CLS STXM and Ptychography Data Analysis

Dr. Jian Wang
10ID-1 Spectro-Microscopy Beamline
Canadian Light Source
University of Saskatchewan
December 4th, 2020

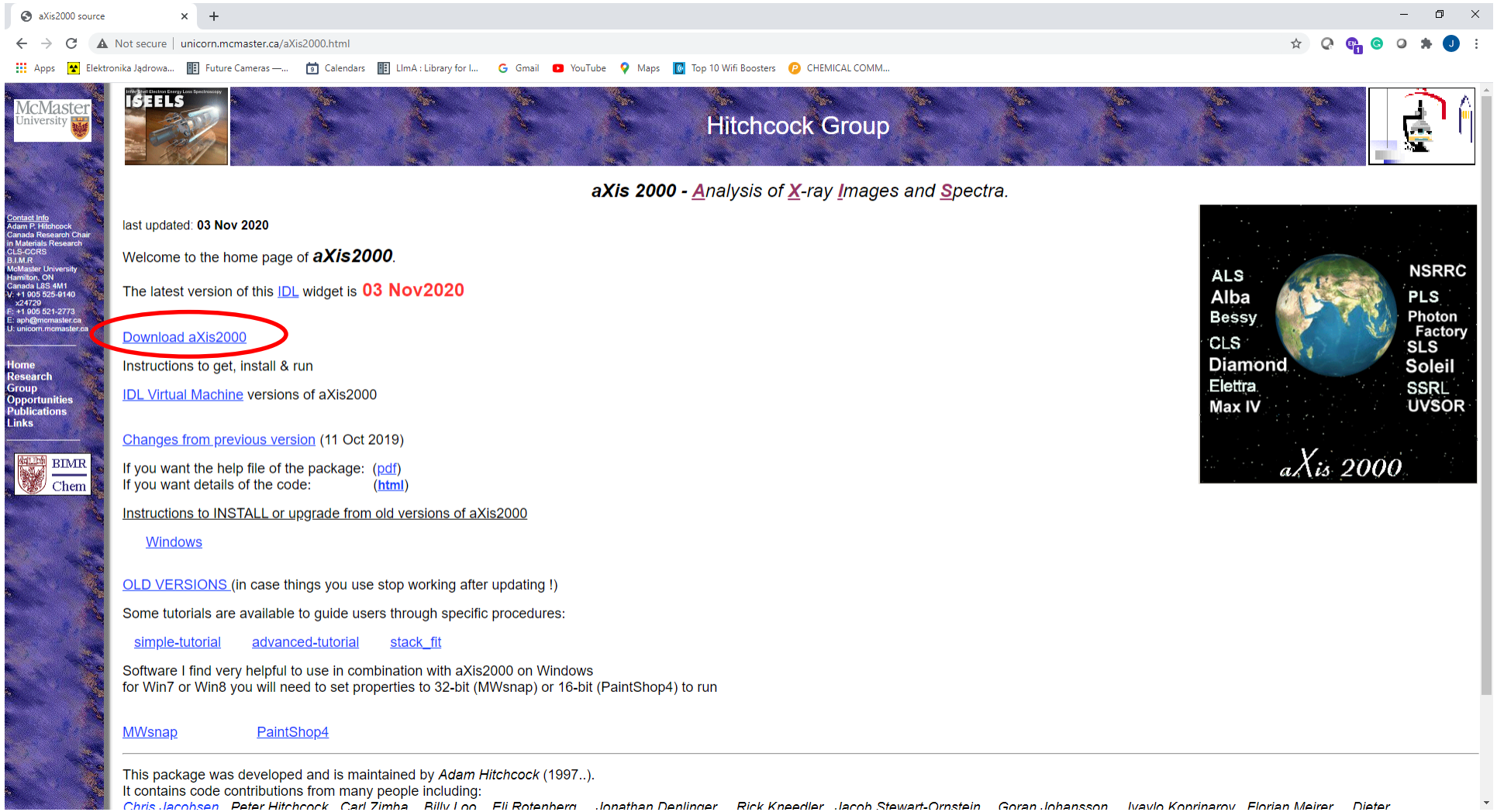
Outline

- aXis2000
  - Installation and Introduction
  - Images
  - Spectra
  - Linescans
  - Stacks: alignment, spectra, mapping, fitting, etc.
  - Cryo-STXM
- PyPIE: STXM-Ptychography
- PCA_GUI and Mantis
- ImageJ Plugins - ScatterJ

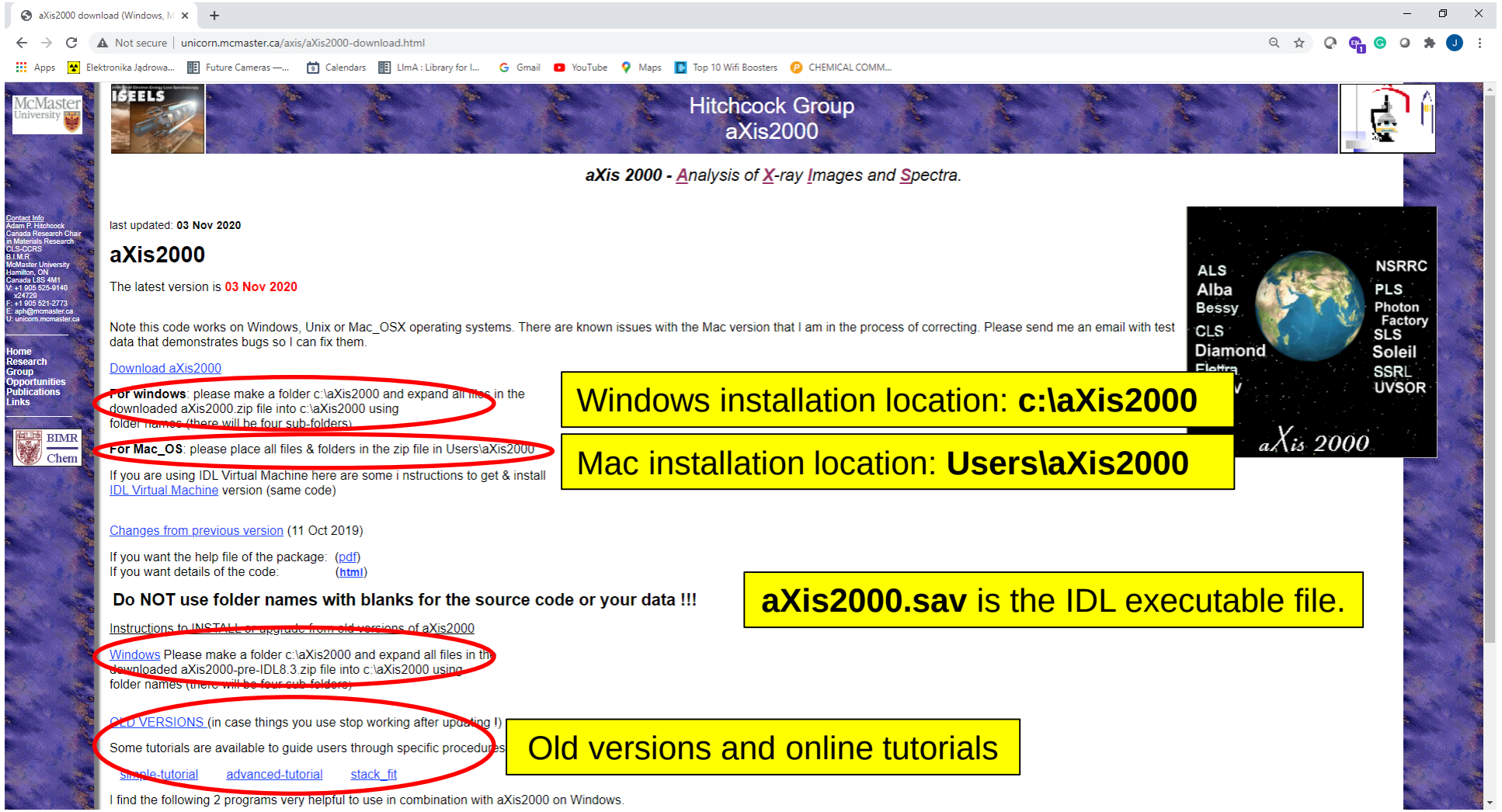
STXM and Ptychography Data Analysis Software

- aXis2000, Interactive Data Language (IDL) Virtual Machine, http://unicorn.mcmaster.ca/aXis2000.html.
- Other IDL based: stack_analyze.sav, pca_gui.sav
- Python based: Mantis 2.3.02
- Ptychography: PyPIE, Sharpcamera, PtychoLib, etc.
- Other imaging processing: ImageJ and plugins, MATLAB, etc.
- Other spectroscopy processing: Athena, Fityk 0.9.8

aXis2000 – **A**nalysis of **X**-ray **I**mages and **S**pectra



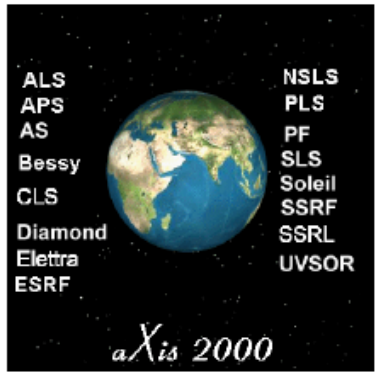
aXis2000 – **A**nalysis of **X**-ray **I**mages and **S**pectra



aXis 2000

Analysis of X-ray microscopy Images and Spectra
(30 June 2014)

aXis2000 - Analysis of X-ray microscopy Images and Spectra - is an IDL widget for viewing, comparing and processing X-ray microscopy images and spectra. IDL stands for Interactive Data Language, a scientific computing platform developed by Research Systems Inc (RSI), currently part of Exelis Visual Information Solutions (http://www.exelisvis.com/). aXis2000 contains scripts developed by Chris Jacobsen, Carl Zimba, Adam Hitchcock and others. The widget platform was written by Adam & Peter Hitchcock. It is maintained and occasionally updated by Adam Hitchcock. It can be obtained from unicorn.mcmaster.ca/aXis2000. It operates on Windows (WIN), Unix (X) and Macintosh (MAC) versions of IDL, although there are problems at present with operating it on a Mac, if you use IDL 7.0 and later versions.



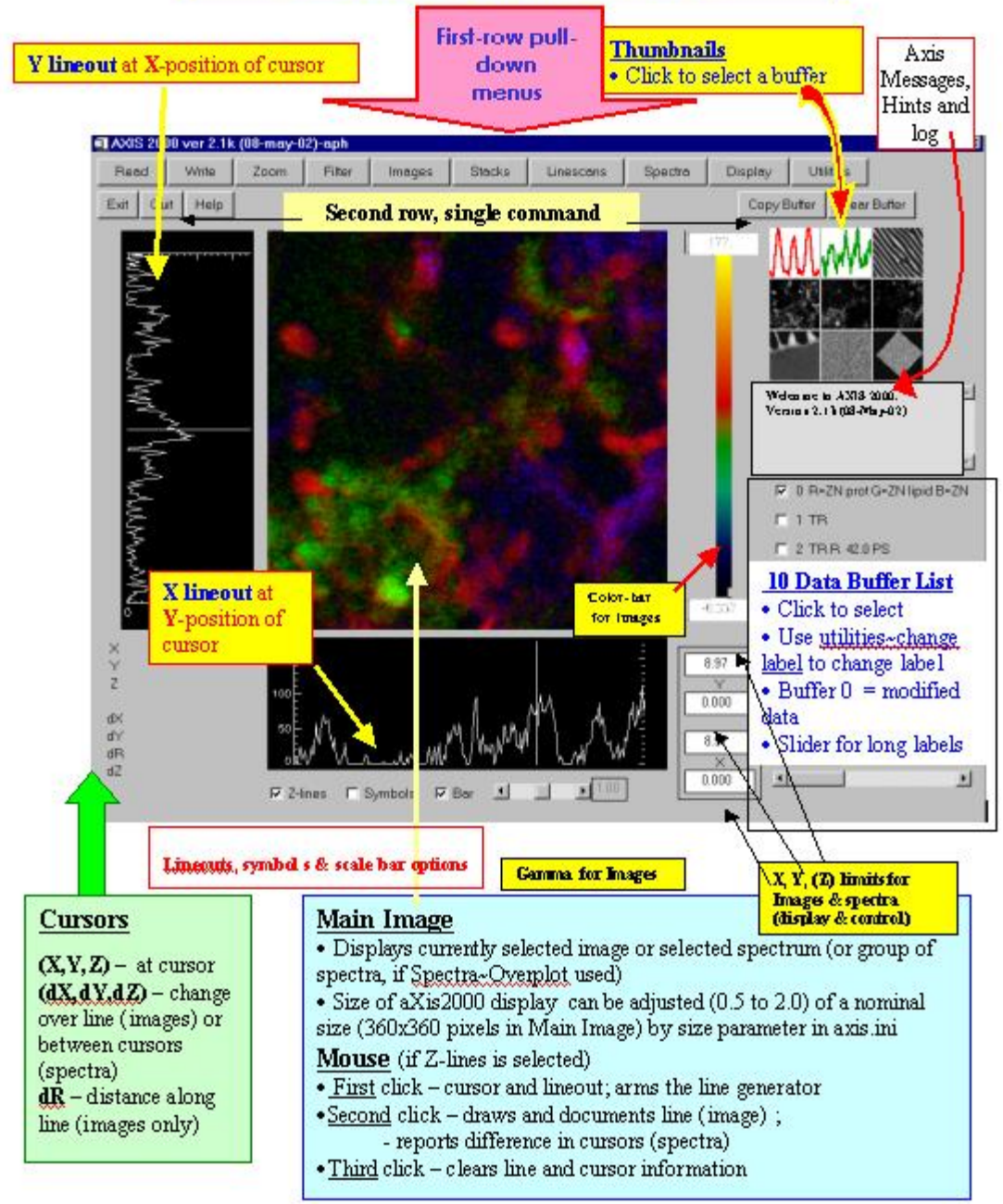
Since May-04 a compiled version (**aXis2000.sav**) for use with IDL Virtual Machine has been available. This allows access to the power of aXis2000 without needing to purchase an IDL license. Please note that there are often features of aXis2000 that work with the licensed version, but not with the VM version, and that the specific details of these problems depend on the version of both IDL and aXis2000.

I would appreciate it if you would notify me by email (aph@mcmaster.ca) about problems with the code or if you wish to make suggestions for improvements. If you make extensions or corrections, I would appreciate receiving a copy of your code revisions with sample data, so I can evaluate and incorporate in future versions.

