

ESRF Data Acquisition Related Activities For Fluorescence Imaging and Spectroscopy

V.A. Solé – ESRF Data Analysis

Acknowledgements

The activities presented in this talk are the work of

- The beamline staff of the ESRF imaging and spectroscopy beamlines ID21, ID22 and ID24
- The ESRF support groups, mostly the Beamline Acquisition and the Data Analysis units

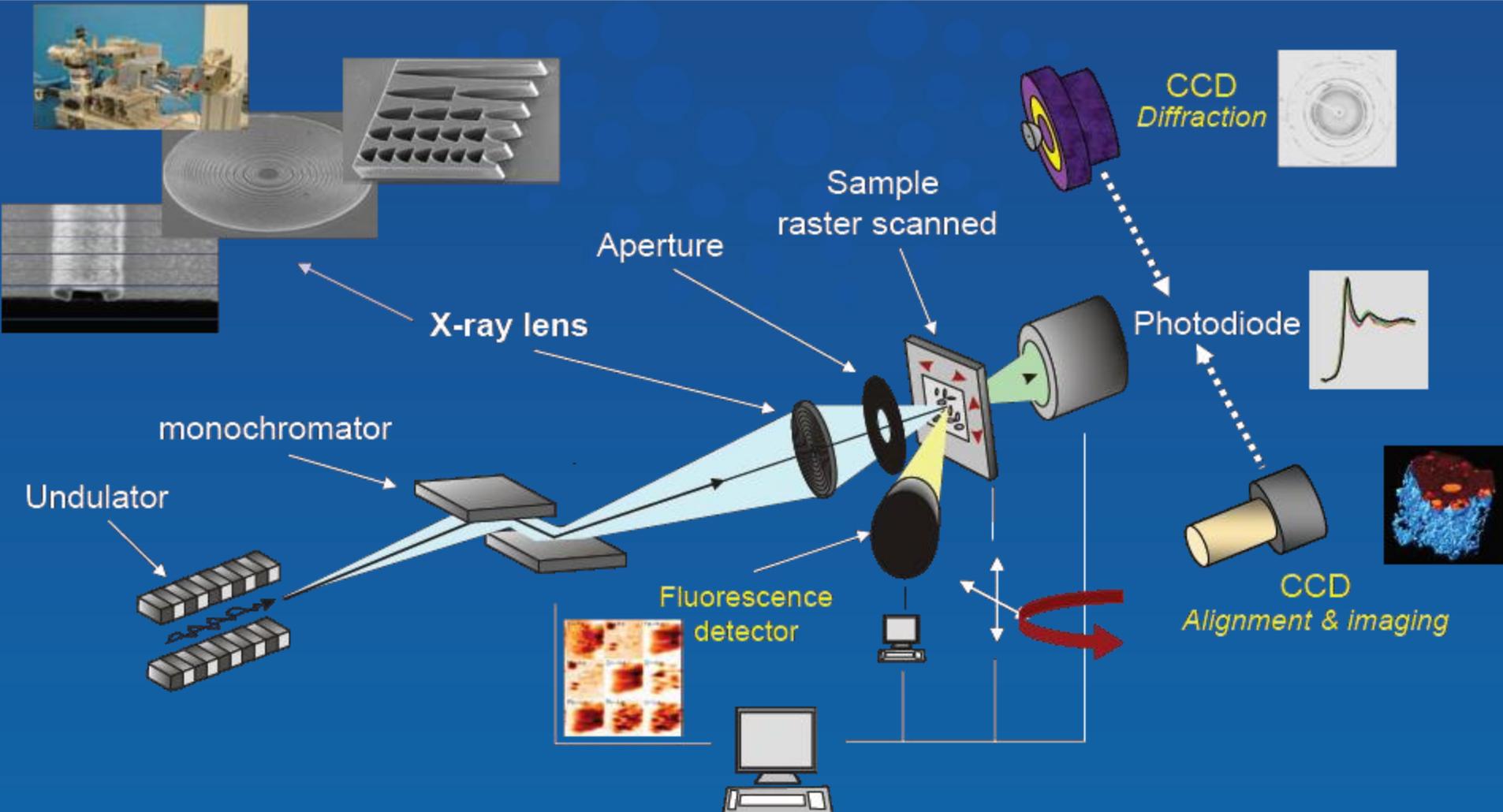
Talk Outline

Current situation

- Techniques
- Data Acquisition
- Data Storage
- Data Analysis

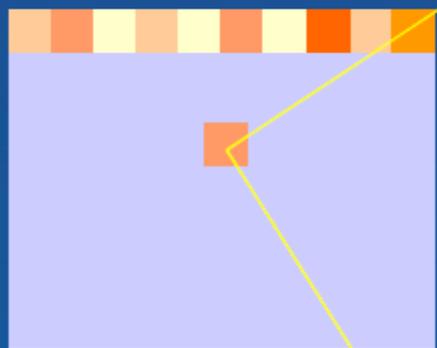
On going activities

- Techniques
- Data Acquisition
- Data Storage
- Data Analysis

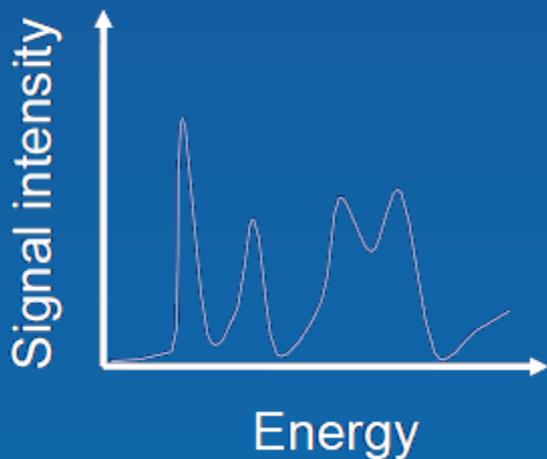


- Spatial resolution : 0.05-2 μ m
- Spectral resolution : $\Delta E/E \sim 10^{-2} - 10^{-4}$
- Averaged flux : $10^9 - 10^{13}$ photons/s/ μ m²

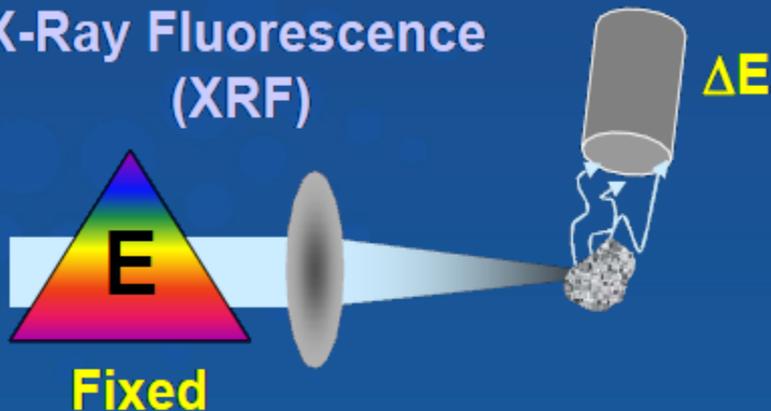
1 - Microscopy: $< \Delta x, \Delta y < 1 \mu\text{m}$



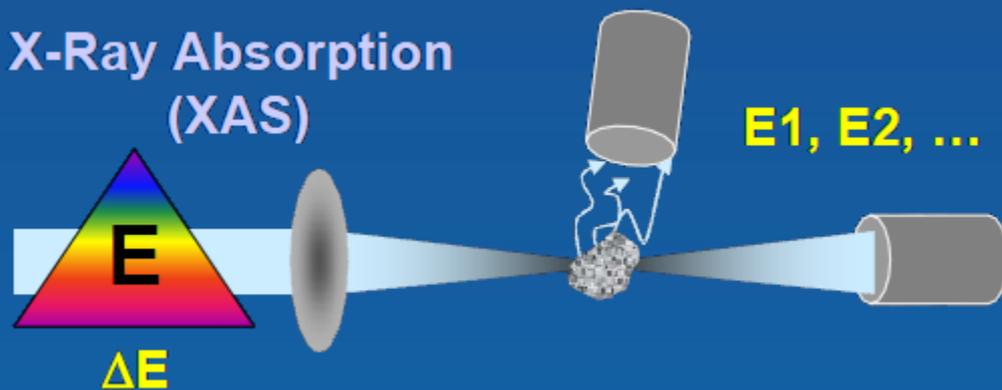
2 - Spectroscopy: ΔE



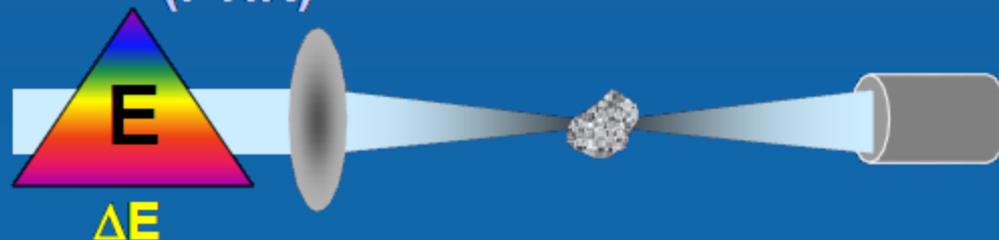
X-Ray Fluorescence (XRF)

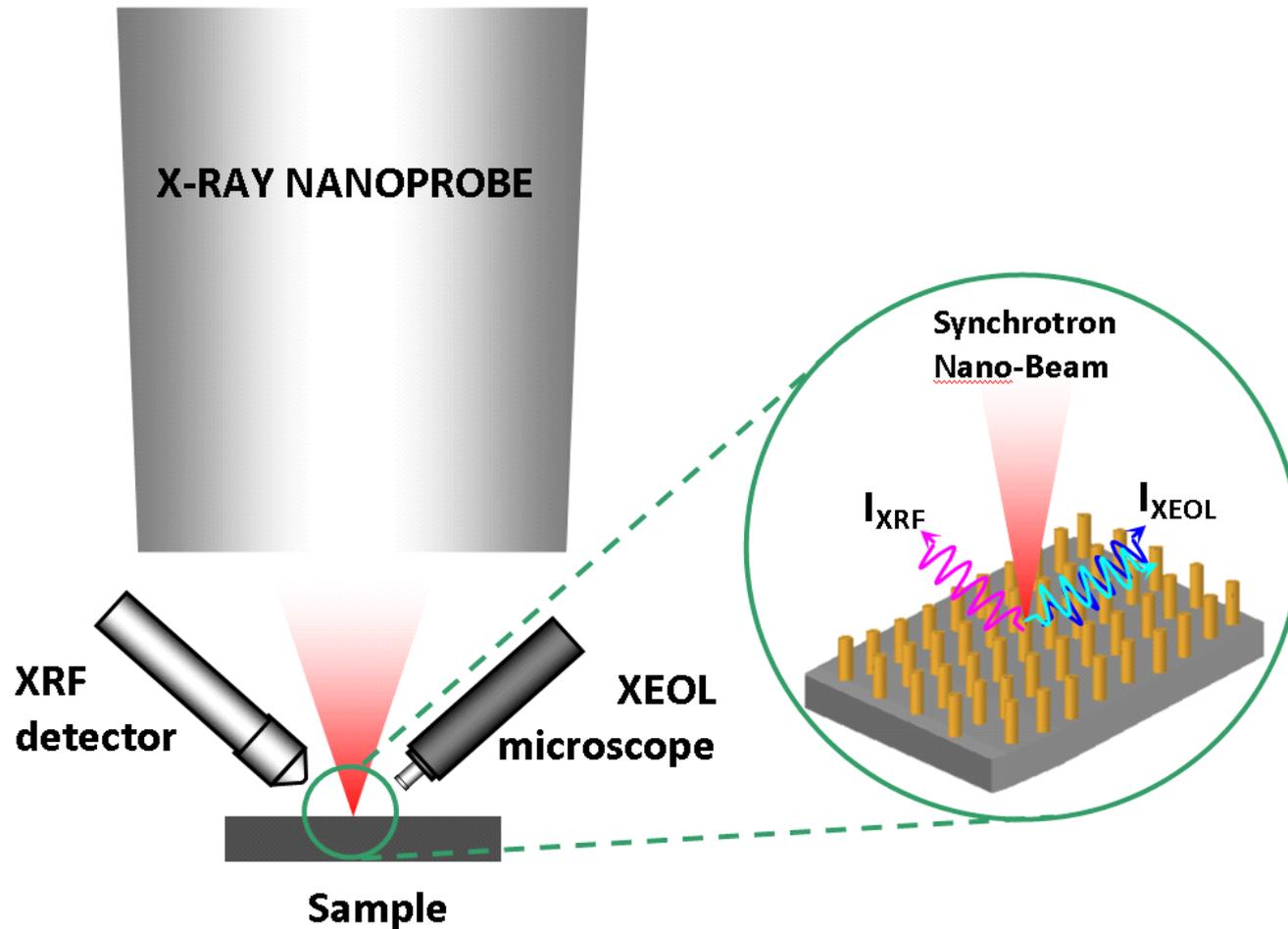


X-Ray Absorption (XAS)



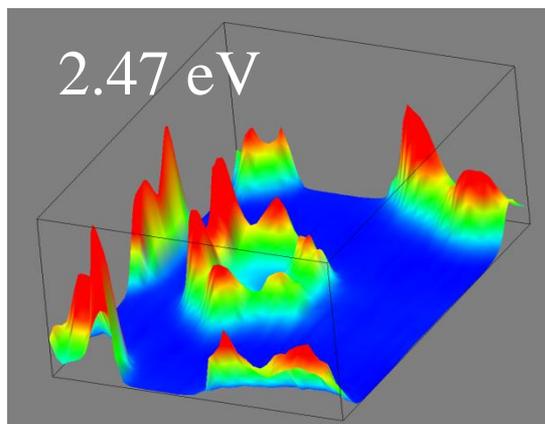
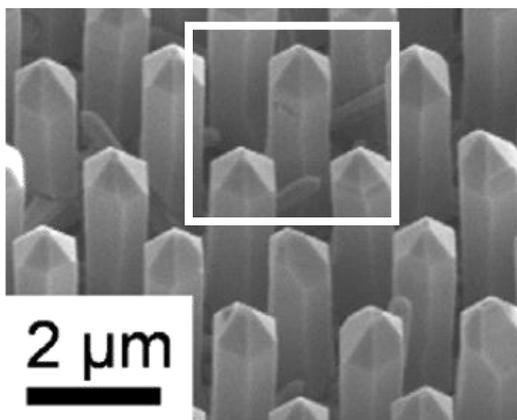
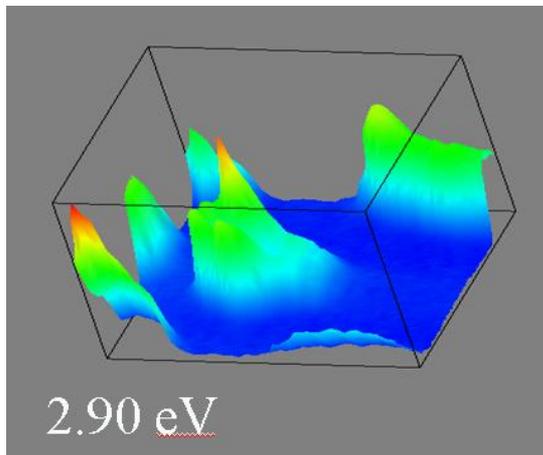
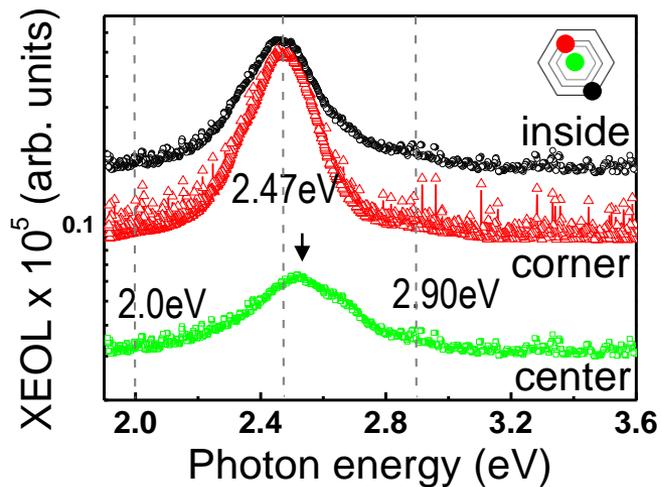
Infrared Absorption (FTIR)



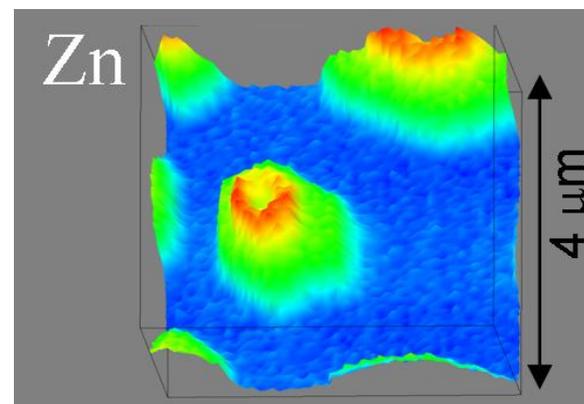
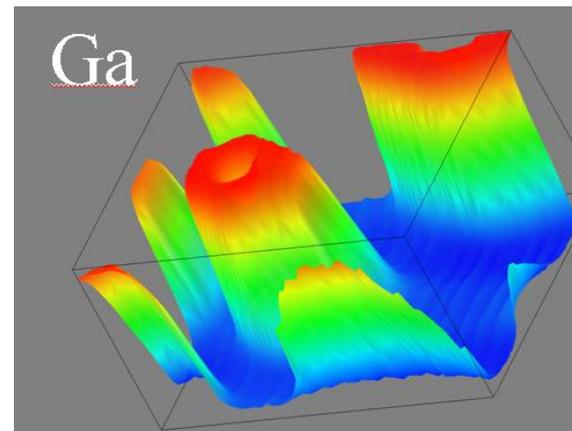


Device Function ↔ Nanostructure Properties

XEOL Imaging



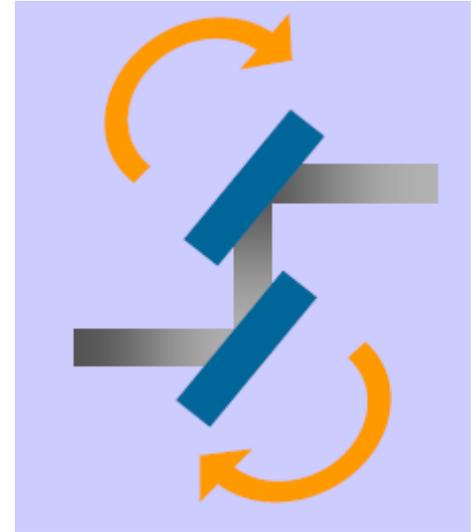
XRF Mapping



XAS Mapping: Energy Scan Approaches

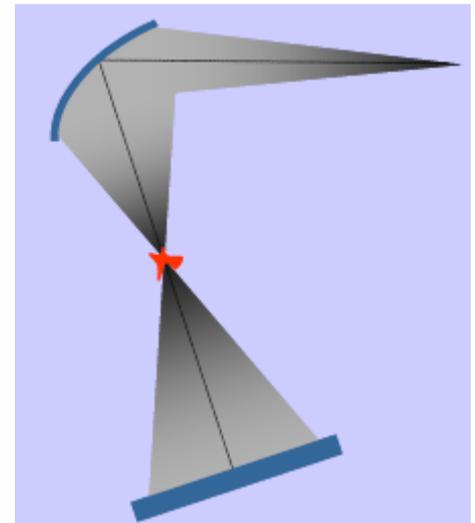
Two-crystal fixed-exit scanning monochromator

