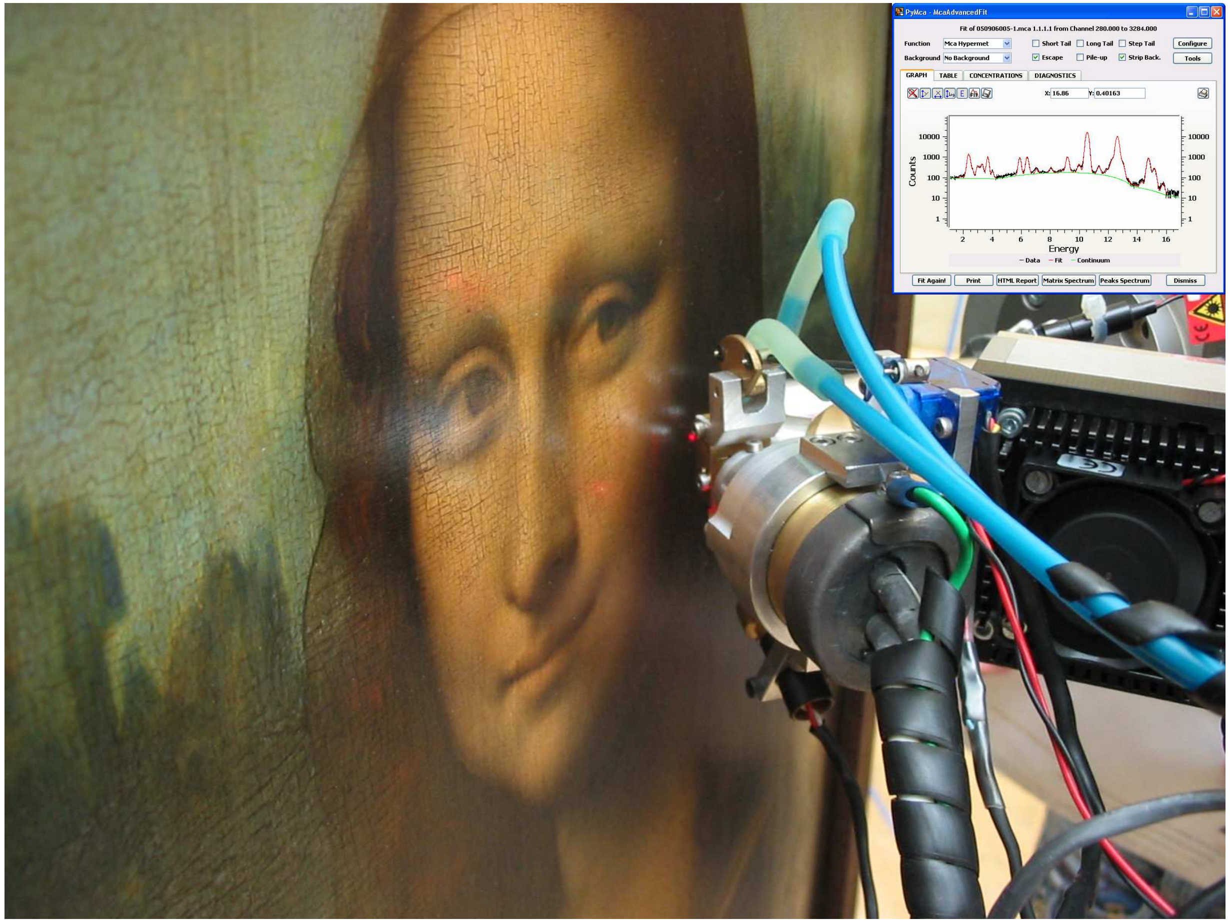


PyMca, Hyperspectral Data and HDF5

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:

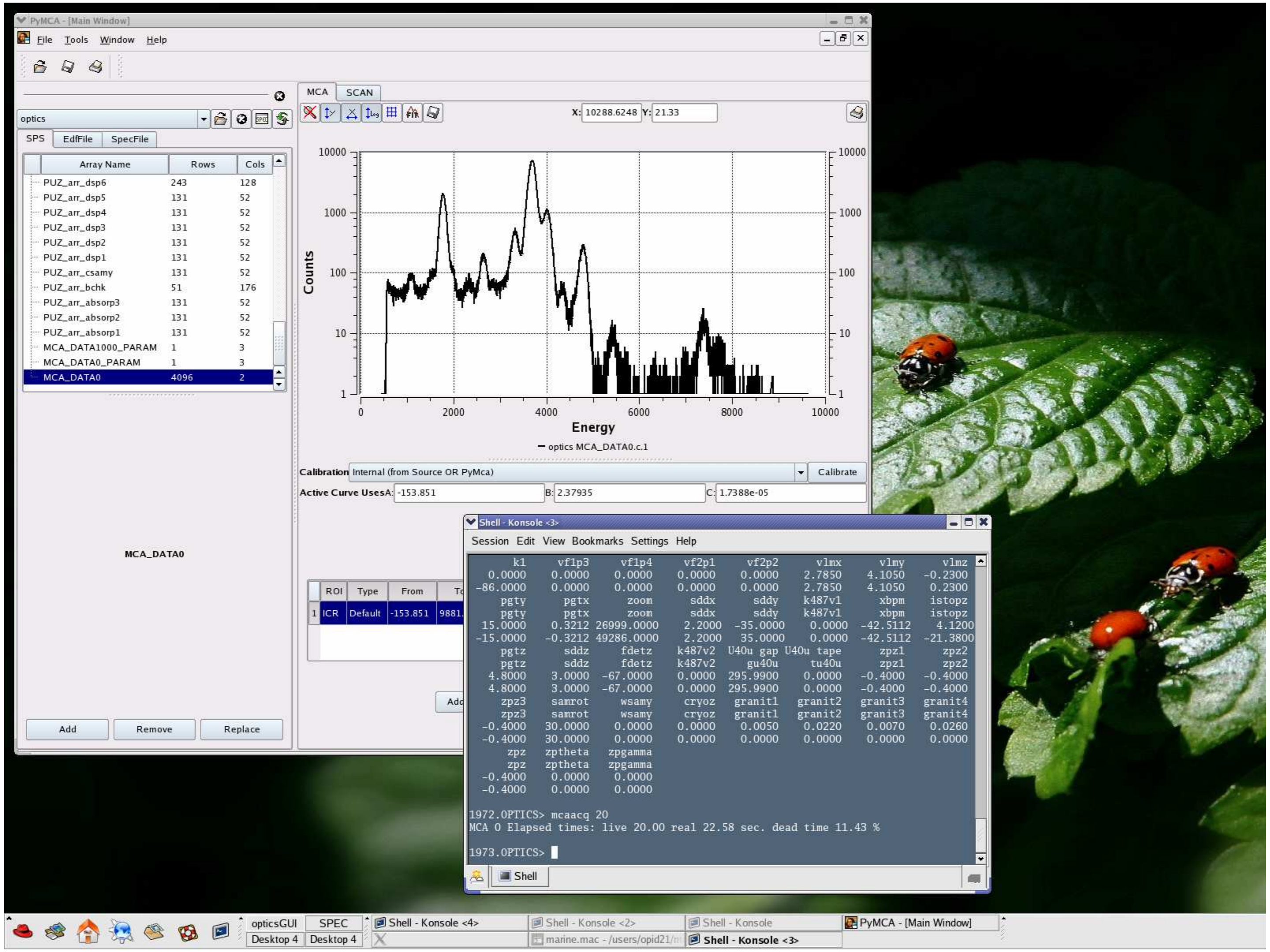
- Spectrum modeling
- Quantification
- ROI imaging
- Fit imaging via batch processing

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

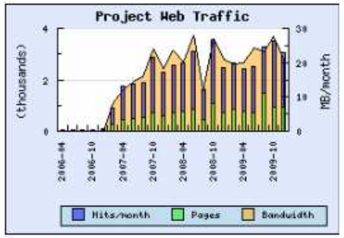
PyMCA

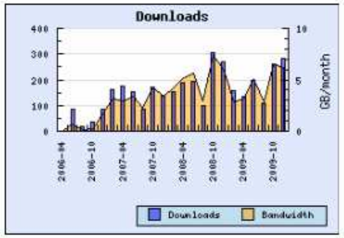
Share

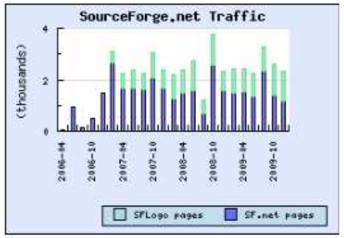
Summary | Files | Support | Develop | Tracker | Mailing Lists | Code

Downloads | Project Web Traffic | Tracker Activity | Forum Traffic | SourceForge.net Traffic | Subversion Activity