*I thank all the people who have written scripts that went into this. **Carl Zimba** (Photons Unlimited) who supplied ZSTACK and extensively improved the overall package in 2000; my son, **Peter** who helped set up the basic widget structure; **Eli Rotenberg, Jonathan Denlinger, Stefano Cerasari, Tolek Tyliszczak, Andy Smith, Andreas Scholl, Göran Johansson, Jacob Stewart Ornstein,** and many others. SPECIAL thanks to **Chris Jacobsen** (Stony Brook, nsls) for sharing his STACK_ANALYZE and PCA_GUI codes, Rick Kneedler, for providing the basis for the stack-fit routine, and Billy Loo (UCSF) for providing SF, the Henke mass absorption routine and the Conjugate Gradient Optimization routine (ax_cgo).*

TO START *aXis2000*: *after installing aXis2000* (see end of this file)
Windows and Mac OS:
Start **IDL** ;
If you have set the Preferences (in IDL) so that **axis2000_batch.pro** is the start file, aXis2000 will launch automatically.
Otherwise, type **axis2000** on the IDL command line.
If you quit aXis2000 and stay in IDL, you can restart by typing **axis2000**

Features of the *aXis2000* widget

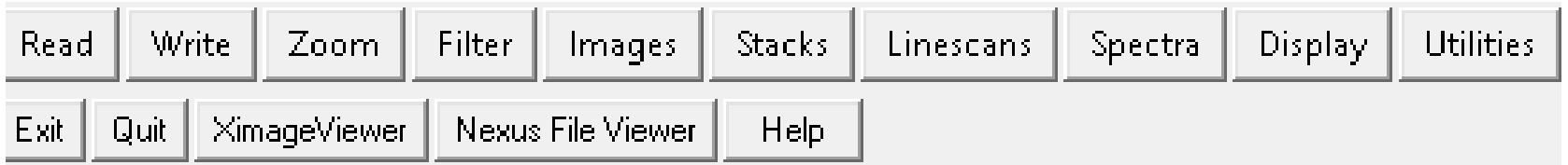


aXis2000 GUI Structure

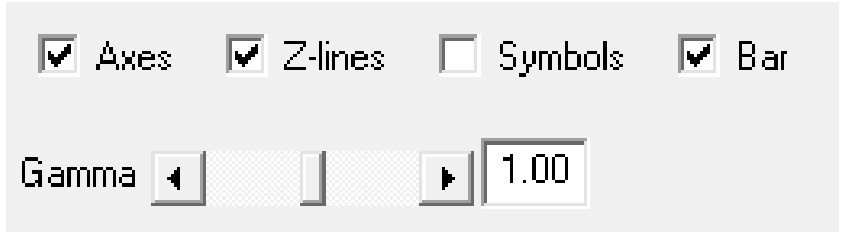
- Think of as a **TOOLBOX** rather than a **Workflow** (Mantis is workflow oriented)
- User can follow pre-set sequence(s)

Tutorials on web site http://unicorn.mcmaster.ca/aXis2000.html (rather dated!!)
aXis2000 Manual (Help) describes function of each button, not workflow

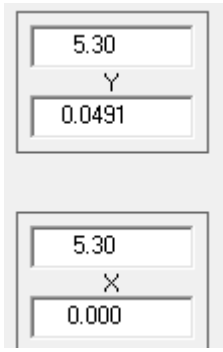
Top level menus organized by **process** (read/write/zoom/filter/utilities)
and by **type of data**: images / stacks/linescans/spectra



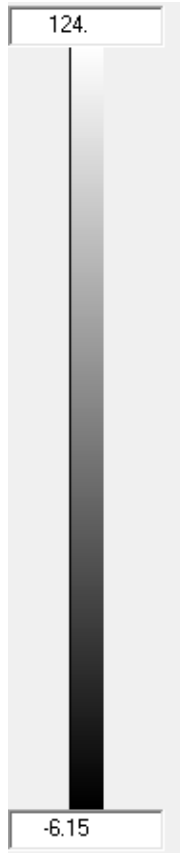
Main panel features



x,y

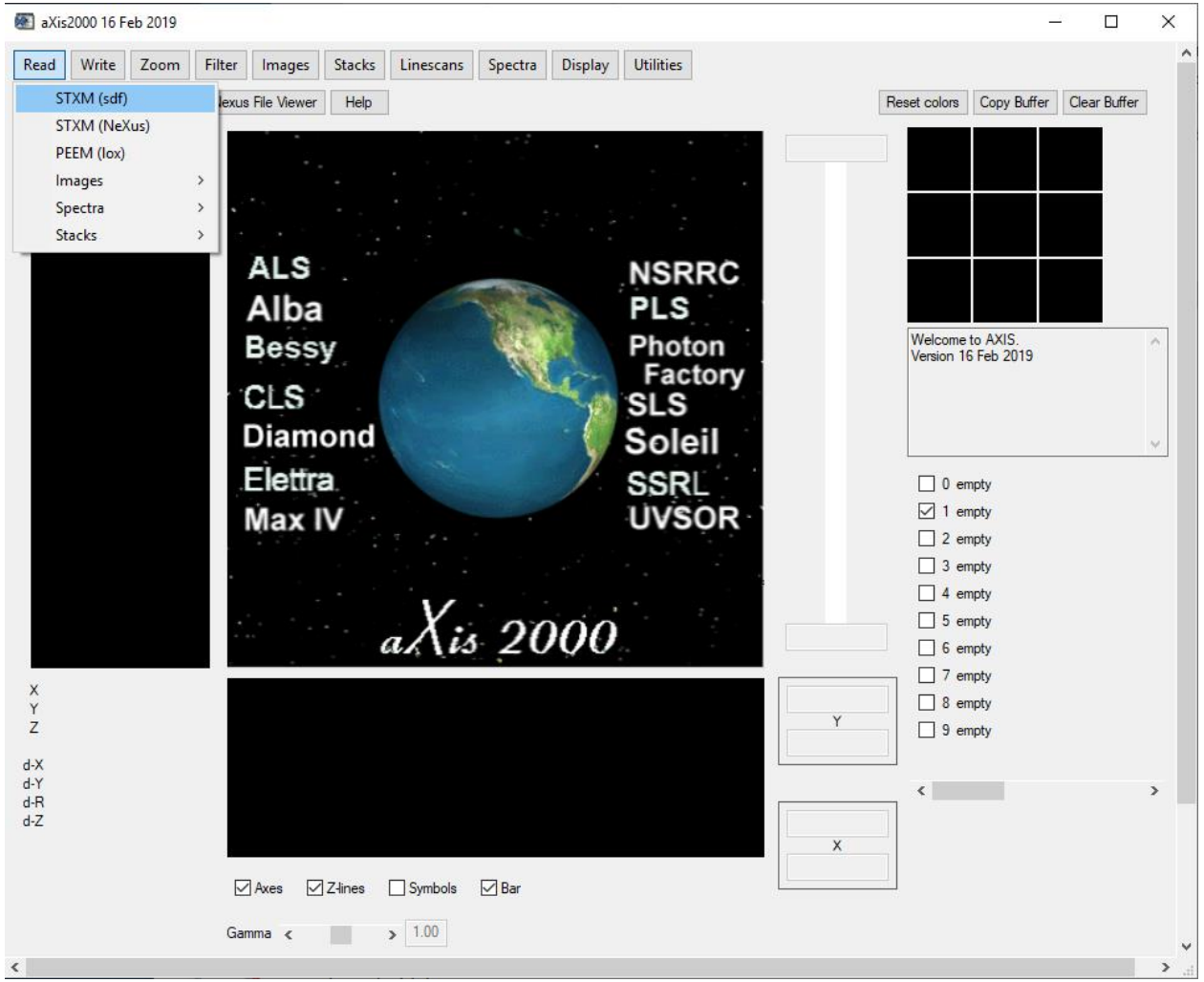


intensity



Images

aXis2000 – Read Data



- **Read → STXM (sdf)**, for standard CLS Ambient-STXM data files
- **Read → STXM (NeXus)**, for CLS Cryo-STXM data files
- **Read → PEEM (lox)**, for CLS PEEM data files
- **Read → Images**, for many other synchrotron instruments data files, graphics files, etc.
- **Read → Spectra**, for many other synchrotron instruments data files, text files, etc.
- **Read → Stacks**, for many other synchrotron instruments data files

Ambient-STXM Data Format

Header File: e.g. **A091111005.hdr**,

Scans, STXM, and Beamline settings

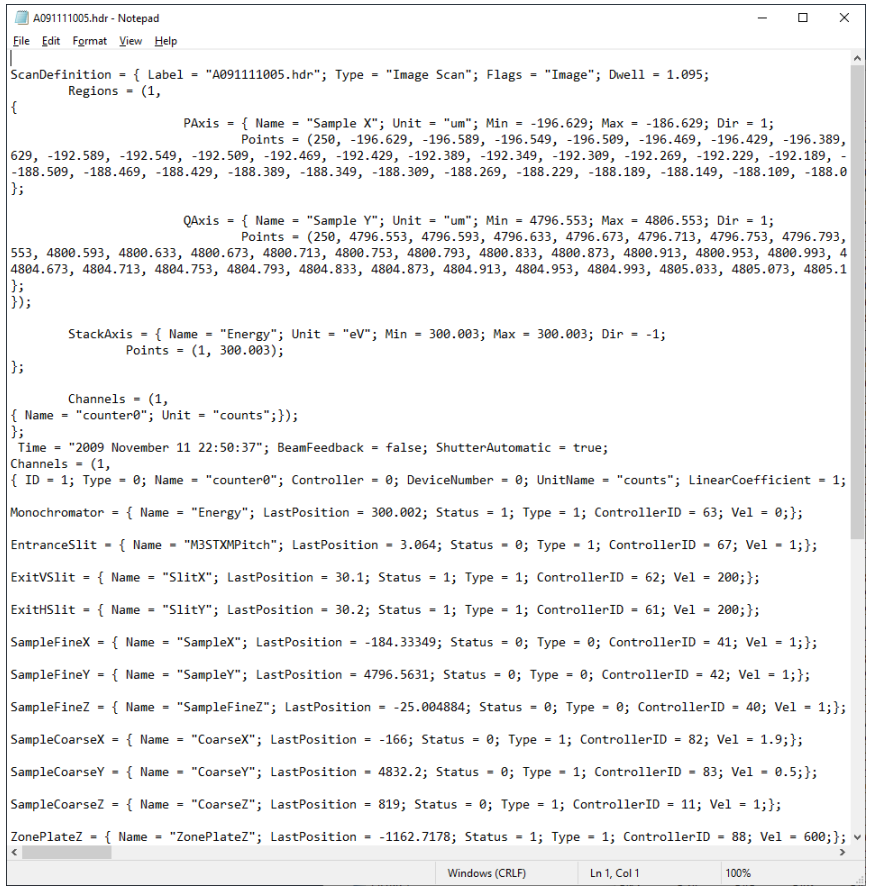
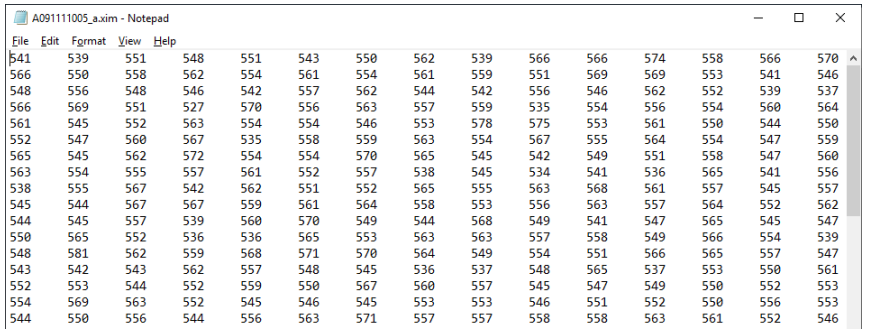


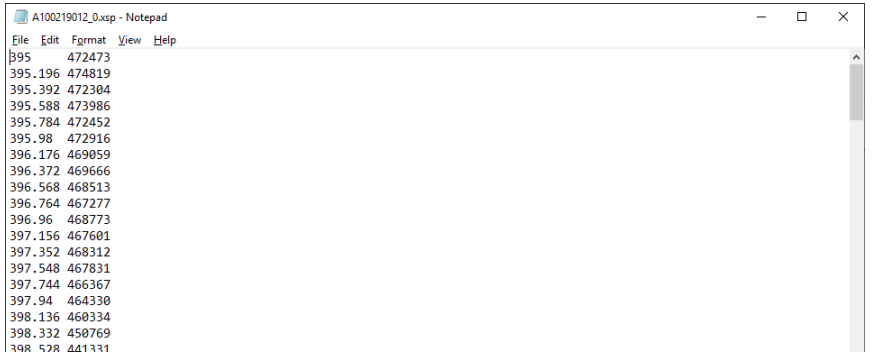
Image Data File:

e.g. **A091111005_a.xim**

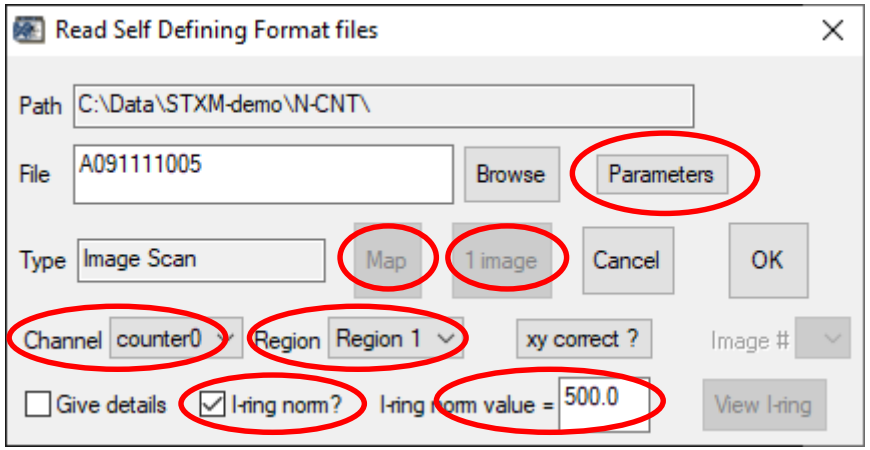


Spectrum Data File:

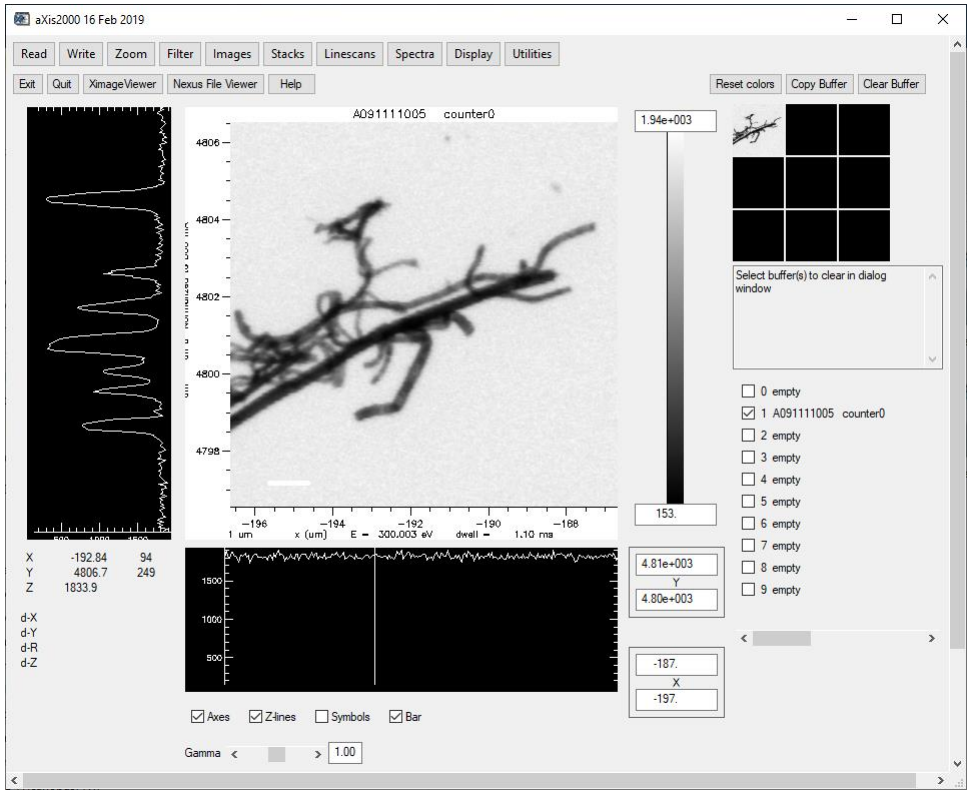
e.g. **A100219012_0.xsp**



aXis2000 – Read Ambient-STXM Images

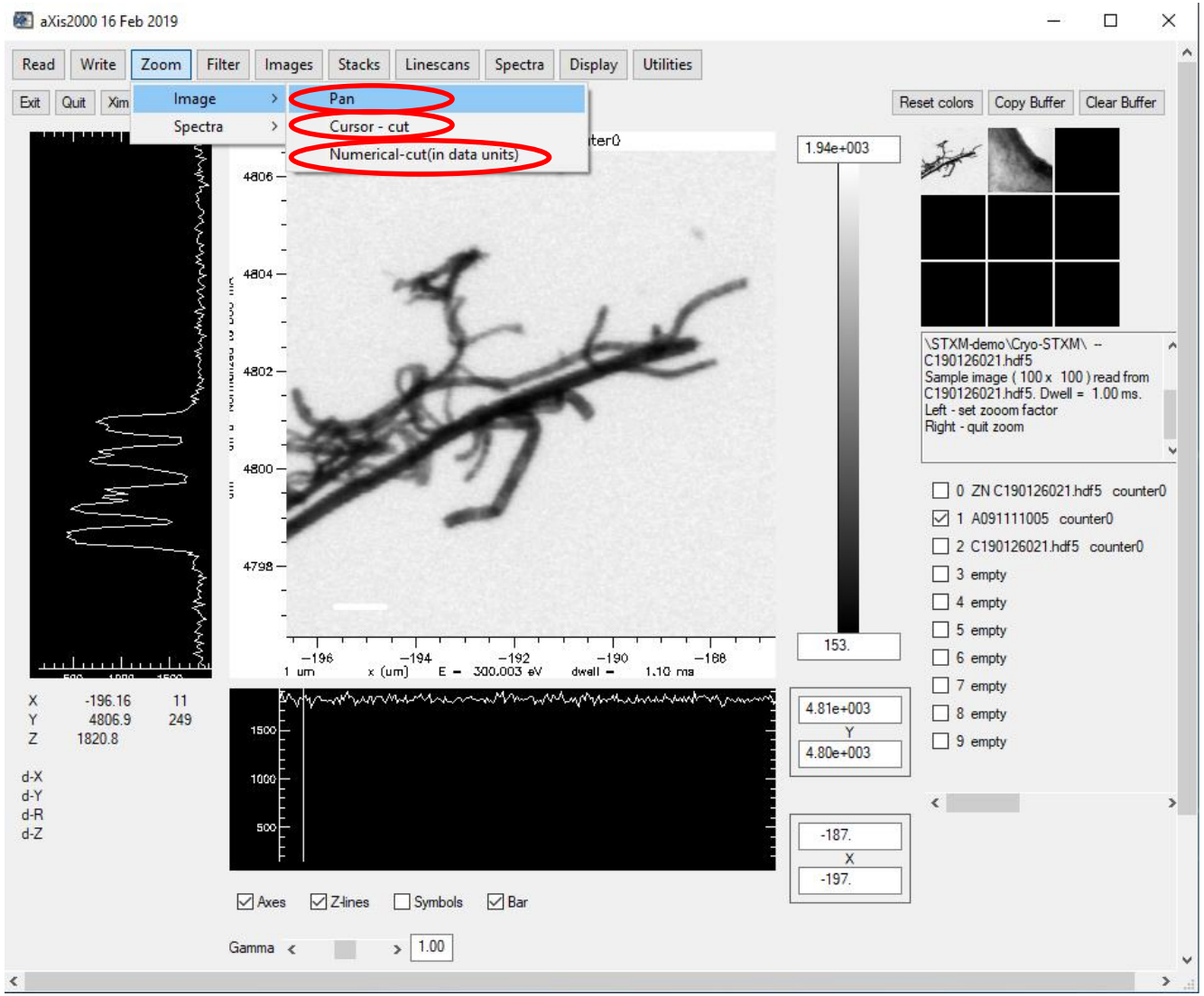


- **Parameters**: access all information in the header file
- **Map**: obtain a two-image subtraction map directly without alignment
- **1 image**: read an image from a selected photon energy for a stack
- **Channel**: select data channel if more than one detector is used
- **Region**: select sample region if more than one image region is defined
- **I-ring norm?**: normalization to I-ring
- **I-ring norm value**: CLS 220 mA



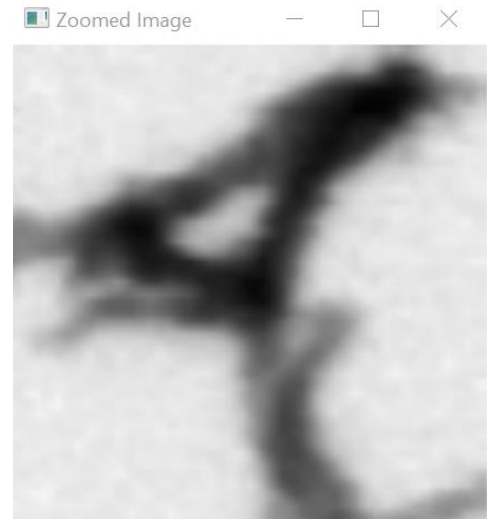
- **First click on image**: select the starting point
- **Second click on image**: select the ending point, then calculate the d-X, d-Y, d-R, and d-Z of the two points
- **Third click on image**: clear the selected points

aXis2000 – Image Zoom



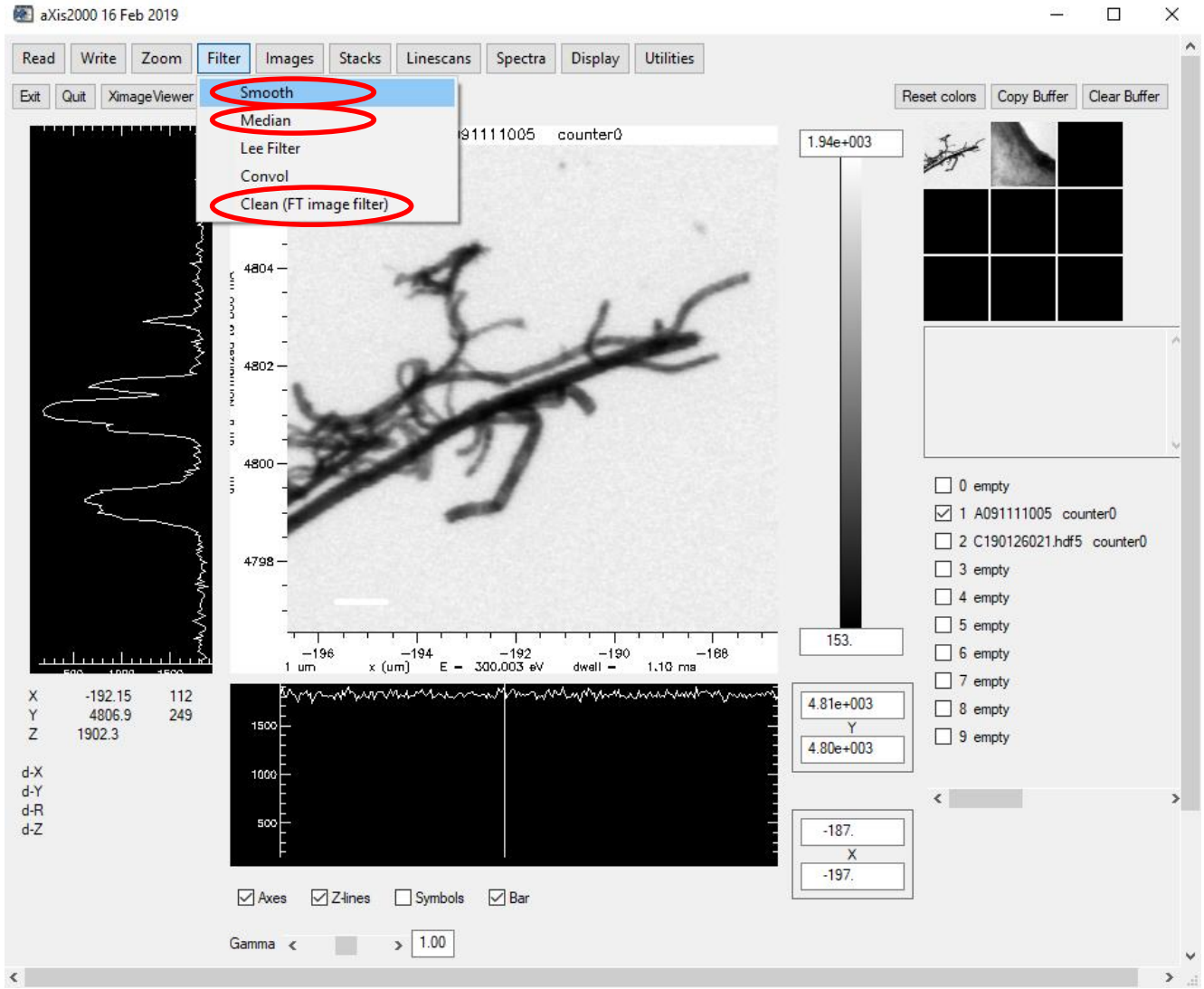
Functions

- **Pan**: pops-up a zoom window with a ~3x expansion of the region around the cursor



- **Cursor - cut**: use cursor to define a stretchable box on the image to cut
- **Numerical - cut**: numerical input of the coordinates X-min, X-max, Y-min, and Y-max for image cut

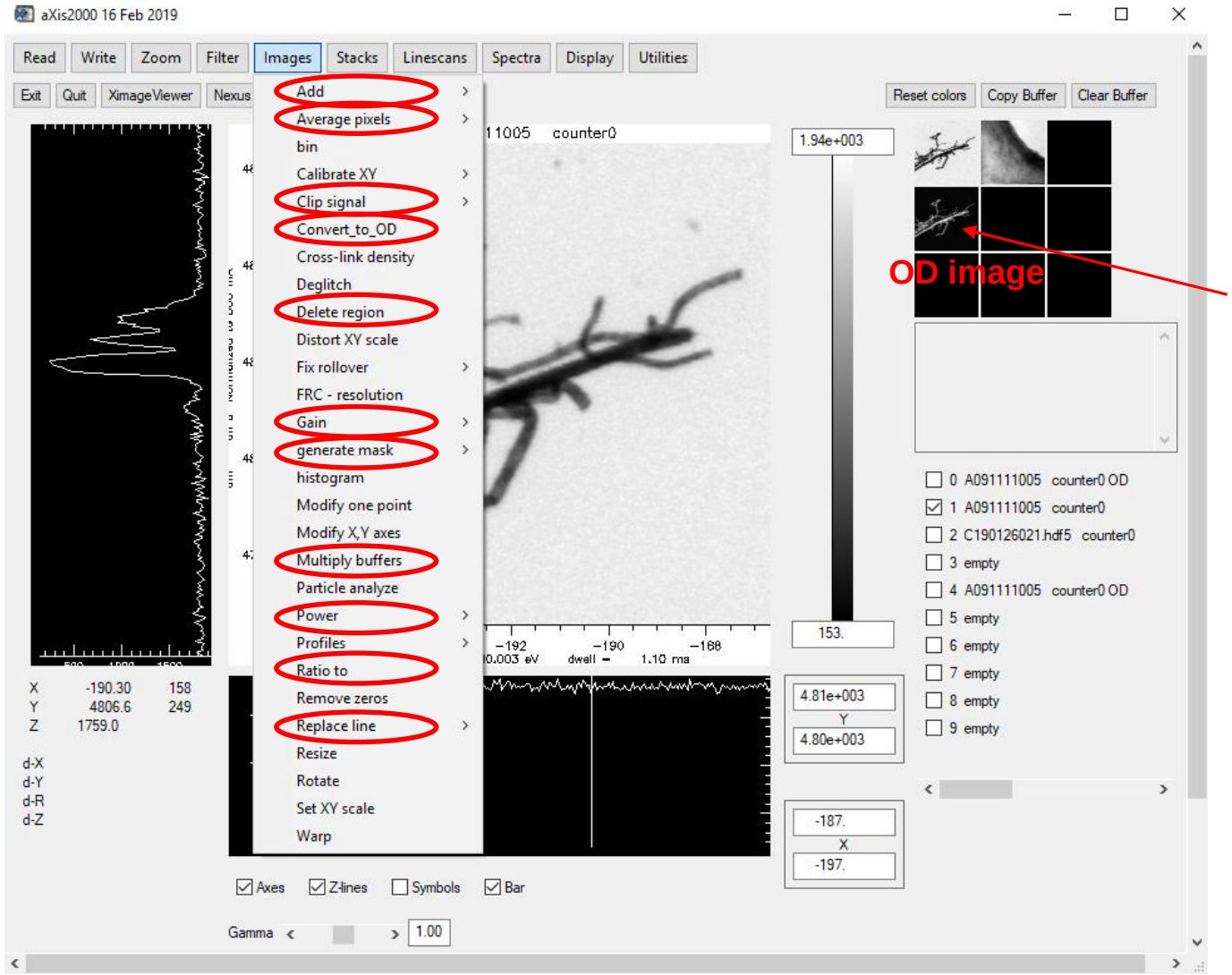
aXis2000 – Image Filter



Some Useful Functions

- **Smooth**: Boxcar average over n-points
- **Median**: n-point Savitsky-Golay averaging
- **Clean (FT image filter)**: 2d-FT filter. The FT is displayed on a 1:1 pixel format.