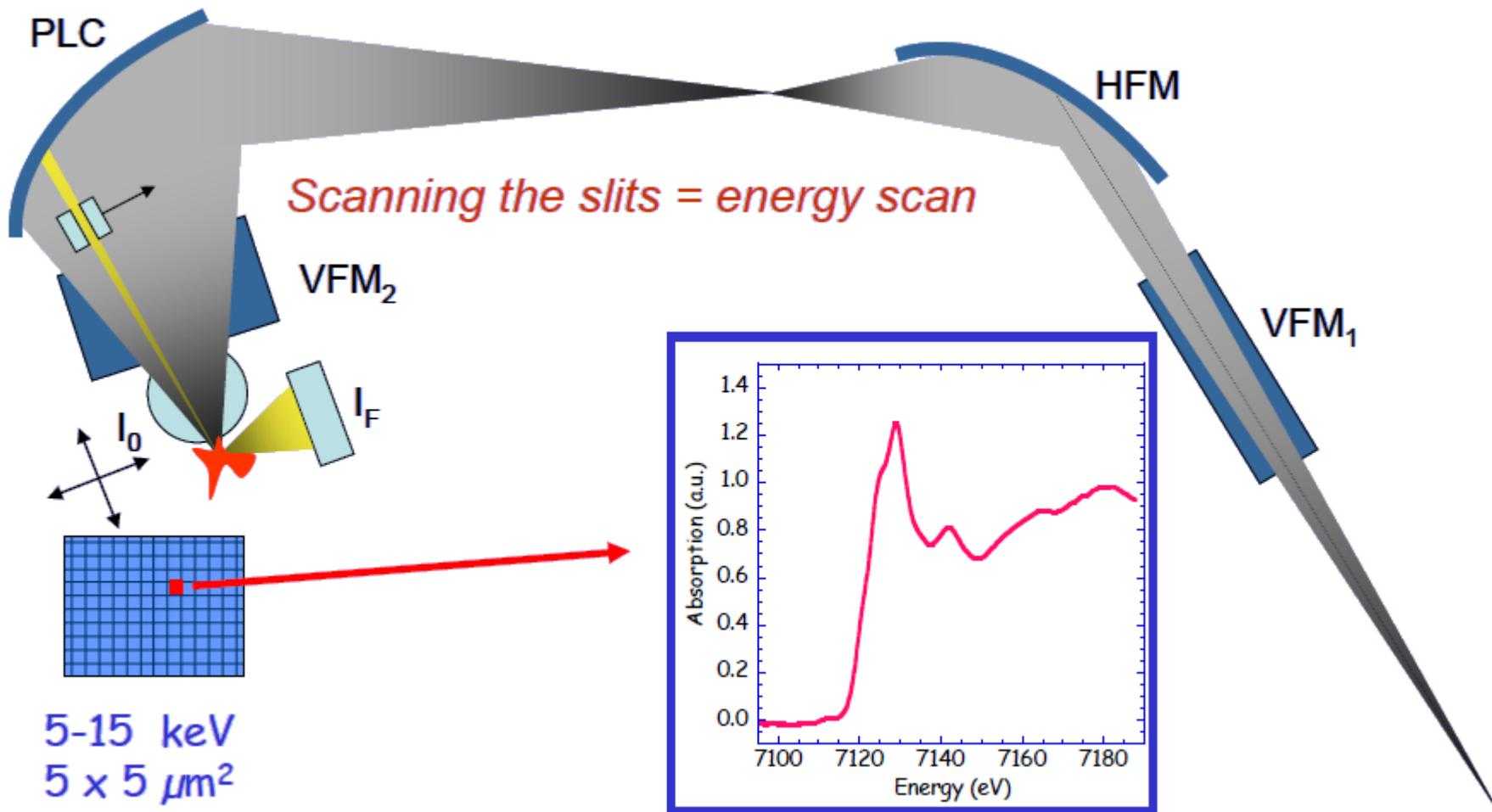
- + Large energy range for XAS
- + High flux
- + Mature technology
- Source of non-statistical noise
- Relatively slow (10s – 1s)



Wavelength dispersive polychromator

- + Fast (1s – 1 μ s)
- + Stable
- Low flux
- Limited energy range
- Fluorescence mode more difficult





Slits 50 mm/s – 1000 eV/s

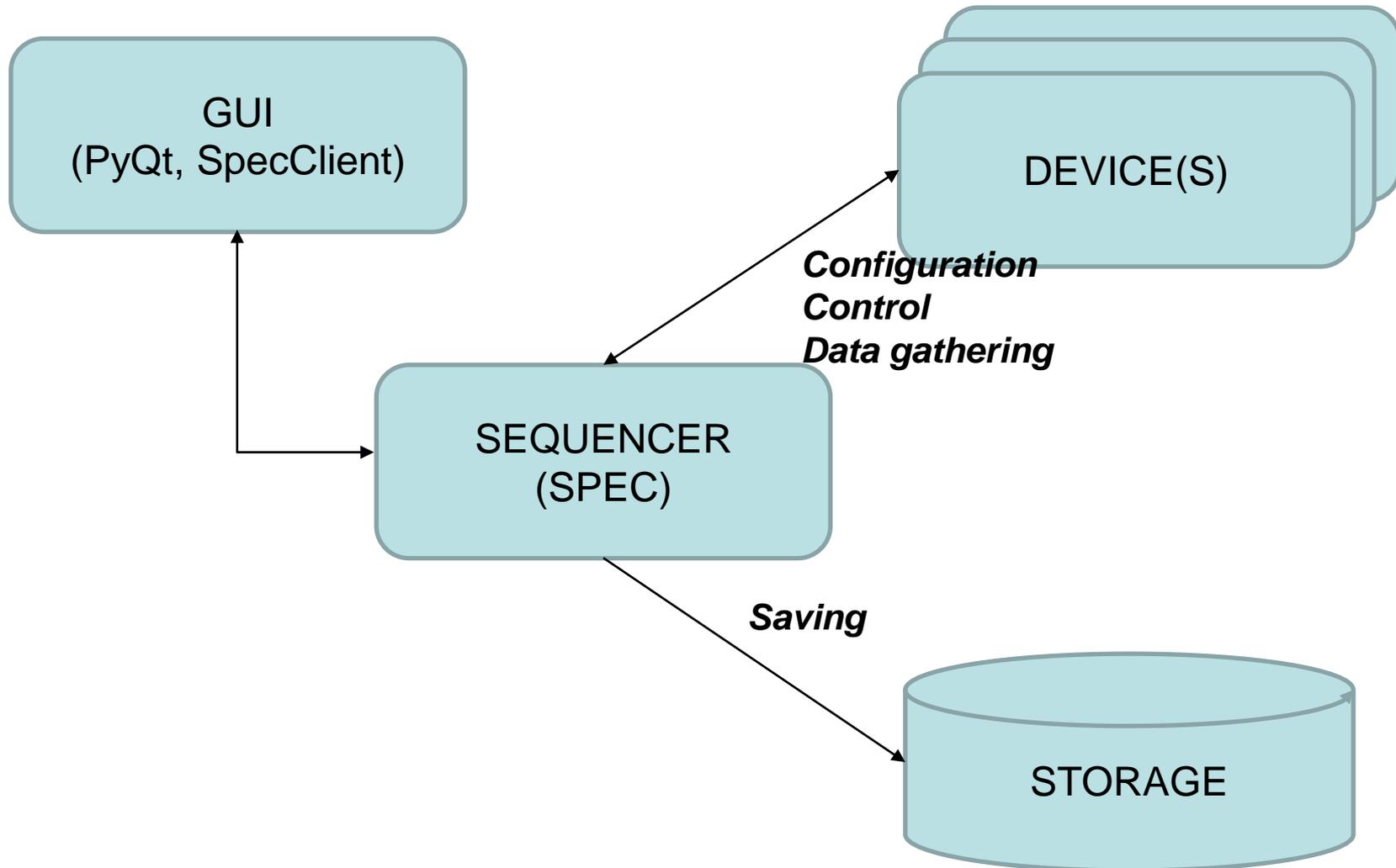
$\Delta E/E$: 10^{-1} – 10^{-4}

Tuning of flux versus $\Delta E/E$

100 x 100 pixels map in 2 hours

Pascarelli et al., J. of Synchrotron Radiation, 2006

Current Situation: Data Acquisition



ID21 Data Acquisition GUI

Video Display | X Ray Display

X:1093 Y:1026 Grid grab, select the area

Spec | SXM Config | SXM KB Config | Sample

```

Error on "tango:id21/prosilicacato_6/00" sending DevCcdXSize.
It is currently not allowed to read attribute widthFailed to read_attribute on device id21/prosilica_6/00, attribute widthFailed to ex
t on device id21/prosilicacato_6/00, command DevCcdXSize (61).
esrf_io() server error.
Error on "tango:id21/prosilicacato_6/00" sending DevCcdYSize.
It is currently not allowed to read attribute HeightFailed to read_attribute on device id21/prosilica_6/00, attribute heightFailed to e
out on device id21/prosilicacato_6/00, command DevCcdYSize (61).
*** setting up detector motors
Reconfiguring the CCD camera0 ...
*** setting up KB motors
*** setting up WDX
1538,OPTICS> piwa_uu
mot. name | mot. pos (mm) | enc. pos (um) | angle (mrad) | wago (V) |
kbvf | -0.692 | -692.07 | -6.018 | 5.1093 |
kbhf | -0.656 | -656.94 | -6.109 | 5.5206 |
1539,OPTICS> Warning: motor motor 2, unit 8, "samz" is busy.
1539,OPTICS> Warning: motor motor 1, unit 8, "samy" is busy.
Warning: motor motor 2, unit 8, "samz" is busy.
1539,OPTICS> []
                    
```

Zap Image | Zap Energy | Zap Scan | z3escan | Generated Scans

Sample Zap Image

Video Grab

X-Ray Grab

	Motor	Start	Stop	Nb. Points	Interval
Horizontal	samy	7.254	6.852	27	0.015
Vertical	samz	1.786	2.159	55	0.007

Pixel Dwell Time (ms)

Vertical as Fast Axis Zap Puzzle Zap Lookup

Uni-Directional

Fluorescence Det.

Fast Shutter on Image

	samx	samy	samz	sampy	sampz
Current	0.000	7.049	1.965	50.00	50.00
Start	0.000	7.254	1.786	50.00	50.00

Time Spent Time Left

Files | Counters | Keithley | Xia Roi | Mca Roi | Mca Config | Calibration | WDX

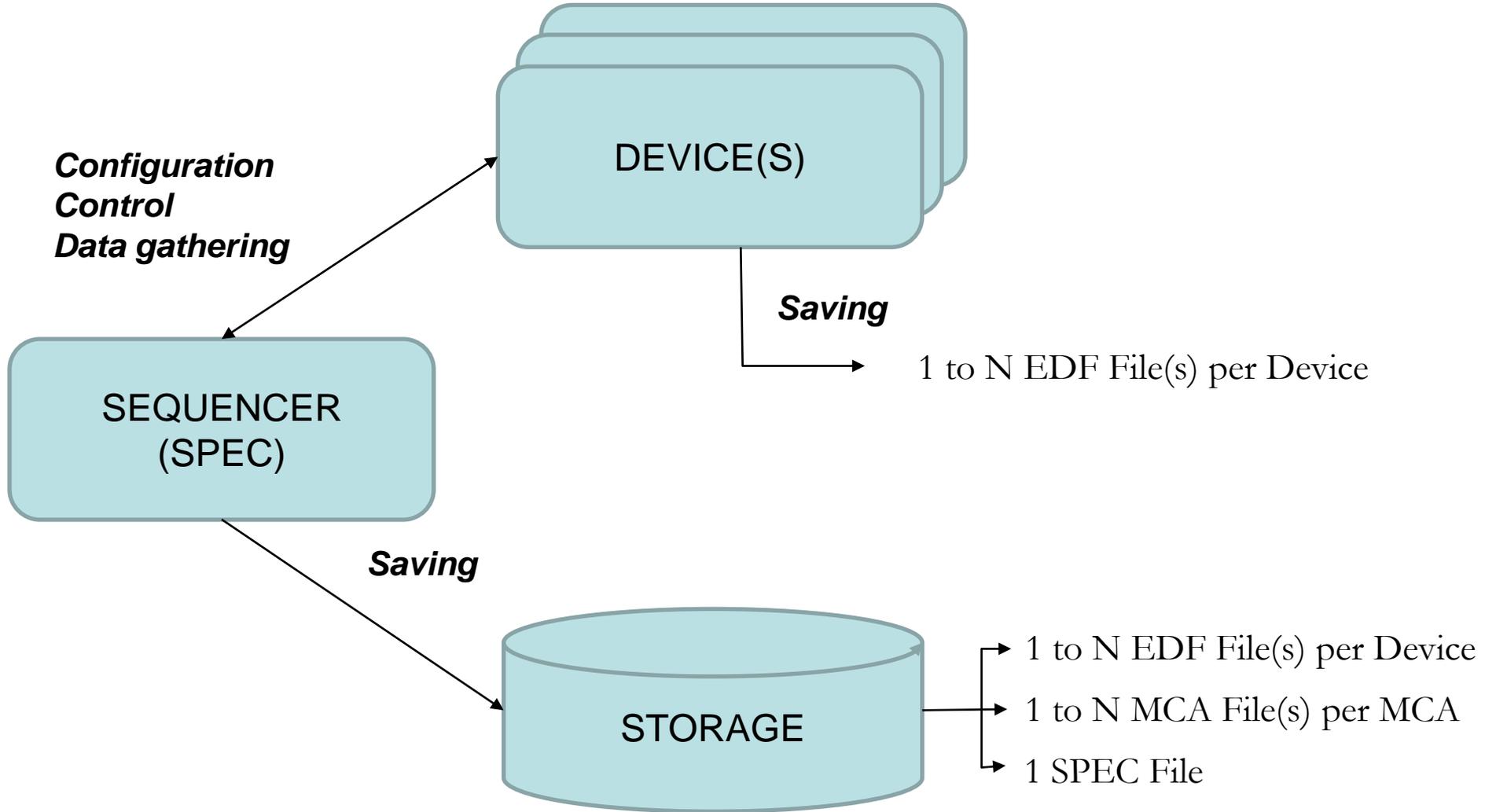
Mca Roi

	Counter	Name	MCA	Ch. From	Ch. To	En. From	En. To
1	dsp2	S	0	1000	1098	2247	2492
2	dsp3	P	0	865	950	1909	2122
3	dsp5	K	0	1370	1500	3172	3498
4	dsp6	Cl	0	1113	1185	2530	2710
5	dsp1	Fe	0	2550	2700	6124	6499
6	dsp4	Ca	0	1525	1630	3560	3823

N

Calibration A = B = C =

Current Situation: Data Storage



Current Situation: Acquisition File Formats

EDF (ESRF Data Format):

- ASCII Header followed by Binary Data
- Mainly used for Images and MCA
- It can contain multiple images and 3D arrays
- Readable by PyMca, Oxidis, ImageJ

SPEC File:

- ASCII
- Mainly used for motor positions and scalar counters
- It can contain multiple MCA
- Readable by PyMca, Newplot and custom user tools

XRF Data Analysis: ESRF needs in 2004

Offline

- **Complete and freely distributable XRF analysis**
 - To reduce beamline staff work load
 - To implement new features

Online

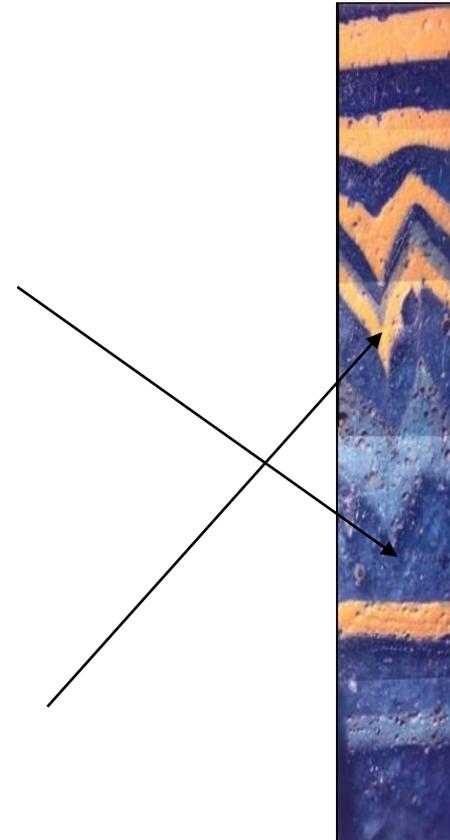
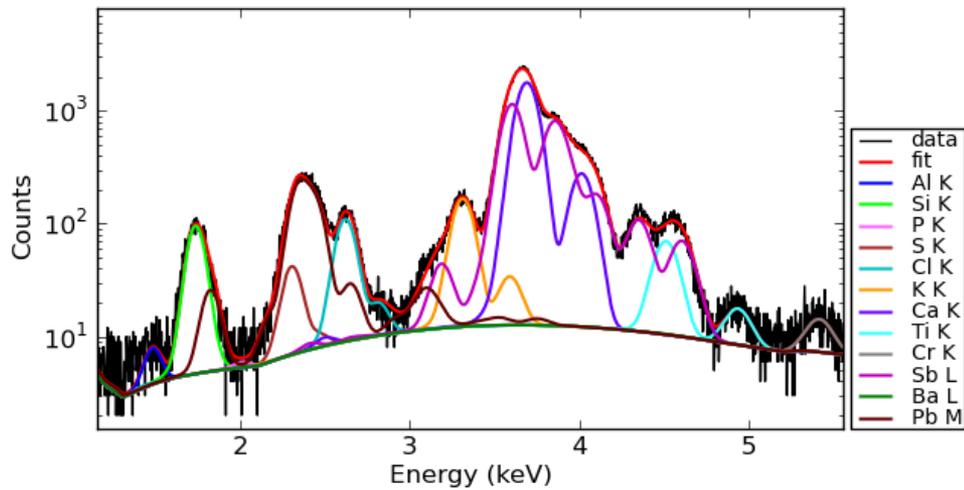
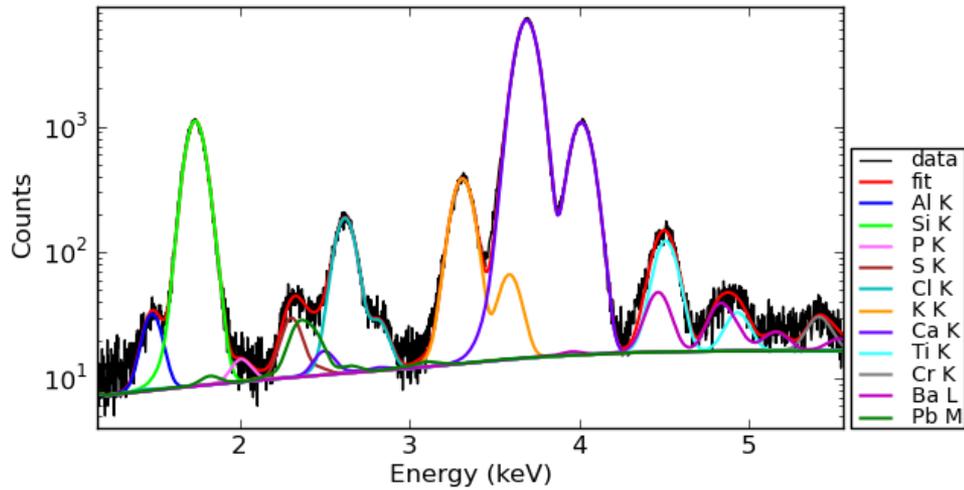
- **Visualization of Multi-Channel Analyzer data**
 - Peak search
 - Energy calibration
 - Peak Area estimation
- Scan visualization
- Simple X-ray fluorescence analysis (XRF)
 - Peak identification

PyMca (2004 - ????)

