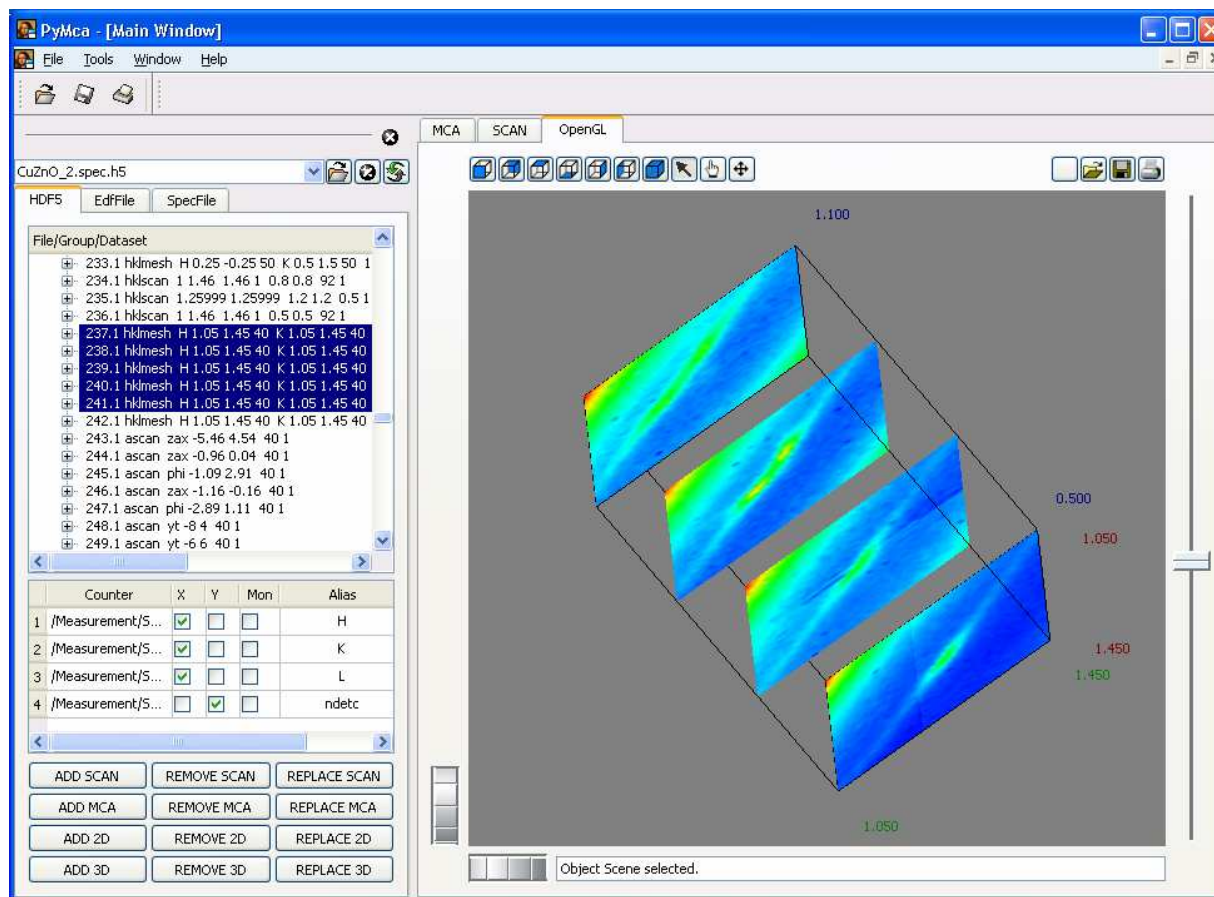
It can combine information from different simultaneously acquired datasets, for instance XRD and XRF data or PIXE and RBS.

PyMca HDF5/NeXus



SOLEIL NeXus Data courtesy of J.A. Sans and G. Martínez

Generic HDF5 Visualization



Generic: Select what to plot, how to plot it, and plot it.

The PyMca HDF5 problem

PyMca can do a lot more than “just” fully analyze XRF spectra

PyMca can read HDF5 data ... but often has no clue about what to do without asking

Properly defined NXdata groups can provide default visualization and more

It would be great to automatically identify the different MCA detectors, with their counts, their energy axis, their preset time, their elapsed time, their calibration, ... NeXus NXdetector has almost everything needed to describe an MCA, but still needs user interaction to know that detector is an MCA.

It would be very simple to add two attributes to datasets. One defining the “natural” data dimensions and other one defining those dimensions are the first or the last ones of the dataset.

What can two attributes bring?

Programs would automatically provide better user choices

Programs would know what to do with datasets! Despite being able to perform 4D plots, PyMca does not know (it could just guess) what to do to visualize datasets with dimensions 1 x 2 x 512 x 1024.