Usage Statistics For PyMCA





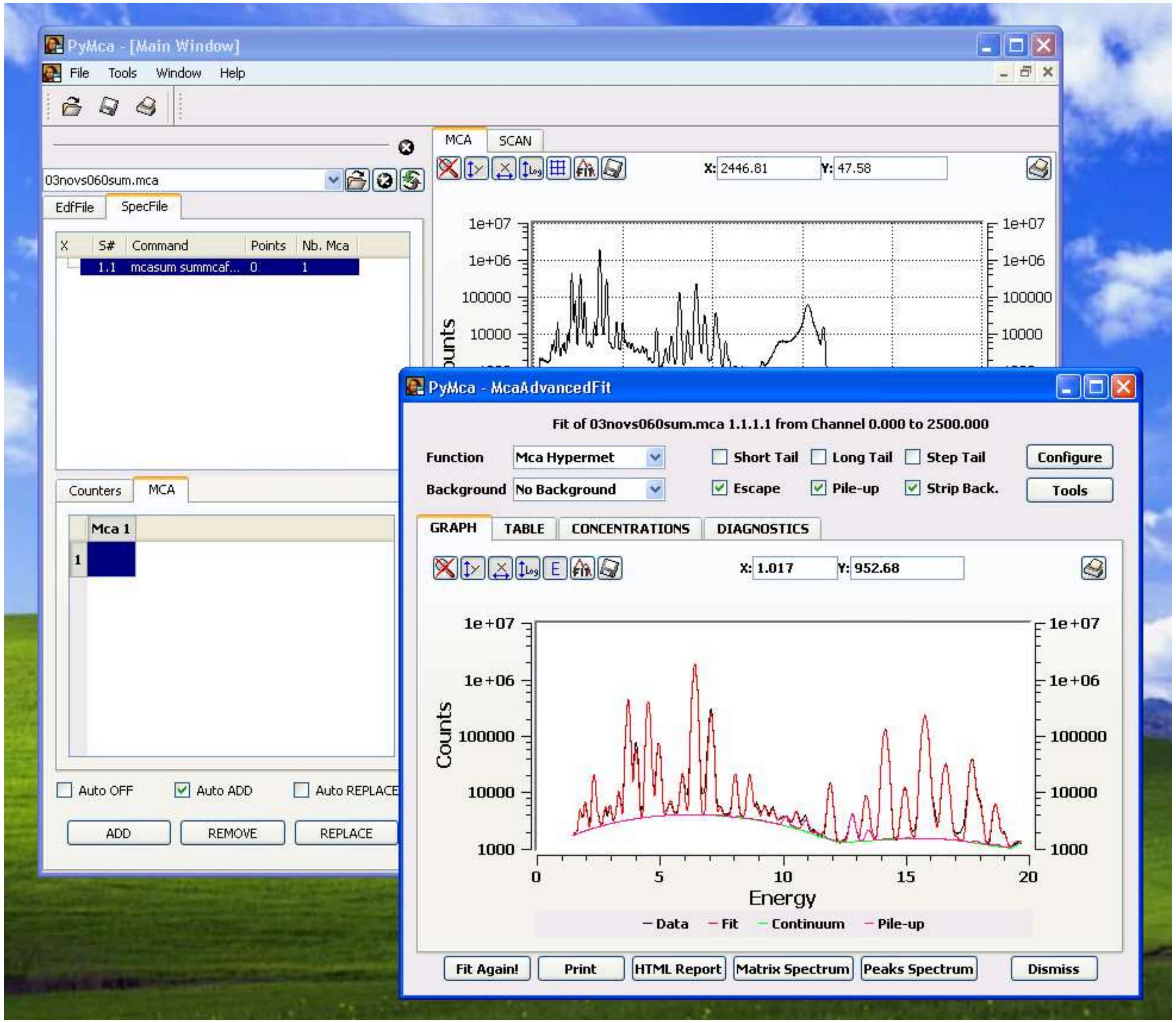


Click a graph for more data

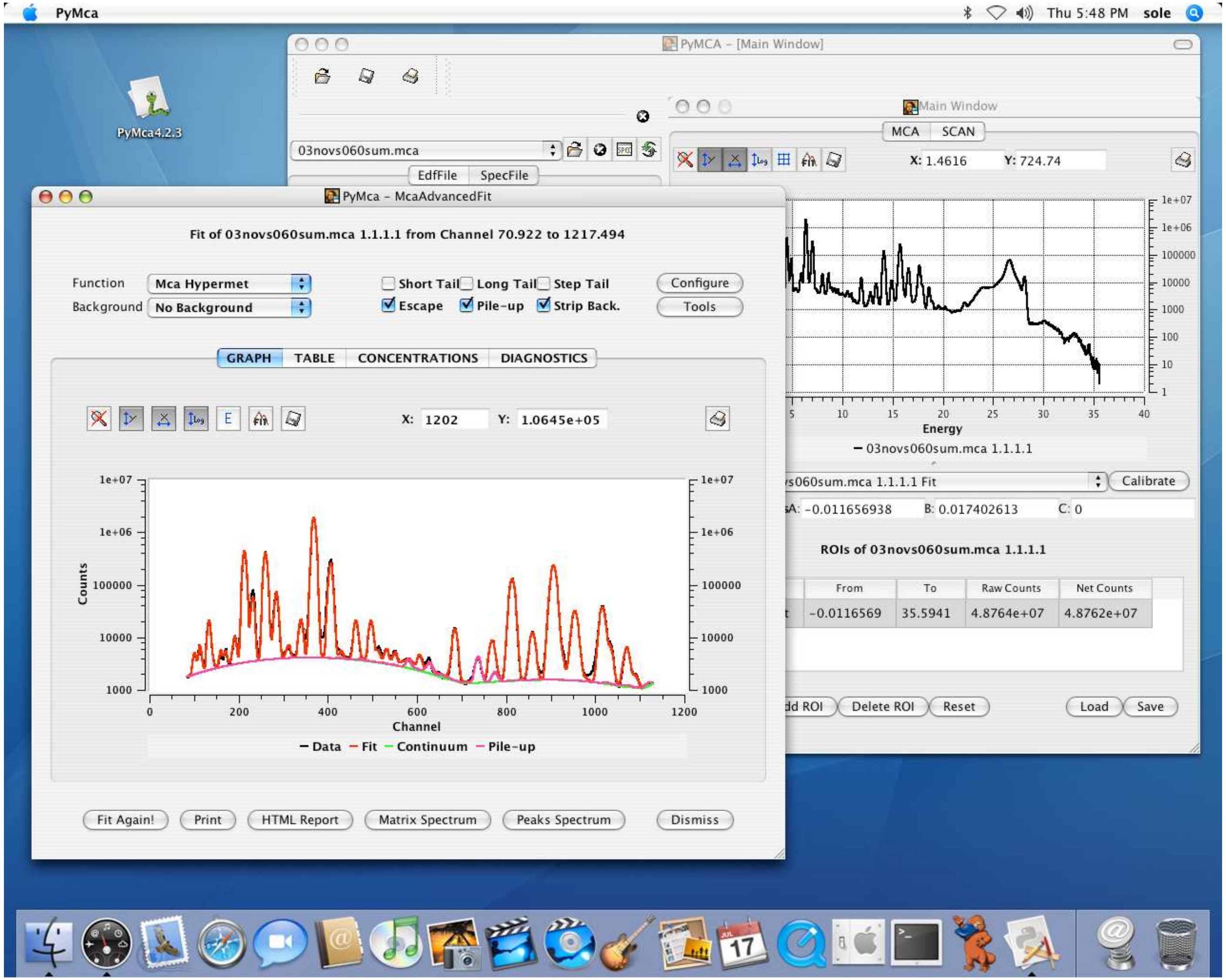
Statistics for All Time | Change View

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

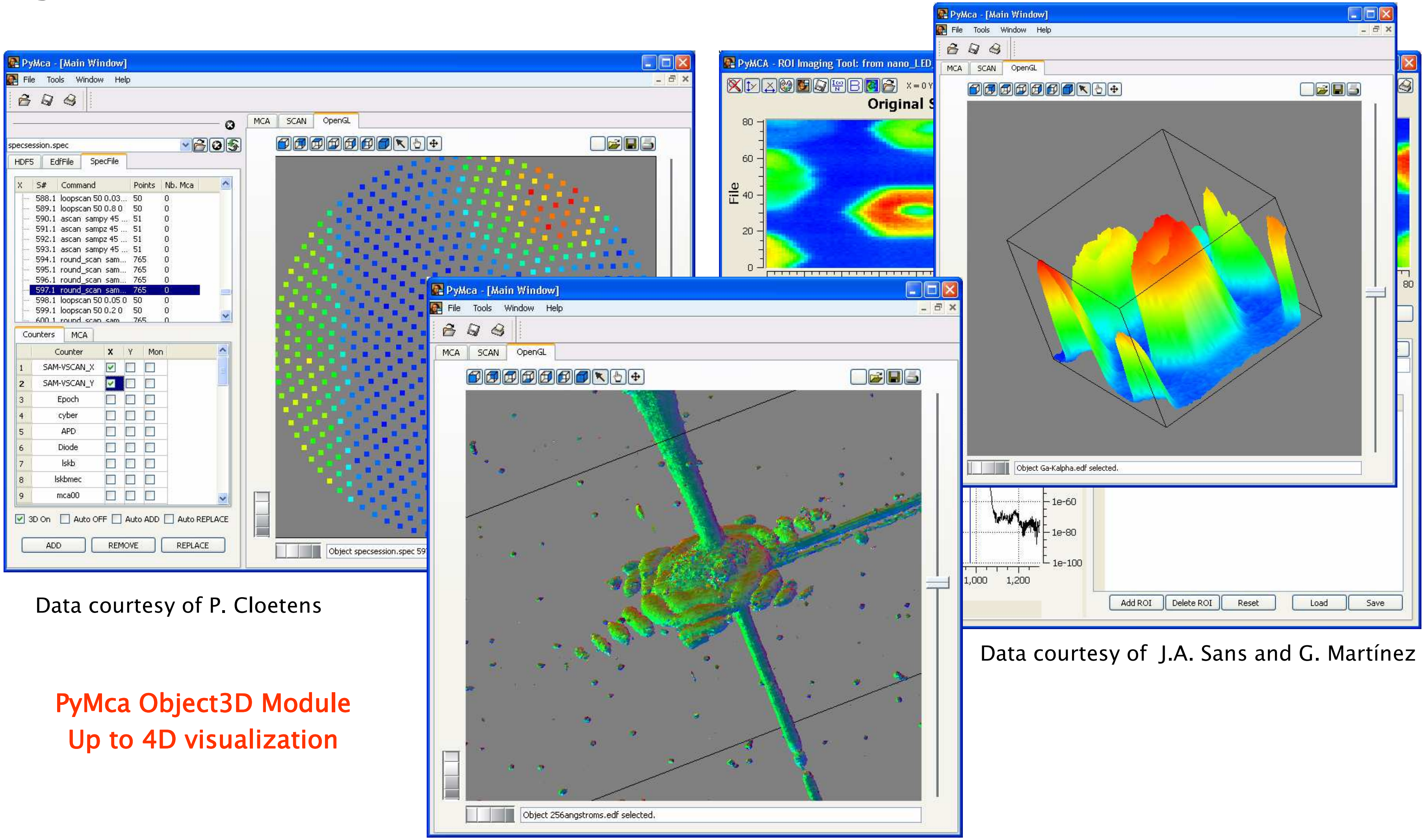
Multiplatform



Multiplatform



PyMca Visualization



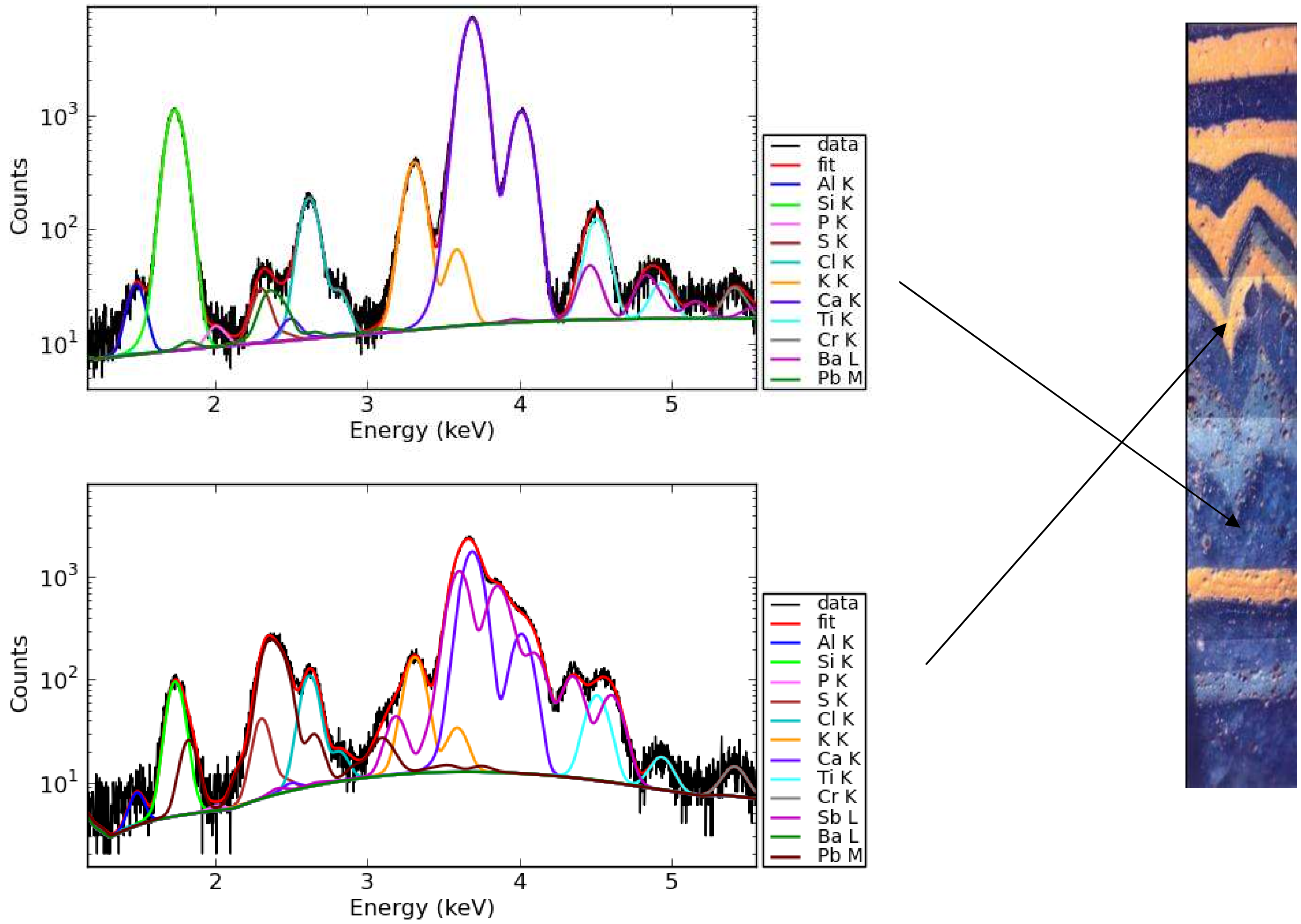
Data courtesy of P. Cloetens

Data courtesy of A. Díaz

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

PyMca Object3D Module
Up to 4D visualization

Complete M-shell support



Use of (sub-)shell mass attenuation coefficients

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

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

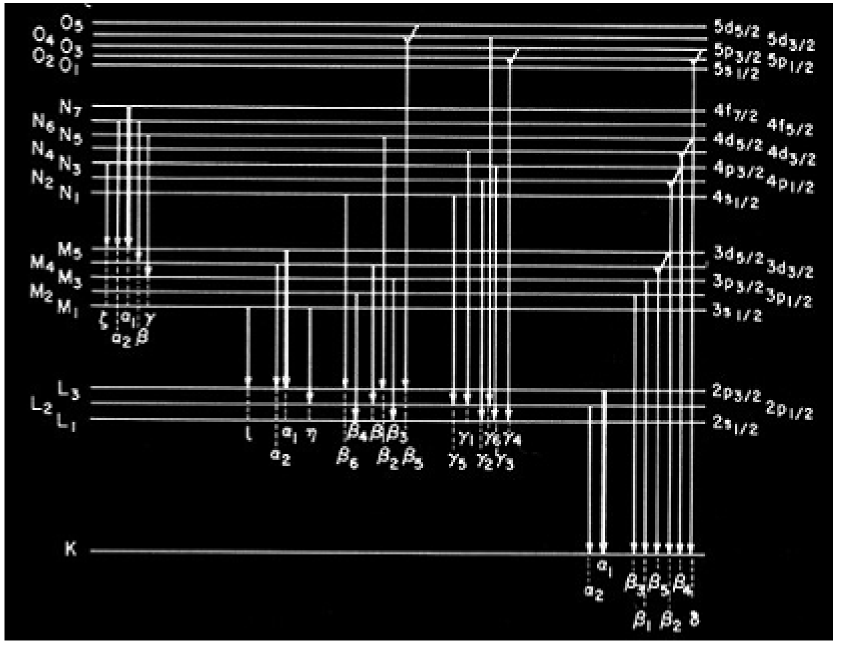
Term between parenthesis for Pb

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

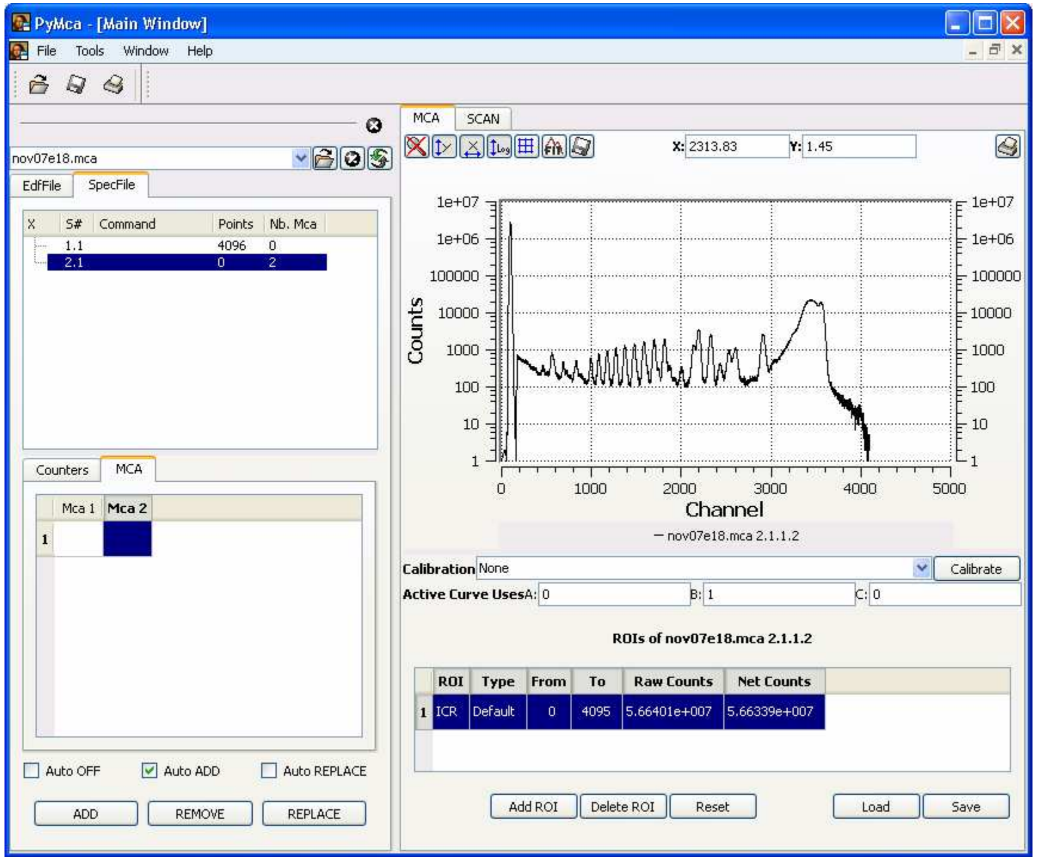


PyMca considers the vacancies produced in an atomic shell or sub-shell by the de-excitation process of an inner shell or sub-shell.

Radiative and Coster-Kronig transitions correctly considered.

Auger transitions approximated.