- Easy to embed Python modules
- Source code available from sourceforge
 - Official releases
 - SVN repository
- Minimal dependencies
 - Easy distribution
 - Multiplatform
- End user not expected to be a programmer
 - Easy installation
 - Distribution of ready-to-use binaries



Complete M-shell support



Use of (sub-)shell mass attenuation coefficients

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

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

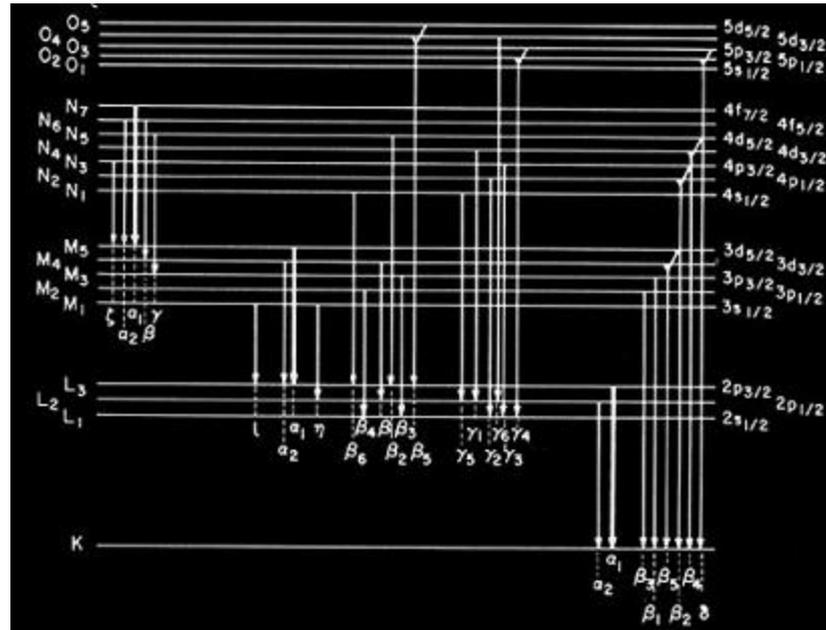
Term between parenthesis for Pb

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

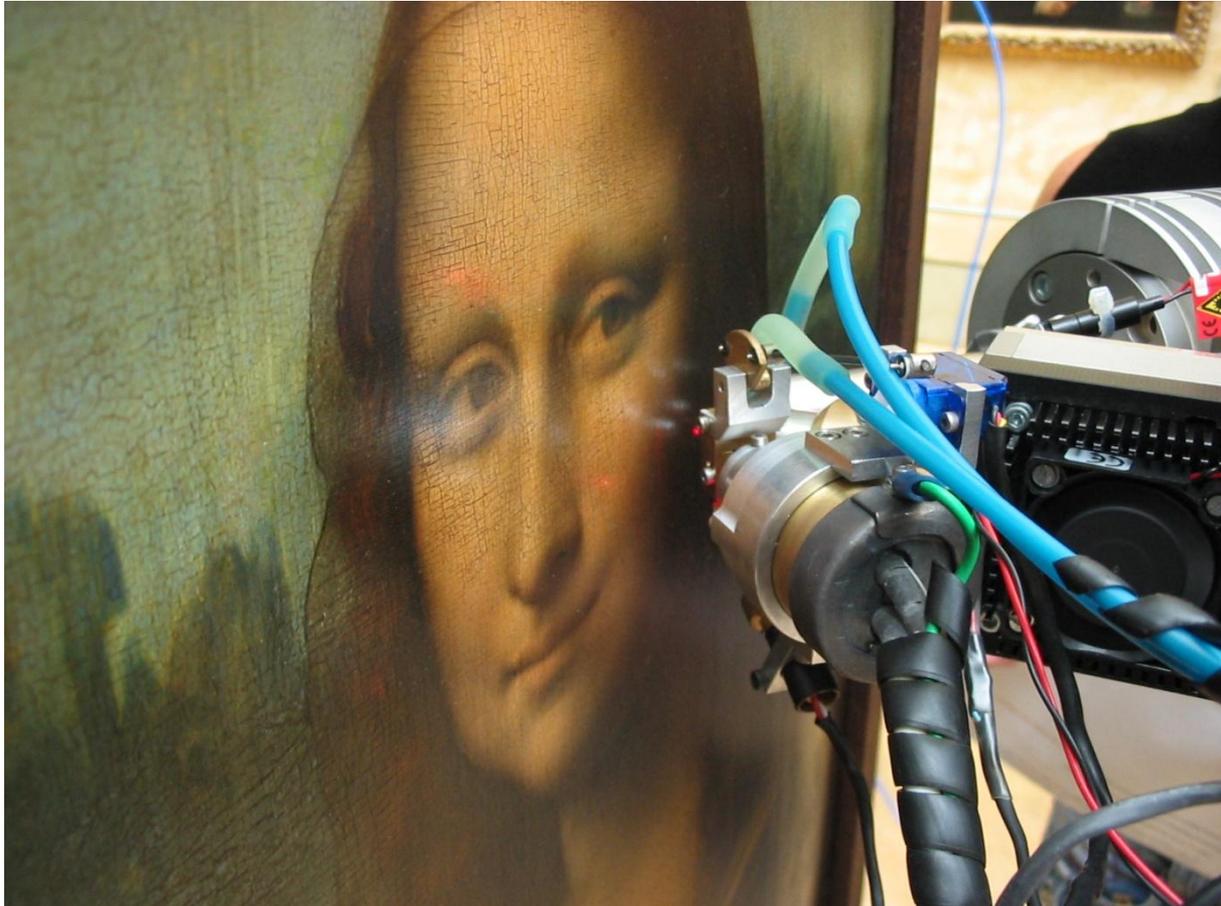


PyMca considers the vacancies produced in an atomic shell or sub-shell by the de-excitation process of an inner shell or sub-shell.

Radiative and Coster-Kronig transitions correctly considered.

Auger transitions approximated.

Cascade and M Shell combined with Multilayer Support



Revealing the sfumato Technique of Leonardo da Vinci by XRF spectroscopy

L. de Viguierie, Ph. Walter, E. Laval, B. Mottin, V.A. Solé. *Angew. Chem. Int. Ed.* 49 (2010) 6125-6128

Integration into our acquisition system

The screenshot displays the PyMCA software interface. The main window shows a spectrum plot with 'Counts' on the y-axis (log scale, 1 to 10000) and 'Energy' on the x-axis (0 to 10000). The plot shows several peaks, with the most prominent one around 4000 energy units. Below the plot, there is a 'Calibration' section with 'Internal (from Source OR PyMca)' selected and a 'Calibrate' button. The 'Active Curve Uses' section shows values: A: -153.851, B: 2.37935, C: 1.7388e-05.

On the left side, there is a table of array names and their dimensions:

Array Name	Rows	Cols
PUZ_arr_dsp6	243	128
PUZ_arr_dsp5	131	52
PUZ_arr_dsp4	131	52
PUZ_arr_dsp3	131	52
PUZ_arr_dsp2	131	52
PUZ_arr_dsp1	131	52
PUZ_arr_csamy	131	52
PUZ_arr_bchk	51	176
PUZ_arr_absorp3	131	52
PUZ_arr_absorp2	131	52
PUZ_arr_absorp1	131	52
MCA_DATA1000_PARAM	1	3
MCA_DATA0_PARAM	1	3
MCA_DATA0	4096	2

Below the table, there are 'Add', 'Remove', and 'Replace' buttons. At the bottom of the main window, there is a table with columns 'ROI', 'Type', 'From', and 'To':

ROI	Type	From	To
1	ICR	Default	-153.851 9881

A 'Shell - Konsole' window is open in the foreground, displaying a list of parameters and their values:

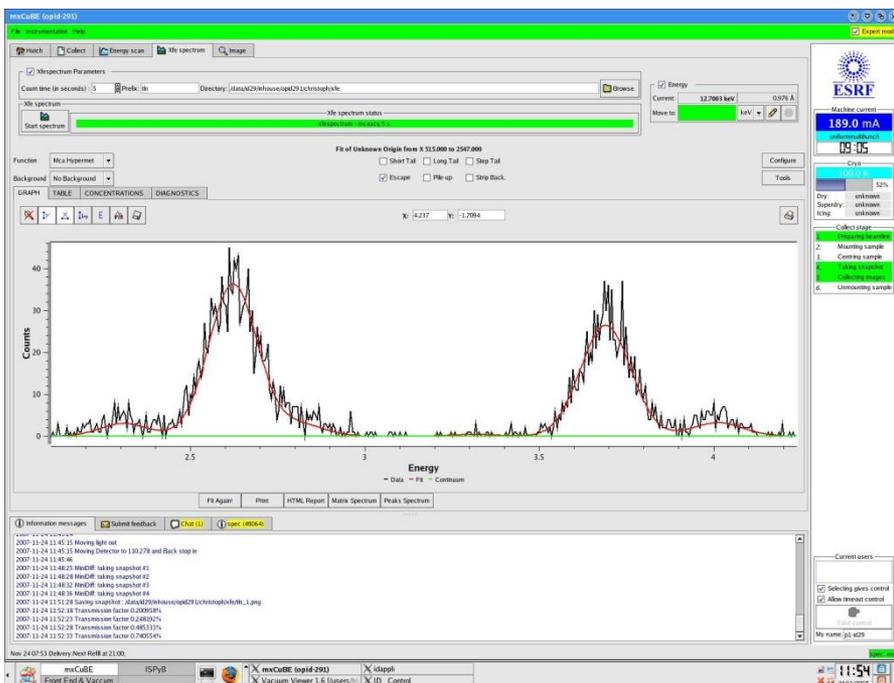
```

k1      vflp3      vflp4      vf2p1      vf2p2      vlmv      vlmz
0.0000  0.0000    0.0000    0.0000    0.0000    2.7850    4.1050  -0.2300
-86.0000 0.0000    0.0000    0.0000    0.0000    2.7850    4.1050    0.2300
pgty    pgtx      zoom      sddx      sddy      k487v1    xbpn      istopz
pgty    pgtx      zoom      sddx      sddy      k487v1    xbpn      istopz
-15.0000 0.3212 26999.0000 2.2000  -35.0000  0.0000  -42.5112  4.1200
-15.0000 -0.3212 49286.0000 2.2000  35.0000  0.0000  -42.5112  -21.3800
pgtz    sddz      fdetz      k487v2    U40u gap  U40u tape  zpz1      zpz2
pgtz    sddz      fdetz      k487v2    gadou     tu40u     zpz1      zpz2
4.8000  3.0000  -67.0000  0.0000  295.9900  0.0000  -0.4000  -0.4000
4.8000  3.0000  -67.0000  0.0000  295.9900  0.0000  -0.4000  -0.4000
zpz3    samrot    wsany     cryoz     granit1   granit2   granit3   granit4
-0.4000 30.0000  0.0000    0.0000  0.0050   0.0220   0.0070   0.0260
-zpz    zptheta   zpgamma
-zpz    zptheta   zpgamma
-0.4000 0.0000  0.0000
-0.4000 0.0000  0.0000
    
```

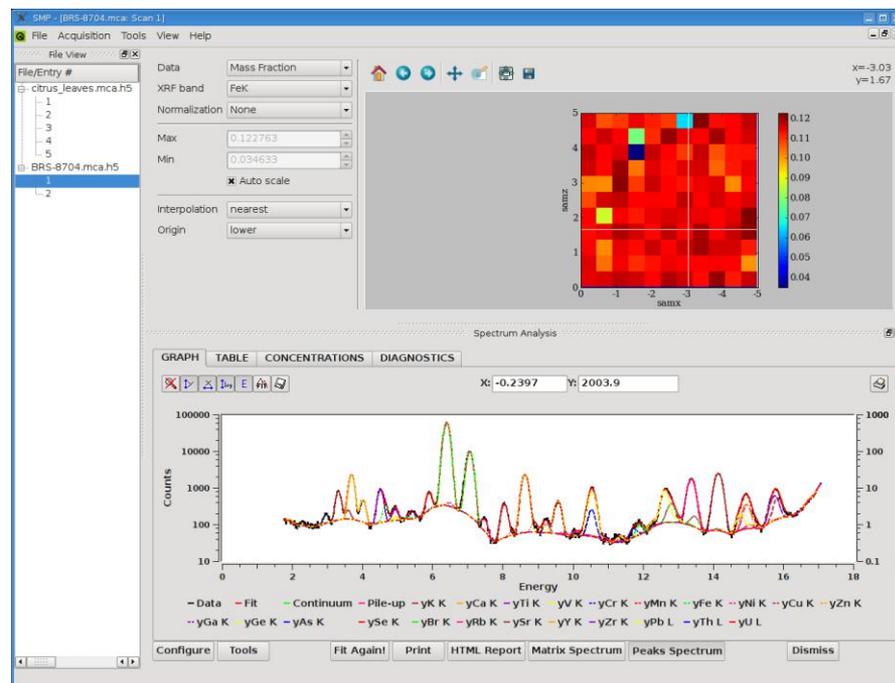
At the bottom of the shell window, there is a command prompt showing the execution of 'mcaacq 20' and the resulting elapsed times: 'MCA 0 Elapsed times: live 20.00 real 22.58 sec. dead time 11.43 %'.

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:

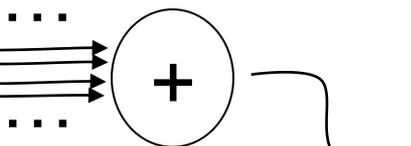
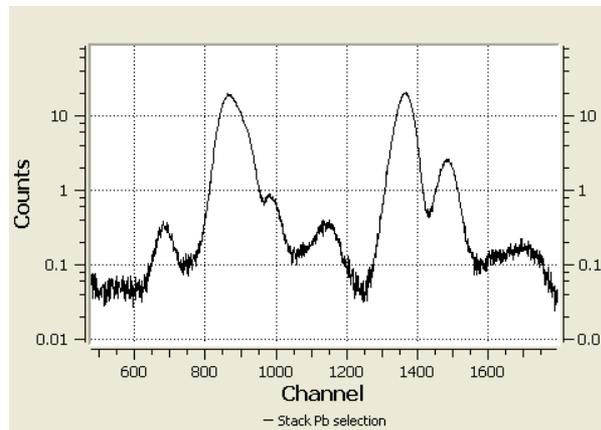
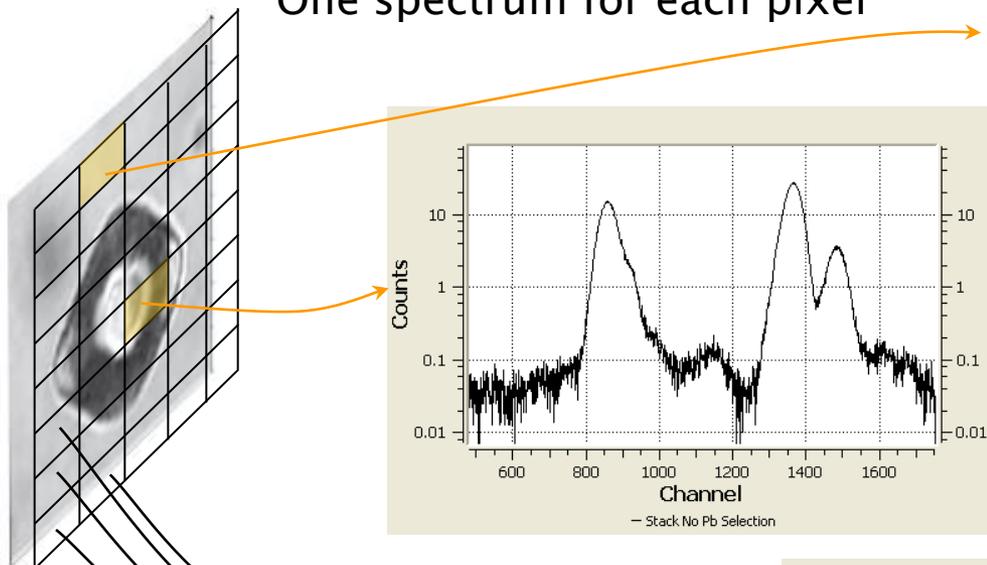
```
from PyMca import McaAdvancedFit

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

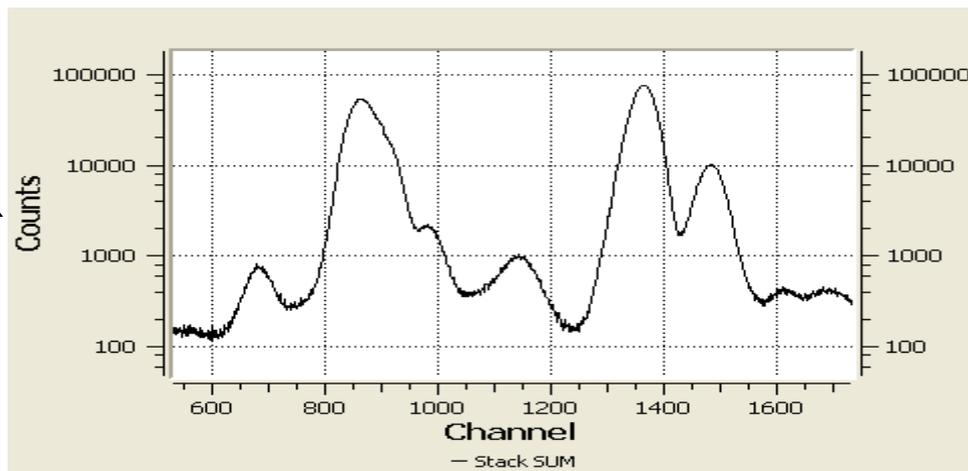
It can be used interactively from ipython just starting it as “ipython -q4thread”

Imaging

One spectrum for each pixel



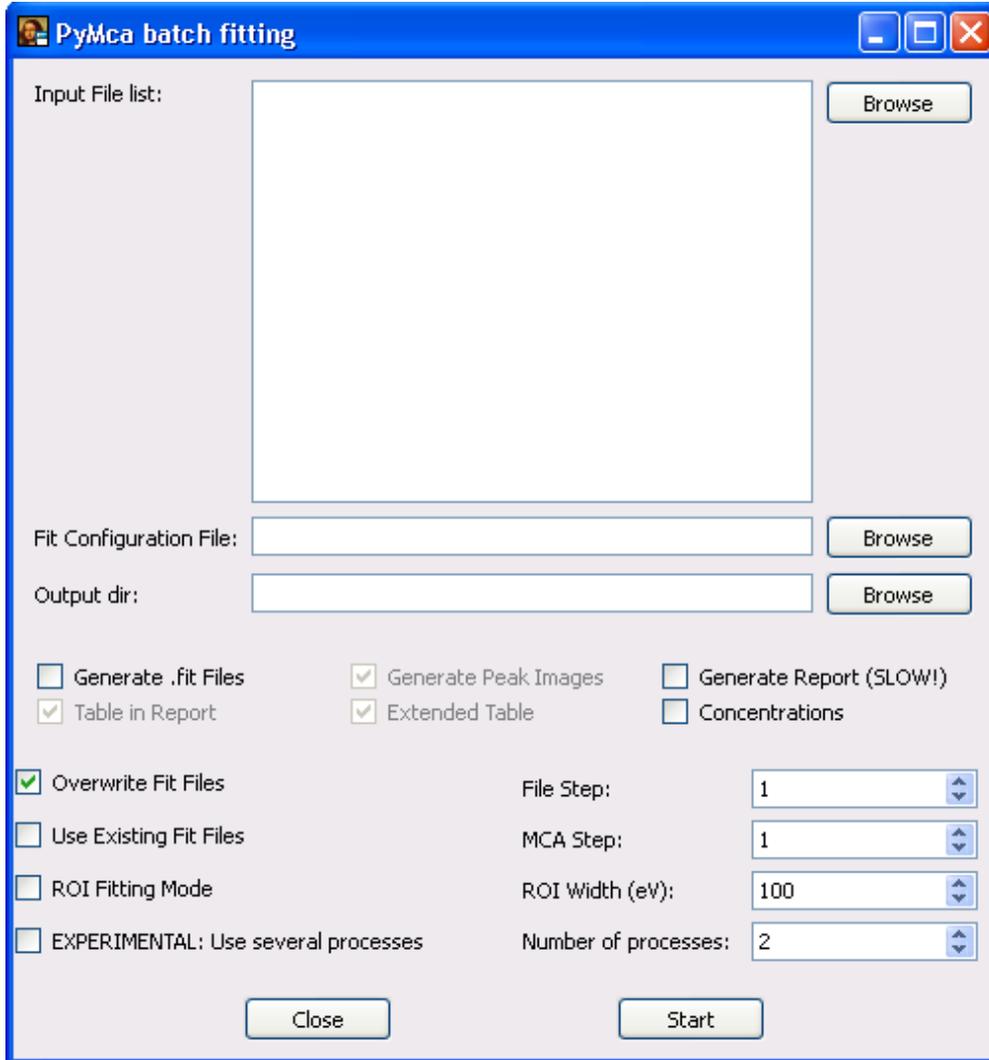
A sum spectrum for the whole image



Fit Imaging

1. We obtain the sum spectrum
2. We analyze the sum spectrum
3. **We optimize the fit configuration** for batch processing
4. **We test the configuration** on several pixels
5. **We save the configuration**
6. We start the batch
7. We interpret the results

Batch processing



PyMca batch fitting

Input File list:

Fit Configuration File:

Output dir:

Generate .fit Files
 Generate Peak Images
 Generate Report (SLOW!)
 Table in Report
 Extended Table
 Concentrations

Overwrite Fit Files
 File Step:

Use Existing Fit Files
 MCA Step:

ROI Fitting Mode
 ROI Width (eV):

EXPERIMENTAL: Use several processes
 Number of processes:

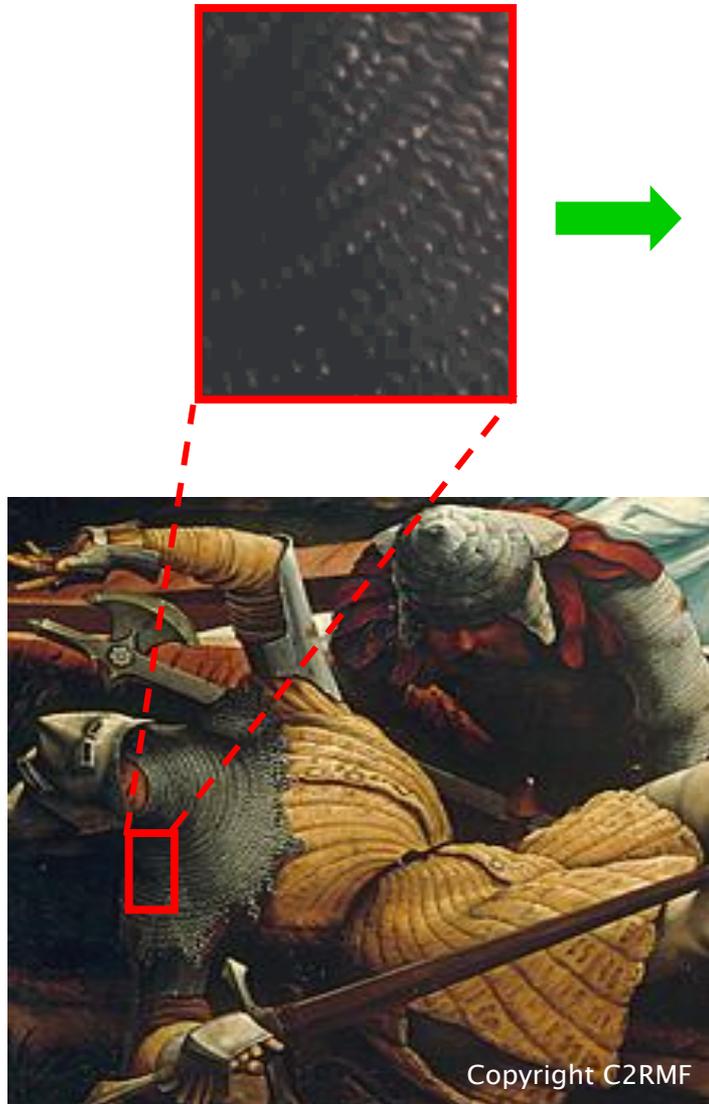
Select the input files

Select the fit configuration

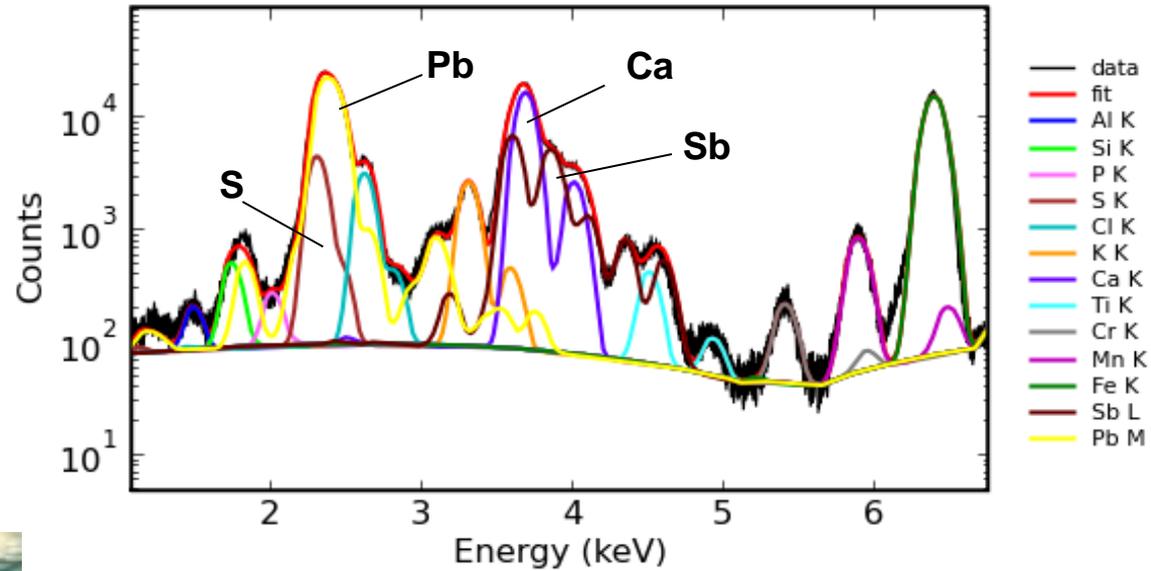
Select the output directory

Select the output options

Start

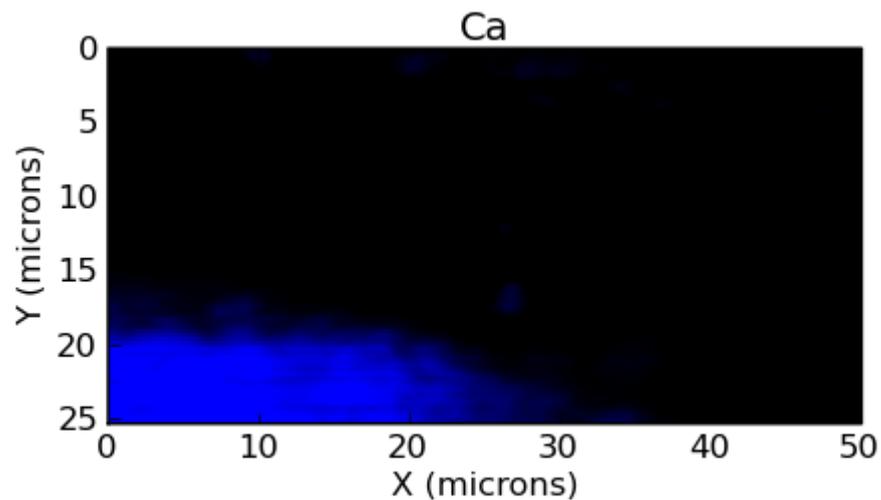
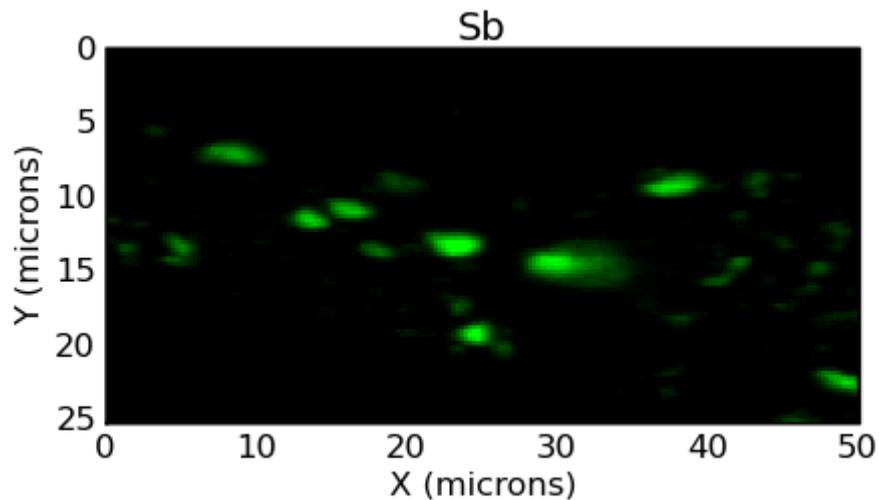
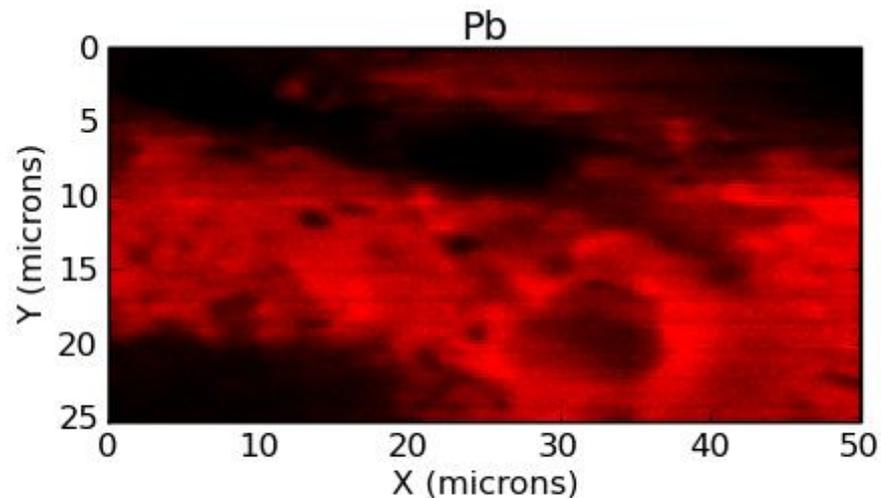
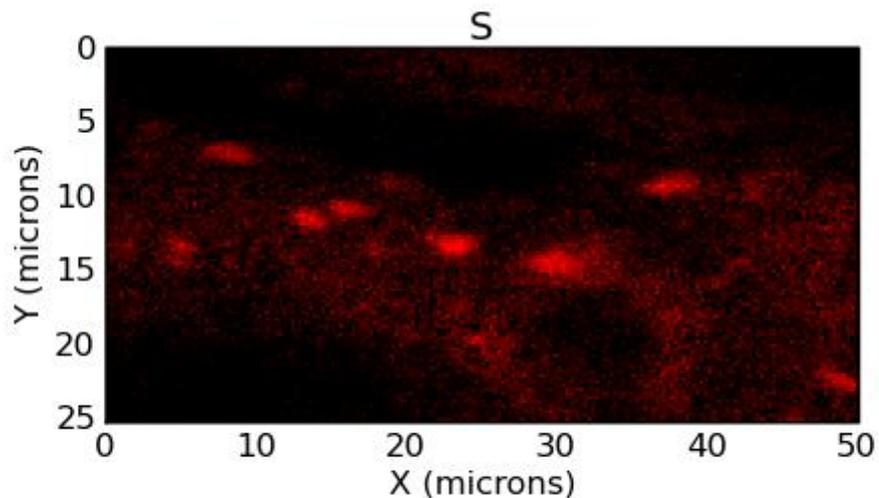


Copyright C2RMF

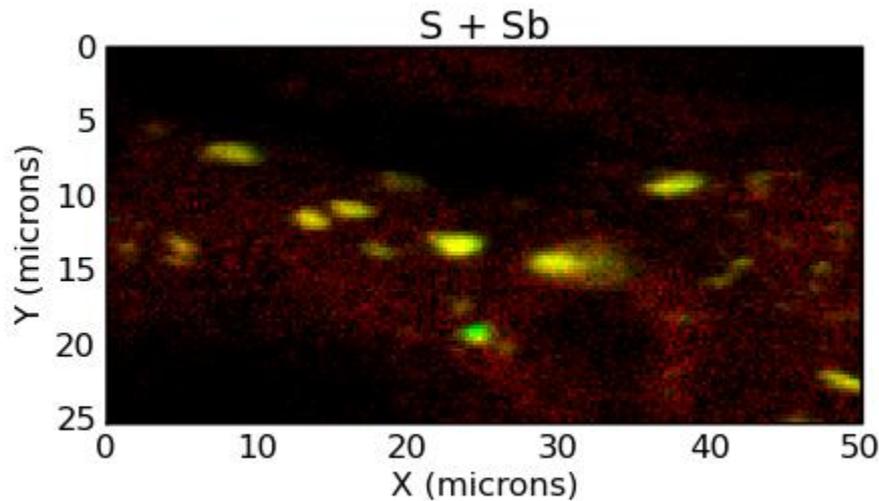


We wanted to know what pigments were used in this section of the painting

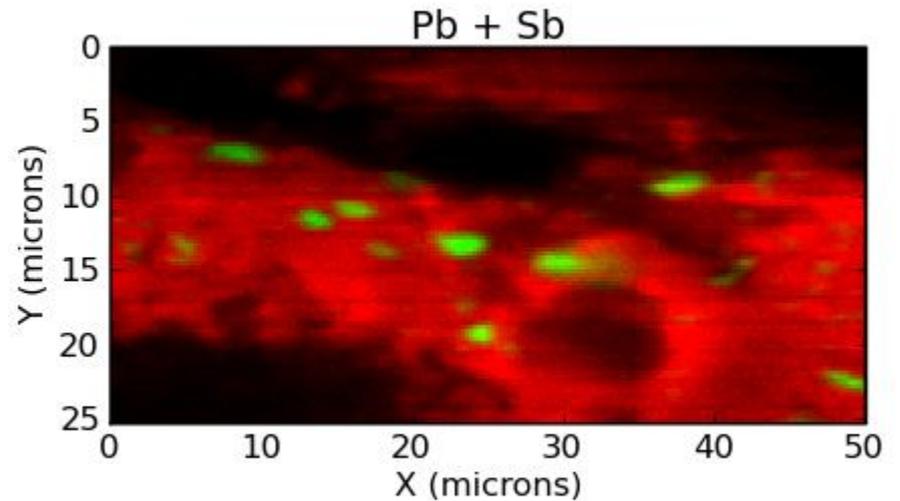
Based on the batch generated element distribution maps ...



... 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.

PyMca Non-XRF Specific Imaging Capabilities

Region of Interest imaging

Very basic multivariate analysis

- Principal component analysis
- Non-negative Matrix Approximation

Image correlation tools

Non-negative Matrix Approximation (NNMA)

