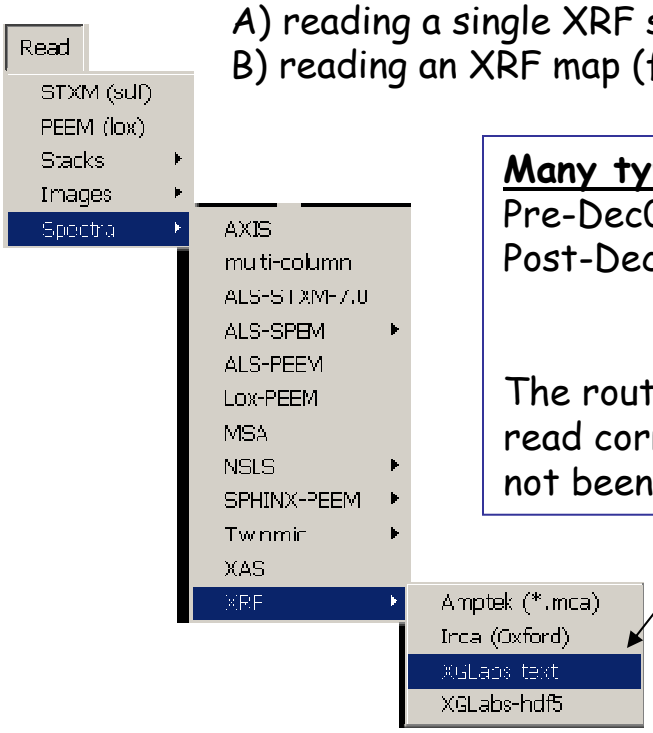
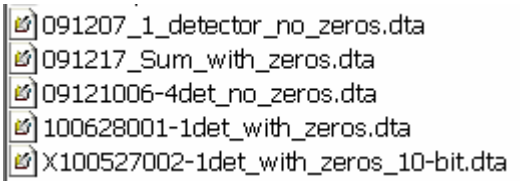
Reading XRF data stored from XGLabs MCA pgm

GOAL: describe procedures used to read XGLabs data flies
A) reading a single XRF spectrum (*.dta, or *.hdf)
B) reading an XRF map (from set of *.dta files or from *.hdf file)



Many types for formats !
Pre-Dec09 - 1 ch, sum, 4-channel
Post-Dec09 - 1 ch, sum, multi-channel
with zero channels, without zeros

The routine is supposed to auto identify and read correctly but all possibilities probably have not been checked