XRF Spectrum Analysis

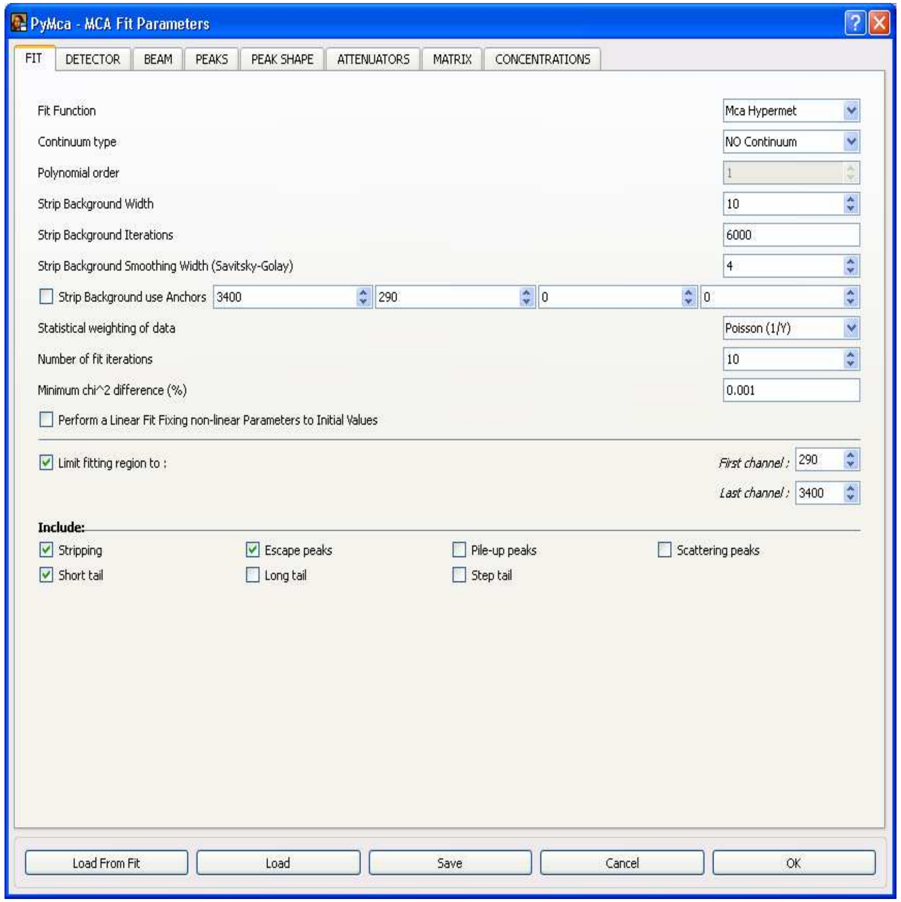


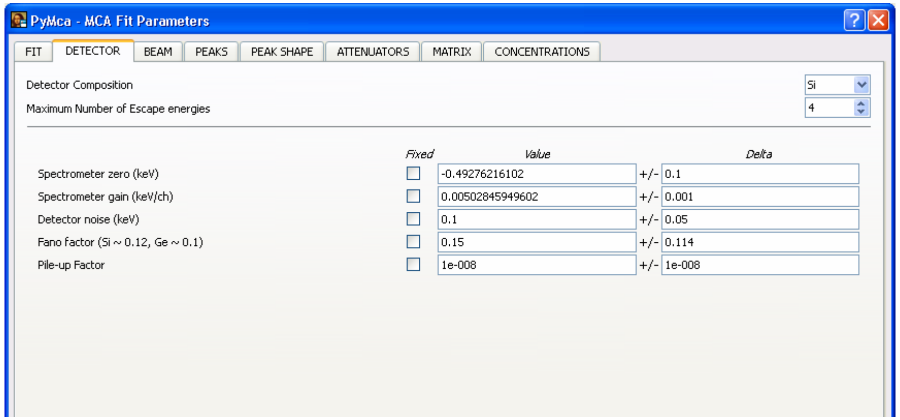
Typical procedure:

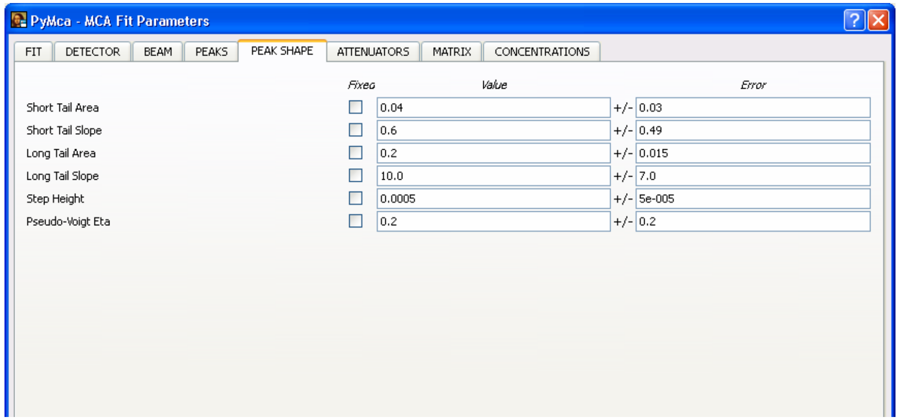
1. Calibration
2. Peak identification
3. Peak area extraction
   Region of interest (ROI)
   Deconvolution (FIT)
4. Quantification

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

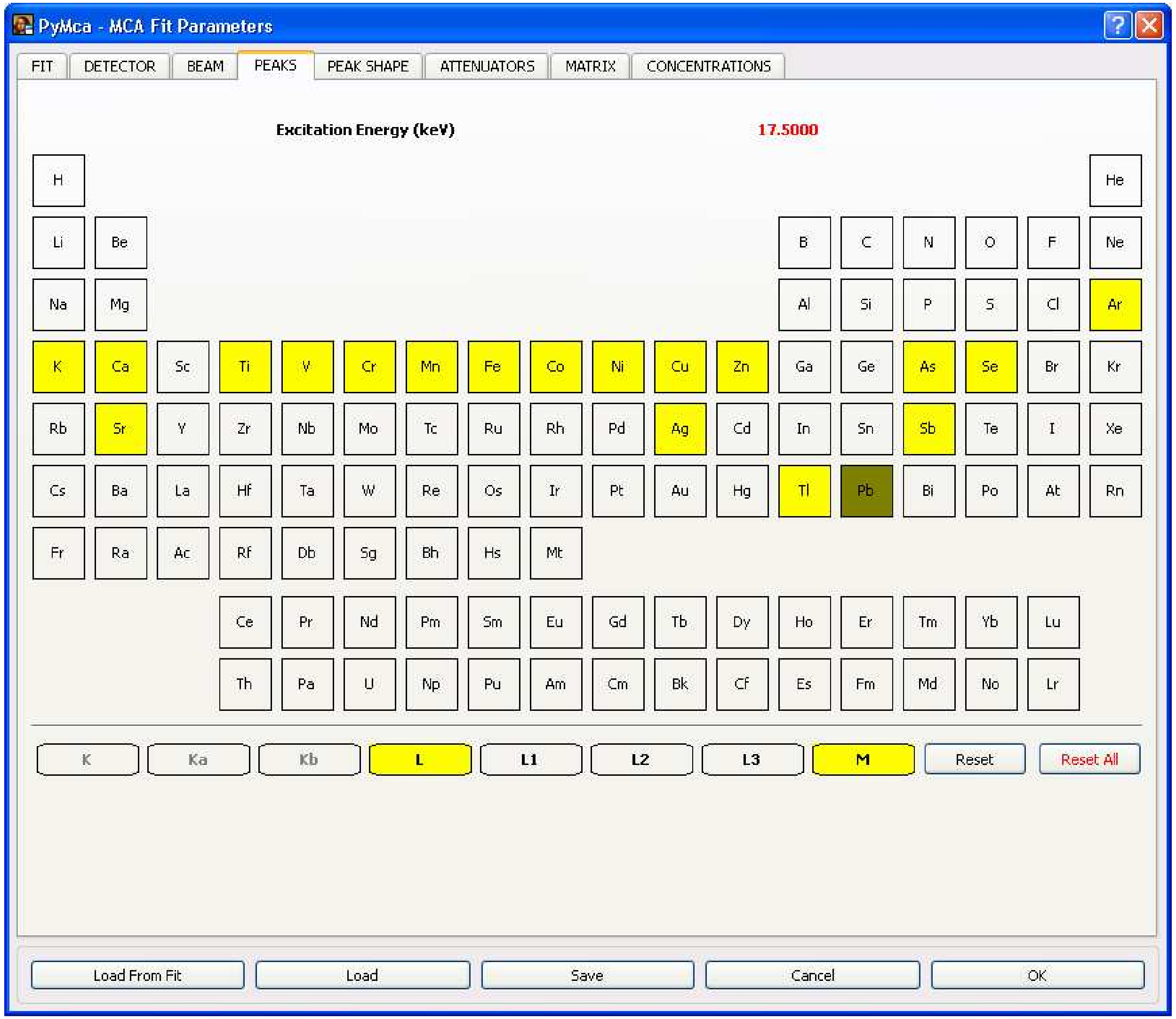
Fit configuration Dialog (I)





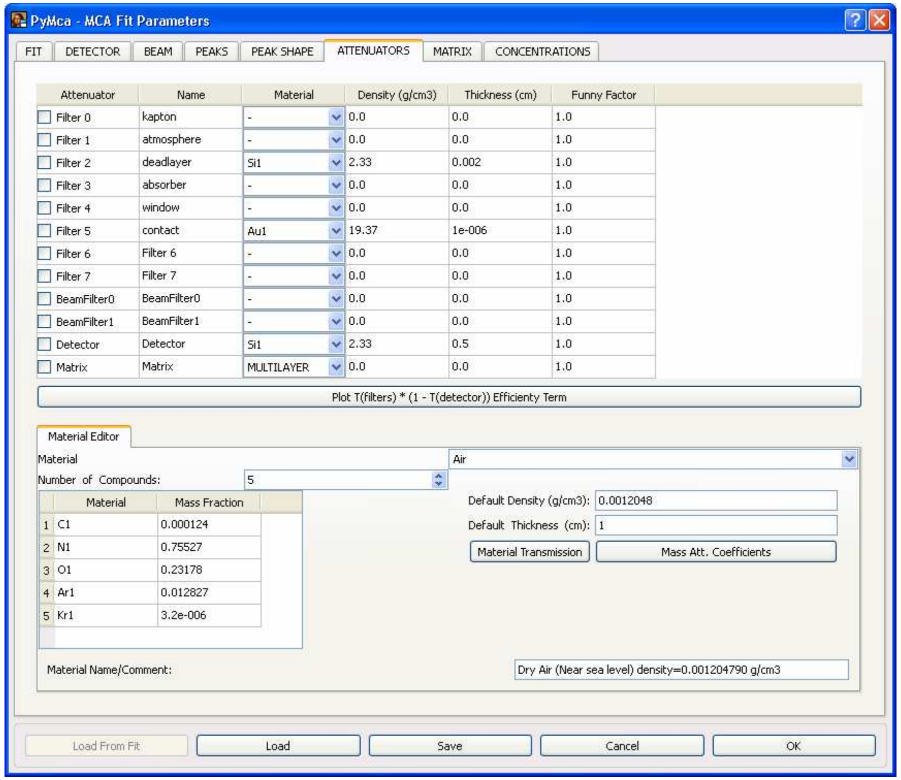


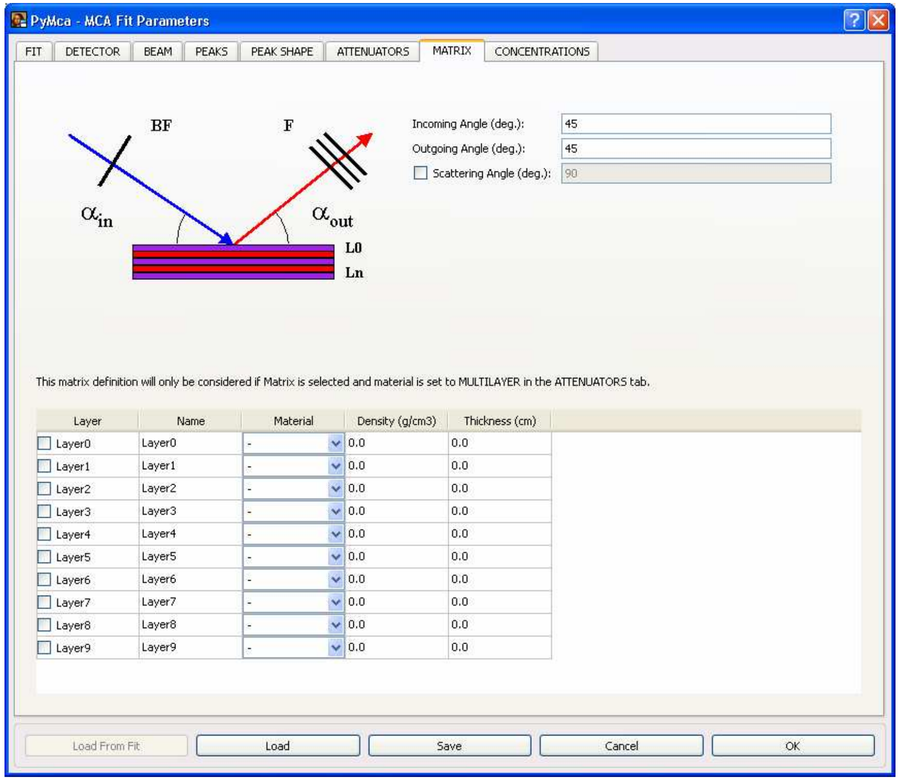
Fit Configuration Dialog (II)



Quantification (I)