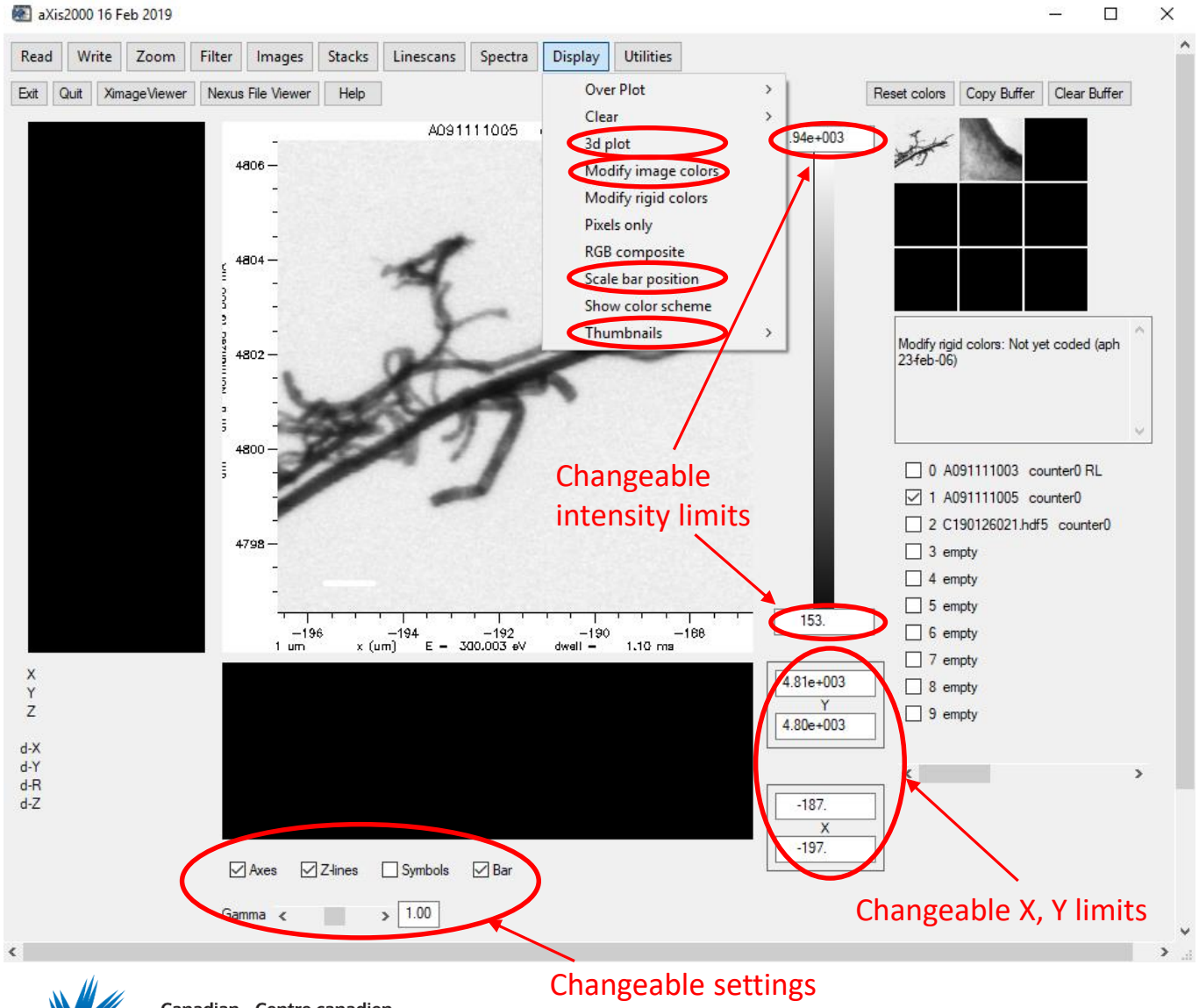
aXis2000 – Image Processing



Some Useful Functions

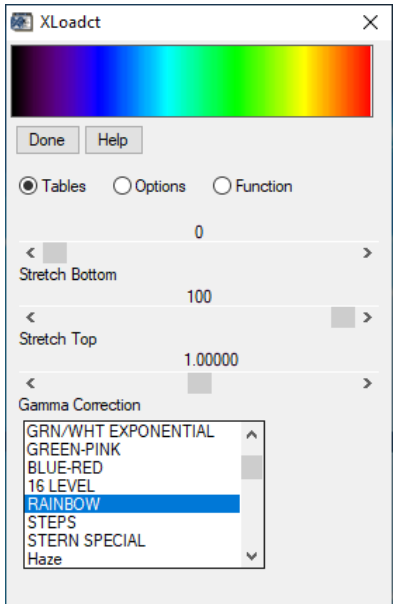
- **Add**: add/append image or constant
- **Average pixels**: average image or region pixels
- **Clip signal**: two clicks to select image intensity range
- **Convert_to_OD**: normalize to the maximum intensity value of the image
- **Delete region**: select an image region and replace the intensity value by arbitrary number
- **Gain**: multiply or divide the image intensity by a number
- **Generate mask**: threshold the image intensity as 0 and 1
- **Multiply buffers**: multiple two images
- **Power**: power the image intensity
- **Ratio to**: divided by another image
- **Replace line**: remove bad/black lines

aXis2000 – Image Display



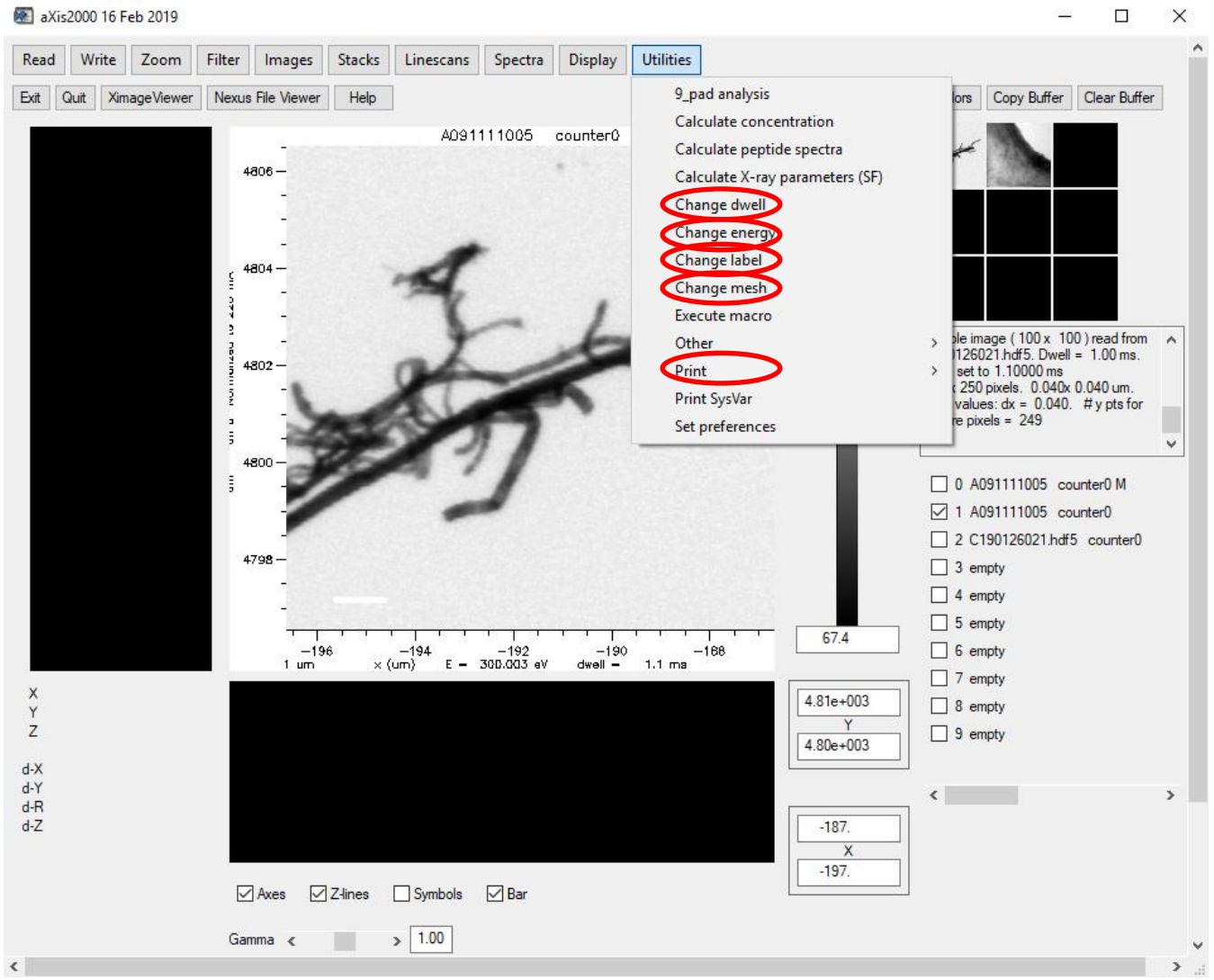
Some Useful Functions

- **3d plot**: X, Y and Intensity 3d image plot
- **Modify image colors**: change image colors



- **Scale bar position**: click on image to manually place the scale bar
- **Thumbnails**: display multiple images together

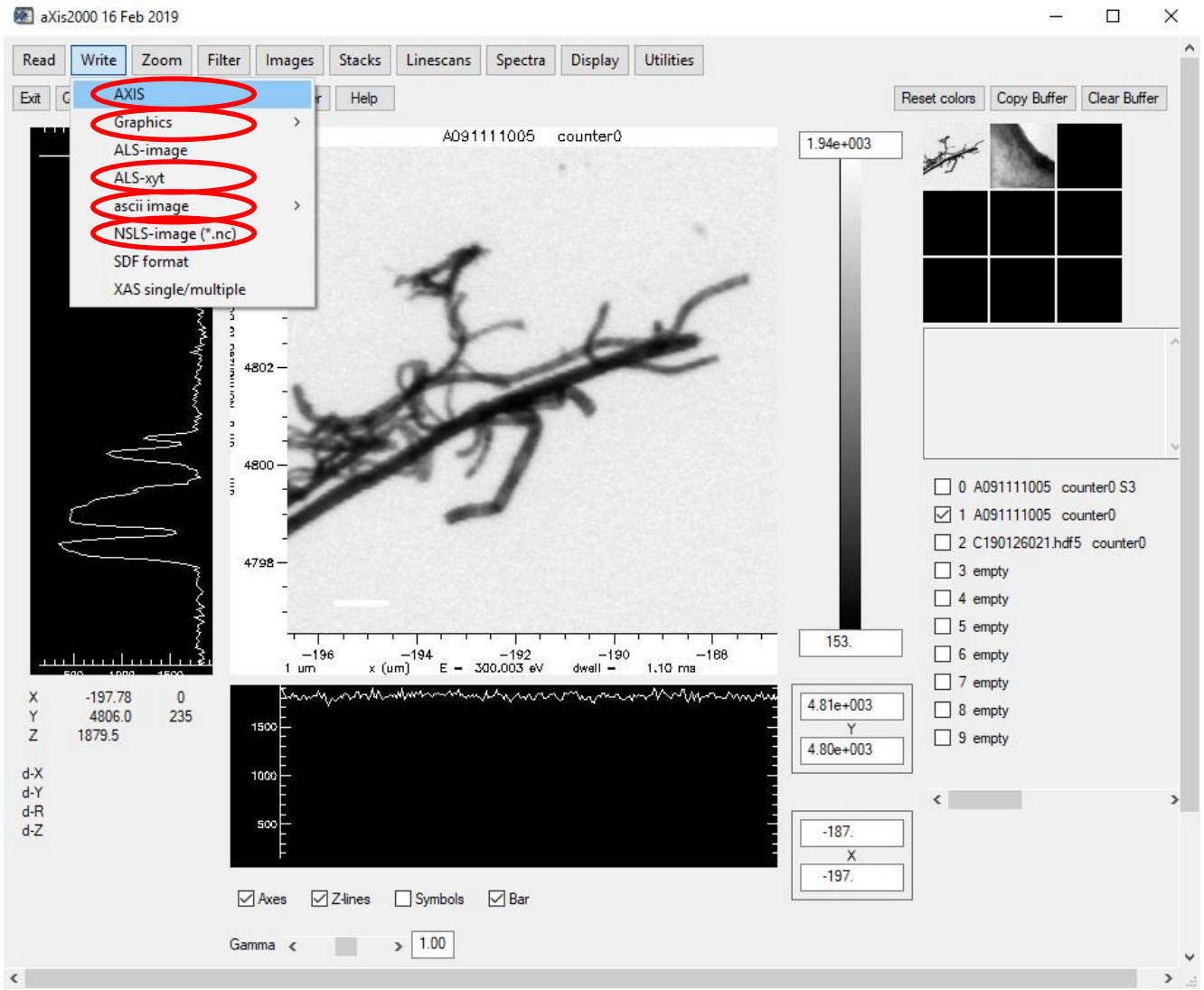
aXis2000 – Image Utilities



Some Useful Functions

- **Change dwell**: change image dwell time
- **Change energy**: change image photon energy
- **Change label**: change label for image or other files like spectra
- **Change mesh**: change image X and Y pixel number
- **Print**: print image or other files like spectra with or without IDL annotation

aXis2000 – Write Images

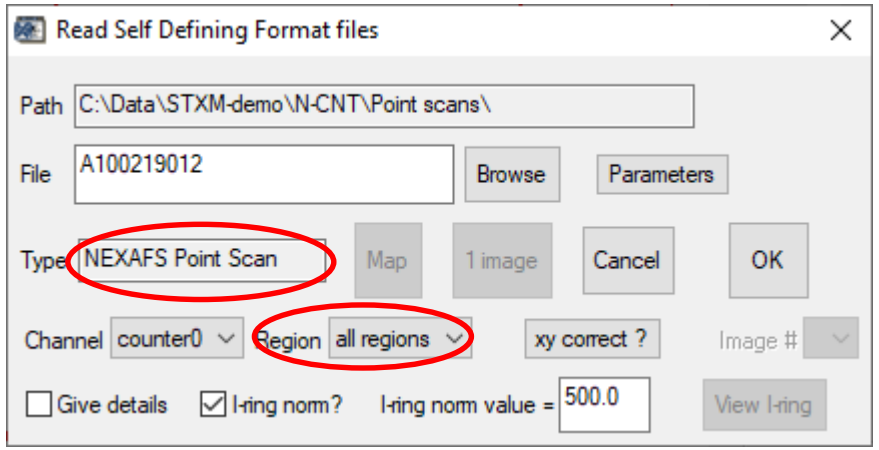


Some Useful Functions

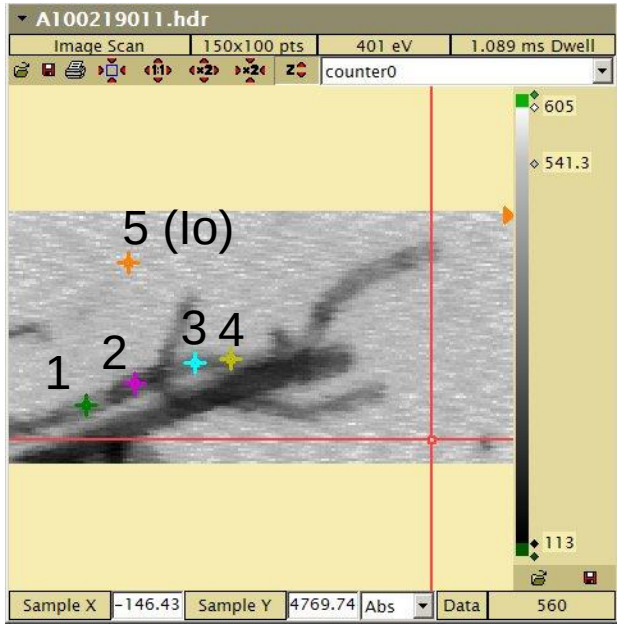
- **AXIS**: images written into *.axb binary format
- **Graphics**: images written into graphics formats, such as GIF, JPG, PNG and TIF, with and without axes or labels
- **ALS-xyt**: images written into X, Y, intensity, X-pixel, and Y-pixel
- **ascii image**: images written into coordinates, and 2d array of image intensity
- **NSLS-image (*.nc)**: images written into NSLS netCDF image format (*.nc)