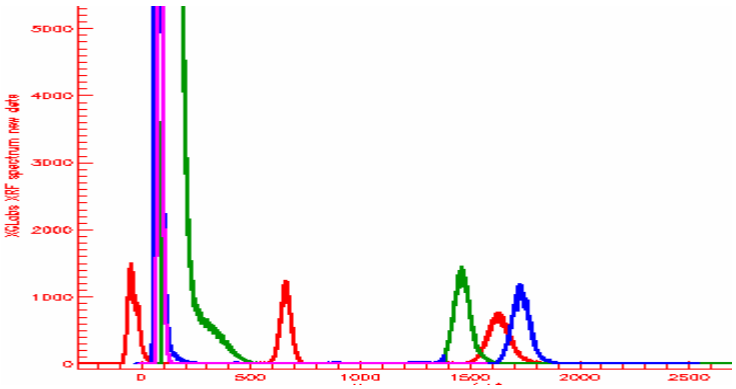


A1 Reading a single XRF spectrum from *.dta

0912006-4det_no_zeros.dta
Each of 4 channels read

(NB data from old Twinmic SDDs when they were having noise problems

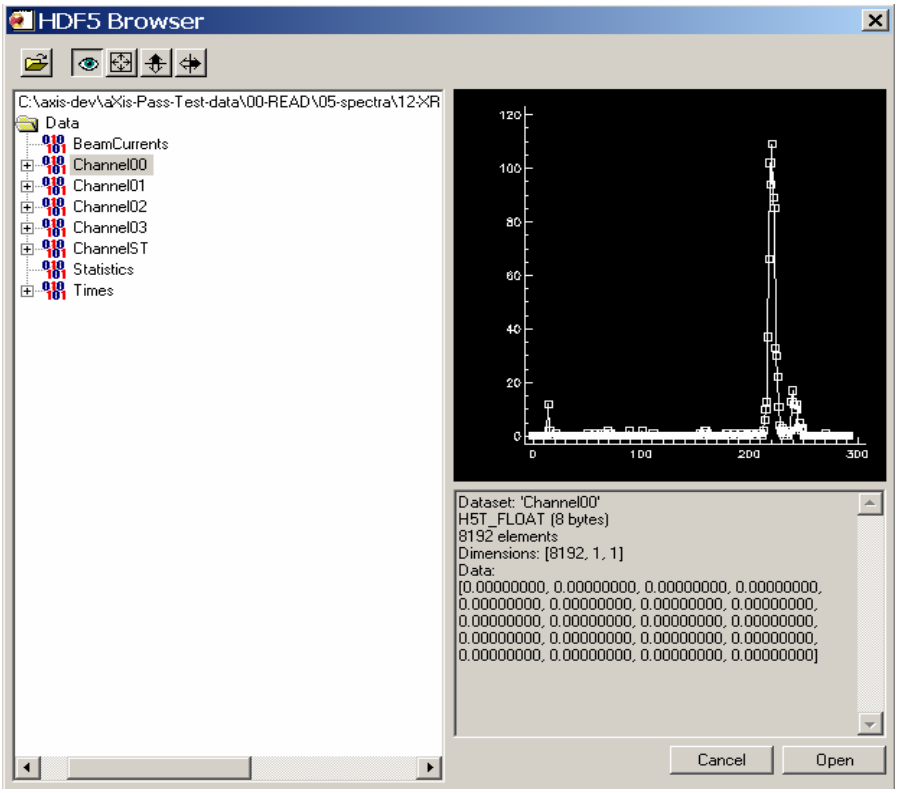
Careful with X-ray energy calibration - if it was done, will be OK, but if not, it is just channel number

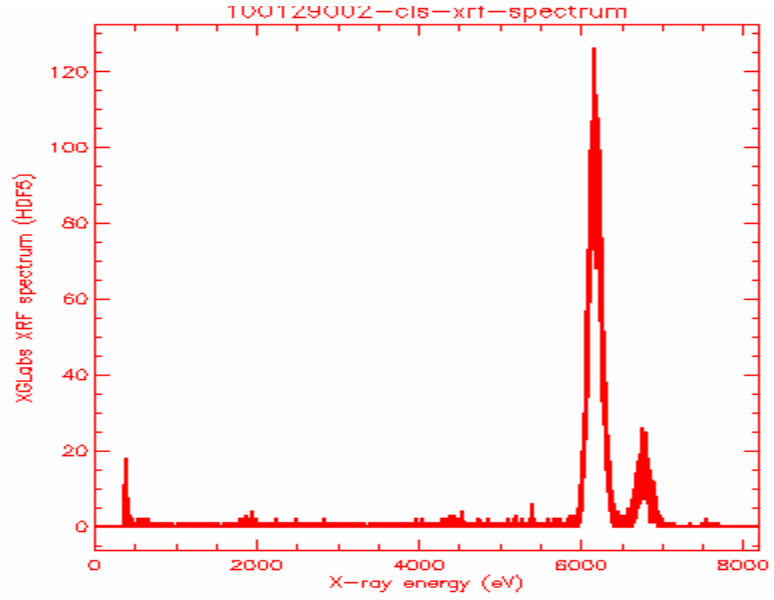


A2 Reading a single XRF spectrum from *.hdf

Read~spectra~XRF~XGLabs-hdf5

Opens HDF5_browser
Navigate to channel of interest (spectrum will be displayed)
Click on 'open'