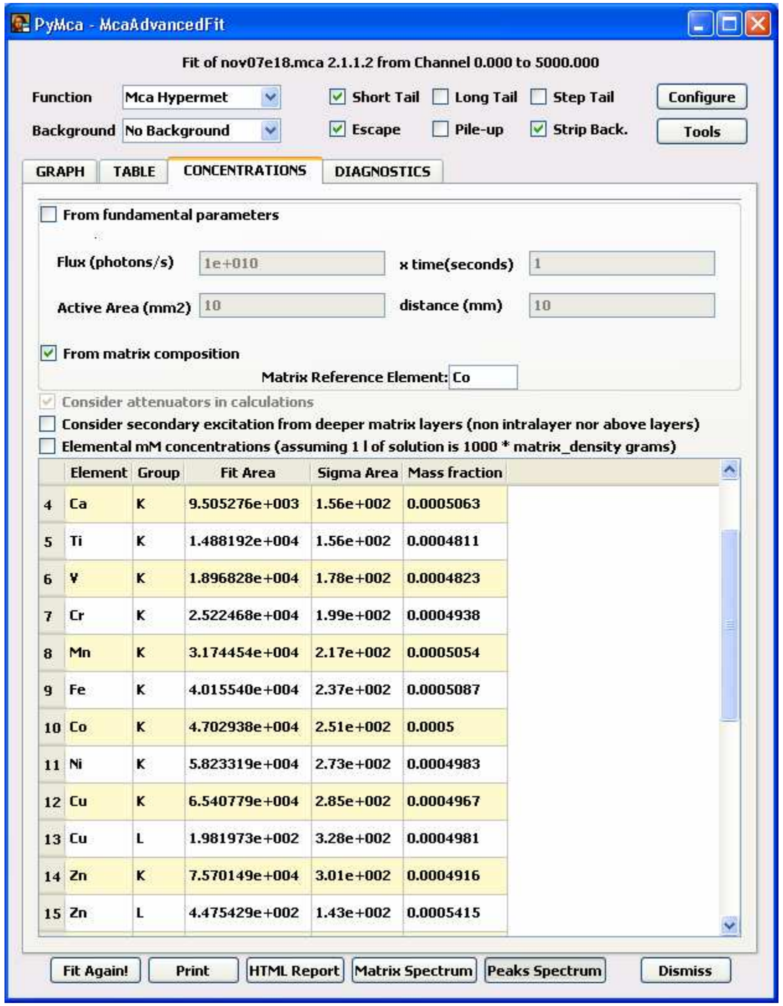
Parallel beam approximation

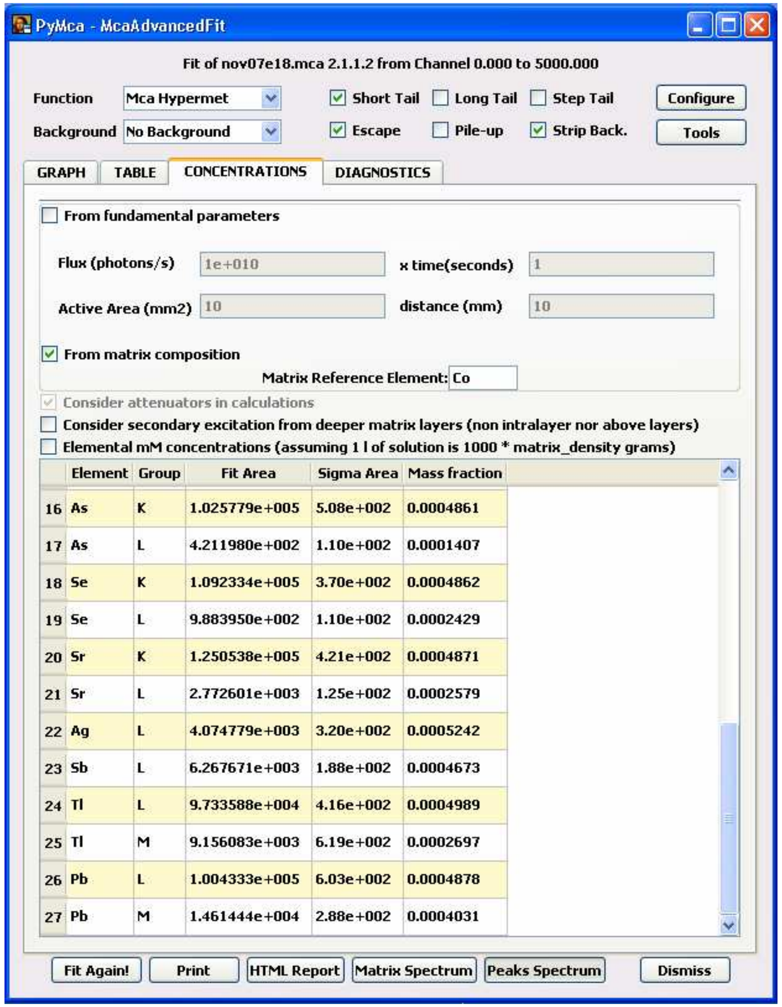




Quantification (II)

Nominal concentration 500 ppm



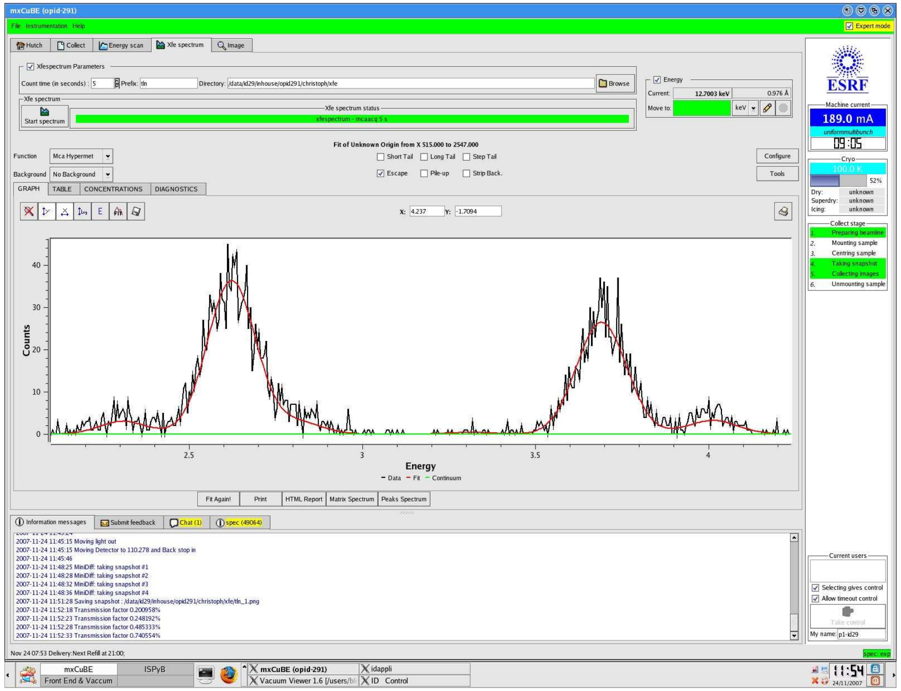


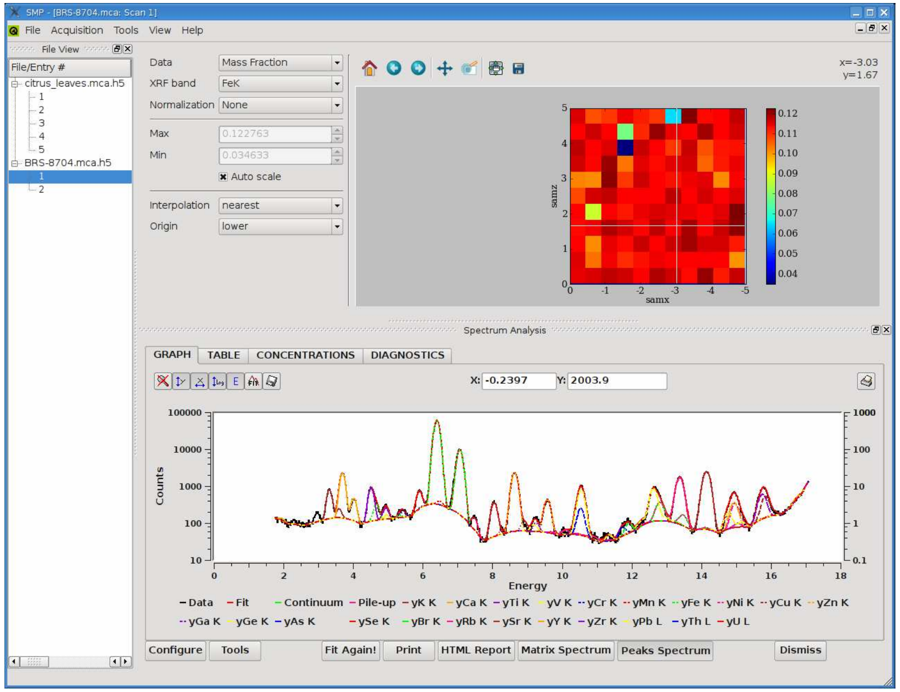
XRF Analysis Integration in other Applications

Integration in mxCuBE (ESRF)

Integration elsewhere

Cornell University
Cornell High Energy
Synchrotron Source





Is it easy to embed?