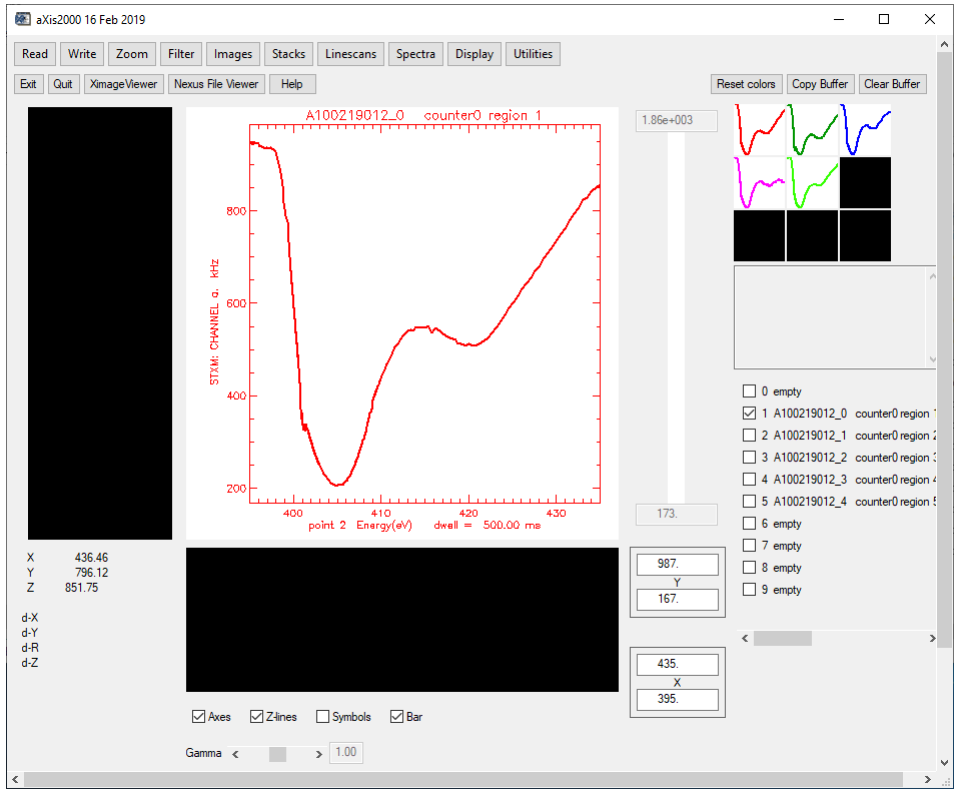
Spectra

aXis2000 – Read Ambient-STXM Point Spectra



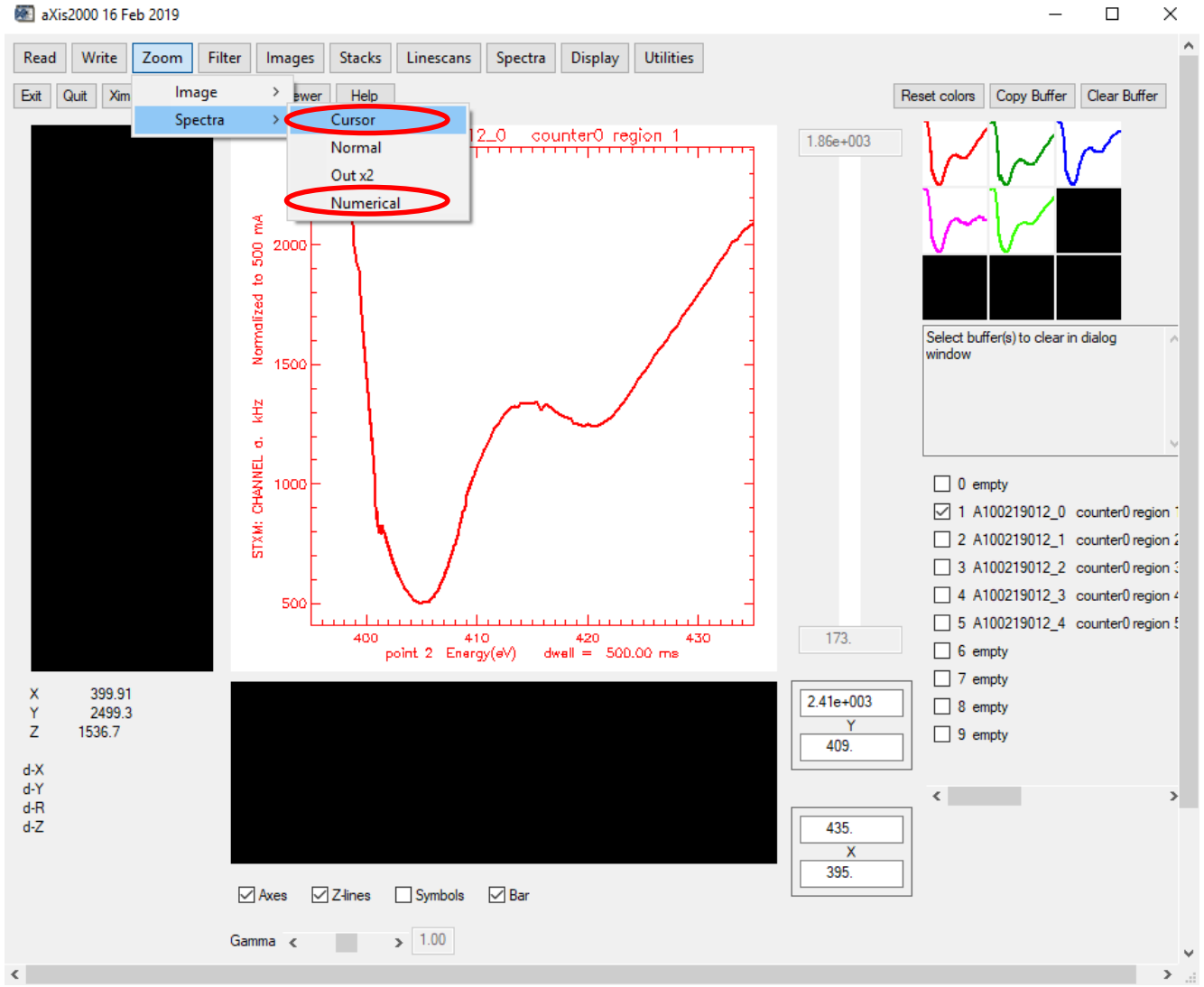
A100219012–point scan–500 ms





- **First click on spectrum**: select the starting point
- **Second click on spectrum**: select the ending point, then calculate the d-X, d-Y, and d-Z of the two points
- **Third click on spectrum**: clear the selected points

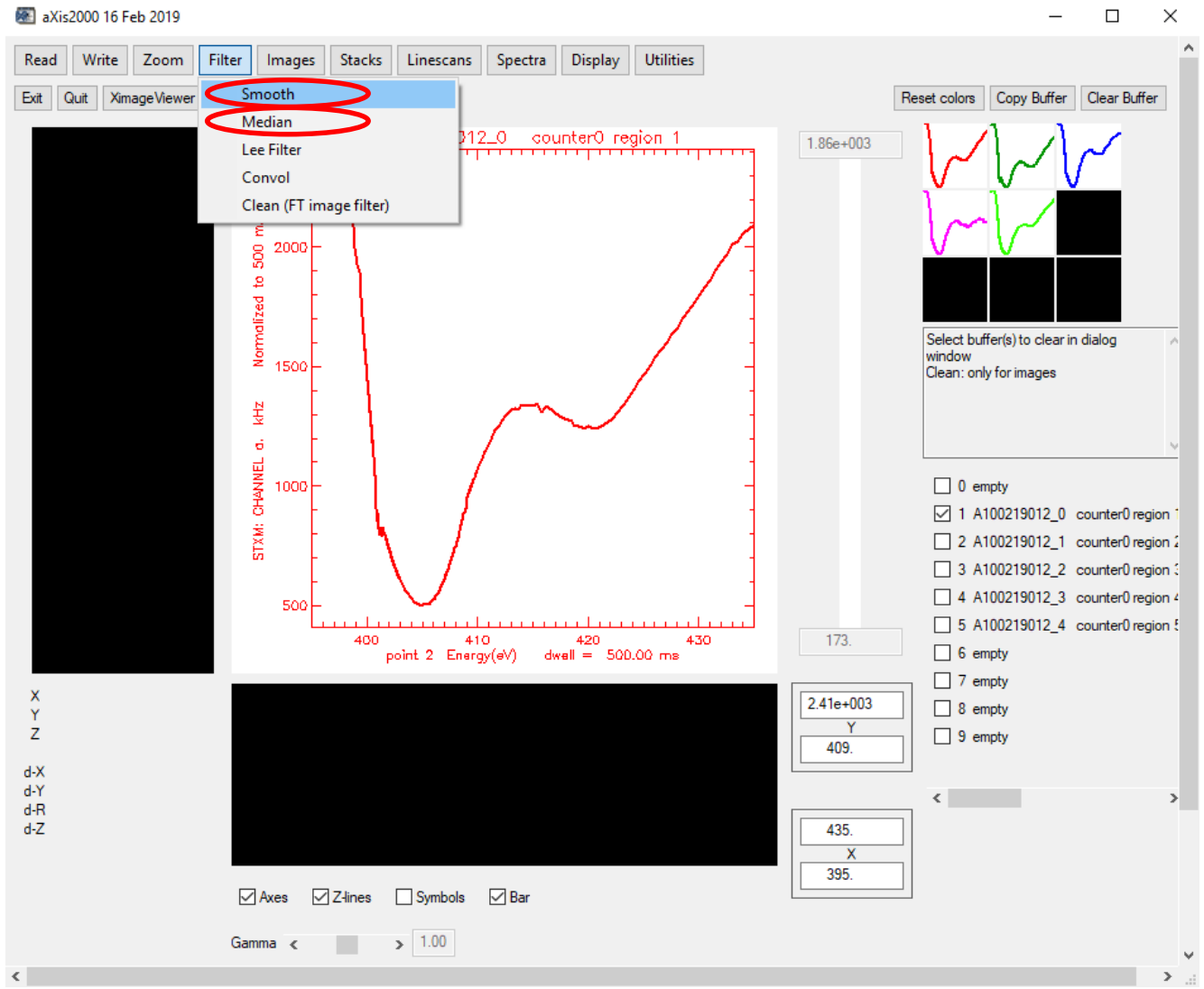
aXis2000 – Spectrum Zoom



Some Useful Functions

- **Cursor**: use cursor to define a stretchable box on the spectrum to cut
- **Numerical**: numerical input of the limits of X-min, X-max, Y-min, and Y-max for spectrum cut

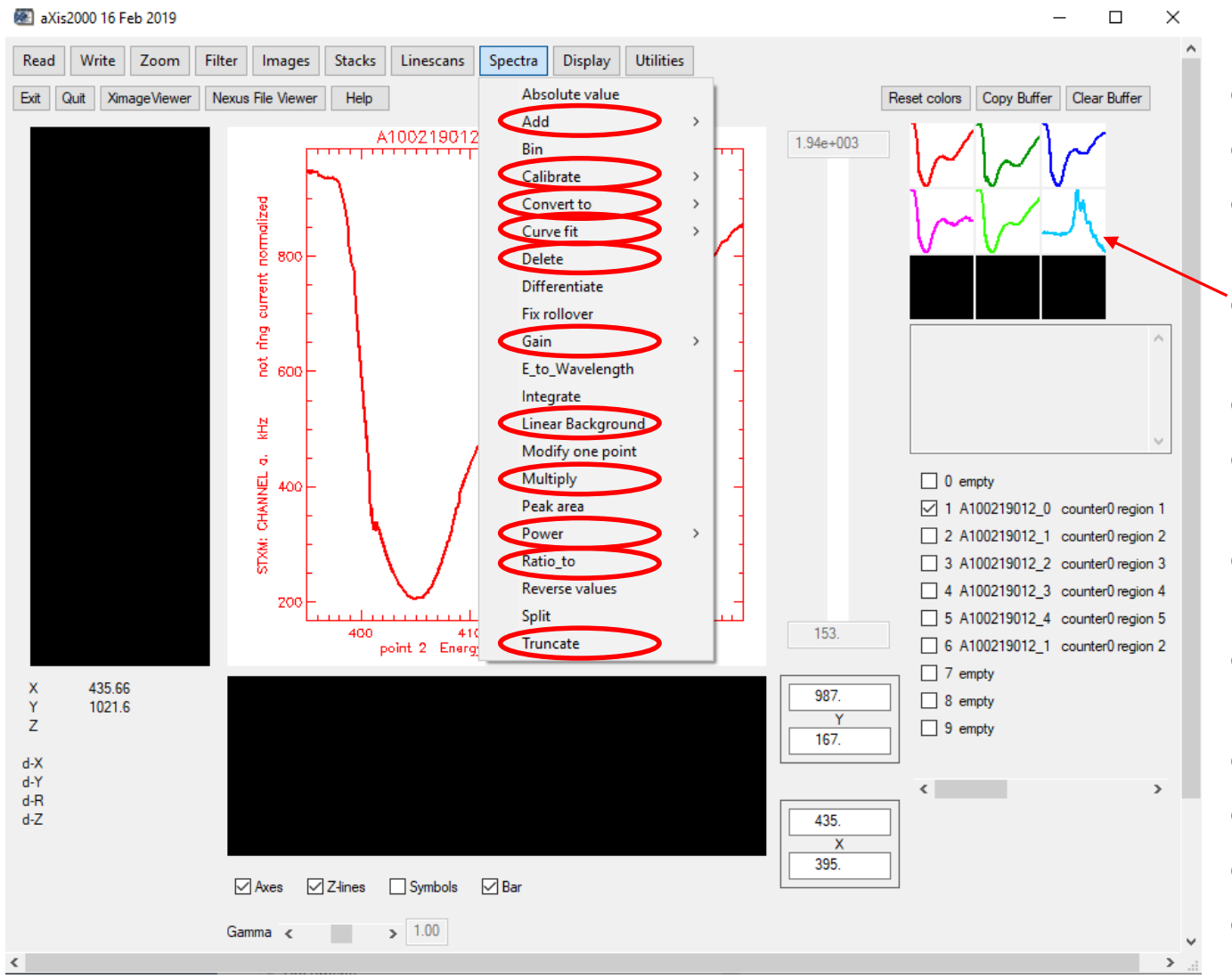
aXis2000 – Spectrum Filter



Some Useful Functions

- **Smooth**: Boxcar average over n-points
- **Median**: n-point Savitsky-Golay averaging

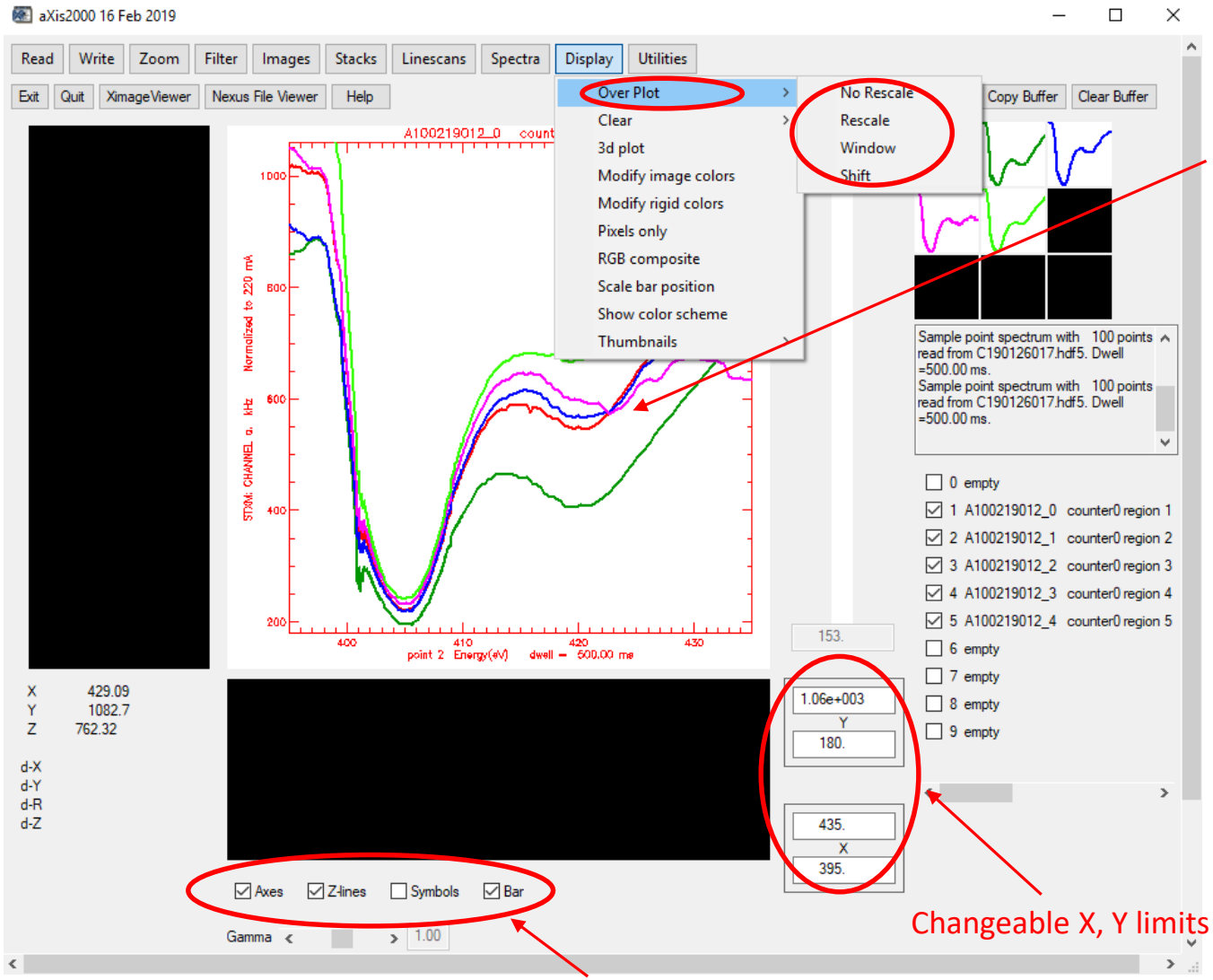
aXis2000 – Spectrum Processing



Some Useful Functions

- **Add**: add/append spectrum or constant
- **Calibrate**: calibrate X or Y scale
- **Clip signal**: two clicks to select image intensity range
- **Convert to**: normalize to an Io spectrum to get OD spectrum
- **Curve fit**: linear combination curve fit and other
- **Delete**: delete the region between two selected points
- **Gain**: multiply or divide the Y scale (intensity) by a number
- **Linear Background**: select two points on spectrum for background baseline
- **Multiply**: multiple two spectra
- **Power**: power the spectrum intensity
- **Ratio to**: divided by another spectrum
- **Truncate**: truncate the spectral regions beyond the two selected points