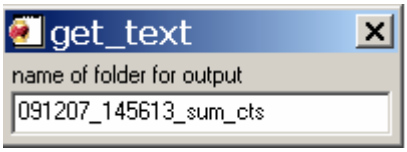


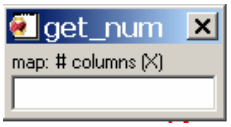


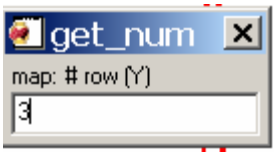
B.1 Reading XRF stack from {*.dta}

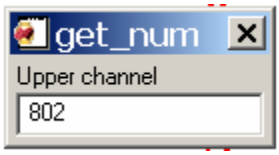
1. select first file of sequence
2. select last file of sequence
3 define folder name for *.cts files
4. define # of columns (example =7)
5. define # of rows (example =3)
6. select detector channel (if 'all')
7. select upper data channel
8. select bin (spatial)

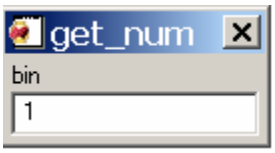
Each *.dta file is then read
Stack is generated
User is asked to provide the stack name
Each spectrum is wriitten as a *.cts file for read into PyMCA

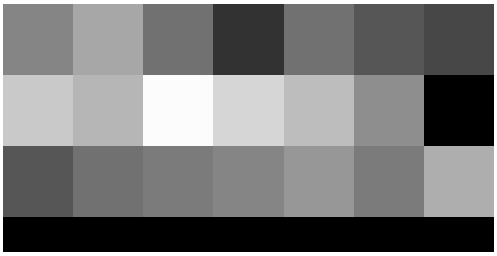


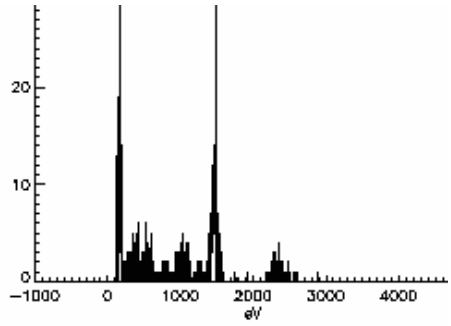