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

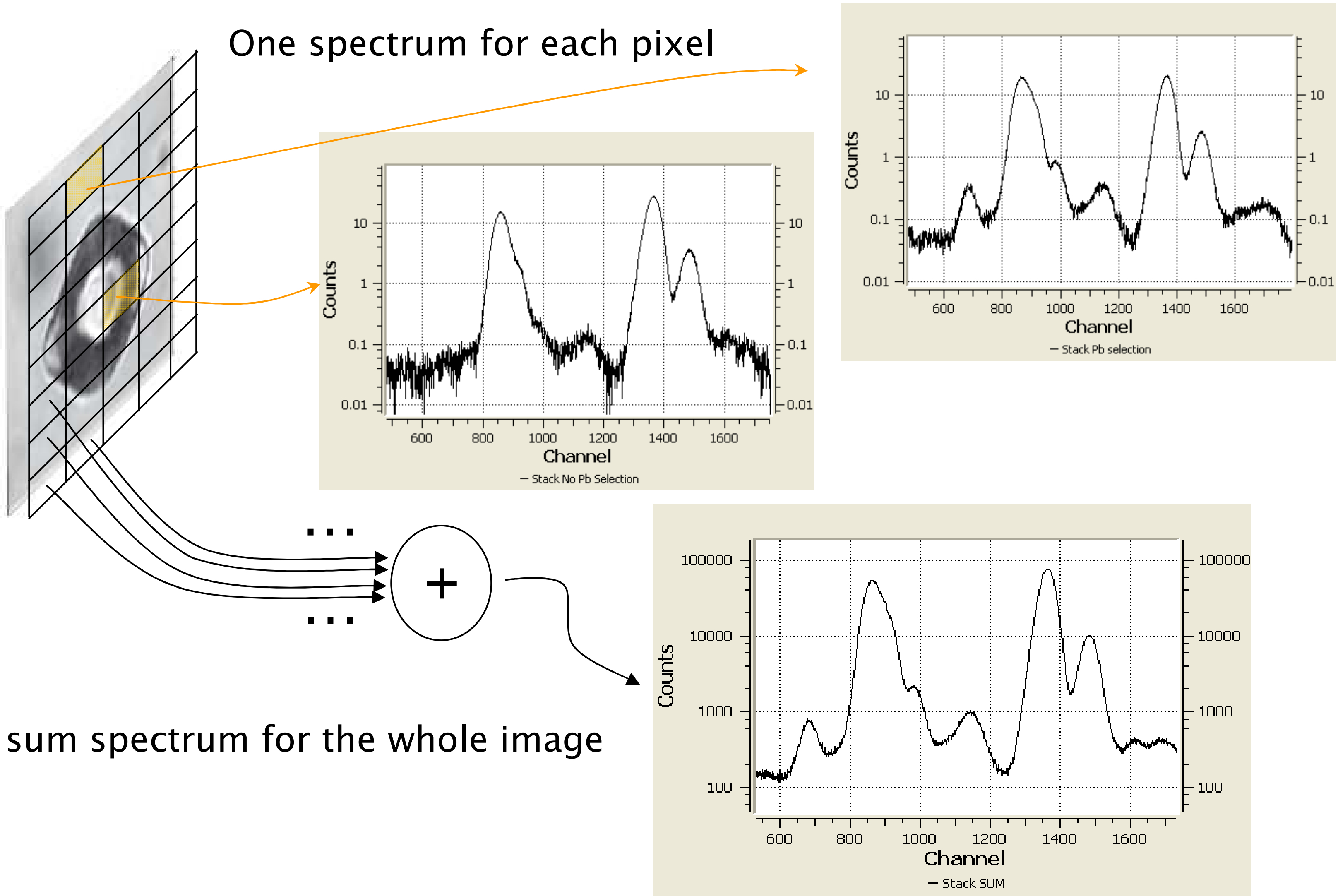
```
from PyMca import McaAdvancedFit

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

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

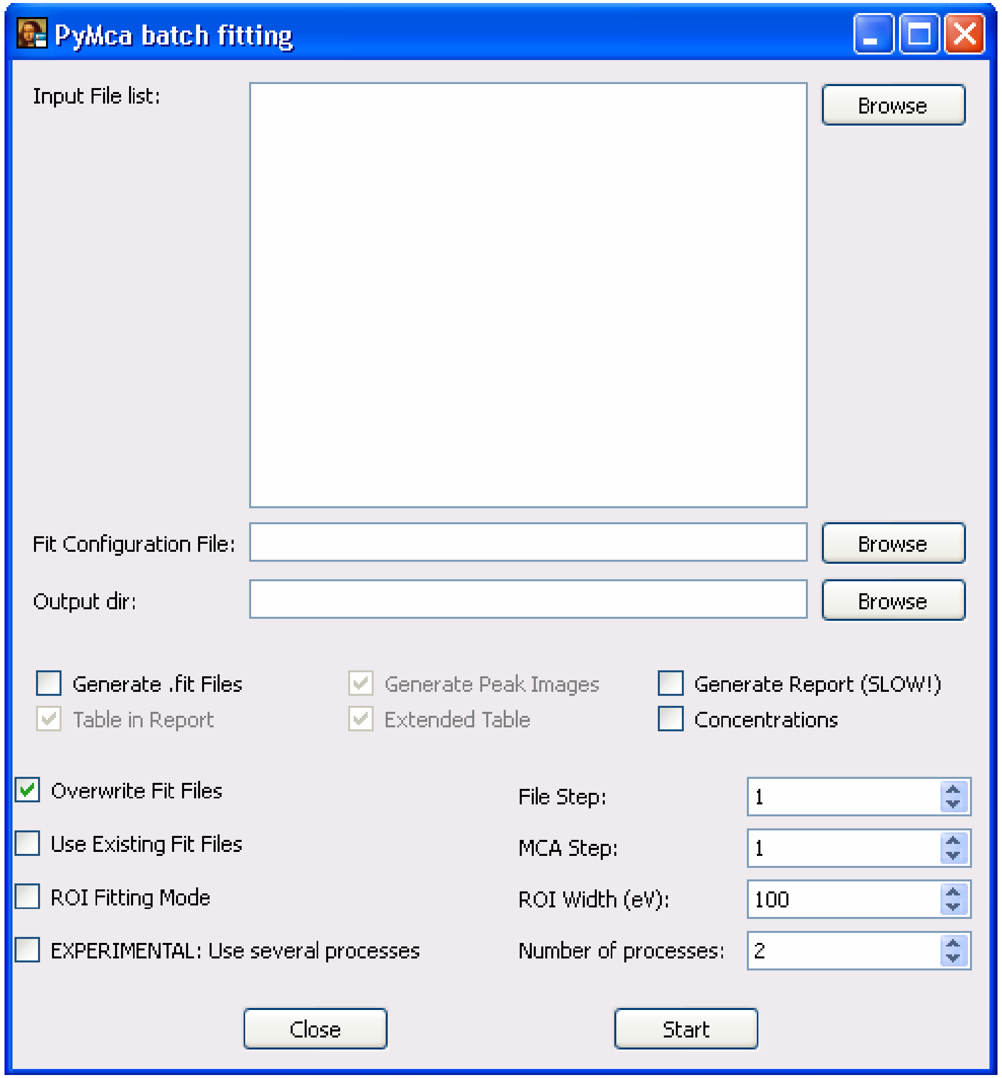
XRF Imaging

One spectrum for each pixel



A sum spectrum for the whole image

Advanced Fit Batch processing



Select the input files

Select the fit configuration

Select the output directory

Select the output options

Start

Output

Images in ASCII and ESRF format

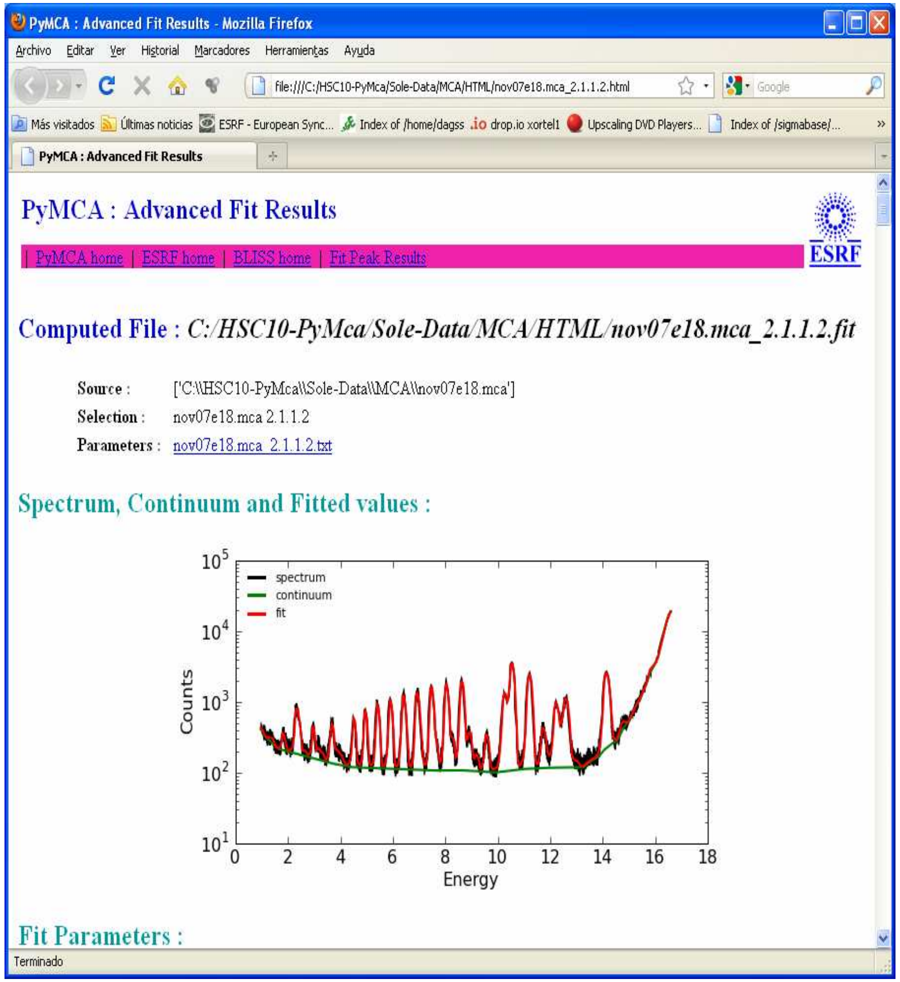
- Easy to import in other programs

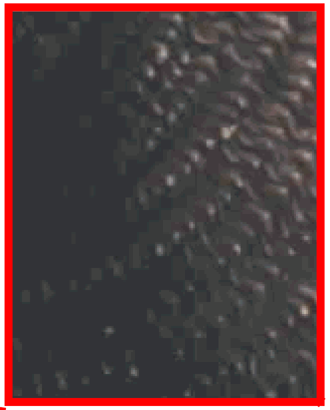
Individual peak contributions in ASCII

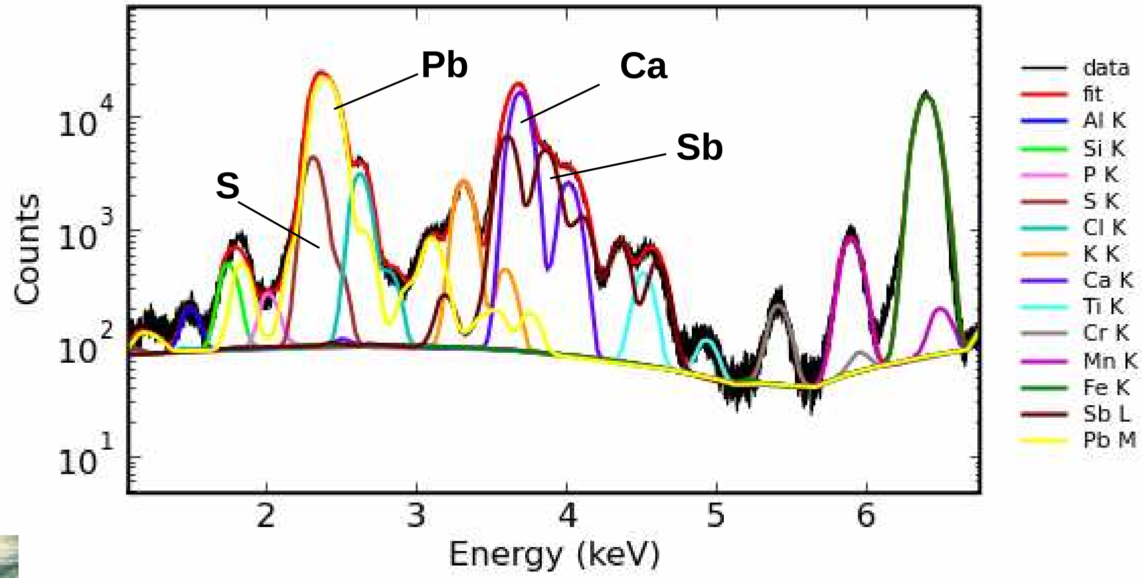