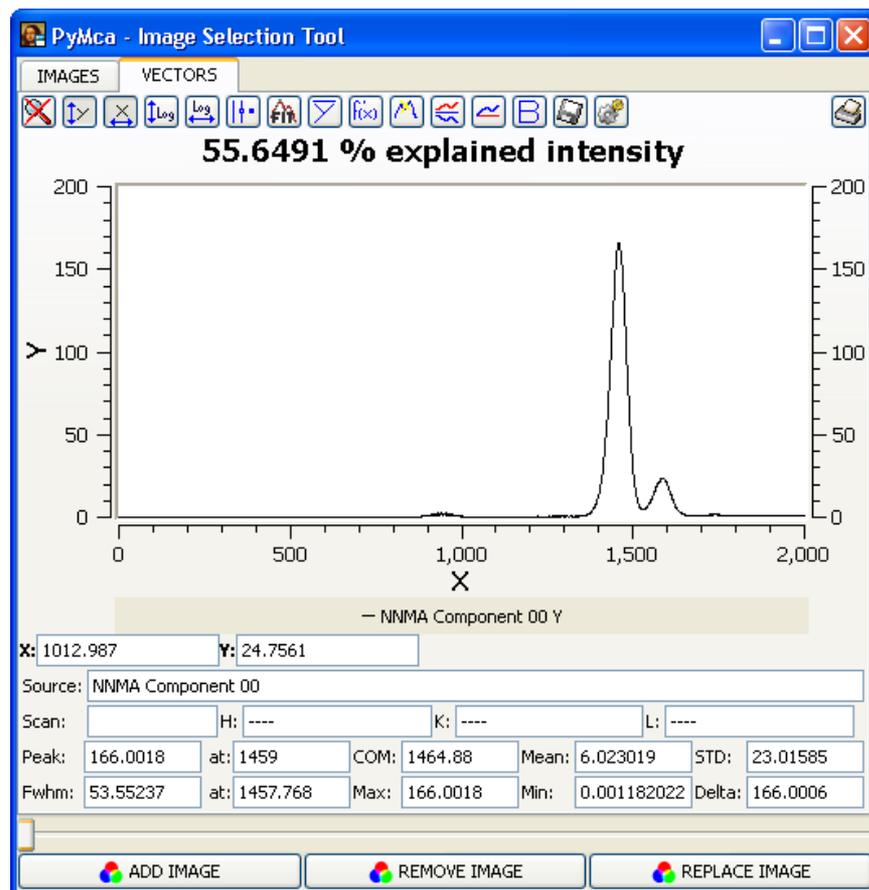
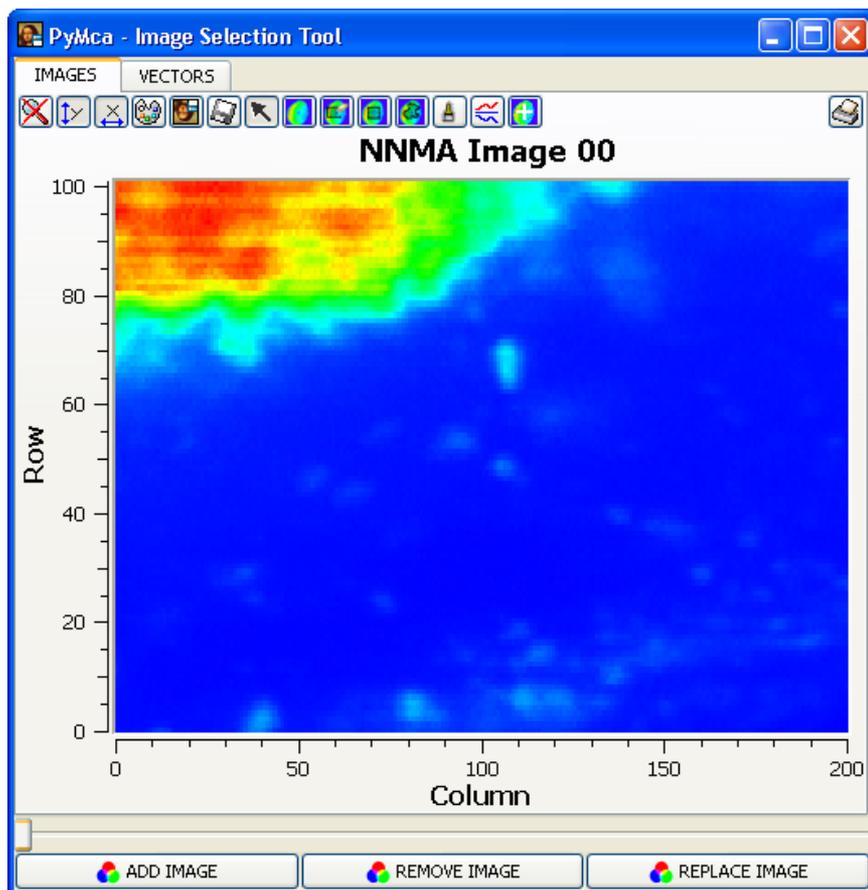
Observed data $Y = A X$ where all terms in A and X matrices are positive or null

Implemented using Uwe Schmitt's `py_nnma` module

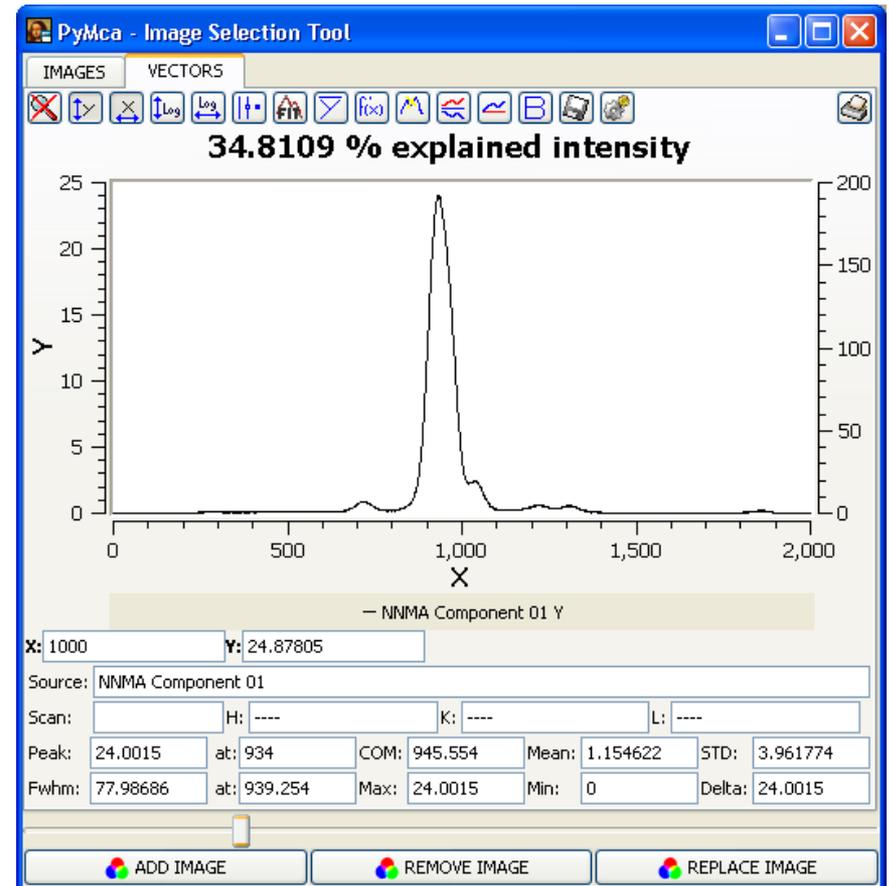
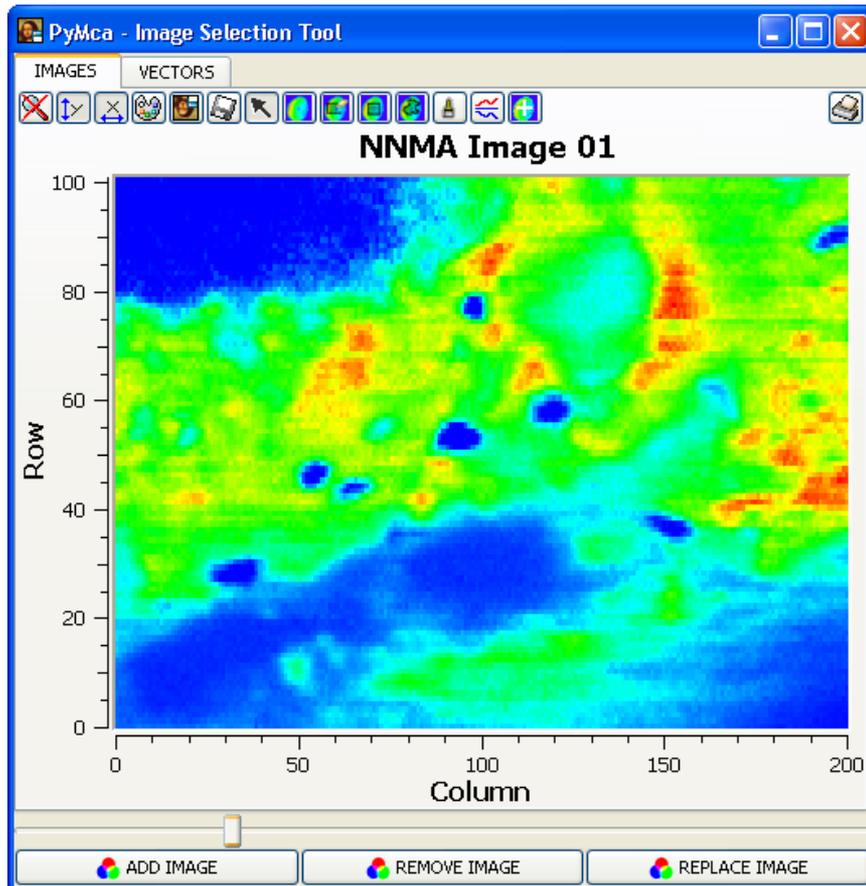
Collaboration with Gerd Wellenreuther (DESY)

PyMca allows to use NNMA on raw spectra and in treated data

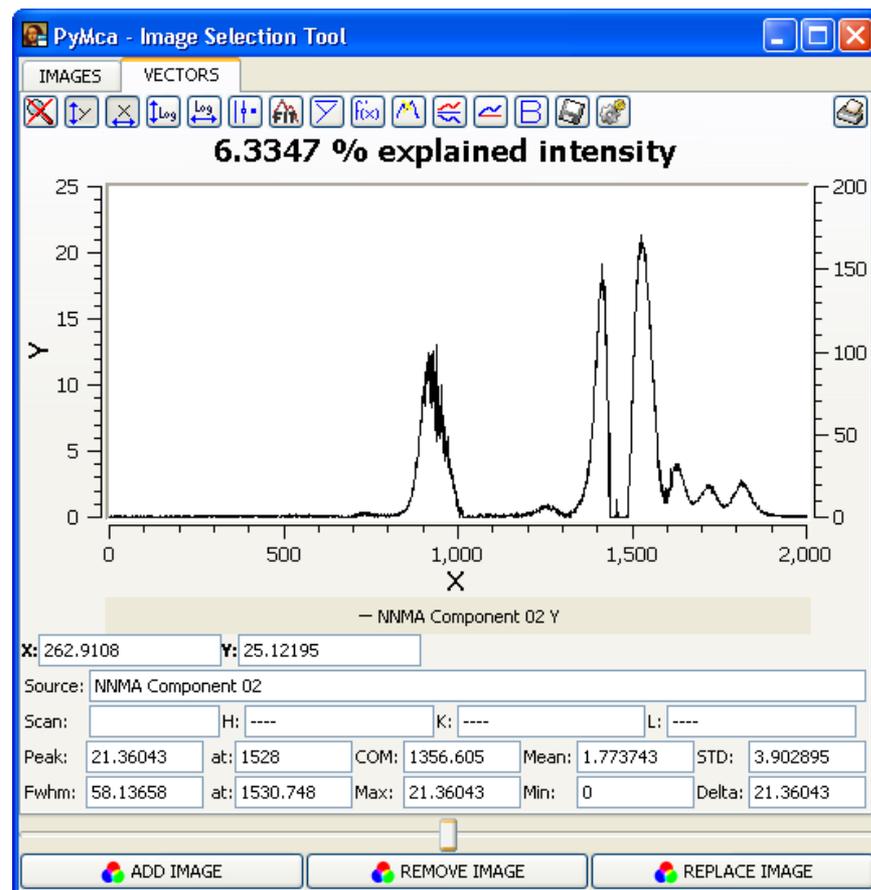
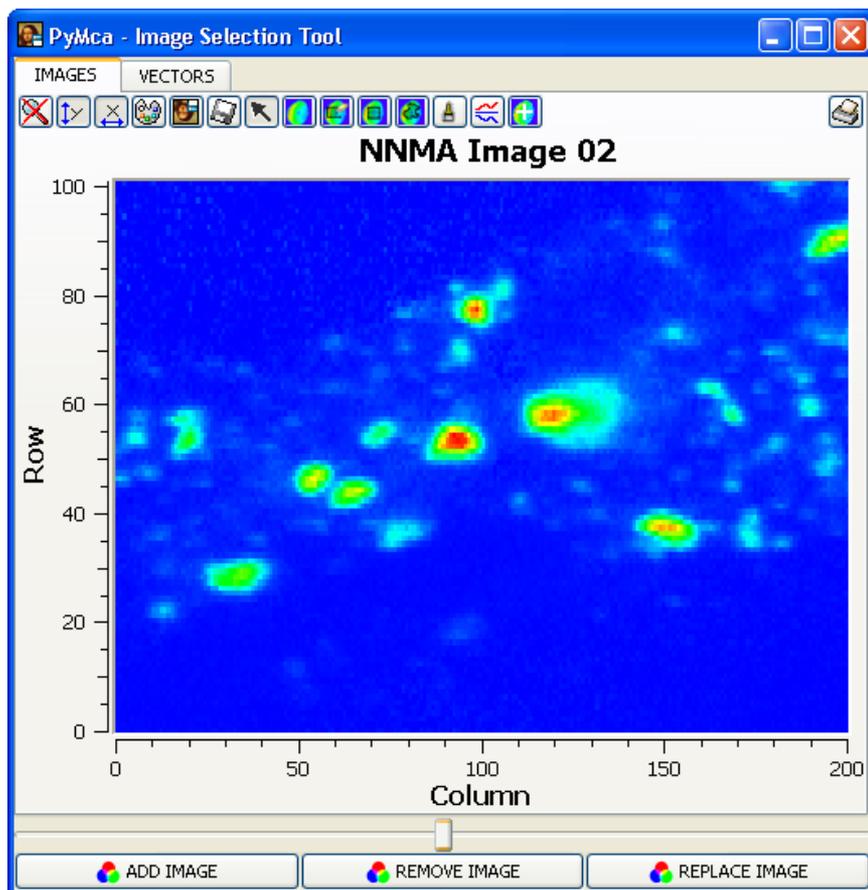
NNMA images and NNMA vectors



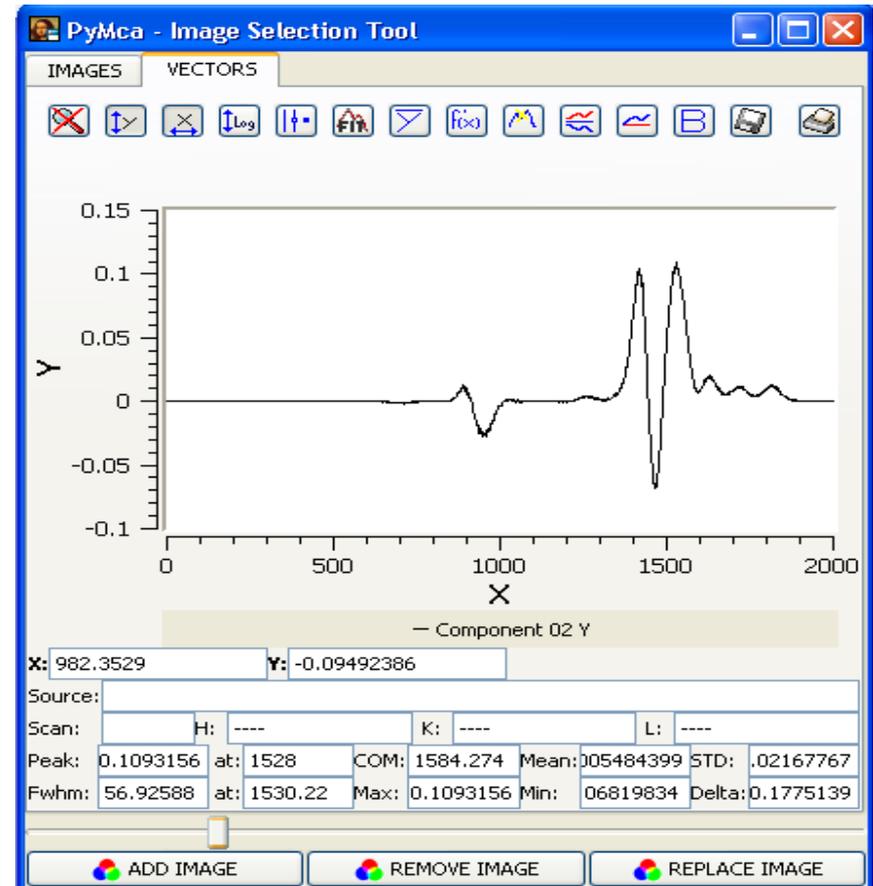
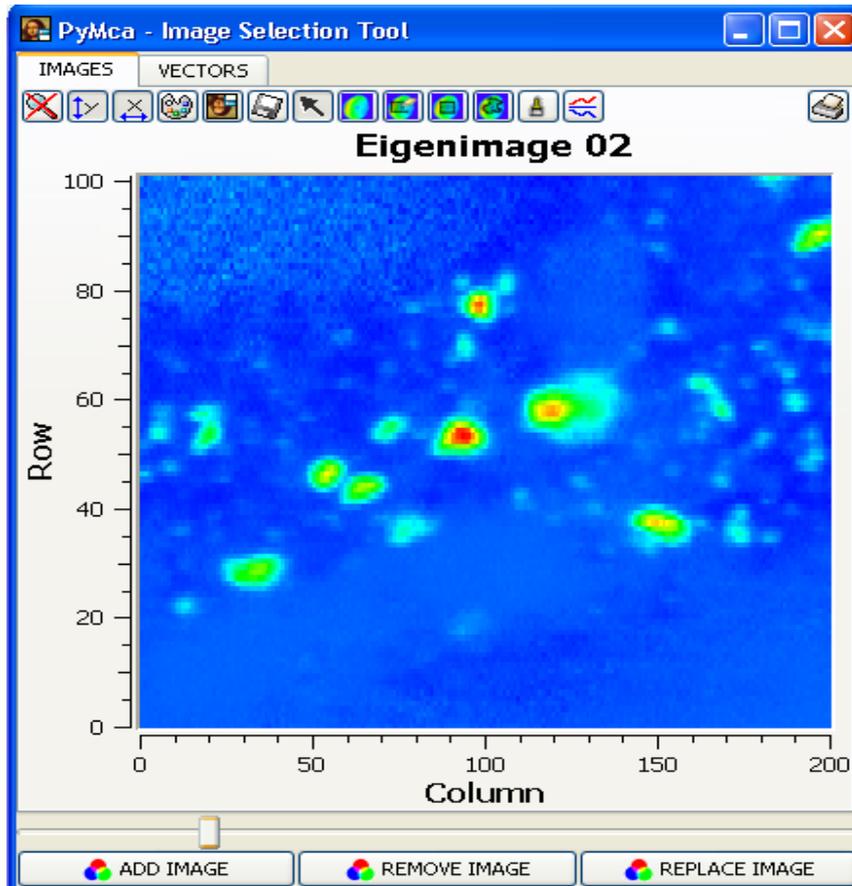
NNMA images and NNMA vectors

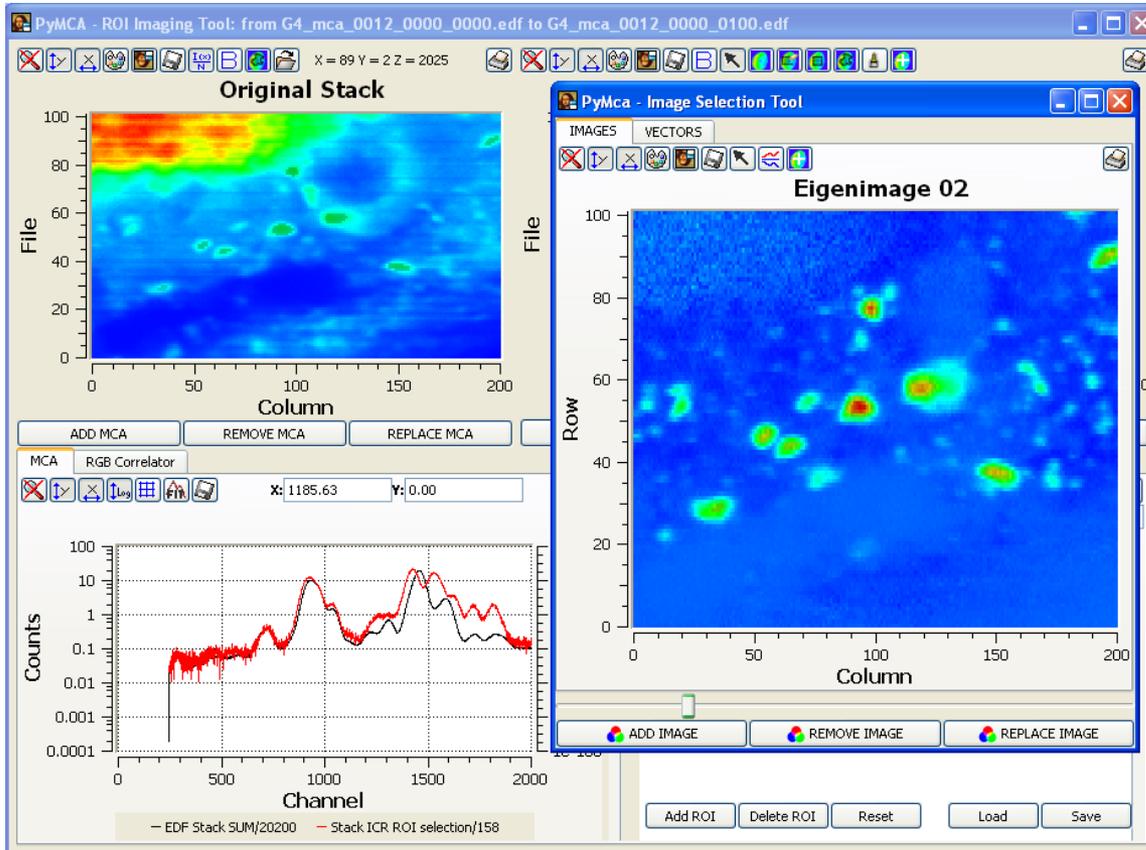


NNMA images and NNMAvectors



Eigenimages and Eigenvectors





We can select a set of pixels on any of the obtained 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).

What have we done?

We have used multivariate analysis to know what sample regions were worth to take a closer look.

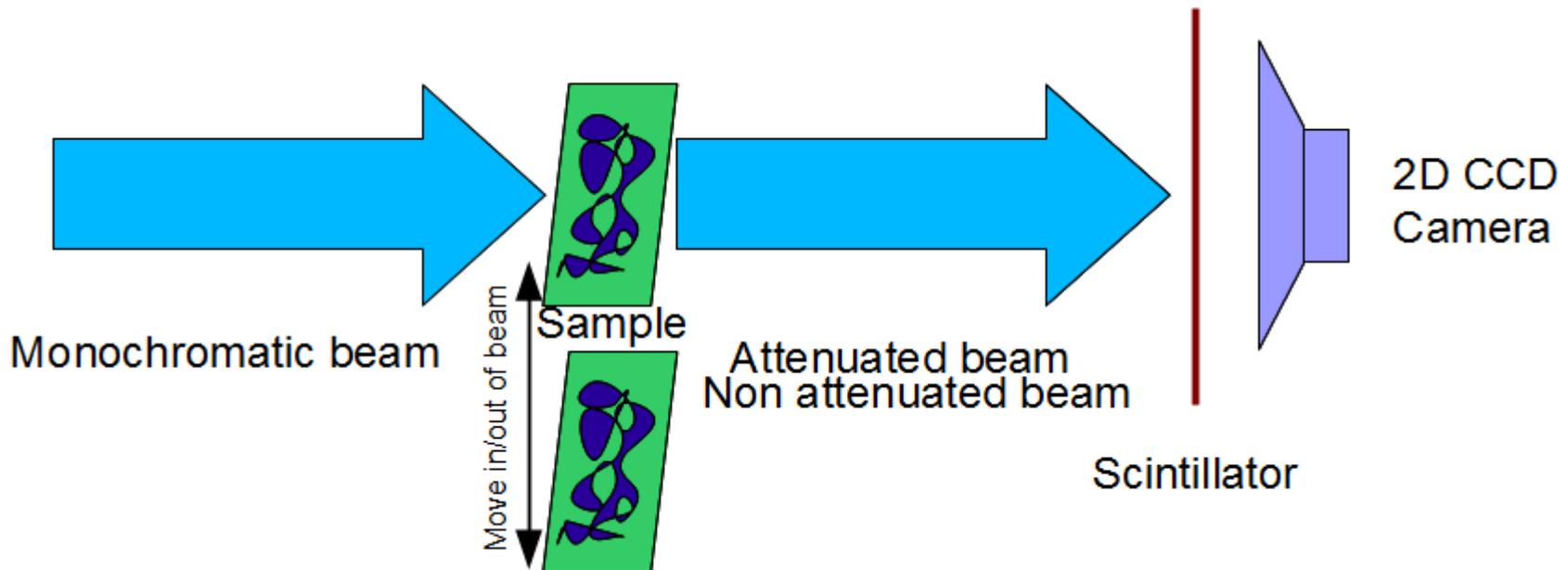
Not bad when you have lots of data ...

This data treatment is totally generic and applicable to other methods of analysis

On going activities: ID21 Full Field XANES

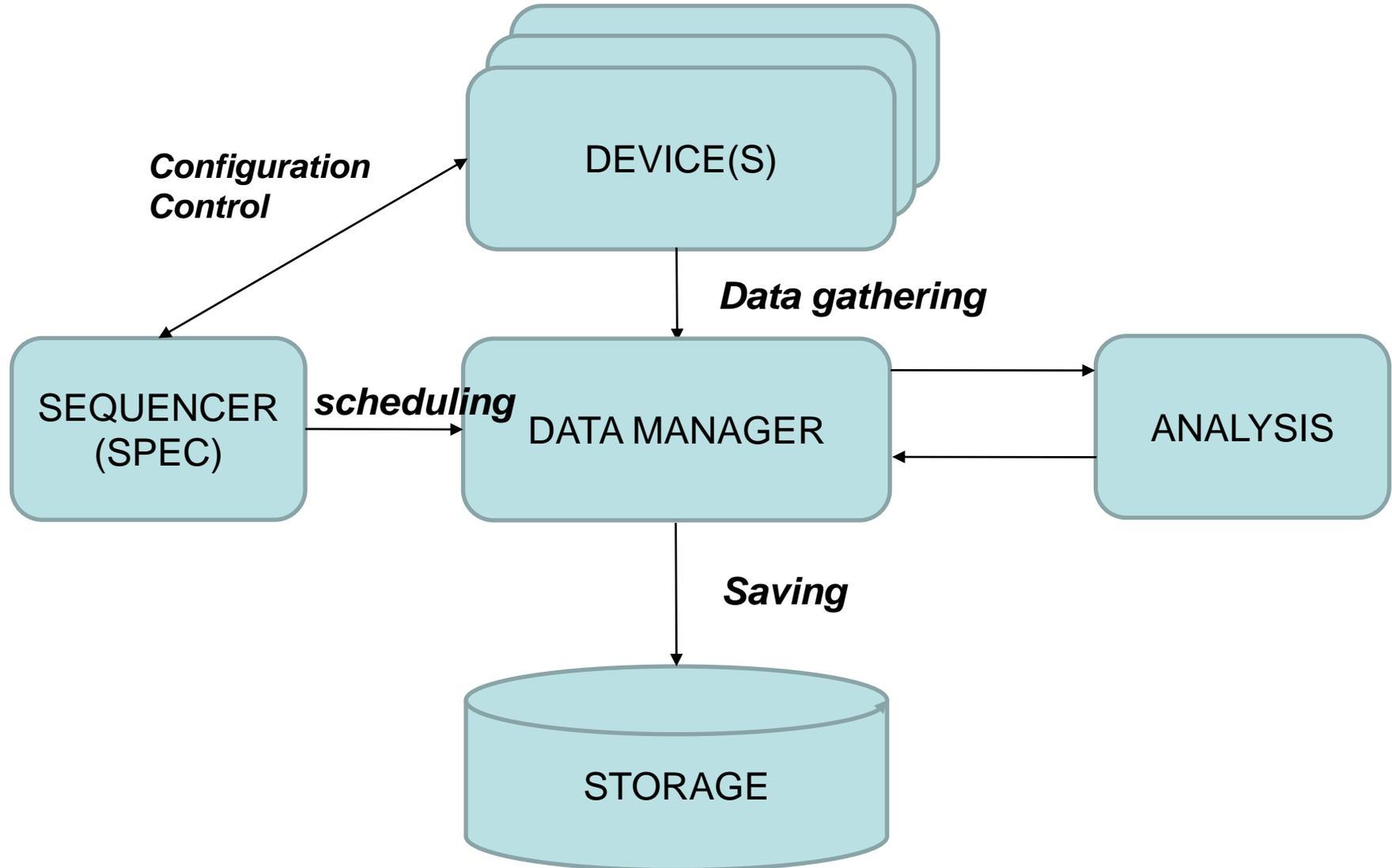
Scan in energy around an absorption edge

Spot size 1 mm
Resolution 500 nm

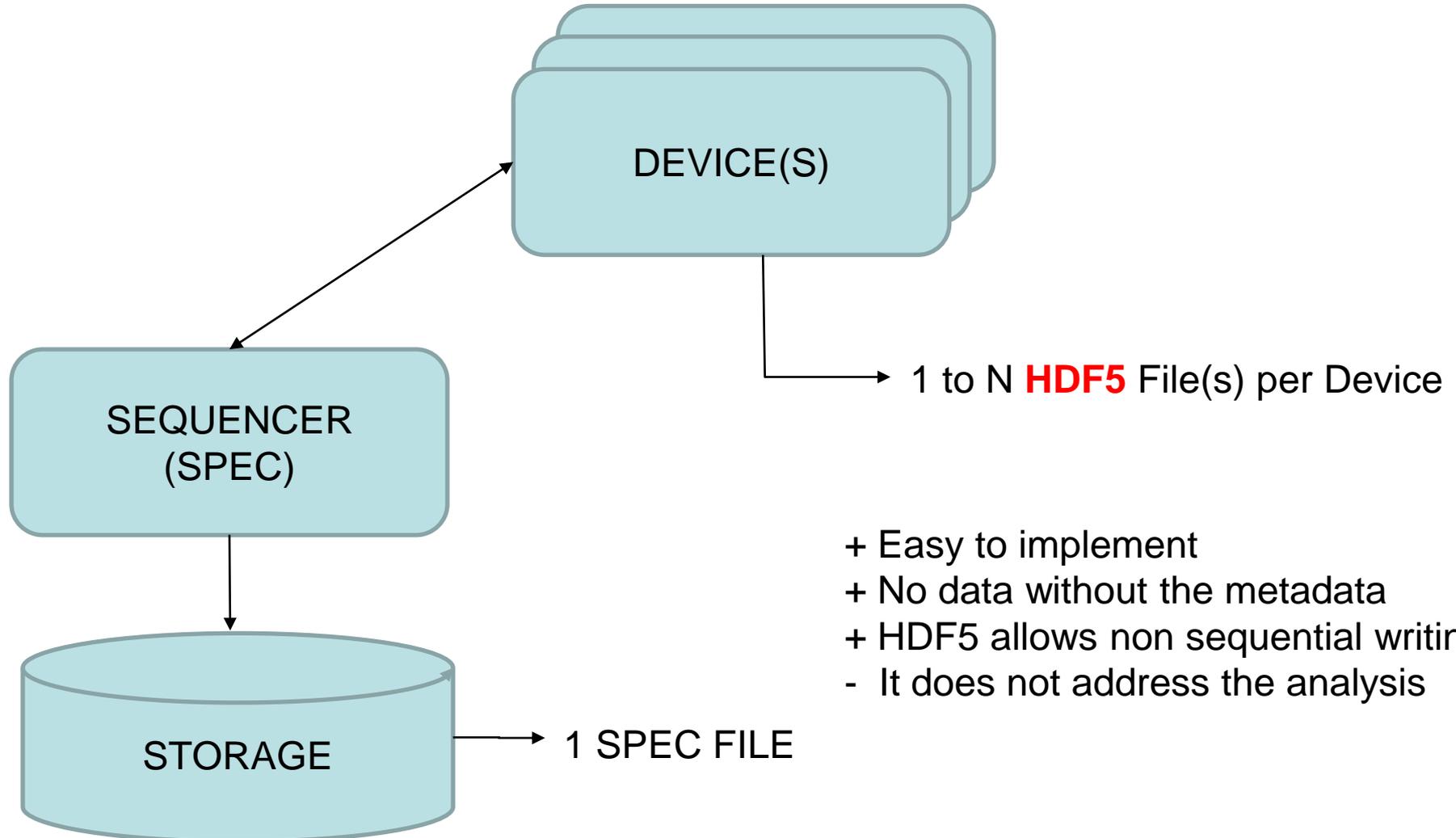


Align sample to correct submicron sample position changes

Acquisition Approaches: Long term target

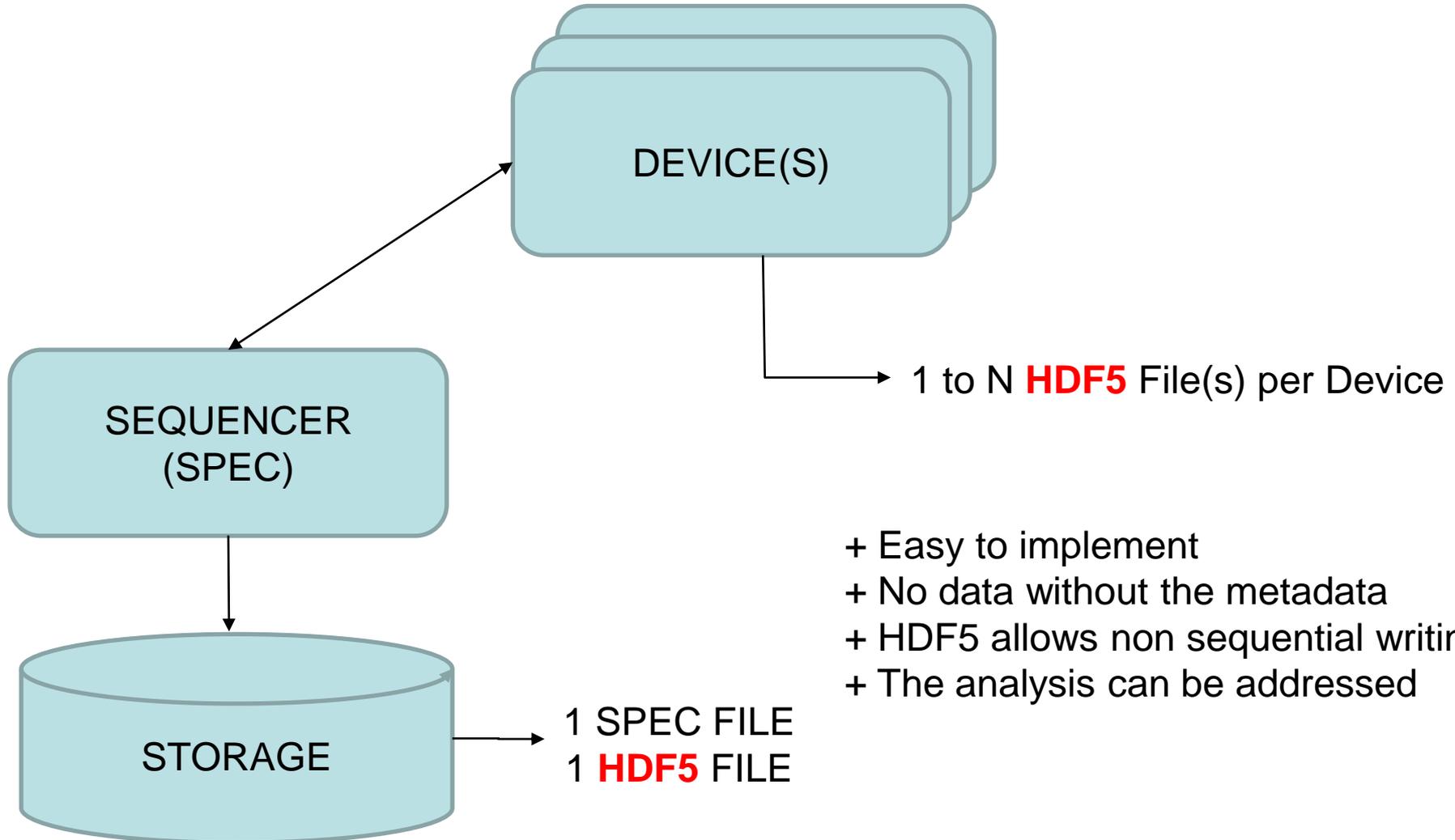


On going: The broadcast approach



- + Easy to implement
- + No data without the metadata
- + HDF5 allows non sequential writing
- It does not address the analysis

On going: The storage based approach



- + Easy to implement
- + No data without the metadata
- + HDF5 allows non sequential writing
- + The analysis can be addressed

A critical component: HDF5

A “sort of” portable file system in a file

It can contain folders (HDF5 Groups)

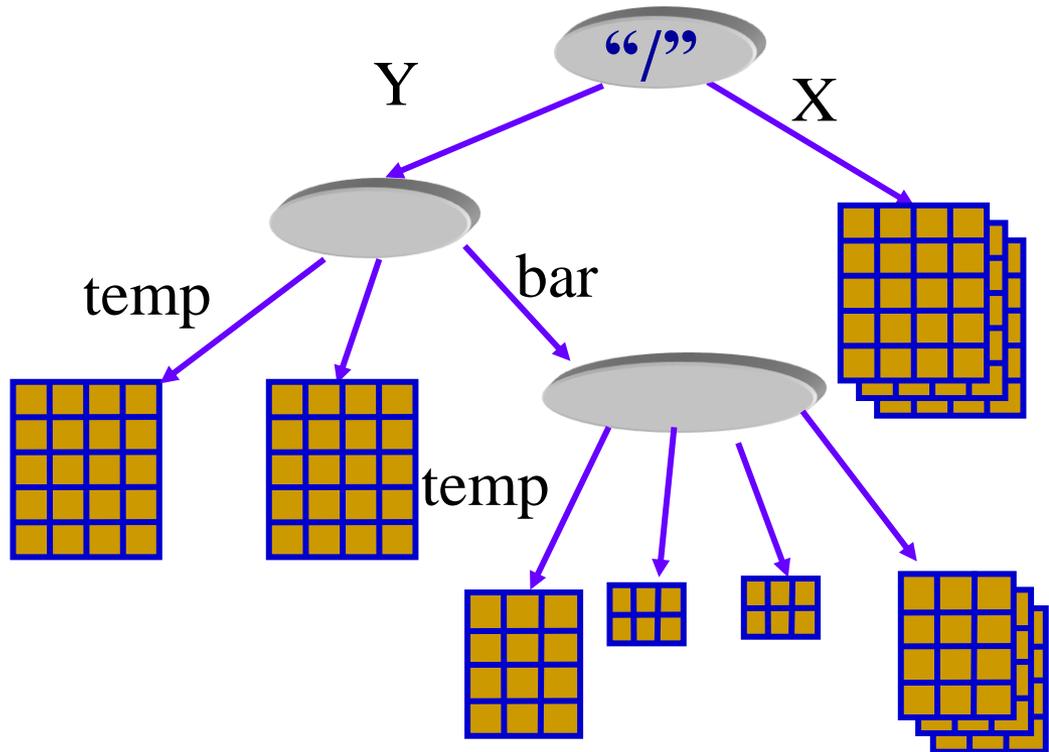
It can contain data files (HDF5 Datasets)

It can contain links (internal or external)