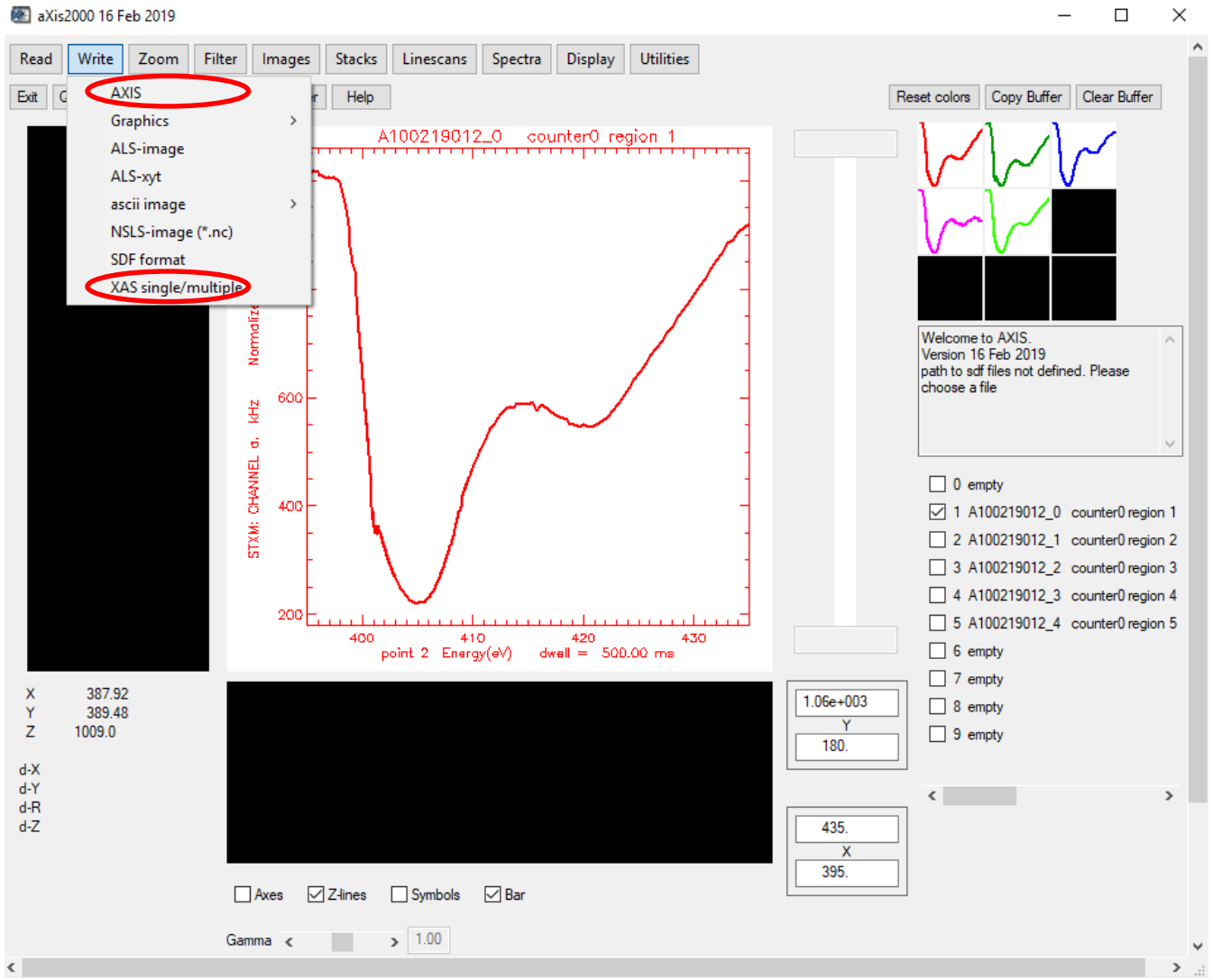
aXis2000 – Spectrum Display



Some Useful Functions

- **Over plot**: display multiple spectra on the same plot based on **No Rescale**, **Rescale**, **Window**, and **Shift**

aXis2000 – Write Spectra

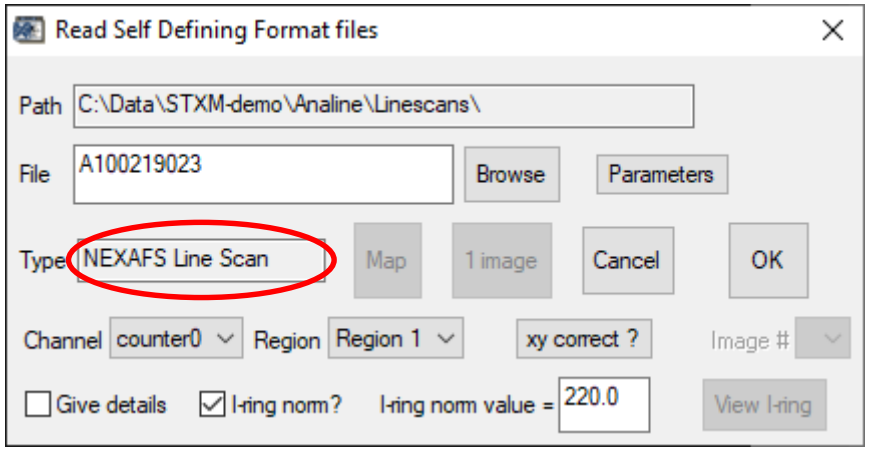


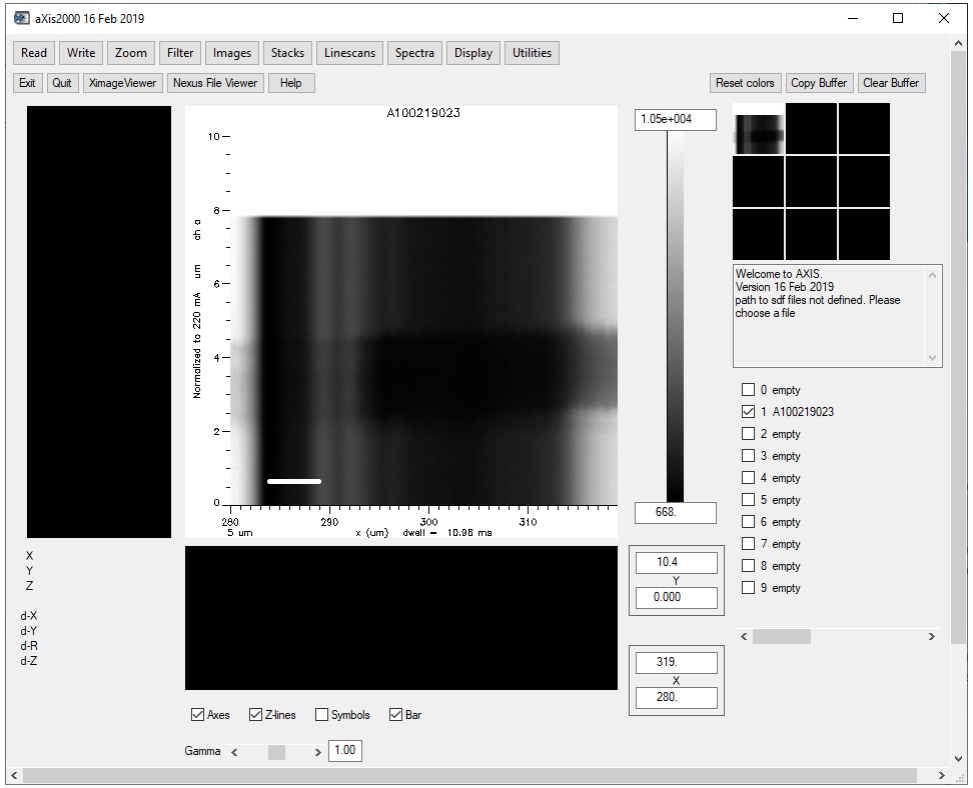
Some Useful Functions

- **AXIS**: spectra written into *.txt ASCII format
- **XAS single/multiple**: images written into NSLS XAS format (*.xas)

Linescans

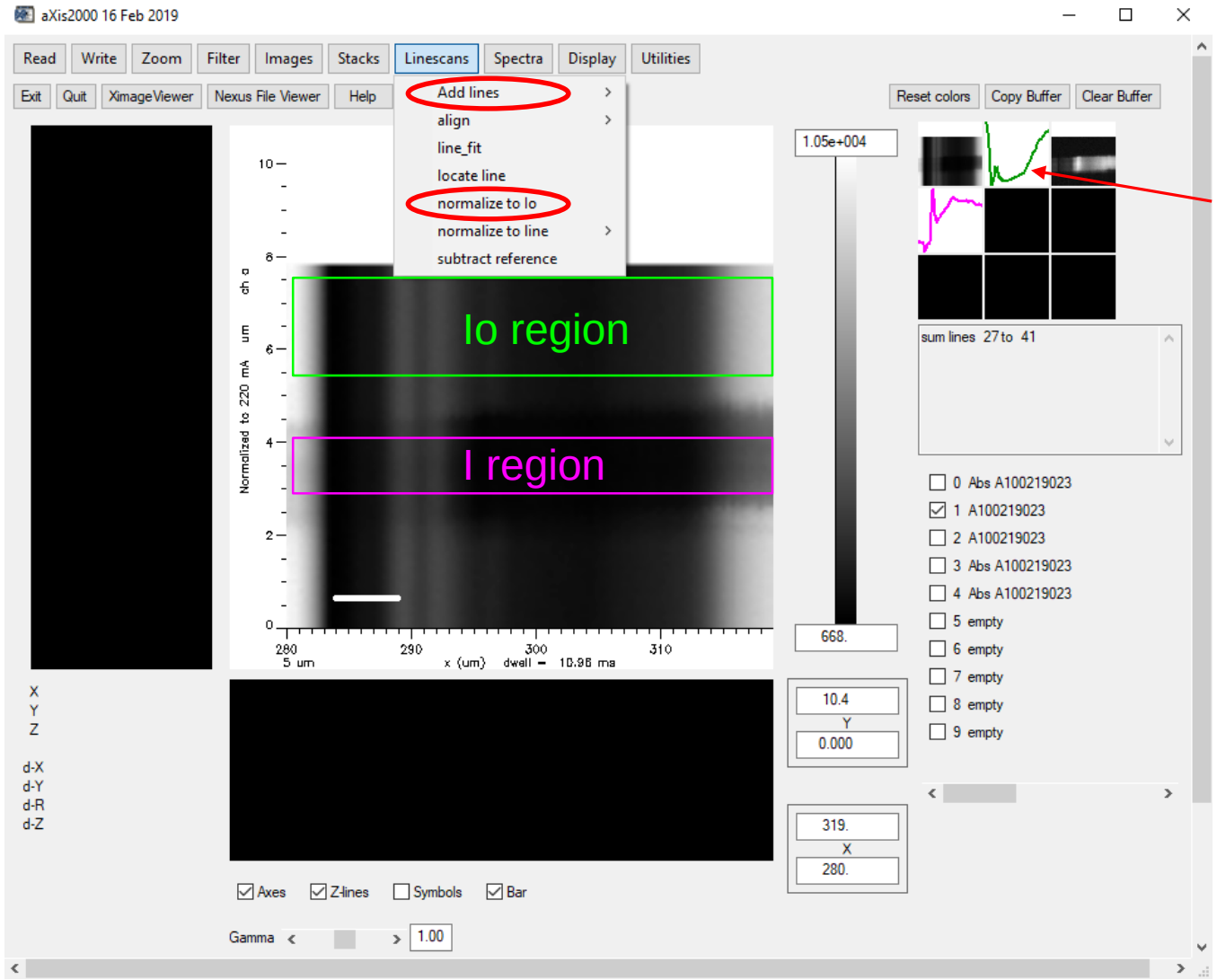
aXis2000 – Read Ambient-STXM Linescans





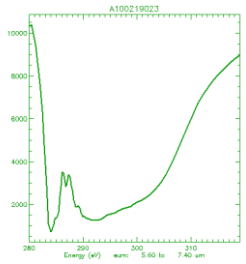
- **First click on spectrum**: select the starting point
- **Second click on spectrum**: select the ending point, then calculate the d-X, d-Y, and d-Z of the two points
- **Third click on spectrum**: clear the selected points

aXis2000 – Linescans Processing

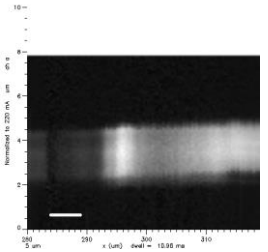


Some Useful Functions

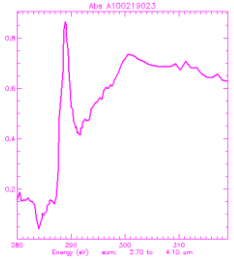
- **Add lines**: define two lines horizontally and average the intensity between the two lines
- **Normalize to Io**: normalize the linescan specto-image to the Io spectrum