One attribute to say “ACTUAL_DIMENSION = 2”

One attribute to say “ACTUAL_DIMENSION_IS_LAST_DIMENSION = 1”

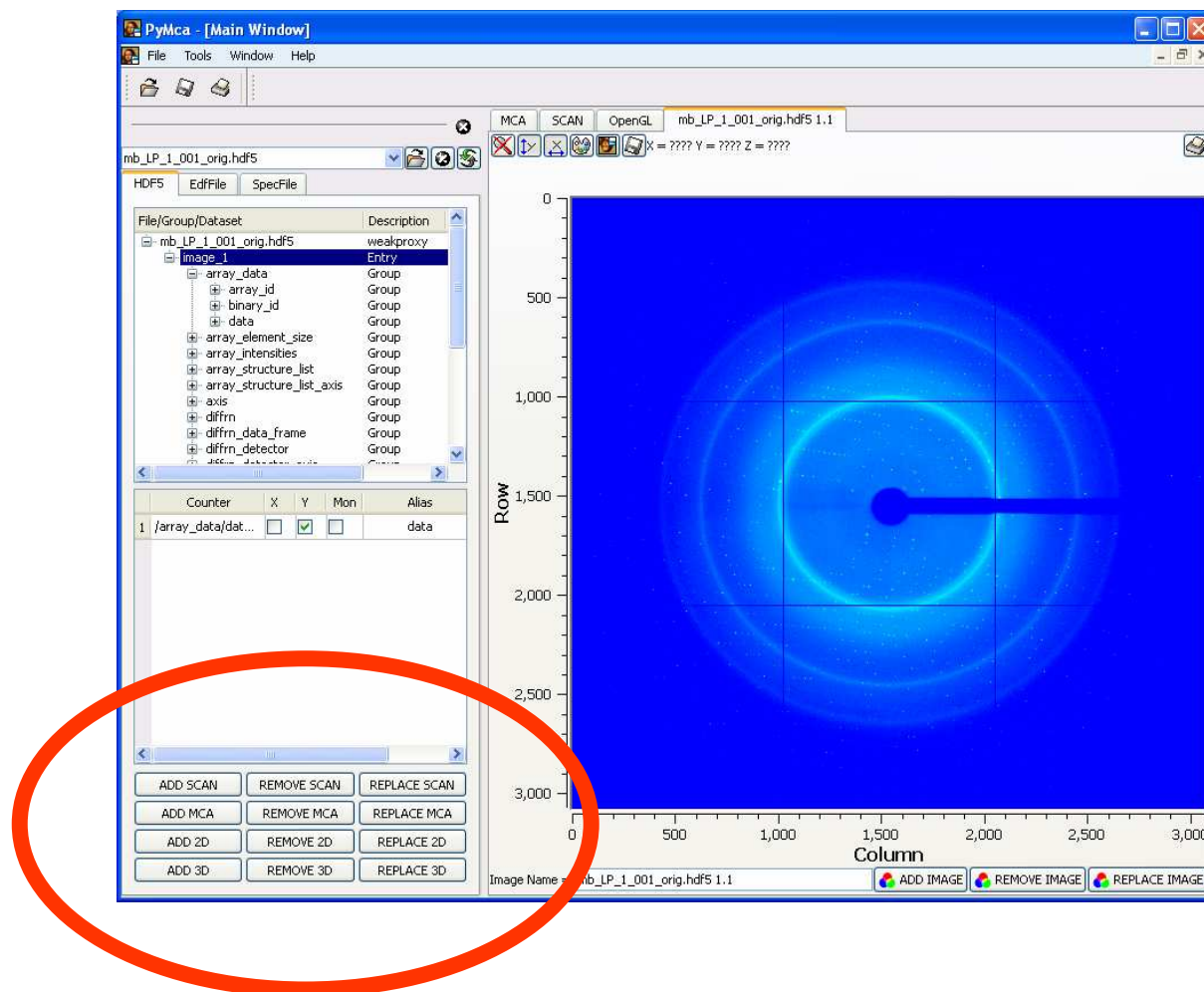
And things are totally different than:

One attribute to say “ACTUAL_DIMENSION = 1”

One attribute to say “ACTUAL_DIMENSION_IS_LAST_DIMENSION = 1”

The alternative looks ugly, doesn't it?

Just imagine a wall made of buttons instead of bricks everywhere



Conclusion

PyMca is a program as well as a toolkit

- Open source and distributed under the conditions of the GPLv2+
- Supports HDF5 among other formats
- Can be used as a multivariate analysis tool
- Can be used as a fitting and visualization tool (for up to 4-dimensional data)
- Allows you to specify a physically meaningful model which can quantitatively determine element concentrations from energy dispersive X-ray spectra
- Provides high level widgets based on PyQt that can be used independently or integrated into your application
- Has an active development funded by the ESRF

Acknowledgements

My ESRF colleagues (mainly software group and microscopy beamlines ID21 and ID22)

Ph. Walter and coworkers from the Centre de Recherche et de Restauration des Musées de France

Darren Dale (Cornell High Energy Synchrotron Source)

The community of Python developers of free software

The PyMca users, for their enthusiasm and their encouragements