It supports compression

Path to HDF5 object in a file

/ (root)
 /X
 /Y
 /Y/temp
 /Y/bar/temp



Metadata

Dataspace

Rank	Dimensions
3	Dim_1 = 4 Dim_2 = 5 Dim_3 = 7

Datatype

IEEE 32-bit float

Storage info

Chunked

Compressed

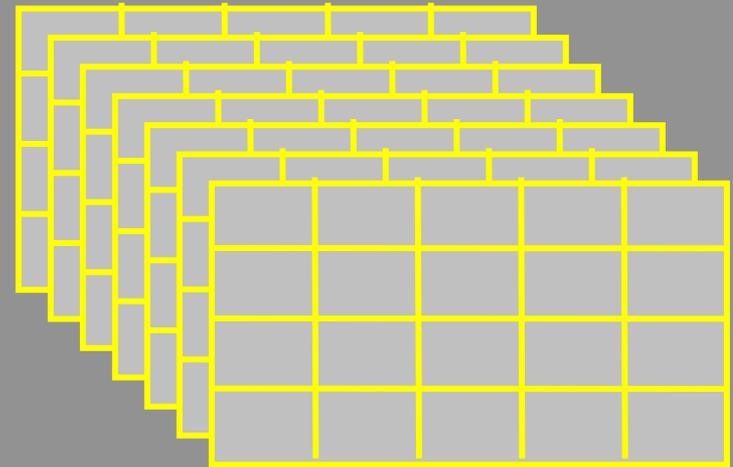
Attributes

Time = 32.4

Pressure = 987

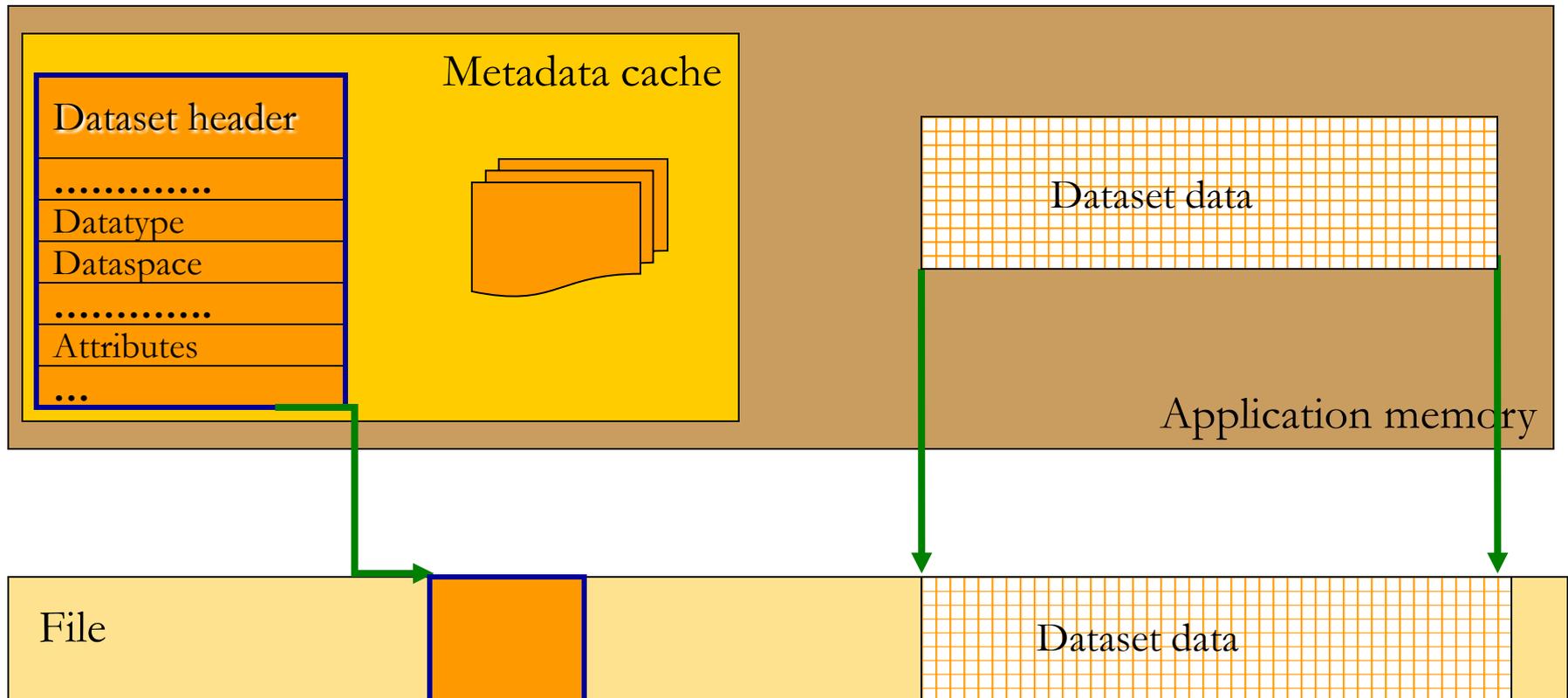
Temp = 56

Data



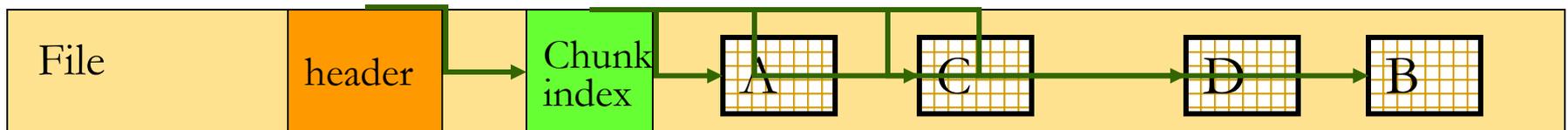
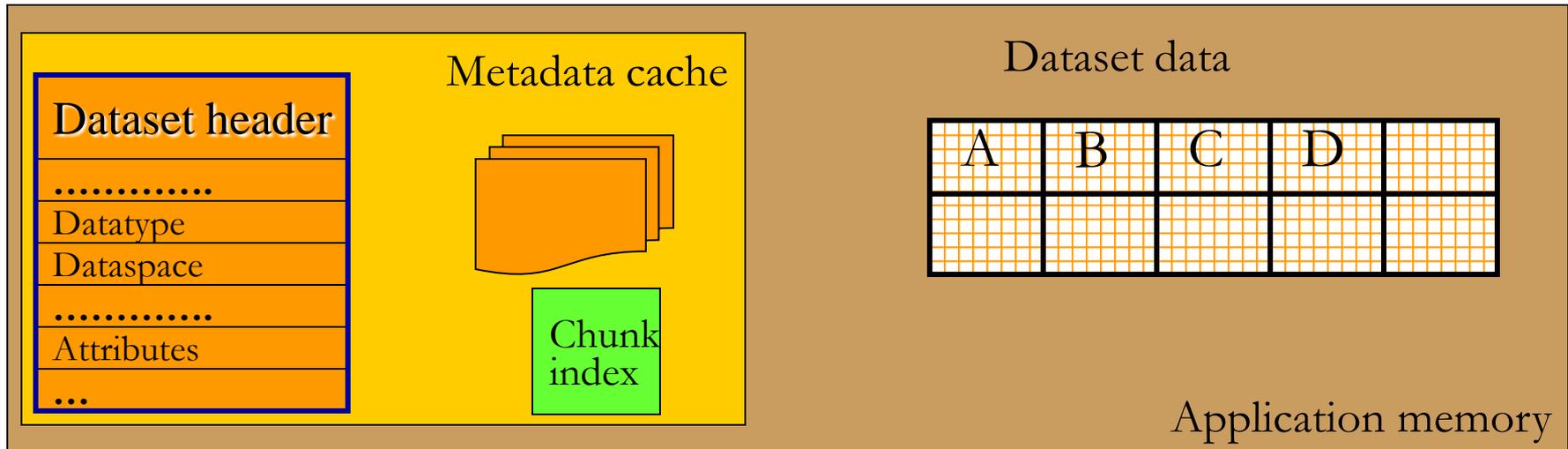
HDF5 Contiguous Storage Layout

- Metadata header separate from dataset data
- Data stored in one contiguous block in HDF5 file



HDF5 Chunked Storage Layout

- Dataset data divided into equal sized blocks (chunks)
- Each chunk stored separately as a contiguous block in HDF5 file
- HDF5 always writes/reads the whole chunk



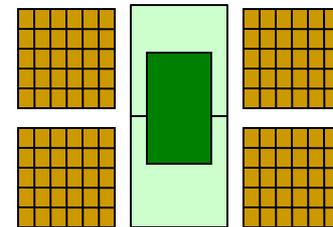
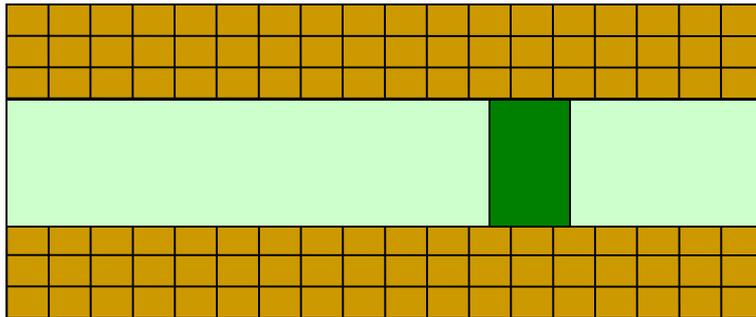
Why HDF5 Chunking?

- Chunking is required for several HDF5 features
 - Enabling compression and other filters like checksum
 - **Extendible datasets**



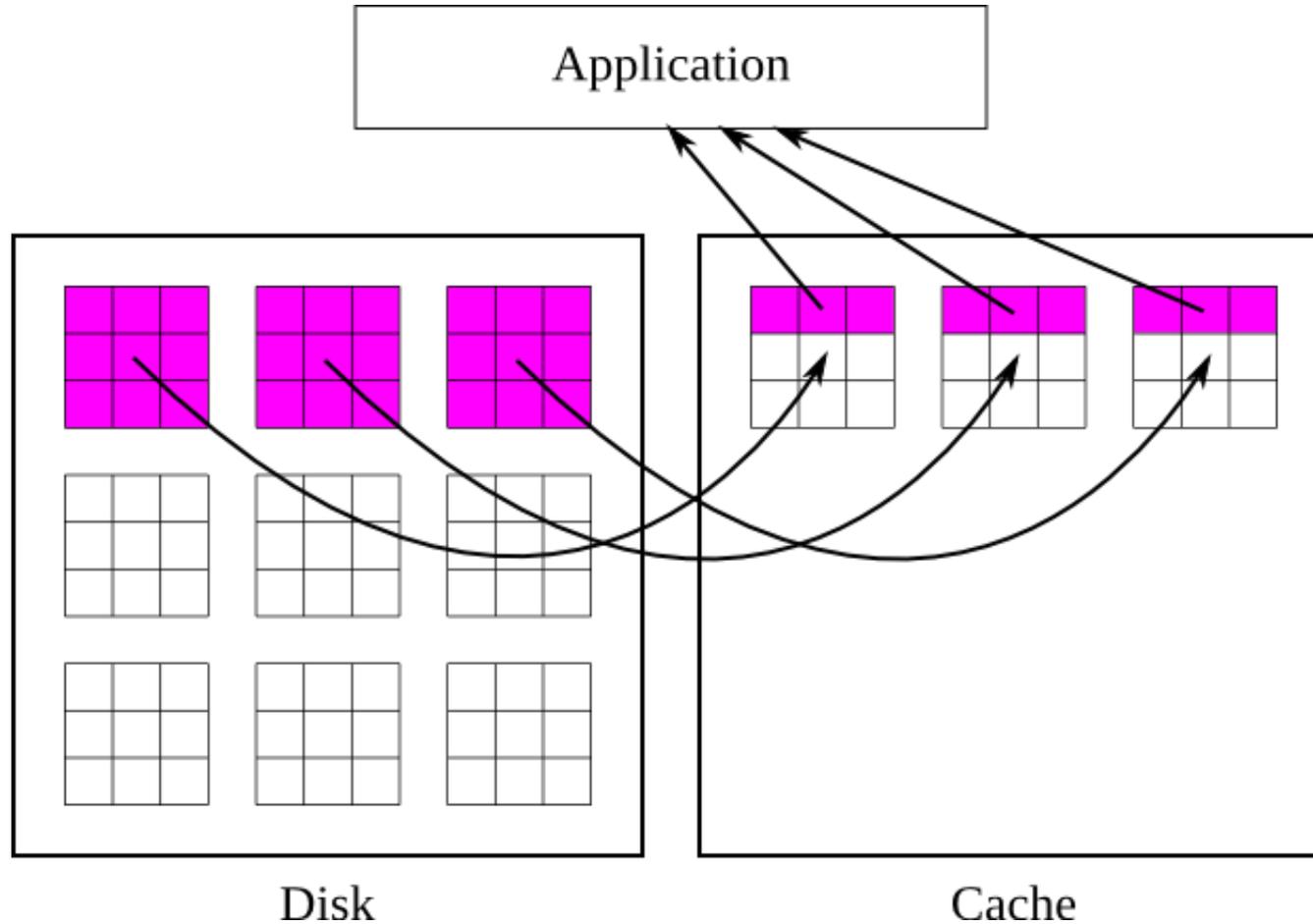
Why HDF5 Chunking?

- If used appropriately chunking improves partial I/O for big datasets



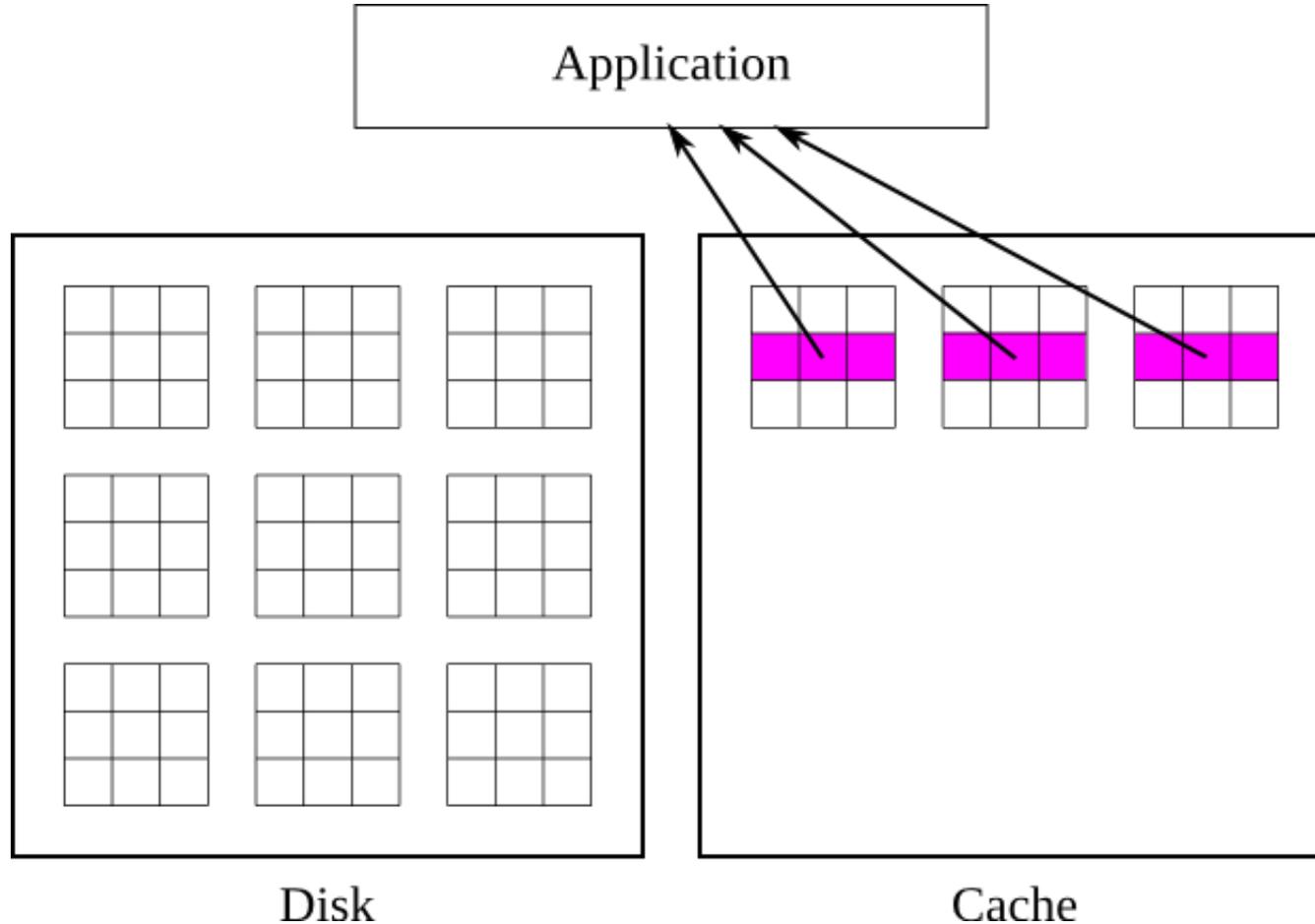
Only two chunks are involved in I/O

Using the chunk cache (I)



Reading from disk with chunk cache enabled

Using the chunk cache (II)



Next readout is taken directly from the cache.

This only works if the chunk cache size was large enough! (default 1MByte)

WARNING

- If NOT used appropriately (for instance not combined with the appropriate chunk cache size) can have very negative effects

MUST READ

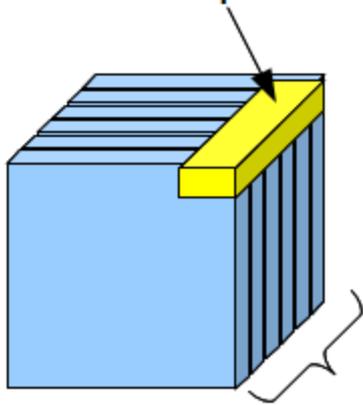
<http://www.hdfgroup.org/HDF5/doc/Advanced/Chunking/>

www.hdfgroup.org/HDF5/doc/Advanced/Chunking/Chunking_Tutorial_EOS13_2009.pdf

How does this affected the ID21 case?

- Dataset extended with images of 4 Mbytes (n_points, 2048, 2048)
- We were using a chunk size of (1, 2048, 2048)
- The dataset had a size of 5 GBytes
- We had to use dynamic reading (32 bit systems ...)