Io Spectrum



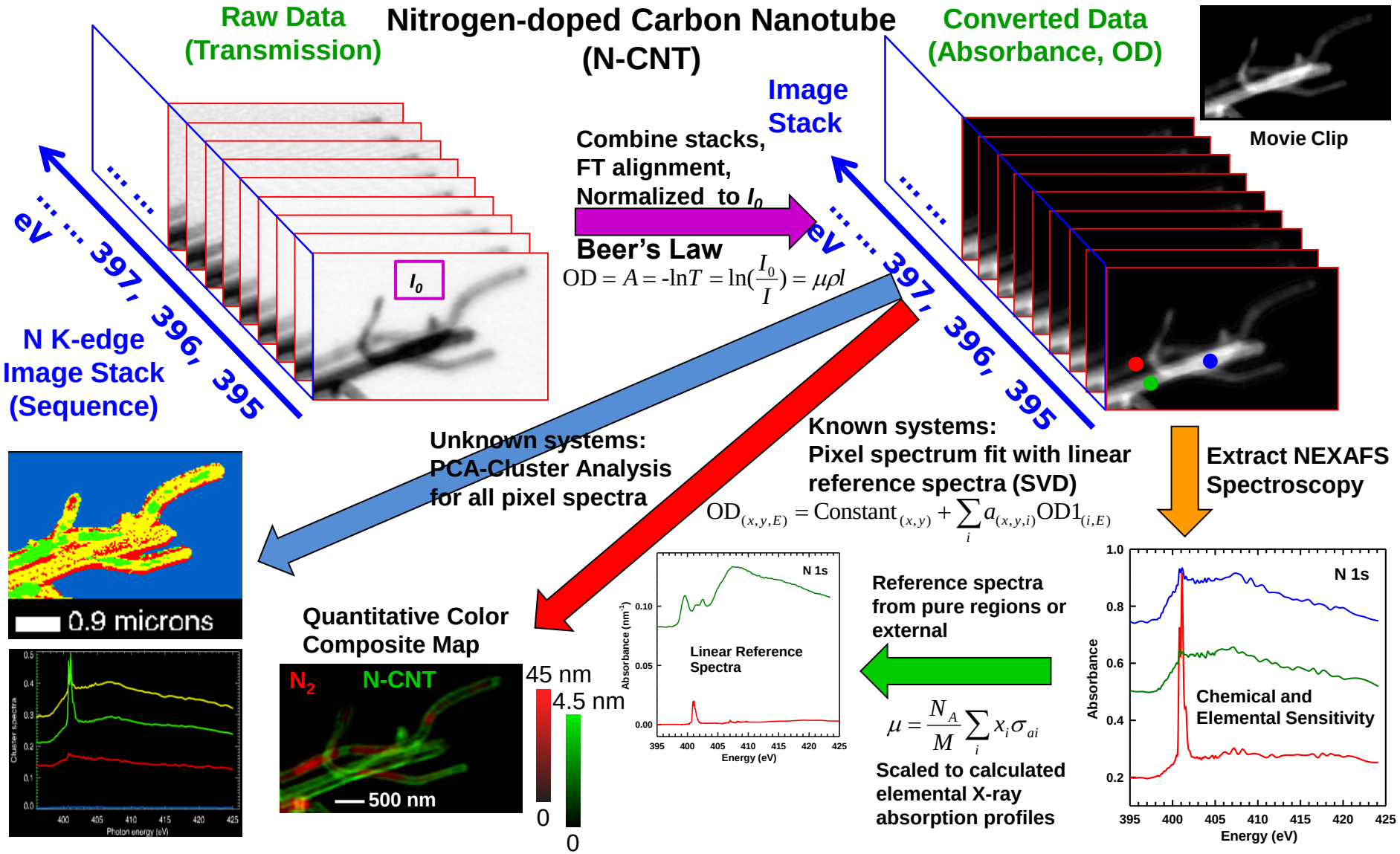
Linescan
OD image



OD
Spectrum

Stacks

STXM Stacks Data Processing

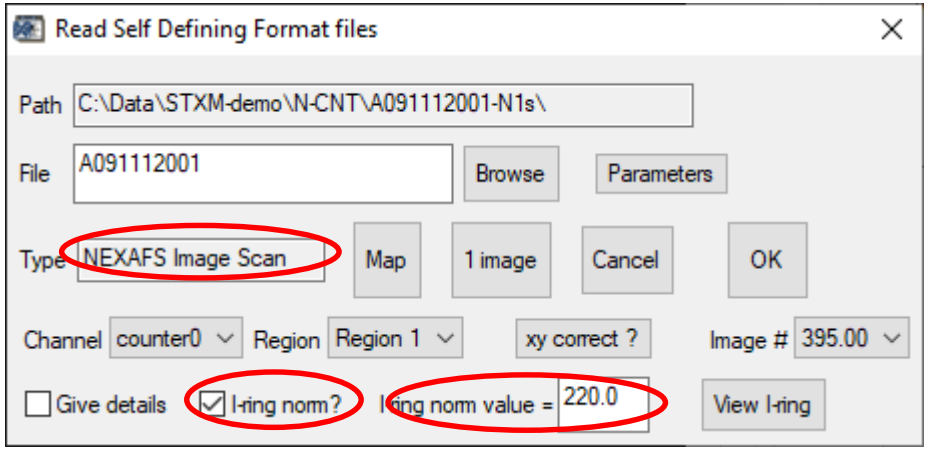


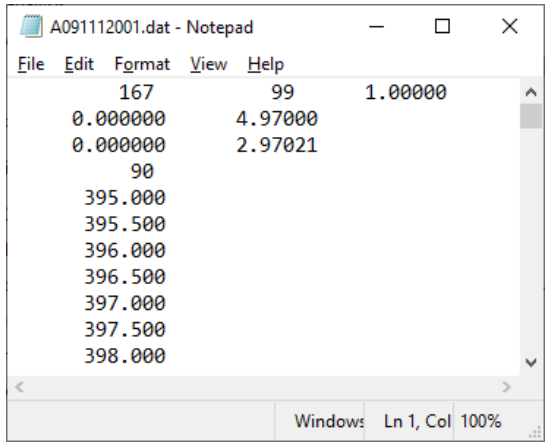
J. Zhou, J. Wang et al. *J. Phys. Chem. Lett.* 1 (2010) 1709.

STXM Data Analysis File Naming Symbols and Rules

Symbols	Description	Examples
A	designate letter for CLS Ambient-STXM data	A161216053.ncb
C	designate letter for CLS Cryo-STXM data	C160927034.ncb
only file number	raw transmission data	A161216053.ncb
a	aligned stack	A161216053a.ncb
aod	aligned stack, then converted to od stack	A161216053aod.ncb
avg	averaged stack image or spectrum	A161216053aod-avg.ncb
c, cali	energy calibrated	A161216053aodc.ncb, or A161216053caod.ncb
cl, clip	clipped, cleaned	A161216053aodc-clip.ncb
ls, l	linescan	lsA161216052.txt
m	remeshed	A161216053aodm.ncb
map	chemical map	A161216051aod-map.ncb
mask	stack mask	A161216053aod-mask.ncb
n	normalized	A161216053aodn.ncb
od	optical density	A161216050od.axb
od1	optical density per nm	A161216053od1-FeO.txt
oda	optical density stack, then aligned	A161216053oda.ncb
p	part of the stack	A161216053aodp.ncb
sm	smoothed spectrum, image	A161216056od-sm-FeO.txt
sf	elemental absorption profile	FeO-sf-od1.txt
t	truncated spectum, stack, clipped stack	A161216053aodt.ncb
x	point scan	xA161216055od.txt
Original File Types	**Derived File Types**	**Naming Rules**
Images	Image	A/C + date + sequence number + symbols (+ sample)
Point Scans	Spectrum	x + A/C + date + sequence number + symbols (+ sample)
Linescans	Spectrum	ls + A/C + date + sequence number + symbols (+ sample)
	Spectrum-Image	ls + A/C + date + sequence number + symbols (+ sample)
Stacks	Stack	(A/C + date +) sequence number + symbols (+ sample)
	Image	(A/C + date +) sequence number + symbols + sample
	Map	(A/C + date +) sequence number + symbols + sample
	Spectrum	(A/C + date +) sequence number + symbols + sample

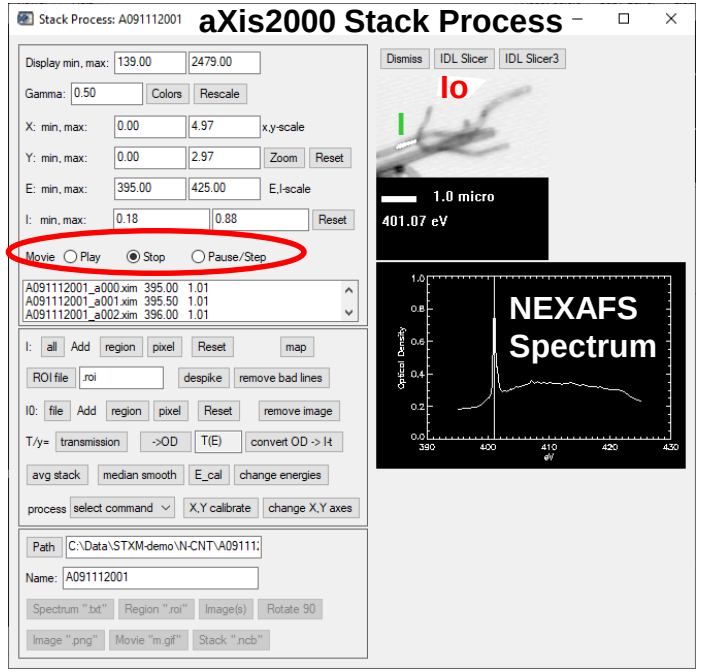
aXis2000 – Read Ambient-STXM Stacks



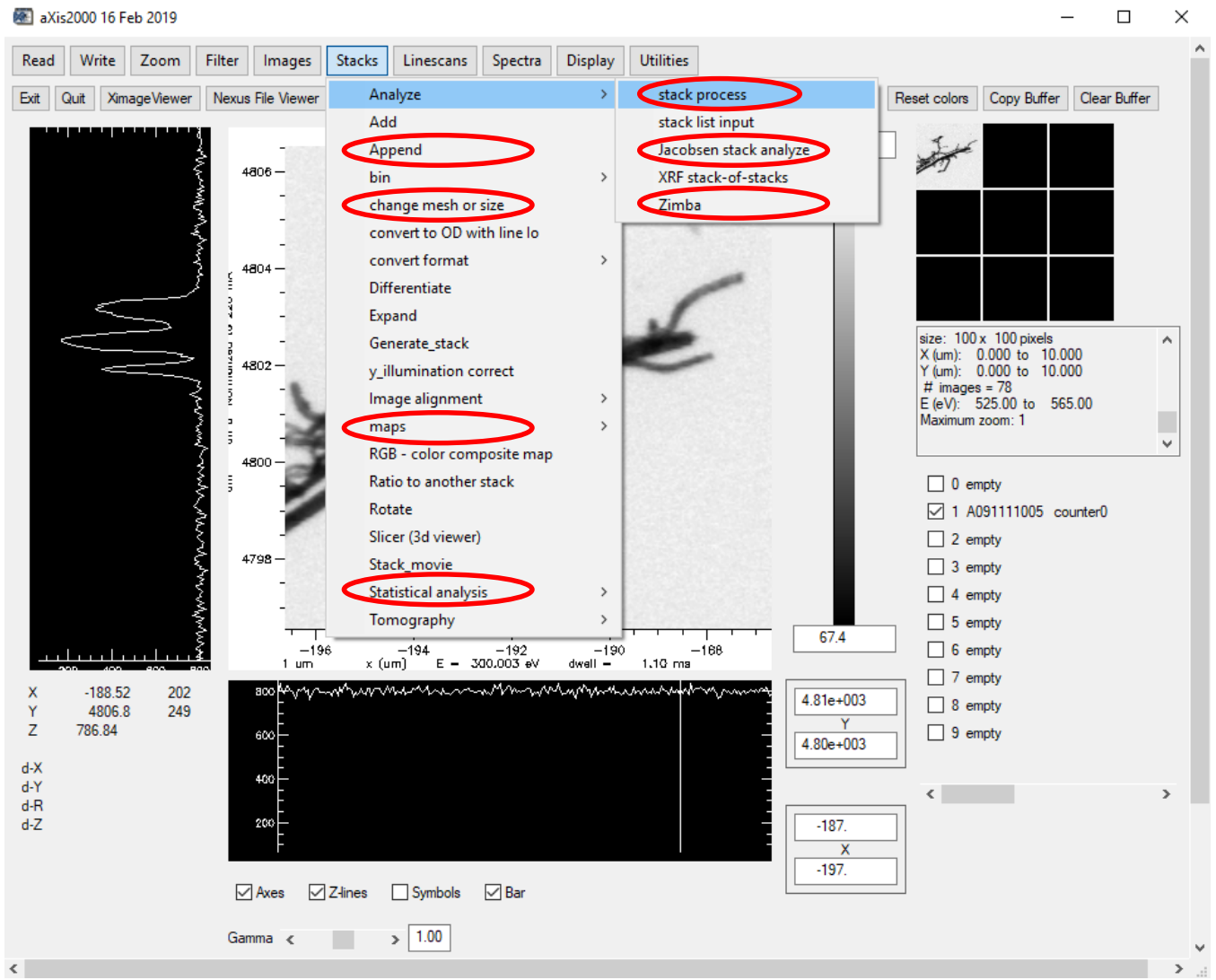


A091111001.dat

- Compile raw Ambient-STXM stack data: aXis2000 → Read → STXM (sdf)
- Give a stack file name, then generate two files:
  - One *.dat file that has stack dimensions, pixel numbers, and photon energies
  - One *.ncb binary stack data file
- Uncheck “I-ring norm?” if ring current information is missing or wrong for some images.
- After stack compiling, the aXis2000 Stack Process will be automatically launched with inputting a ”suggested zoom” number
- Play movie in “Stack Process” to check if the stack is complete in images/energies.



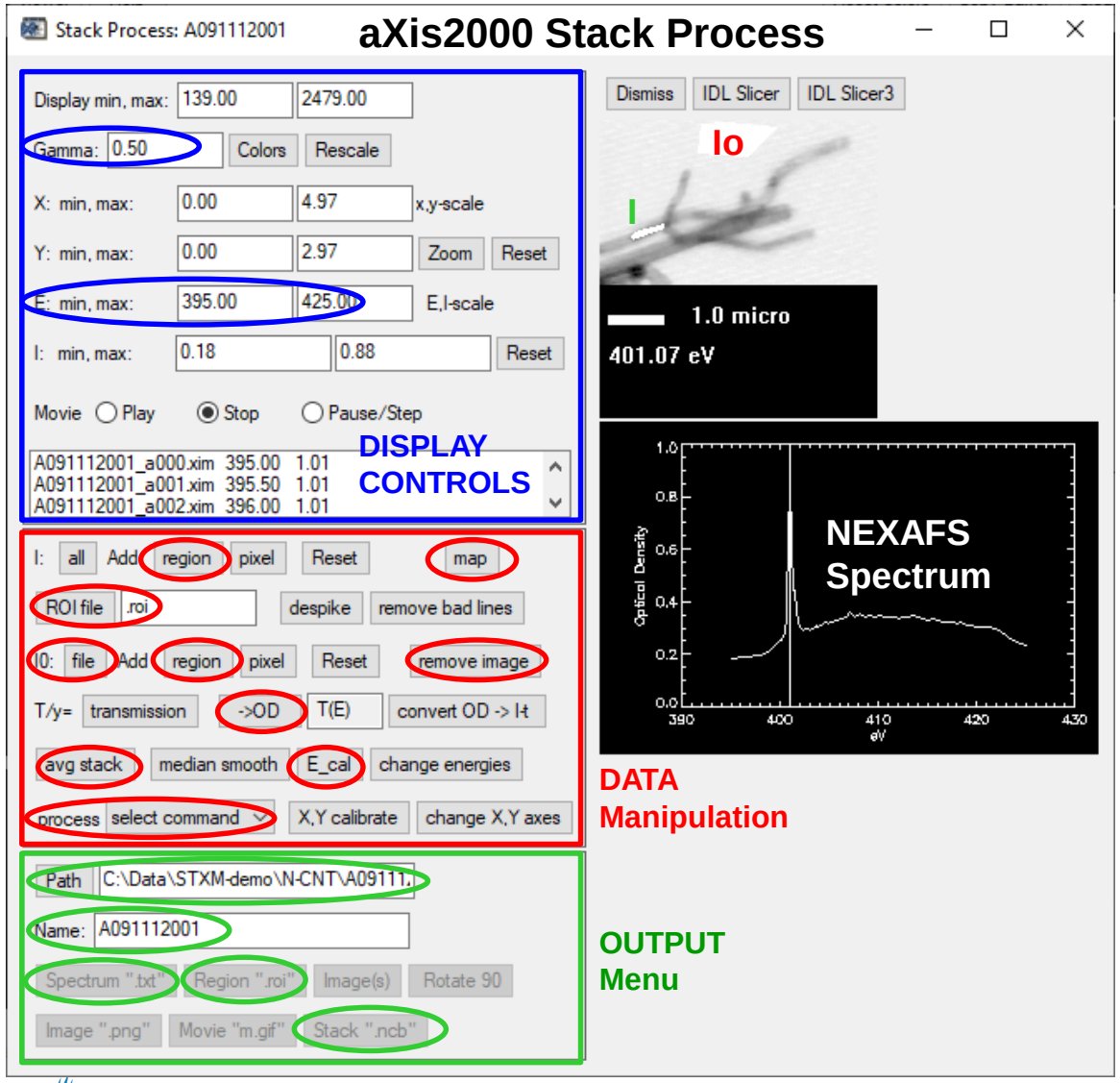
aXis2000 – Stacks



Some Useful Functions

- **Stack process**: stack data display, manipulation, and output
- **Jacobsen stack analyze**: stack data display, alignment, manipulation, and output
- **Zimba**: stack data build, display, alignment, manipulation, and output
- **Append**: append two stacks with the same pixel and physical dimensions
- **Change mesh or size**: change stack pixel number or physical size
- **Maps**: perform fitting like SVD or Stack fit for the stack, and other fittings
- **Statistical analysis**: PCA analysis