- Use your own plotting program

Fully automated HTML report

- Browse your results!

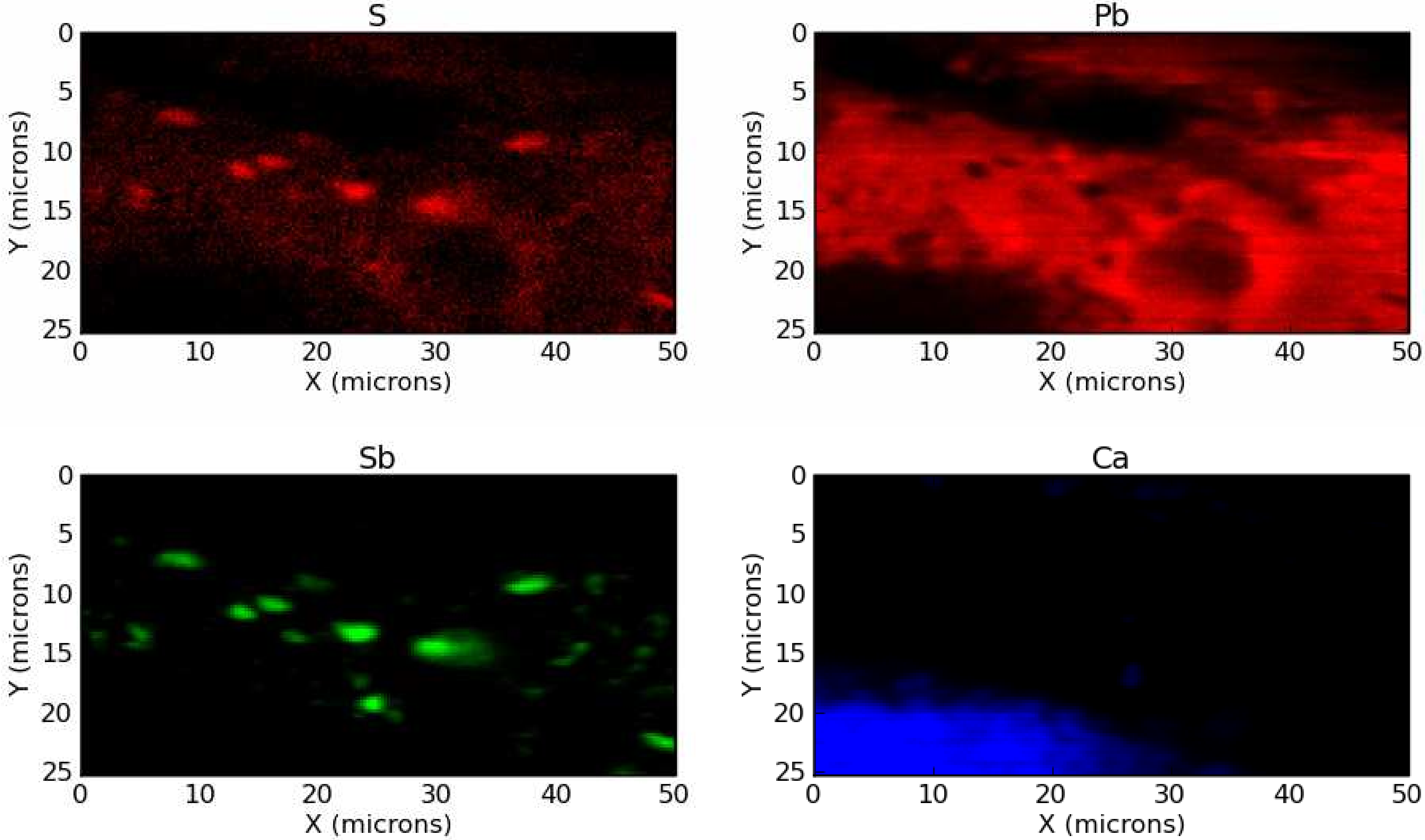




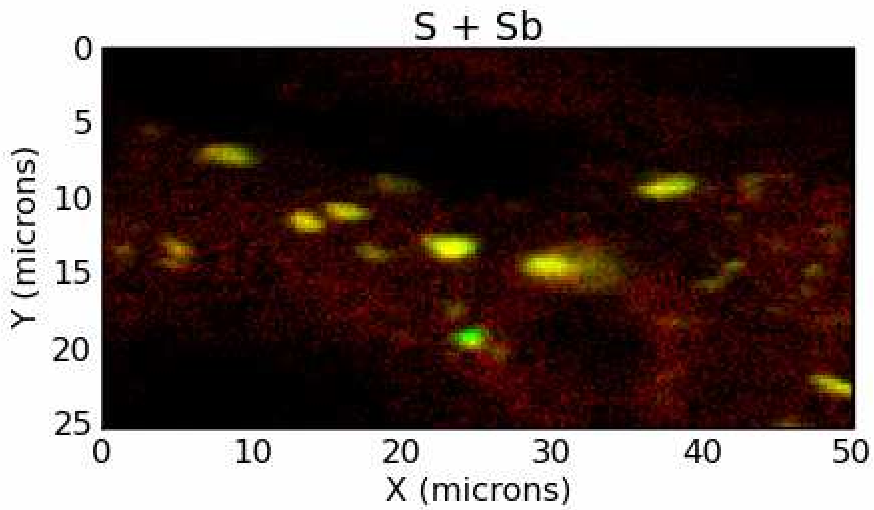


We wanted to know what pigments were used in this section of the painting

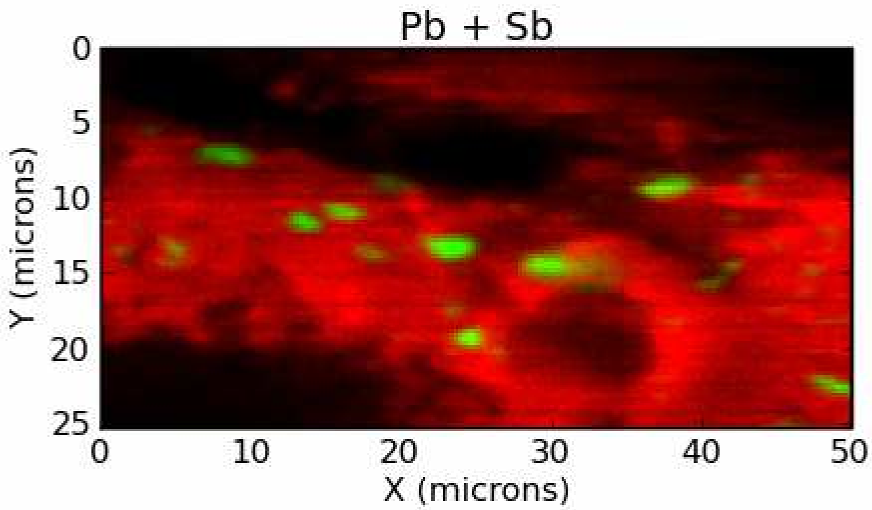
Based on the batch generated element distribution maps …



… and their correlations as shown by the program



Sulfur and antimony correlated

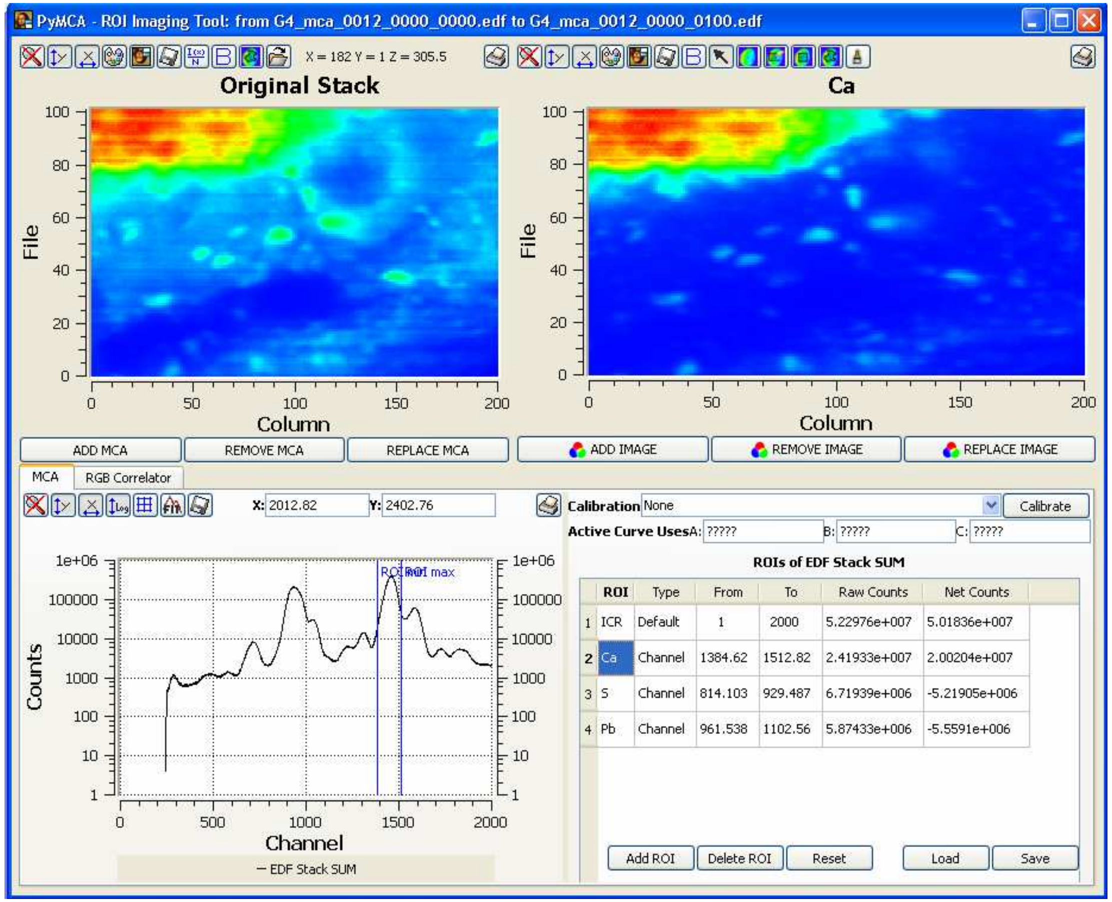


Lead and antimony not correlated

… 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

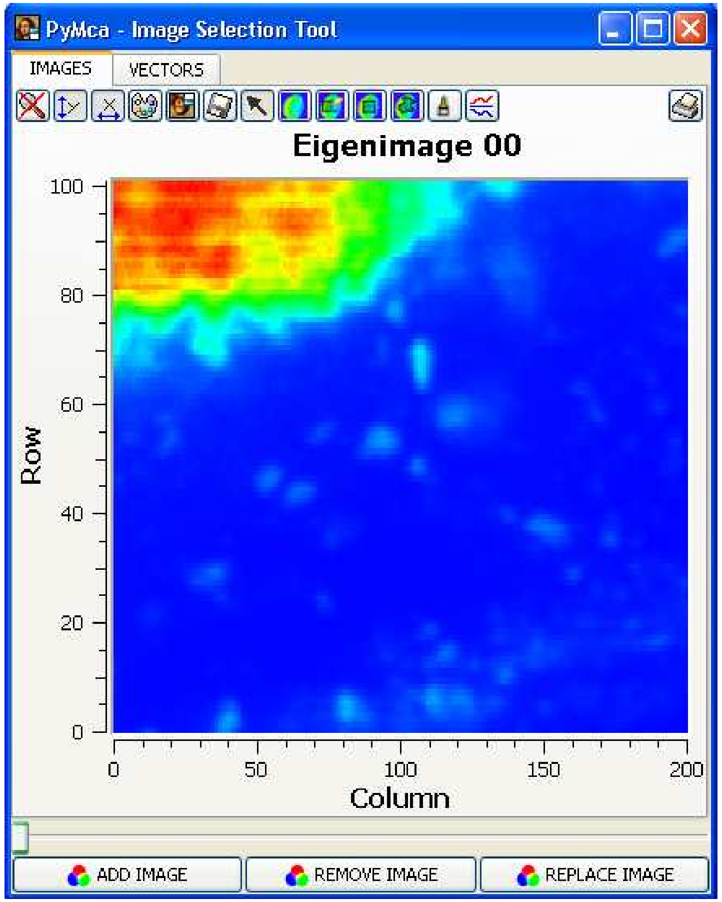
20200 spectra of 2000 channels

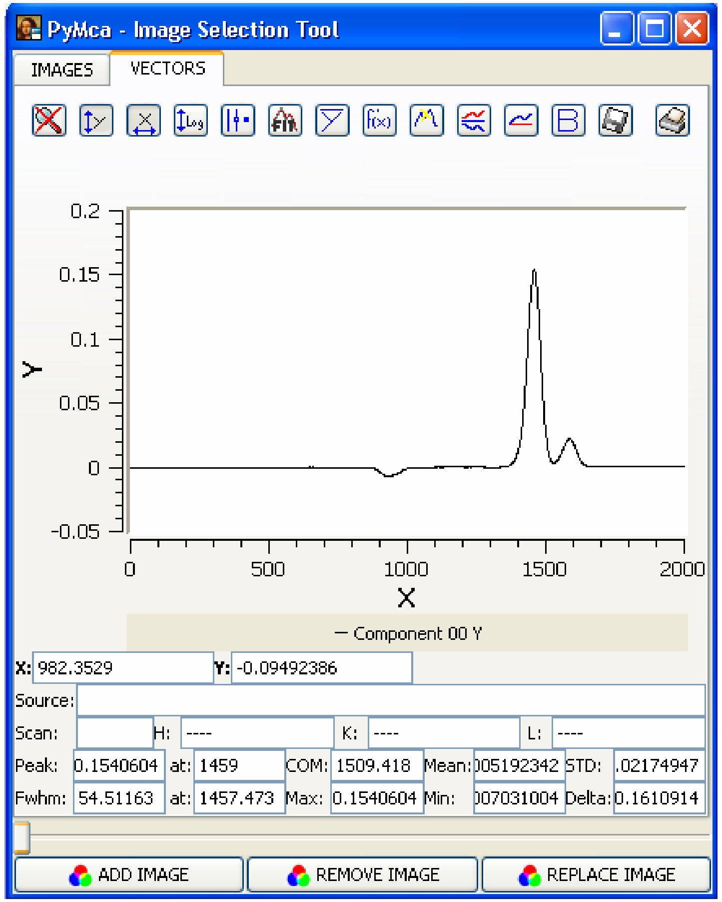
Pixel[i, j] = numpy.sum(Stack[i, j, :])