XANES Spectra



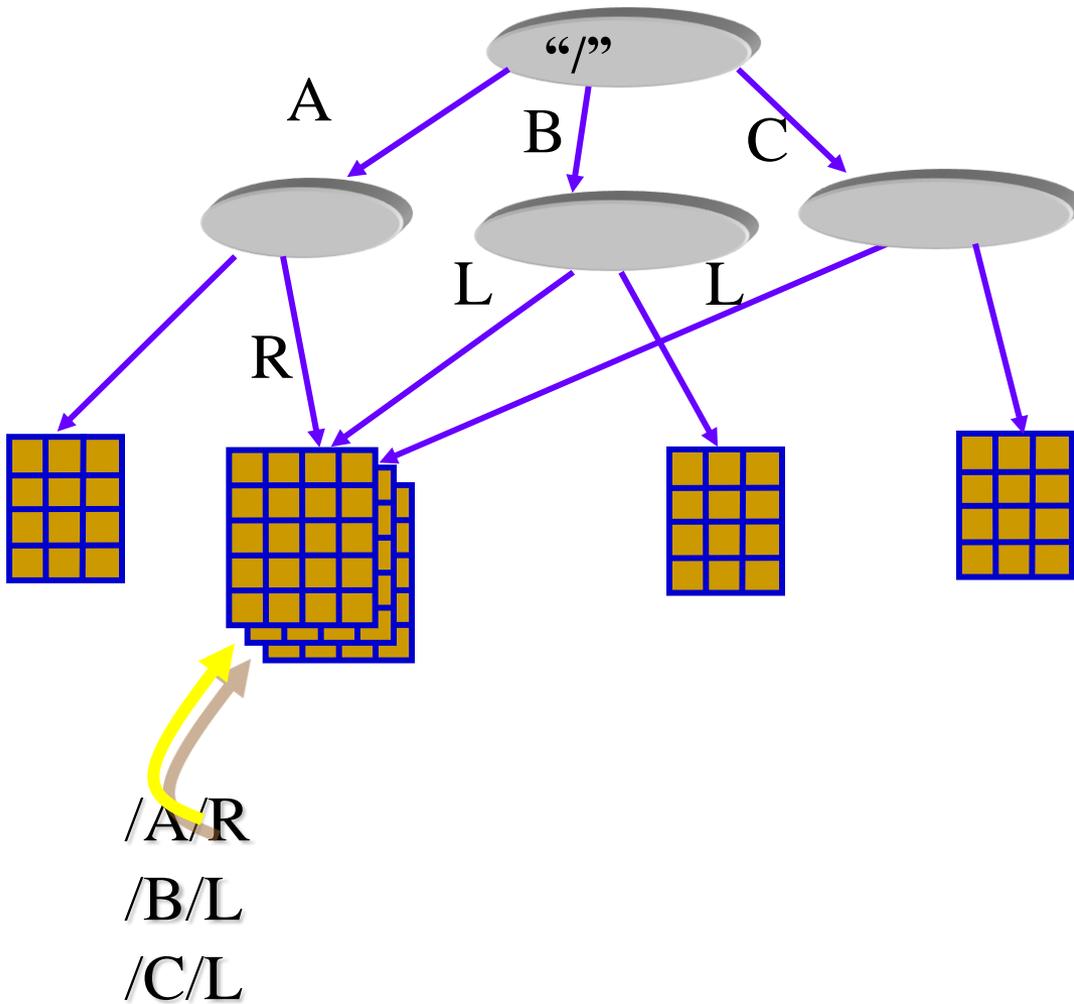
Stack of images

To get the spectrum associated to one pixel of our map we had to read 5 Gbytes

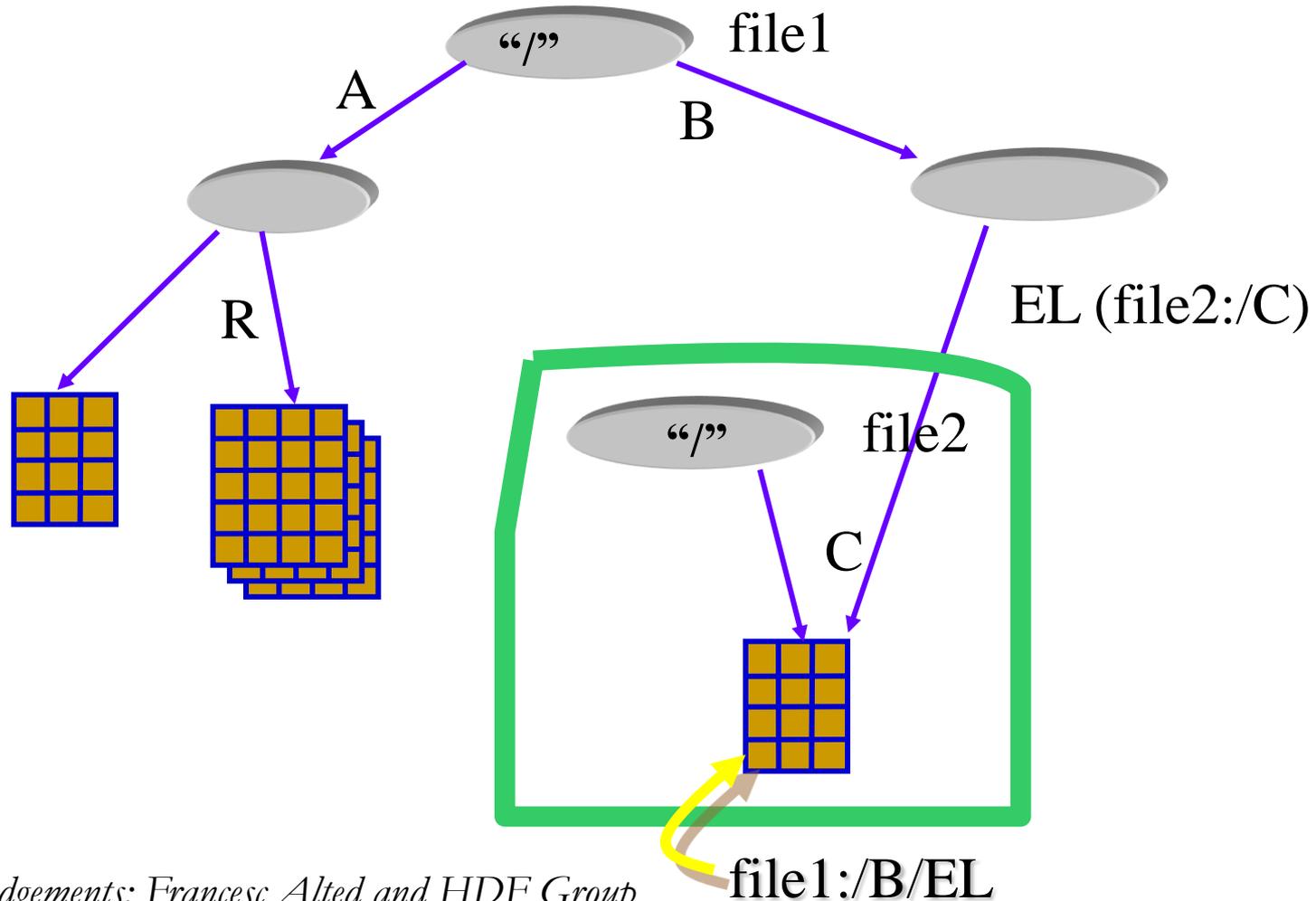
Writing performance can be affected too!

- One can reduce disk access and try to improve compression by using a chunk size of several images (10, 2048, 2048). After all, we expect to read several images to get a spectrum and does not sound a bad idea.
- If the chunk cache size is not modified accordingly to fit at least one complete chunk, HDF5 will write the first image as soon as it arrives and will have to read back the written image to be able to add the second and to compress again ...

HDF5 Links



HDF5 External Links



Acknowledgements: Francesc Alted and HDF Group

How do the links affect us?

- Any set of HDF5 files related to a particular measurement can be made “analysis friendly” at any time. For instance
 - By the sequencer at acquisition time
 - By an analysis application offline

Data Analysis: EDNA

Open source (GPL, LGPL) framework

- Originally developed to automate protein crystallography analysis pipeline
- Web site <http://www.edna-site.org>

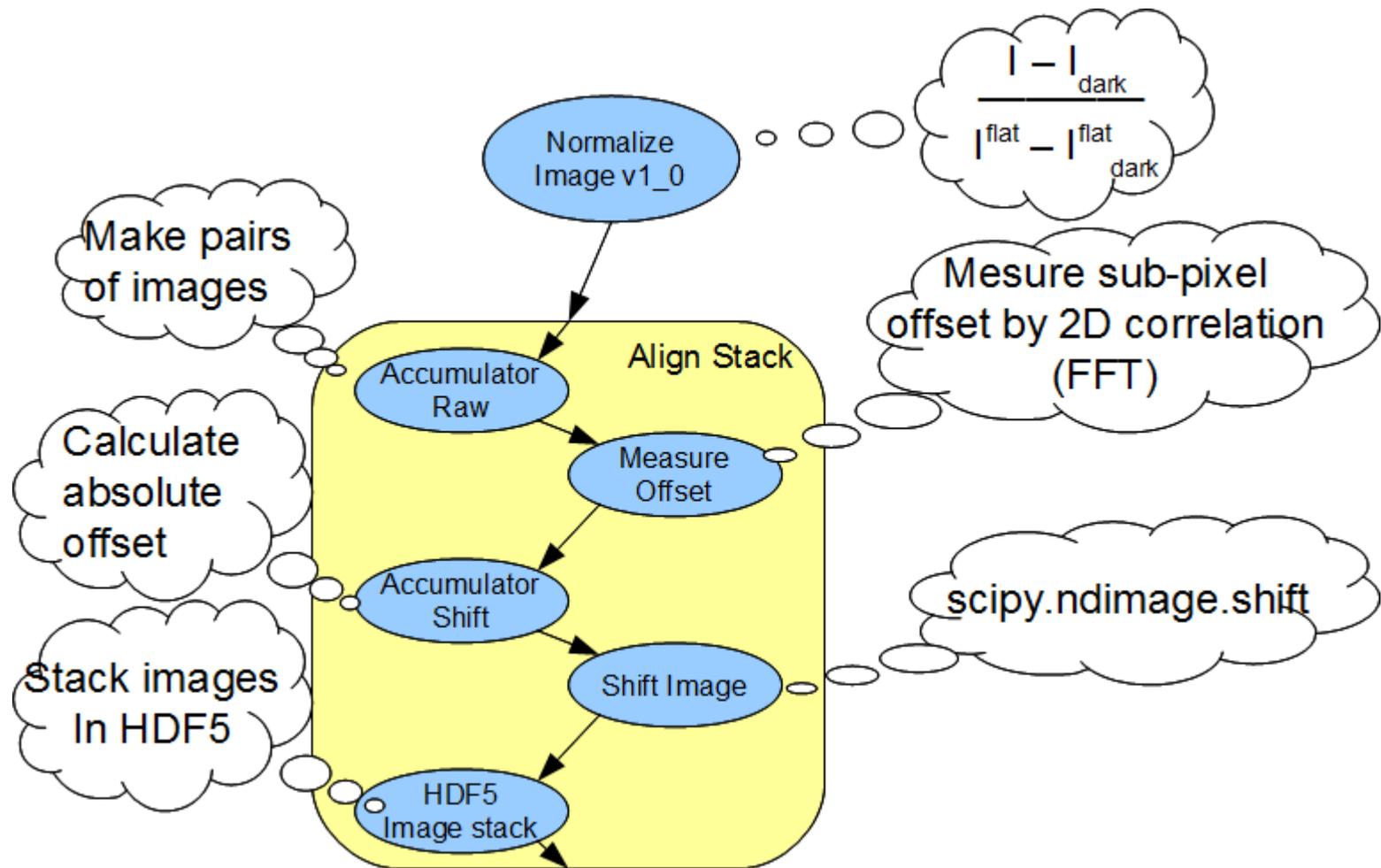
It consists on a (pure) Python kernel and plugins to implement functionalities

- No GUI (A Java based one is under development)
- It uses Numpy, but implements linear algebra routines for Jython use

Its main objective is robustness, not performance

- Nightly builds
- Automated tests

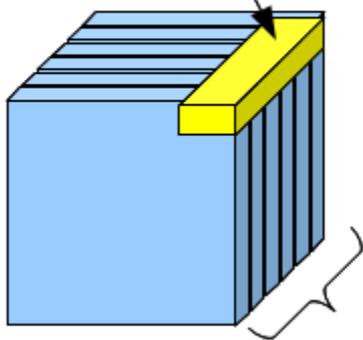
EDNA Workflow for ID21 Full Field



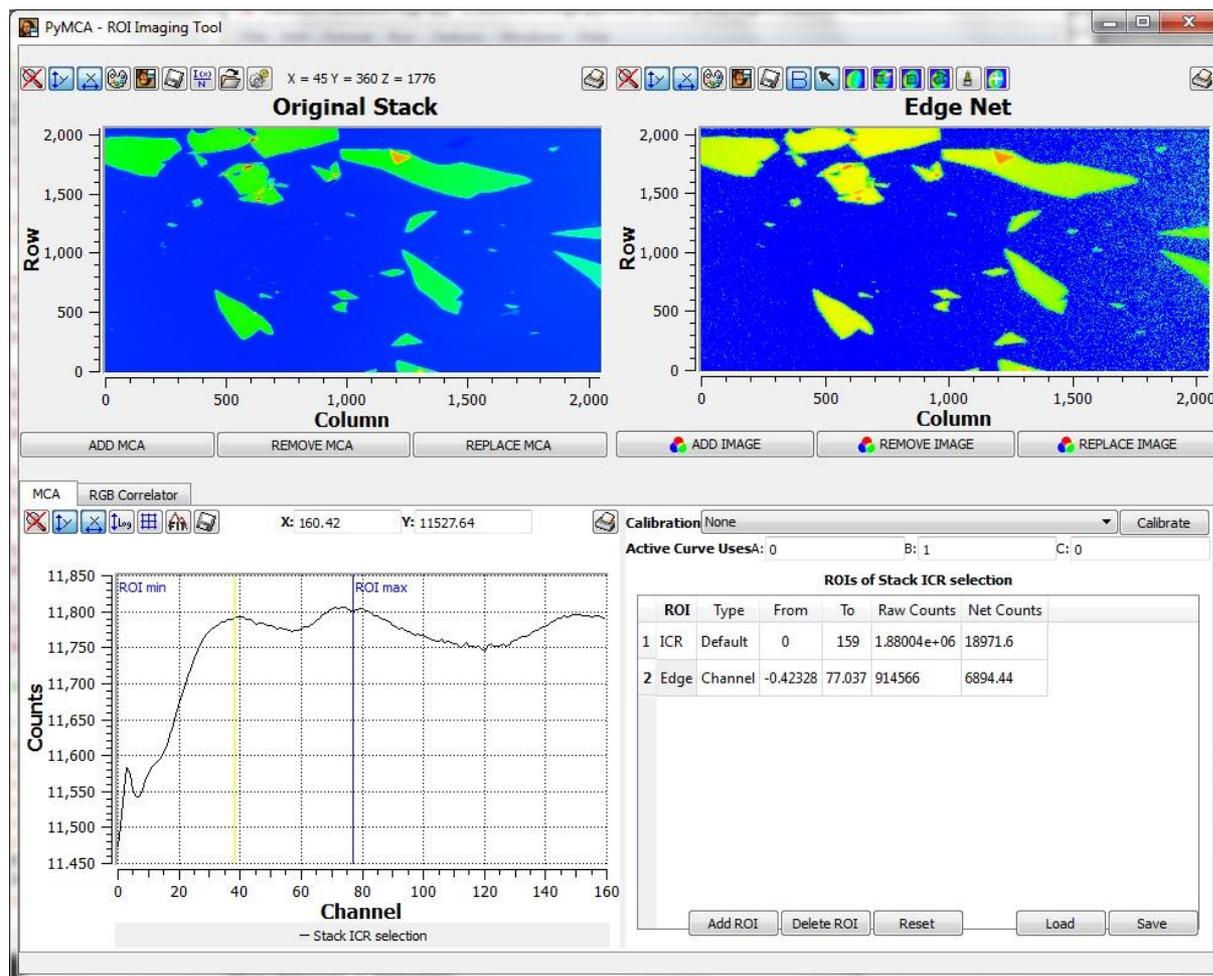
Acknowledgments: Chris Jacobsen

ID21 Full Field Tests

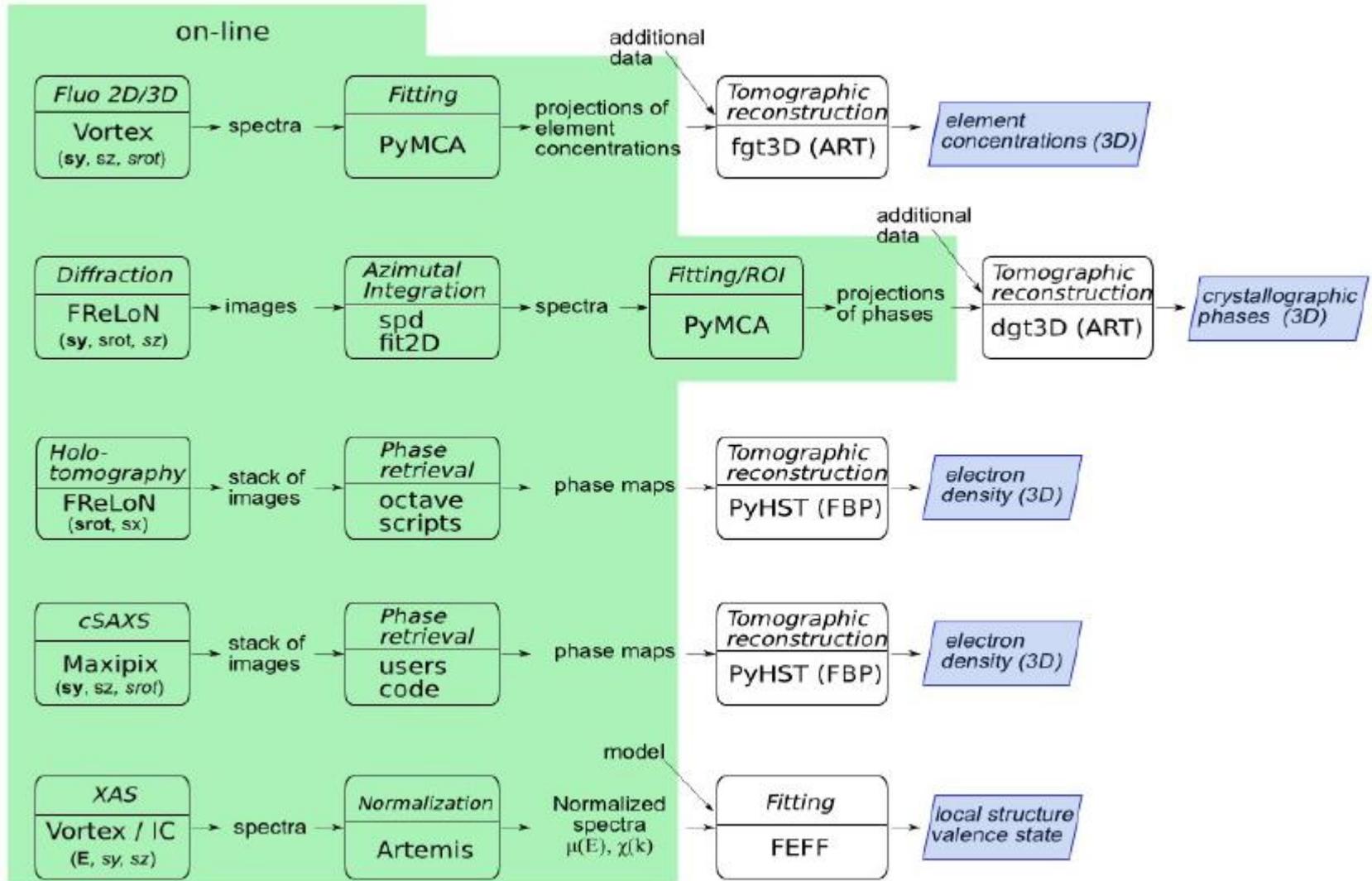
XANES Spectra



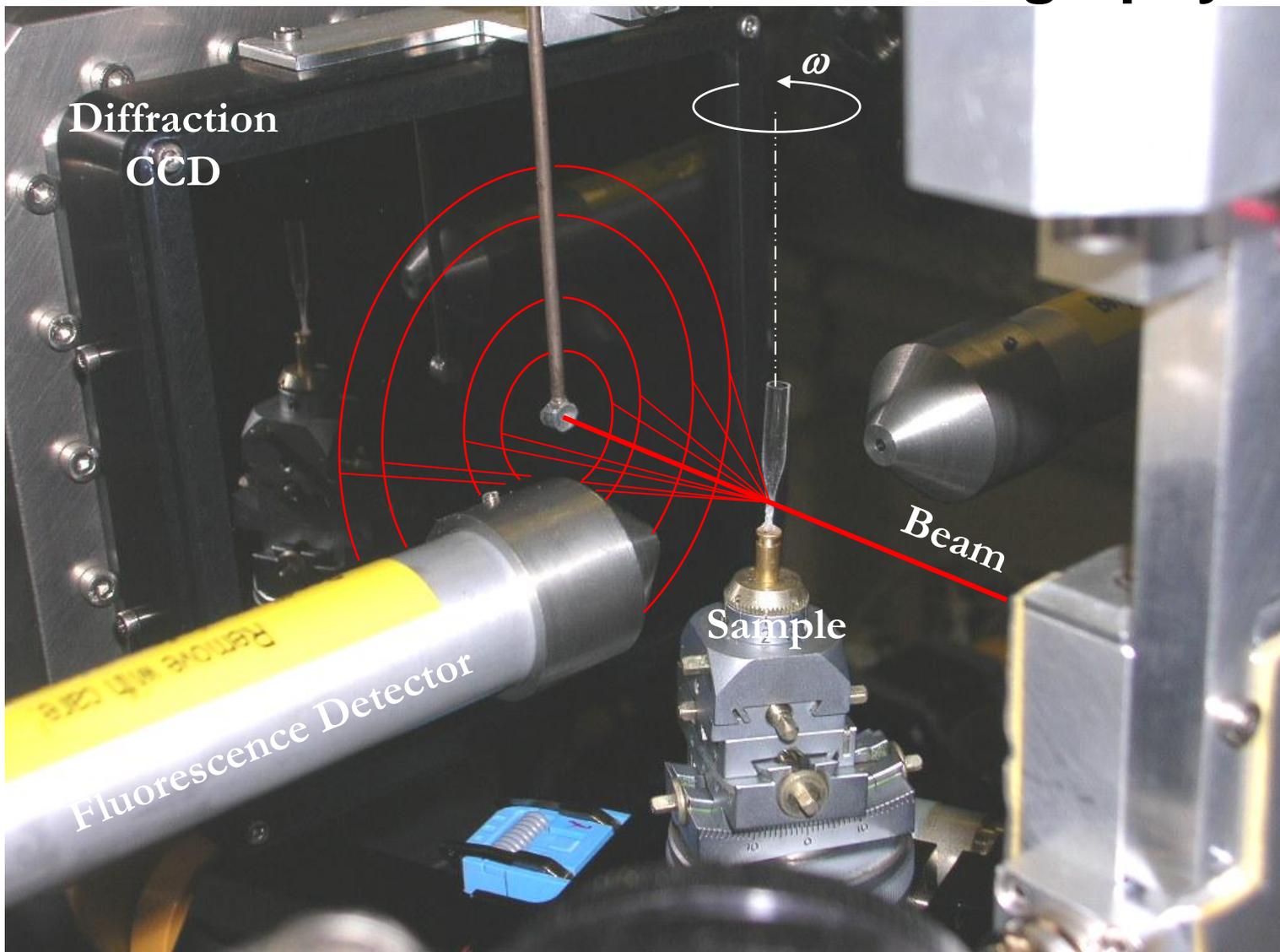
Stack of images

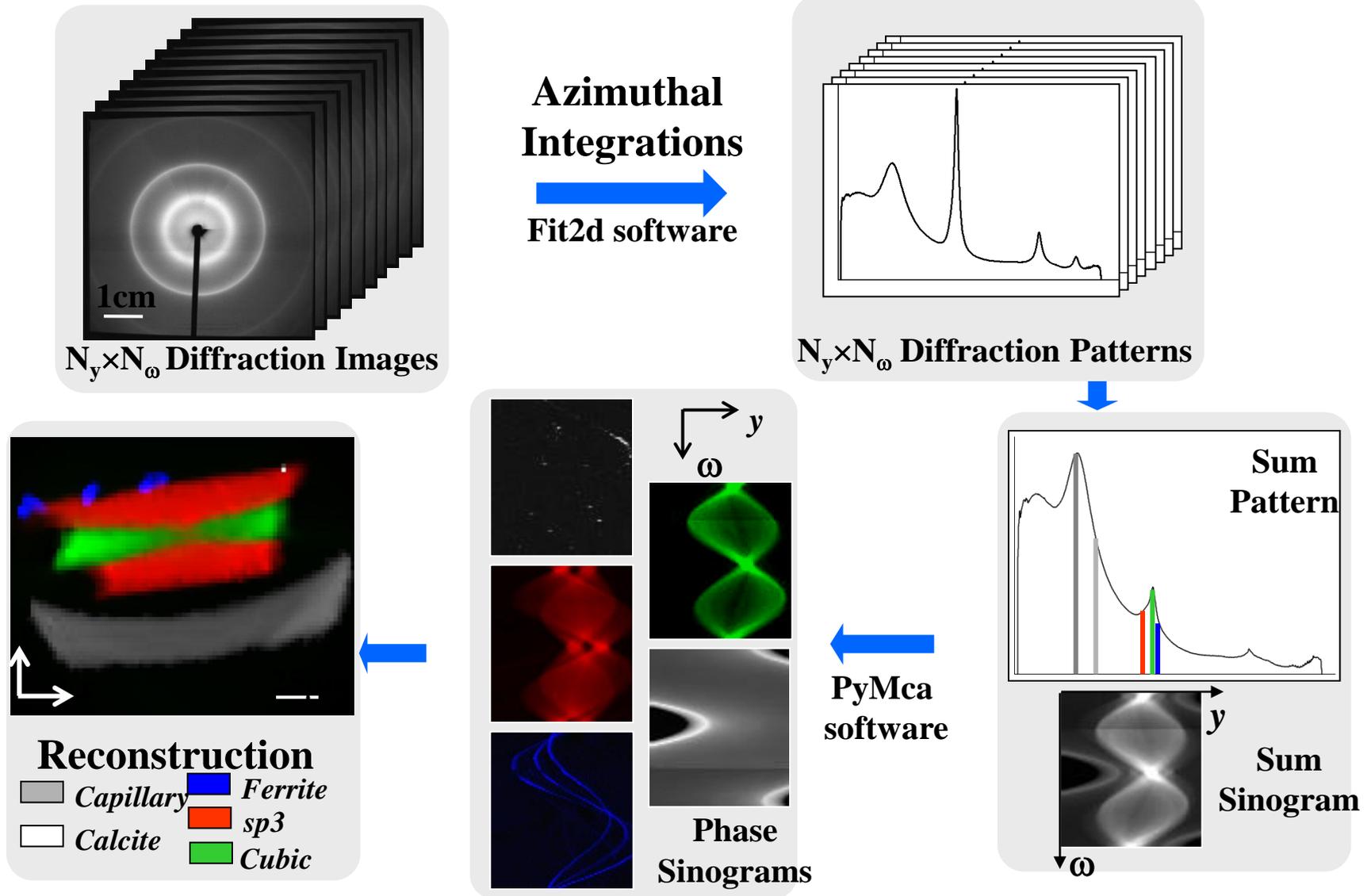


Current Situation: ID22 Data Analysis



ID22 – Fluorescence-Diffraction Tomography





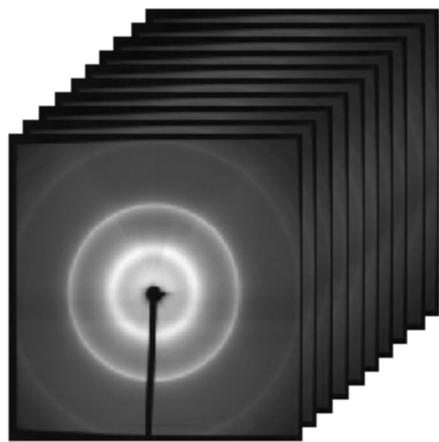
Acknowledgements: Pierre Bleuet CEA - Grenoble