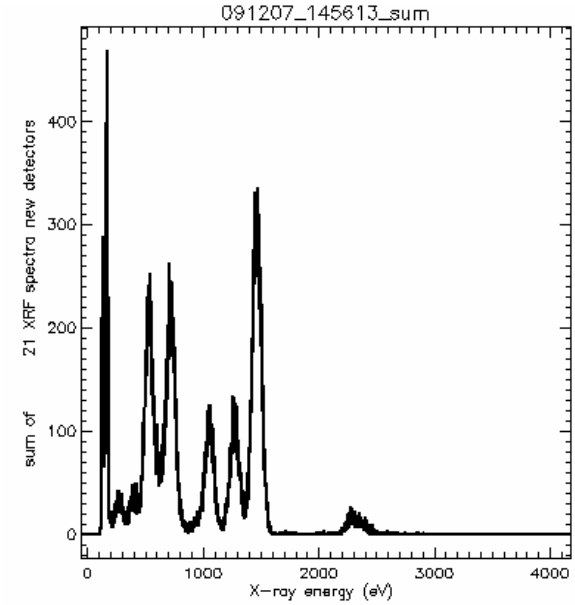










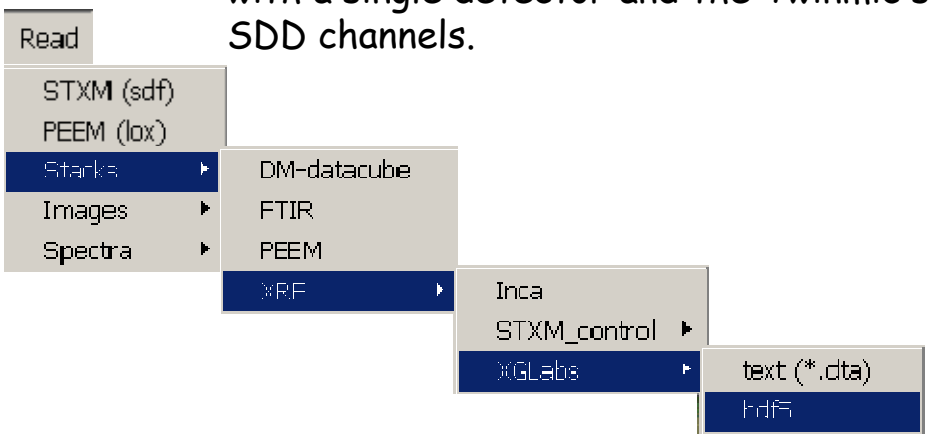


Sum of all spectra is read-in to buffer 0

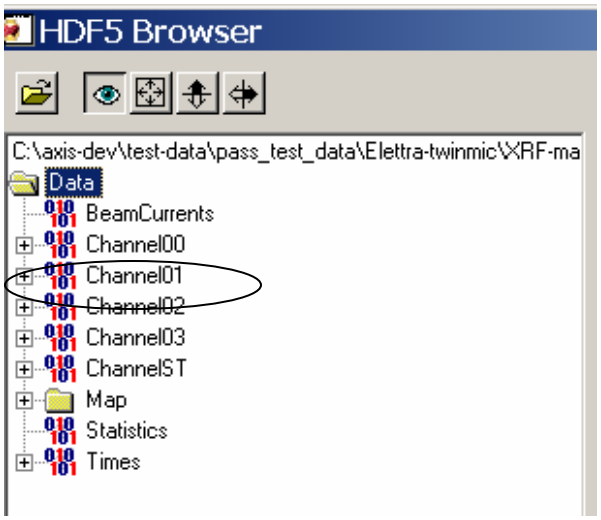
* Set X-ray energy calibration on this spectrum
* Store it as an aXis2000 file
* Use the 'change energies' command in stack_analyze to calibrate the energy scale of the stack

B.2 Reading XRF stack from *.hdf (1)

Method uses IDL routine hdf5_browser to select the specific SDD detector to read since different MCA channels can be used with a single detector and the Twinmic system has up to 8 active SDD channels.



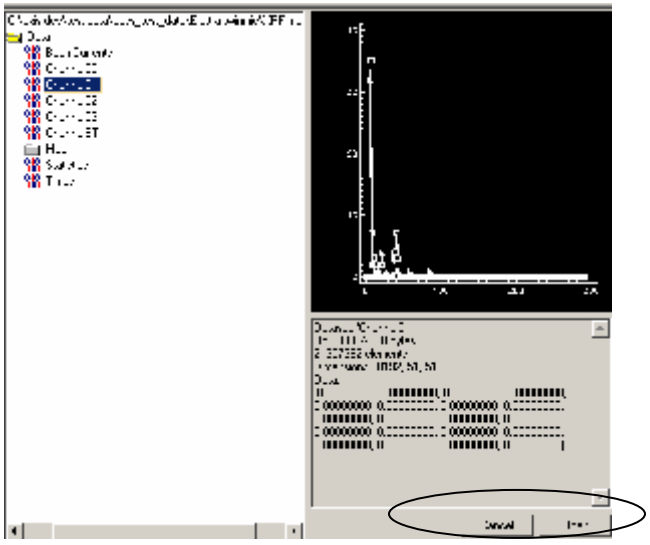
1) Select *.hdf file (Test example = xrf_map09.hdf (17-Dec-09)