Pixel[i, j] = numpy.sum(Stack[i, j, ch0:ch1])

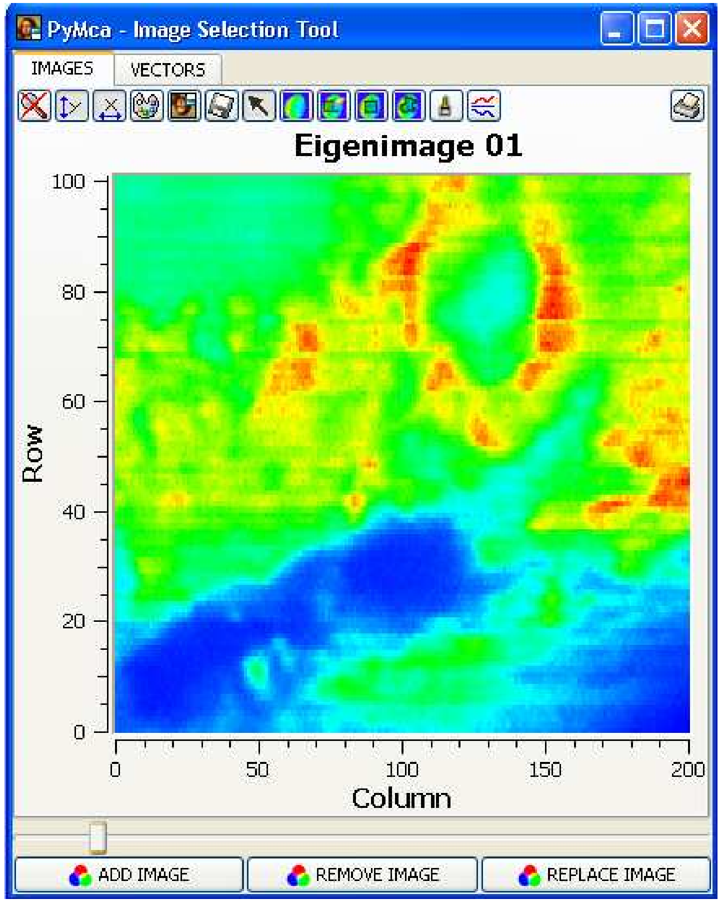
We can generate new images by moving the cursors or defining new ROIs in the table

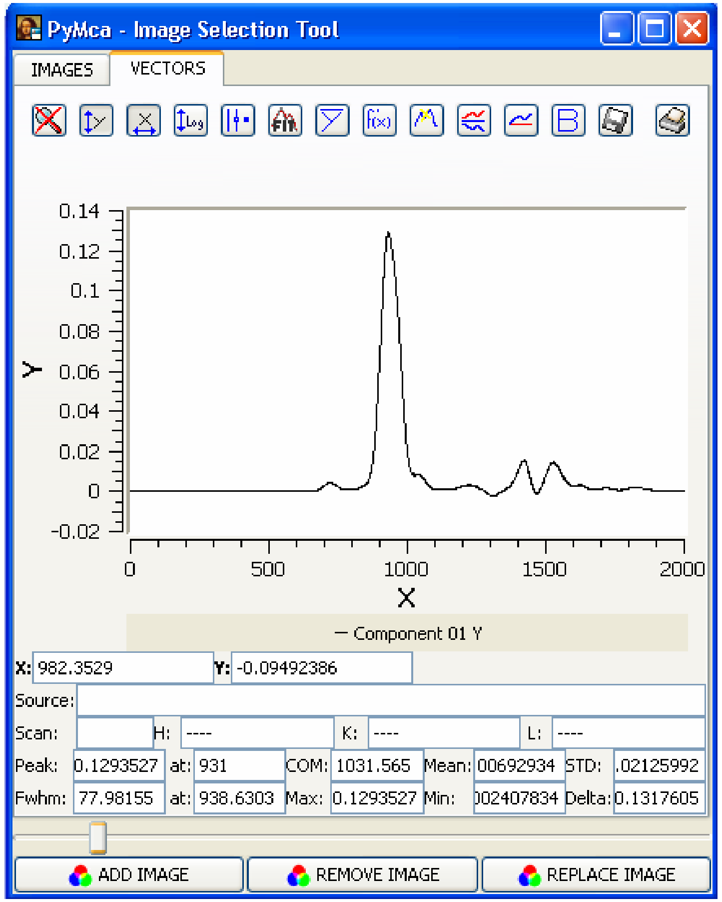
Eigenimages and Eigenvectors



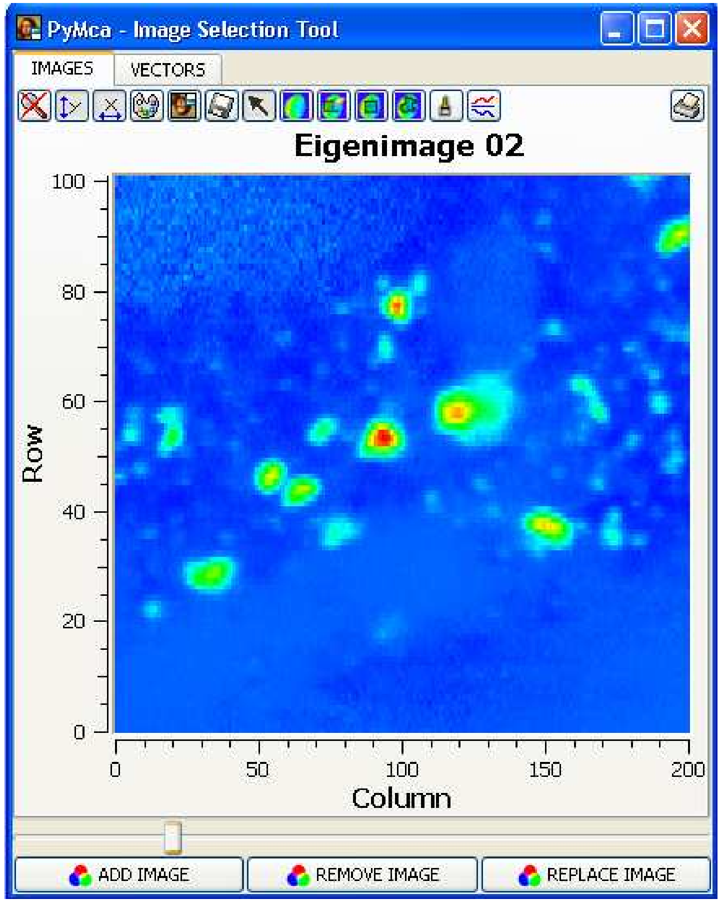


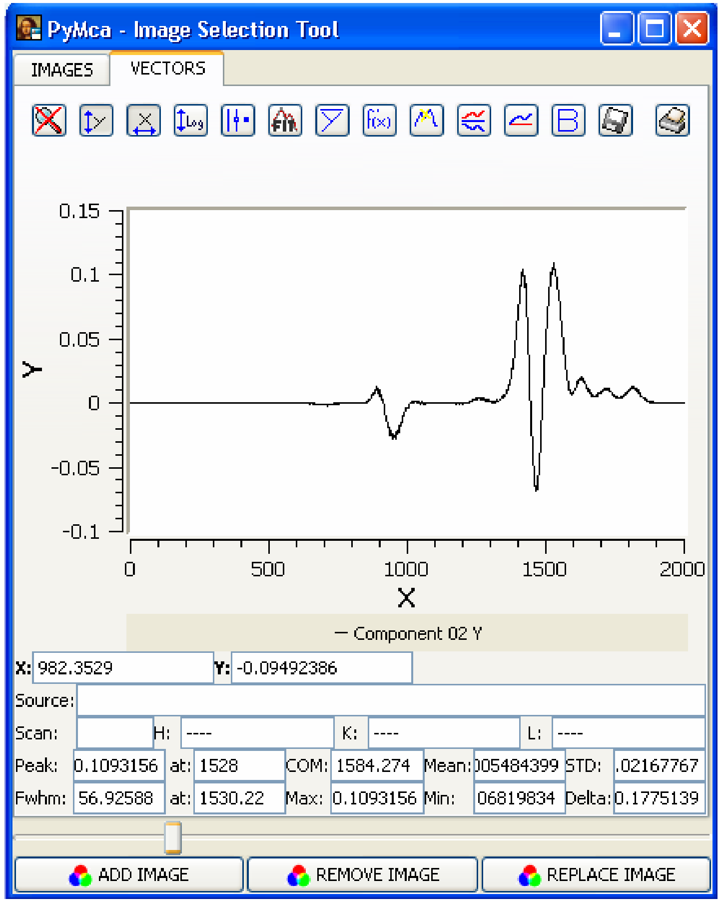
Eigenimages and Eigenvectors



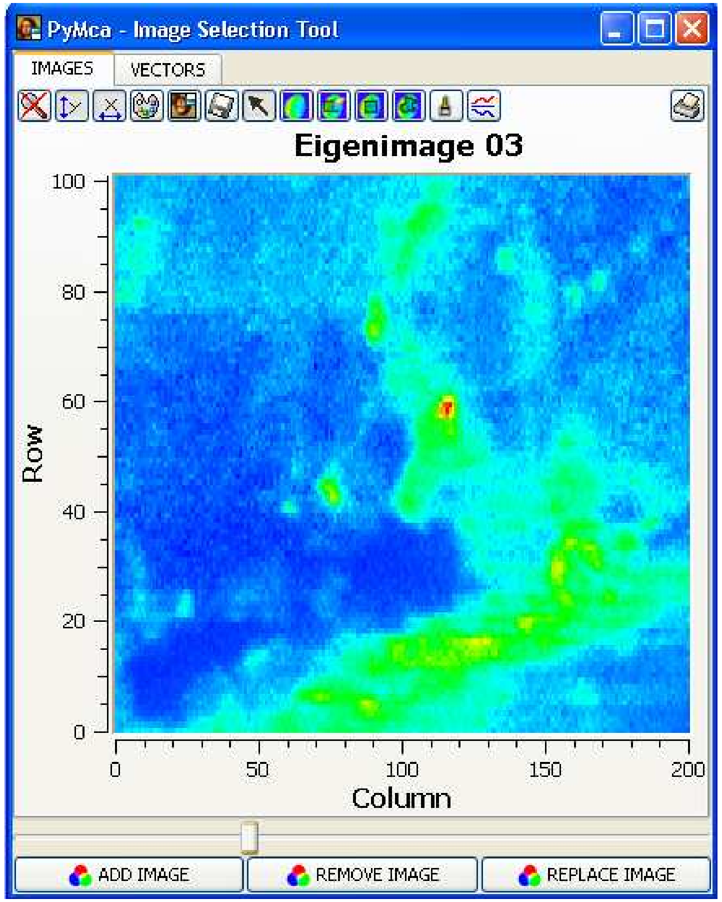


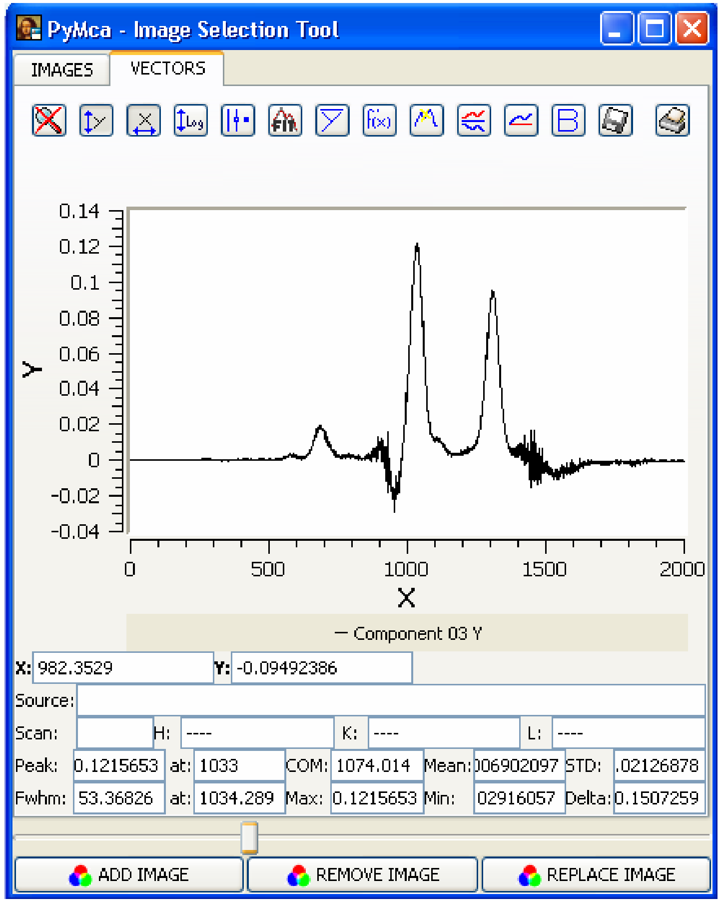
Eigenimages and Eigenvectors



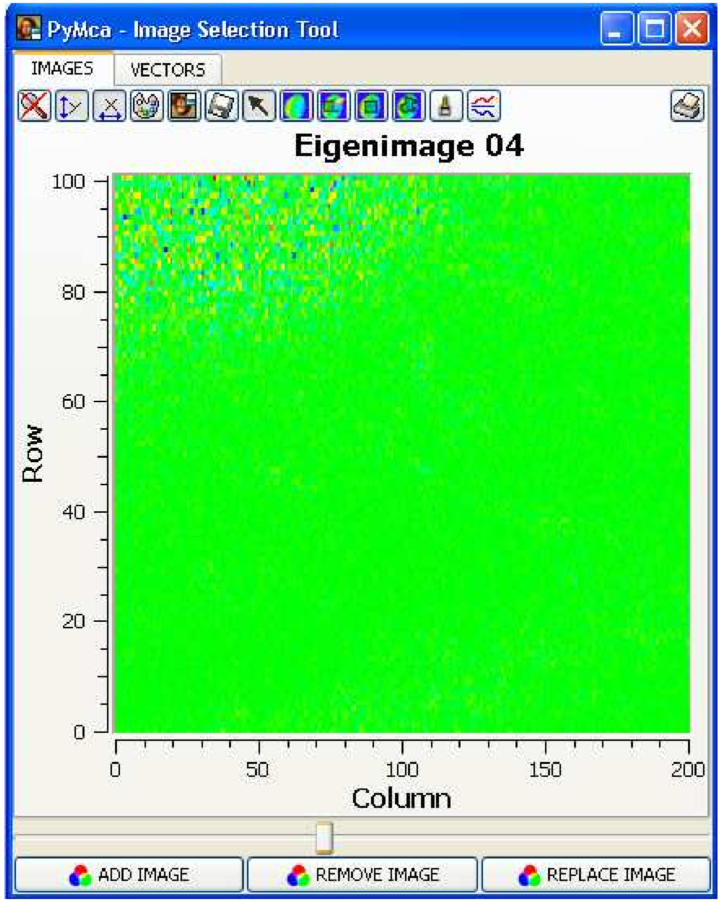


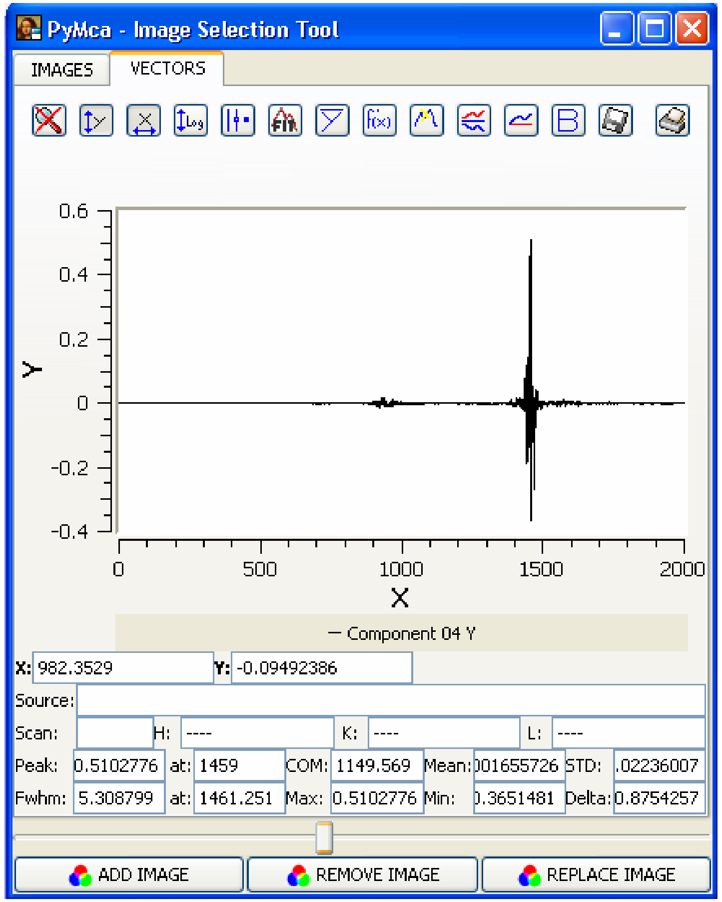
Eigenimages and Eigenvectors



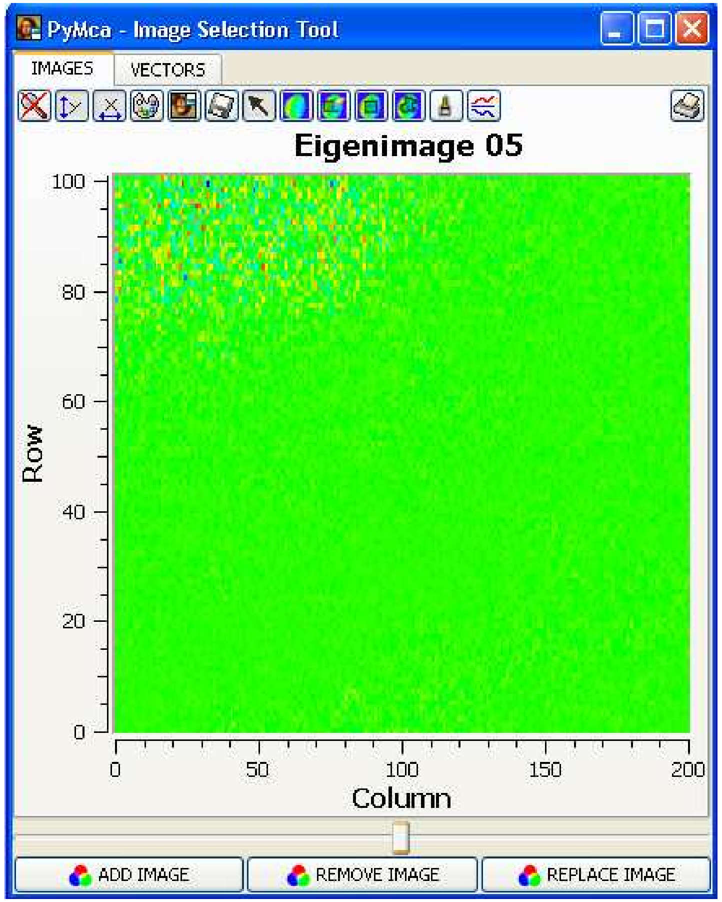


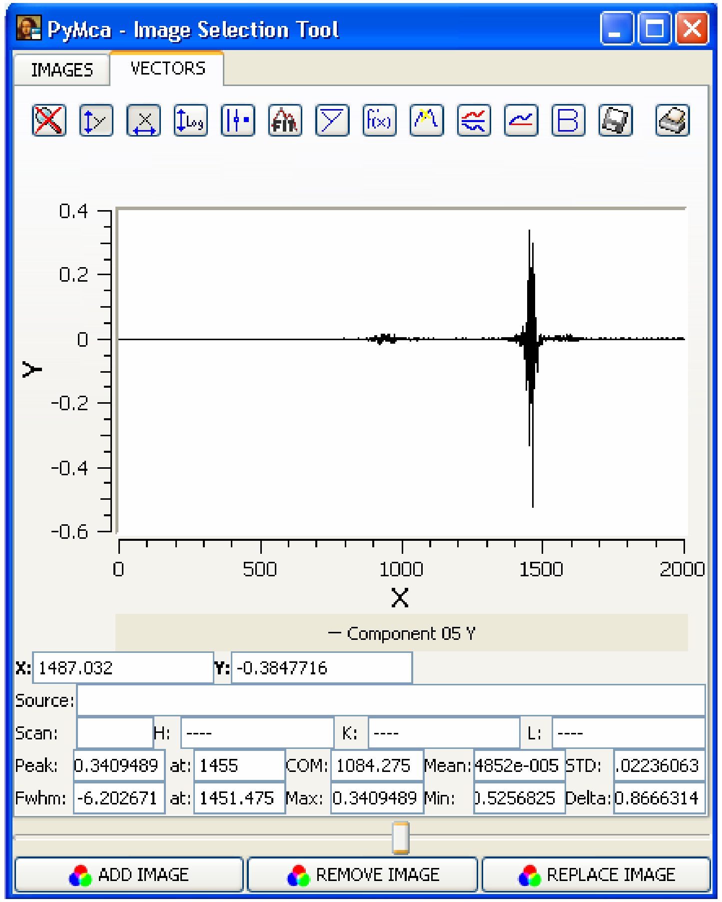
Eigenimages and Eigenvectors



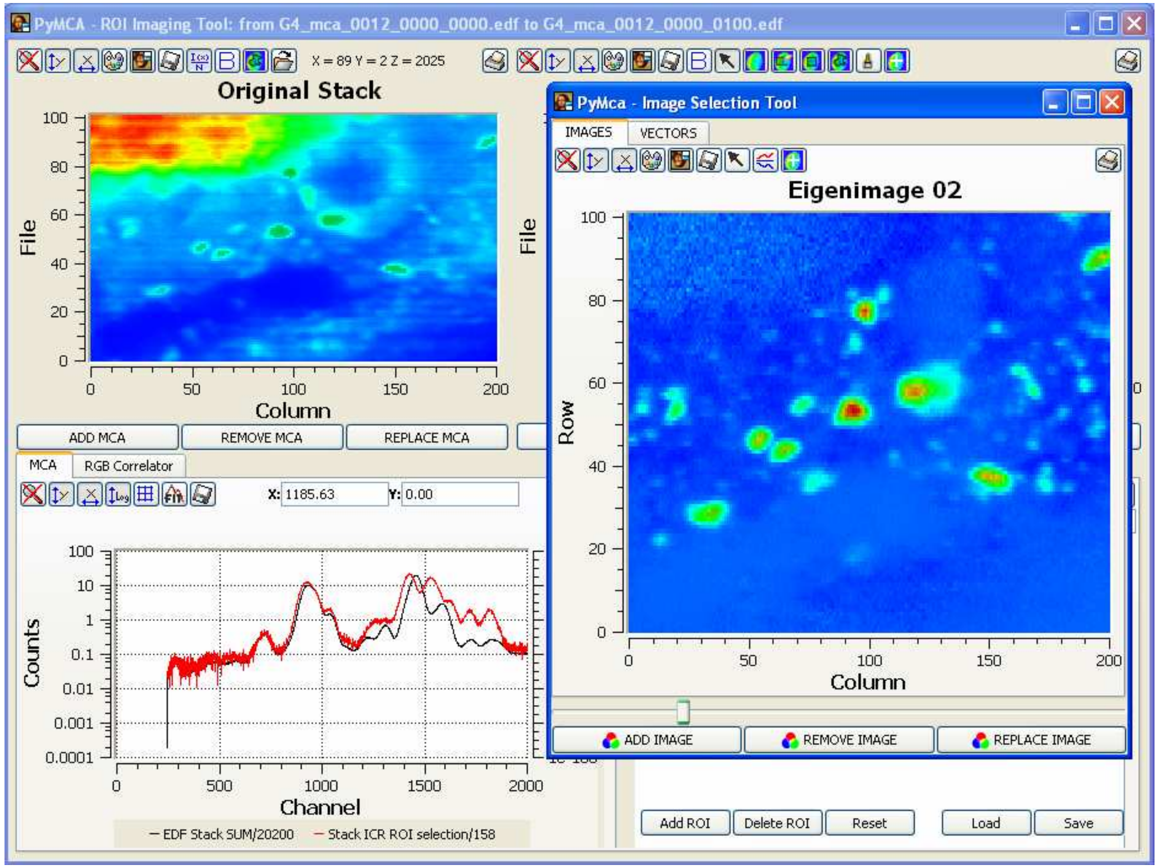