Data format issues

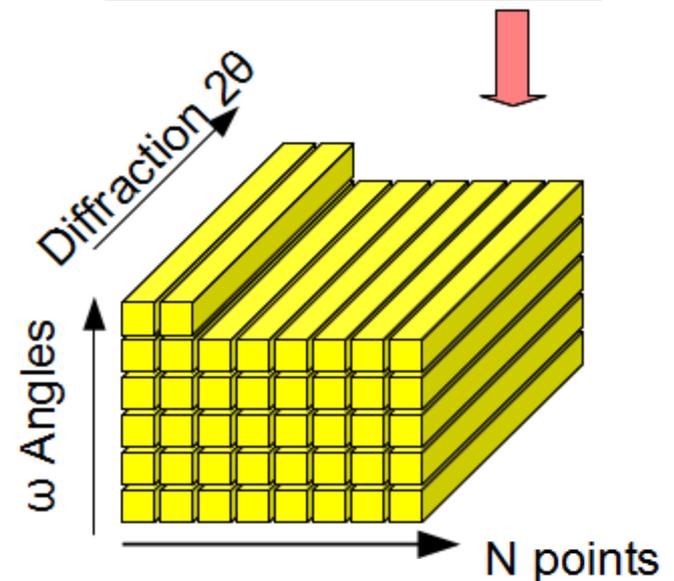
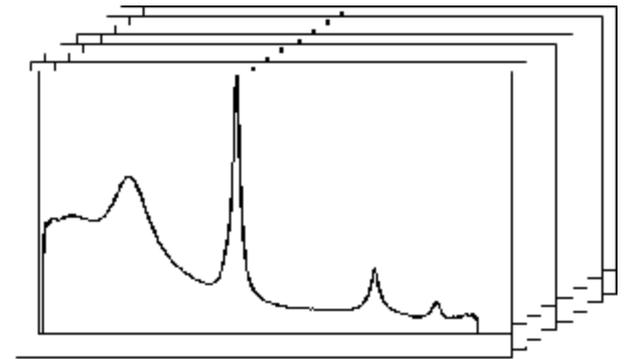
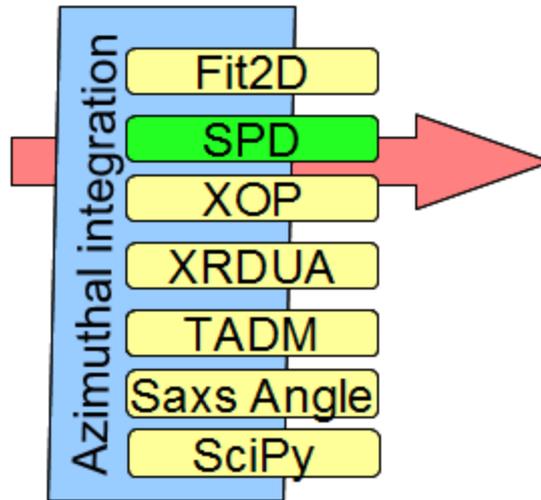
- Diffraction images in EDF or MarCCD format
- Fluorescence data in EDF or SPEC file format
- Scan information in SPEC file format
- Result of azimuthal integration on Fit2D .chi format

EDNA Workflow for ID22 Diffraction Tomography

Reduce each image (2D) to a spectrum (1D)
 Store a 2D mapping of spectra in HDF5
 Store of sinograms in HDF5



$N \times \omega$ 2D
 Diffraction
 Images



HDF5

Today's Situation at the ESRF

The sequencer drives in the same code:

- The experiment control (ex: moving motors, ccd start command)
- The data acquisition (ex: ccd reading)
- The data saving (ex: SPEC file or EDF file writing)

Data are saved at various levels

- SPEC built-in
- SPEC macros
- Device servers
- Treatment tools

Decisions Already Taken

HDF5 will become our “standard” format

- Our device servers have to be able to save data in HDF5 format
- SPEC has to provide minimalistic HDF5 support (groups, datasets and links)
- Our analysis tools have to natively support HDF5

Automated data reduction/data analysis steps will be performed using EDNA

PyMca will be used for mapping of 1D data (XRF, FTIR, XEOL, XANES, ...)

Status

PyMca and EDNA can already read and write HDF5 files

Preliminary HDF5 support in SPEC implemented last week. Tests to follow.

Automated data reduction workflows using EDNA implemented for:

- ID21 Full field XANES image alignment
- ID22 azimuthal integration

Data Manager approach under evaluation

Tomography beamlines begin to manifest interest on using HDF5

Conclusion

The days where control software programmers, data analysis programmers and beamline scientists worked separately seem to be fortunately finishing at the ESRF.

Given the potential amount of data, online treatments should aim to analyze them before they are written to file.

Whatever we finally do, HDF5 is going to play an important role.