2) Double click on name, then Data to expand menu of hdf file content

In this example channel01 was used so the data is in channel01 and ChannelST

When click on Channel01 the average of all spectra is displayed on the right screen



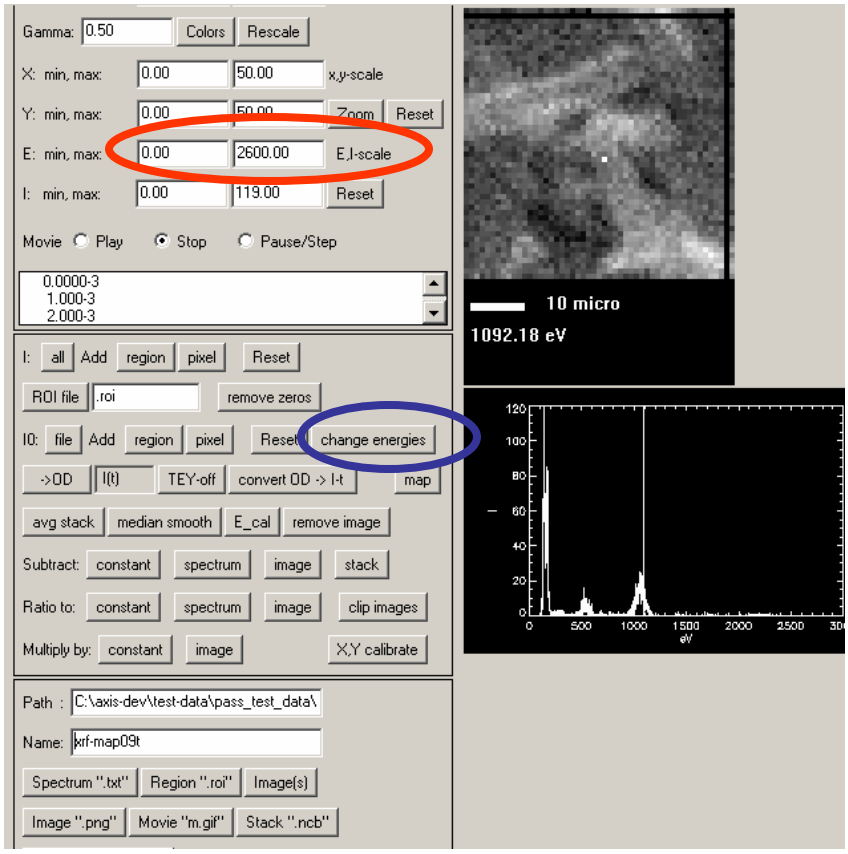
3) Once you have identified the data channel you wish to read, click on 'Open'

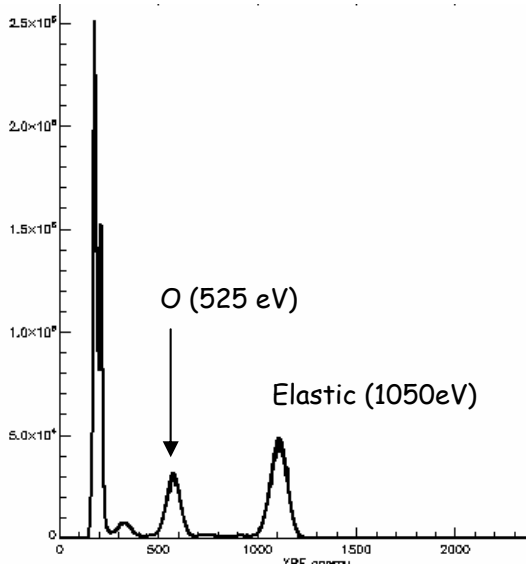
B.2 Reading XRF stack from *.hdf (2)

4) After some time (stacks are typically large - 10-50 Mb as pdf so it takes a while to read), you will be asked the name for the aXis2000 format stack file.