Eigenimages and Eigenvectors



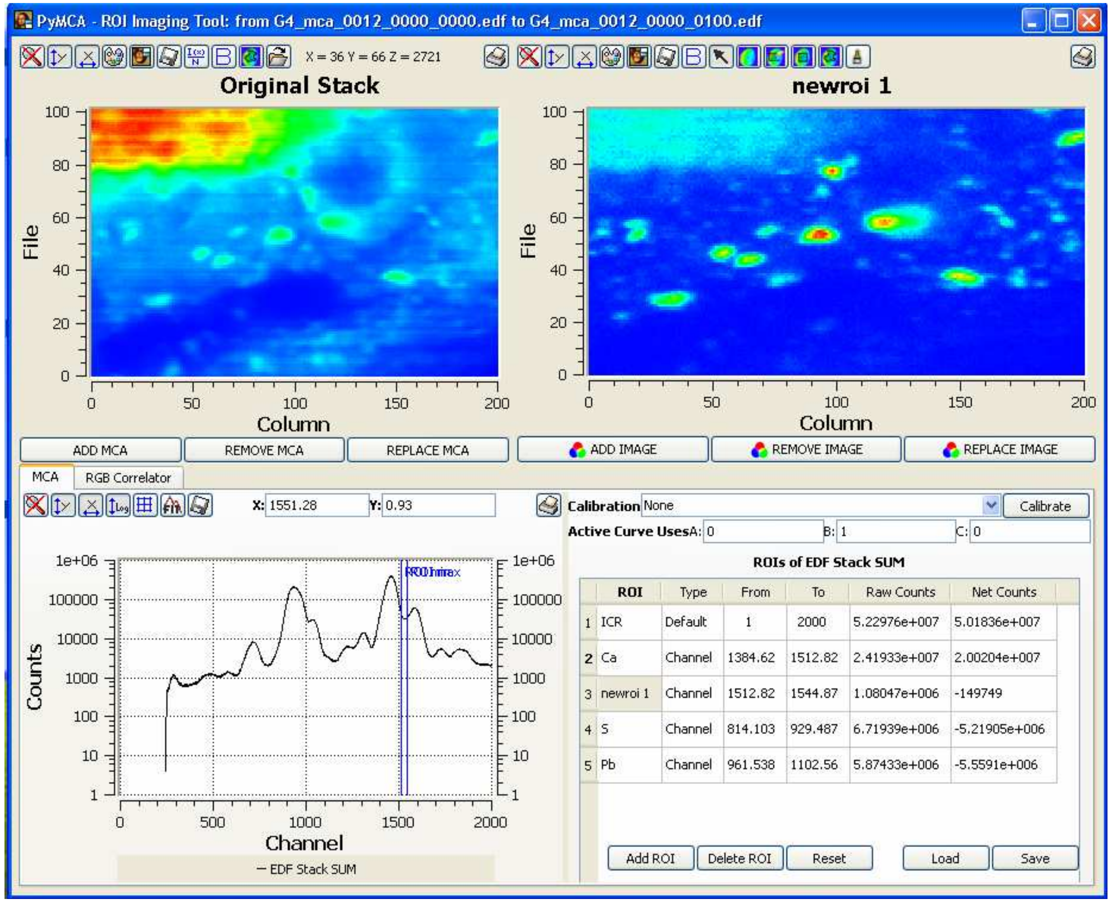


Getting the actual information



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

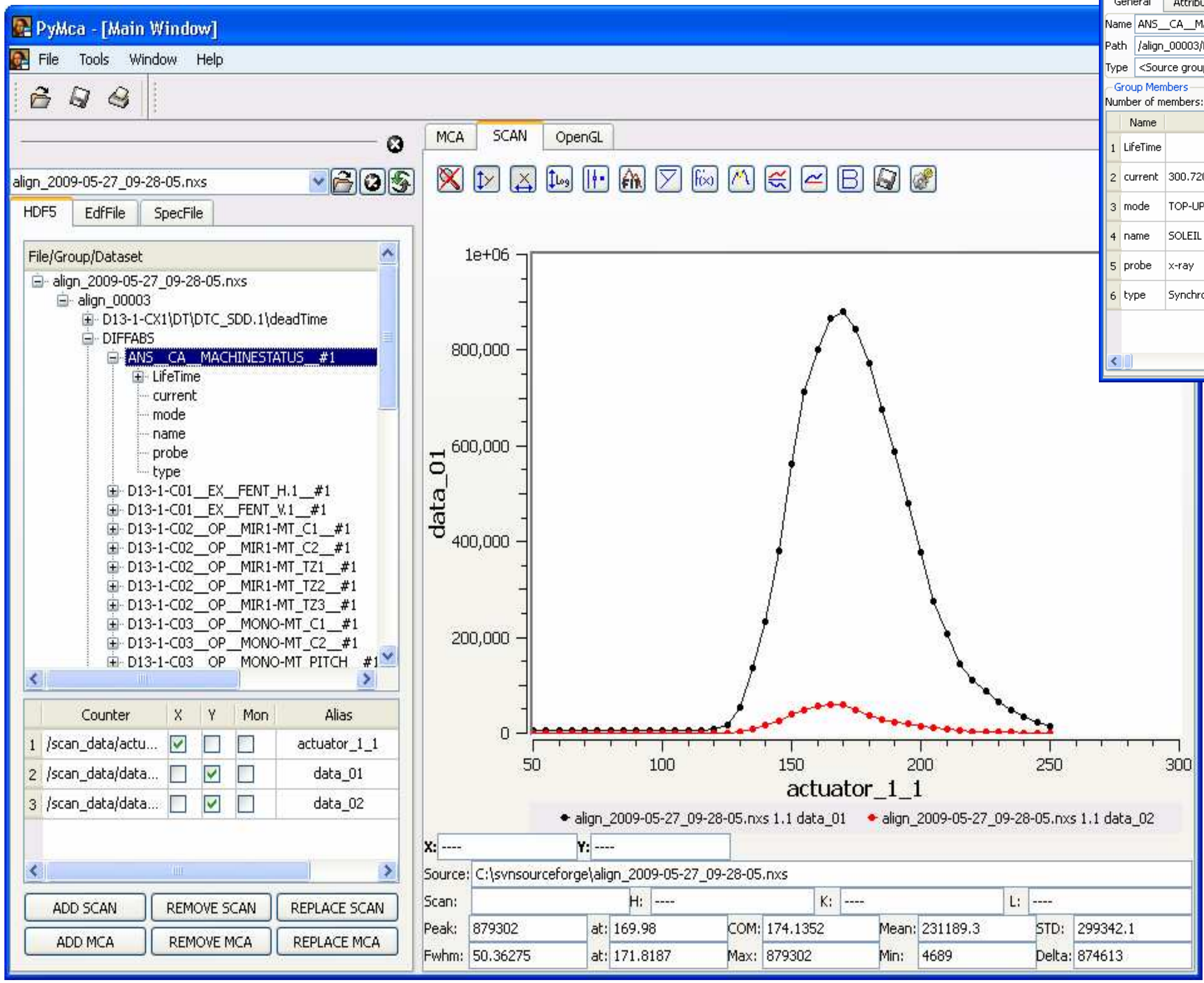
Multivariate Analysis Capabilities

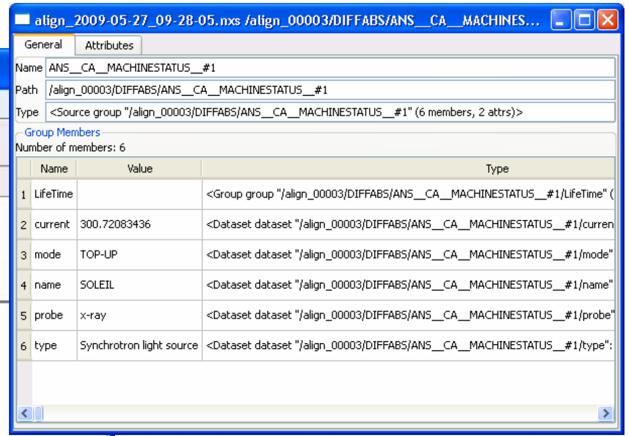
PyMca makes use of PCA, ICA (via MDP toolkit) and NNMA (via py_nnma)

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

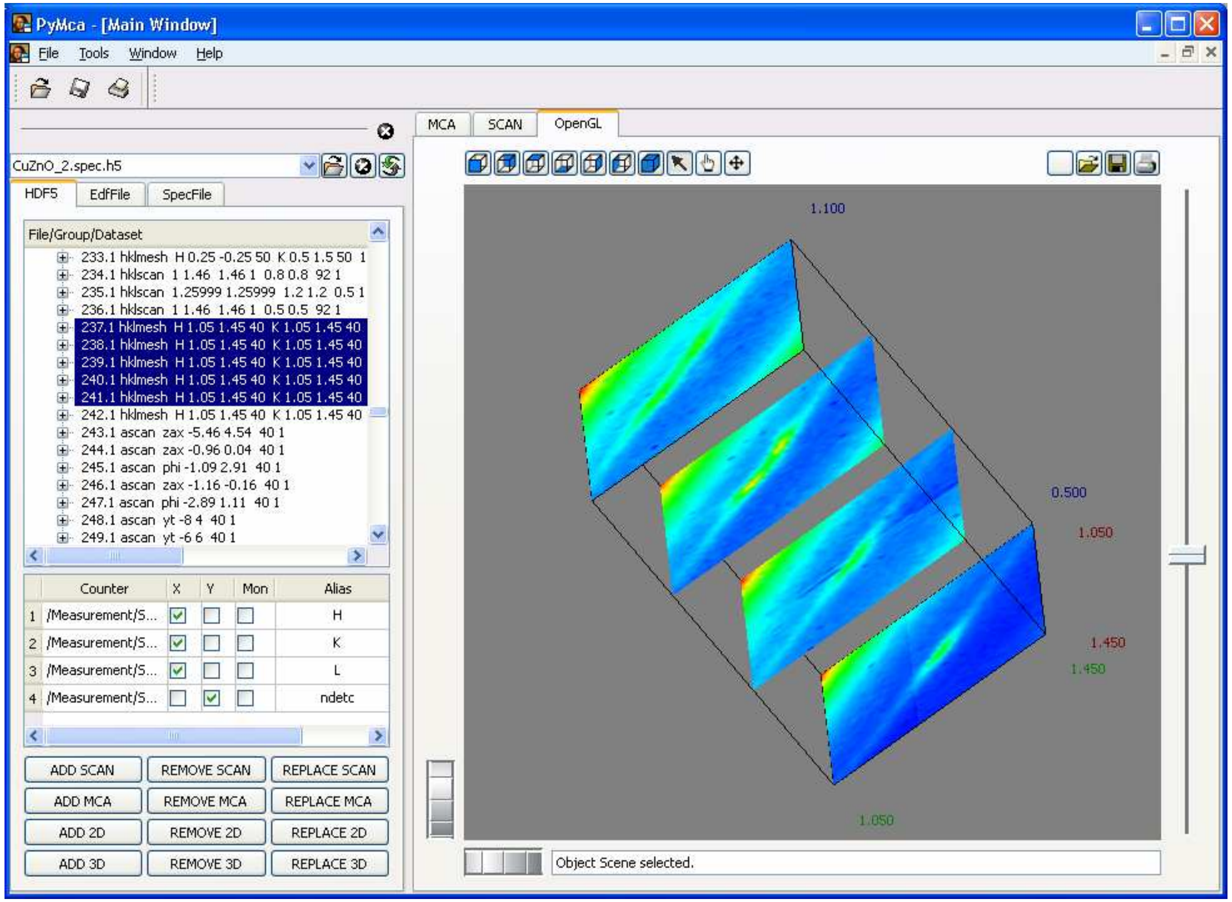




HDF5 Support
Collaboration with D. Dale, CHESS

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.

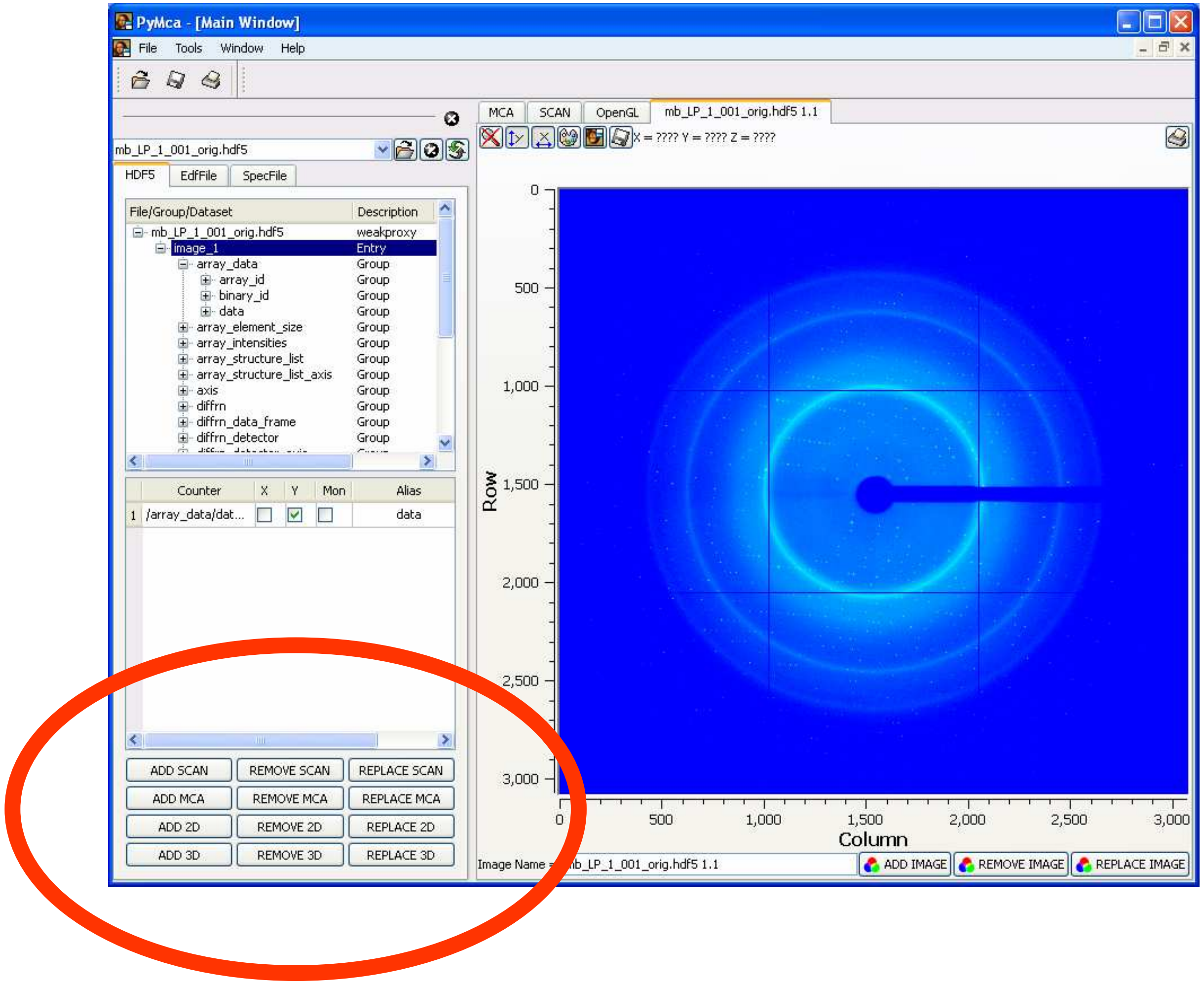
One attribute to say "ACTUAL_DIMENSION = 2"
One attribute to say "ACTUAL_DIMENSION_IS_LAST_DIMENSION = 1"

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