5) Once stored,

* stack_analyze opens up with the stack loaded
* the sum of all XRF spectra is loaded into buffer 0 of aXis2000



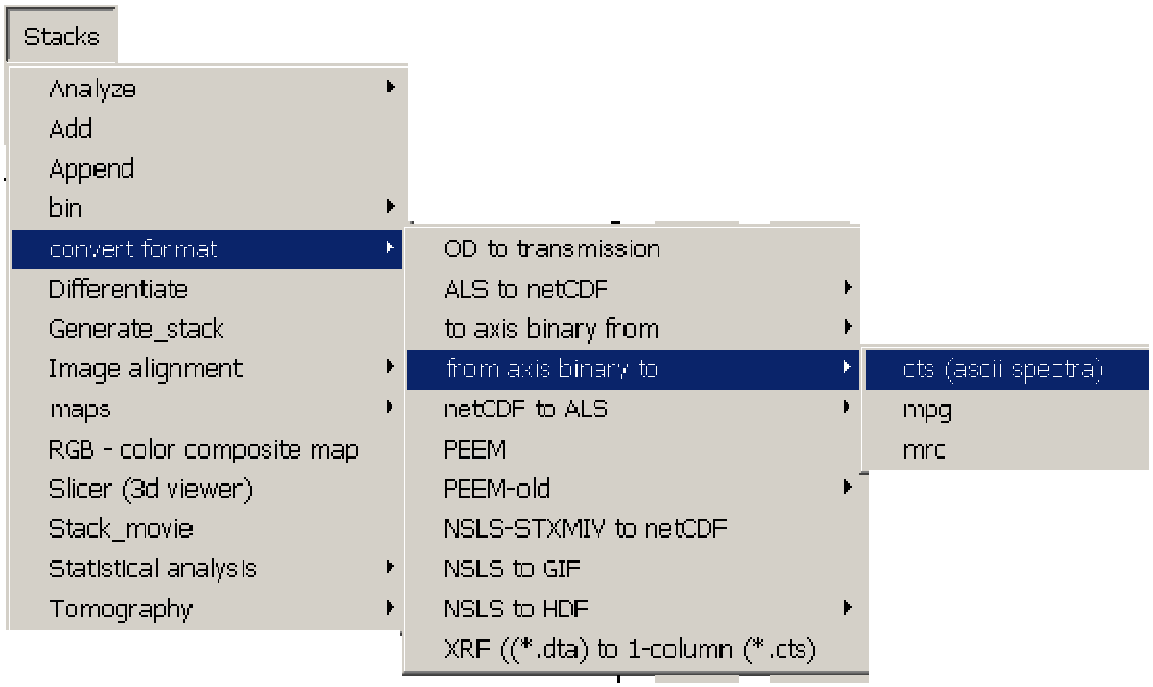


6) The energy scale is channels and typically needs to be calibrated This can be done in aXis using known XRF peaks. If you write out the calibrated spectrum as an aXis2000 file, then you can use the **'change energies'** button in stack_analyze to load the calibrated energy scale.

7) Also you can **truncate the stack to the energy range of interest** (0-1300 eV, or perhaps 2600 eV, since there is higher order light exciting Mg, Al, Si, S Kα peaks, in this example). Do this by selecting lower and upper energy limits, then writing out the stack with the same or a new name

8) since the XGLabs defaults (in Dec09) used 13-bit ADC (0-8192), the peaks are over-sampled. Visualization and analysis will be improved if a 2-fold or 3-fold energy scale binning is done. (**stacks~bin~energy**)

Converting an (XRFmap-stack) to {*.cts}



1. select axis2000 *.ncb binary stack (should be an nXRF map)
2. give name to folder in which to store the *.cts files
3. give main name for the files

Files are written out (1 XRF spectrum per pixel) with sequential names
(0 to #pixels -1)
Sum of all XRf spectra also generated and stored
(useful for PyMCA set up)