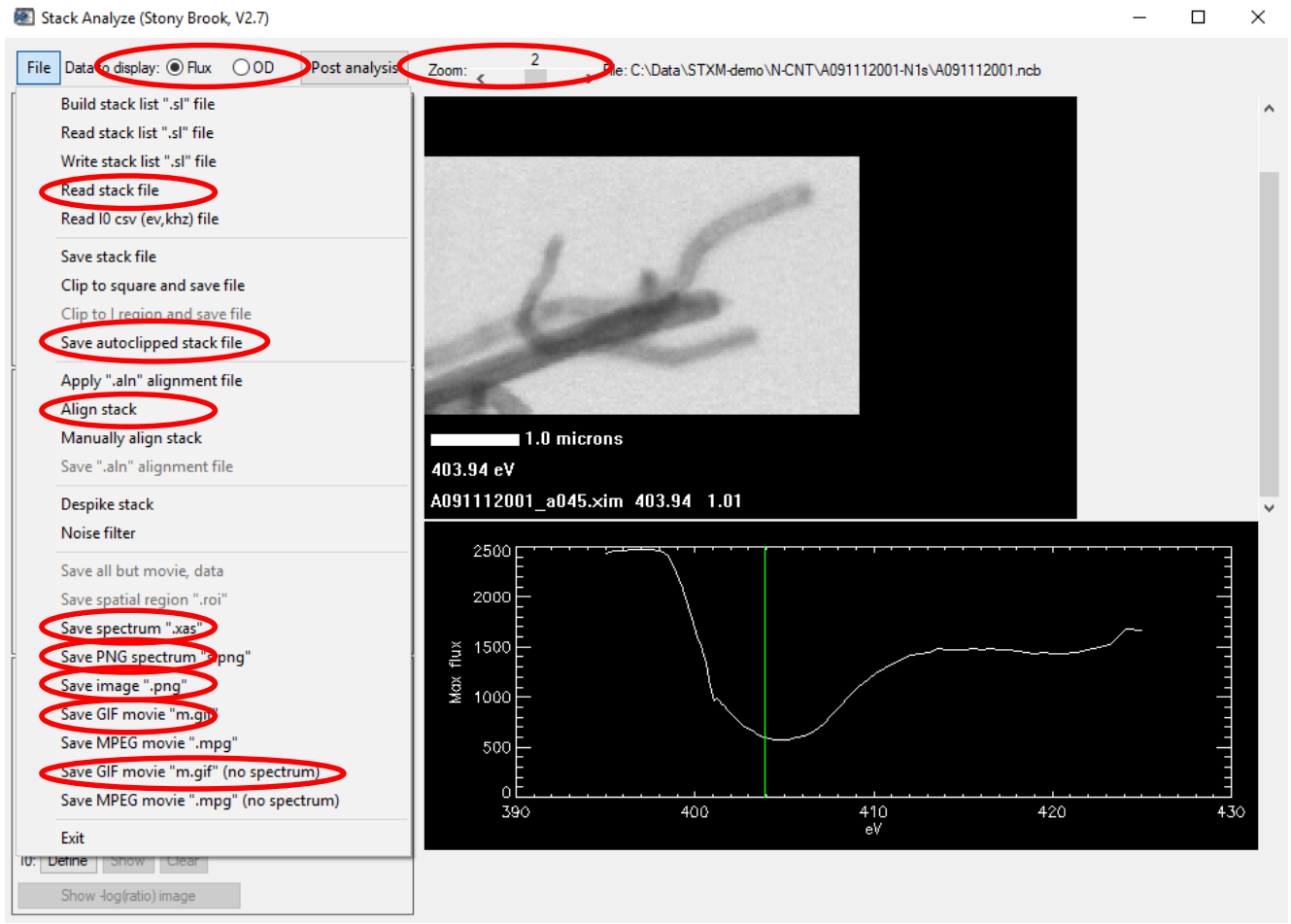
aXis2000 – Stack Process



Some Useful Functions

- **Gamma**: change image Gamma value
- **E**: define stack energy range
- **I: region**: select sample region
- **I: ROI file**: load a sample ROI region file
- **I: map**: generate a two image subtraction map
- **I0: file**: load an Io file
- **I0: region**: select an Io region
- **I0: remove image**: delete an image of stack
- **->OD**: convert to OD stack
- **Avg stack**: average all stack images
- **Process**: multiple math process for stack
- **Path**: saving path, no space in folder names
- **Name**: file name for output, click “enter” to activate the output menu buttons
- **Spectrum, Region, Stack**: commonly saved file types

Stack Analyze (Stony Brook, V2.7)



Some Useful Functions

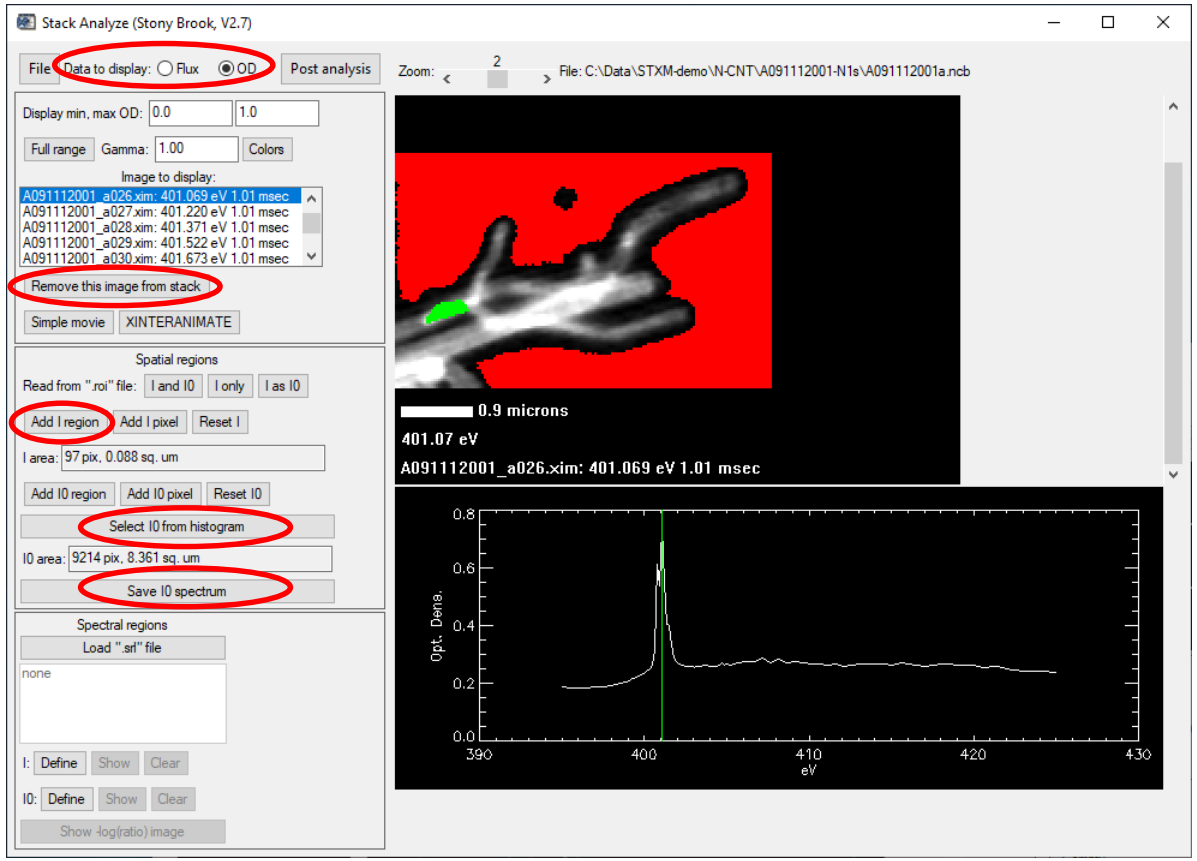
- **Flux/OD**: display in flux (transmission) or OD (absorption) format
- **Zoom**: select the zoom factor for the stack images
- **Read stack file**: read a stack and choose *.ncb format
- **Save autoclipped stack file**: after iterations of alignment, save autoclipped stack file before select Io region.
- **Align stack**: launch the “Stack Align” widget, then process stack alignment
- Save spectrum, image, and movie in formats except the “mpg” format that requires IDL licence

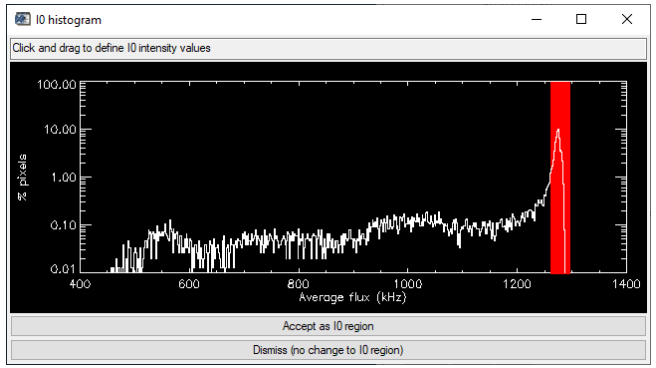
Stack Analyze (Stony Brook, V2.7) – Stack Align

Some Useful Functions

- **Edge enhance before aligning**: choose "None" for most cases unless the image contrast is very poor.
- **Image to align to**: choose "Each preceding image" for most cases; or choose a best contrast image for some cases; use the same reference image for all alignment iterations.
- Use other default settings unless the alignment is very challenging.
- **Save new alignment; go to stack_analyze**: after each alignment, save the alignment back to the stack without a separate alignment file.
- Align the stack until the X-Shift and Y-Shift are zero and fully overlapped, or smaller than 1 pixel.

Stack Analyze (Stony Brook, V2.7)





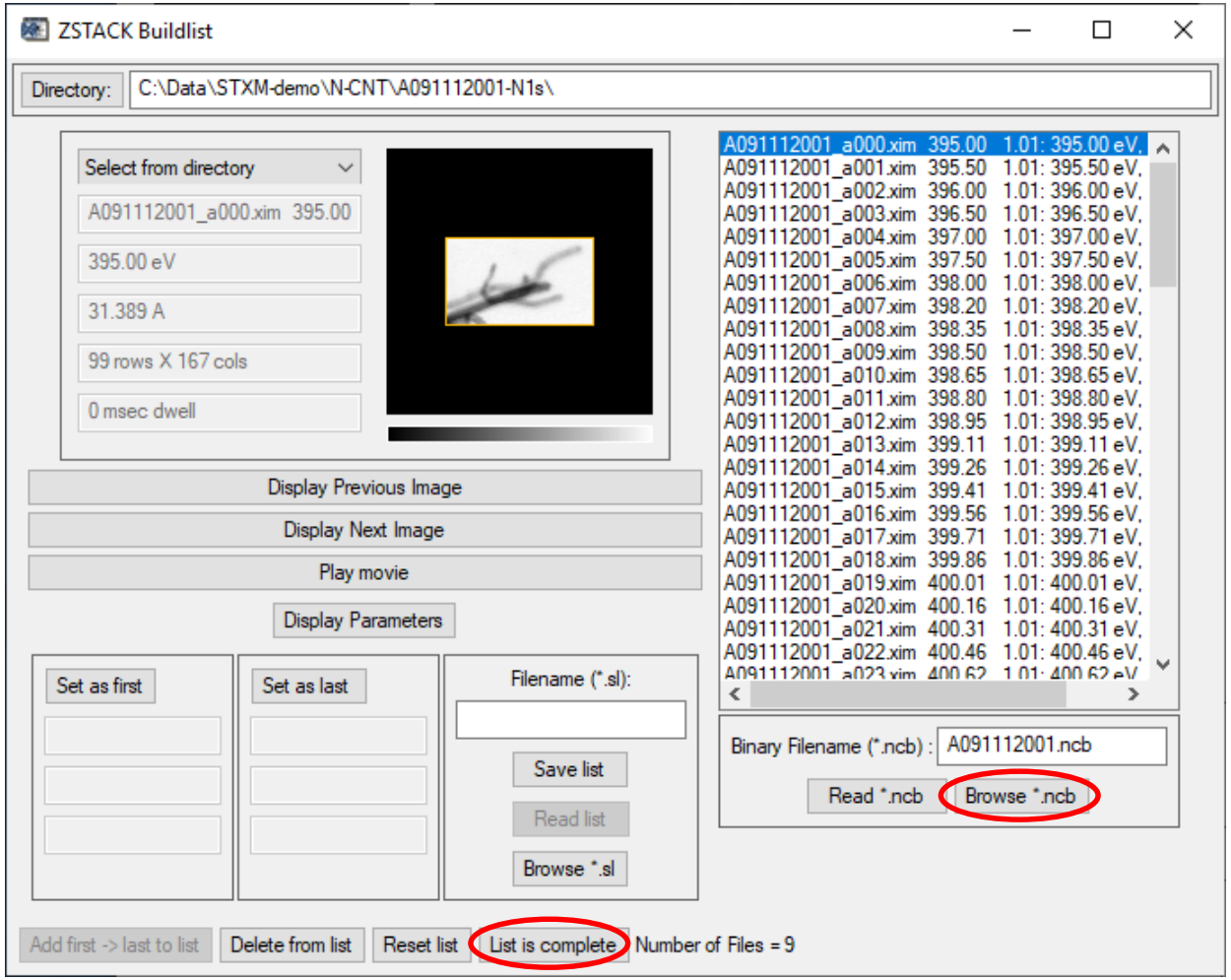
Some Useful Functions

- **Flux/OD**: display in flux (transmission) or OD (absorption) format
- **Remove this image from stack**: delete bad images or unwanted images even before stack alignment
- **Add I region**: select a sample region to check NEAXFS spectroscopy
- **Select I0 from histogram**: select the highest intensity peak for Io, as shown in the window above
- **Save I0 spectrum**: save the Io spectrum in *.xas format

stack_analyze.sav → File → Read stack file; File → Align stack → Save autoclipped stack file → Select I0 from histogram (click-hold-drag) **→ Save I0 spectrum**

aXis2000 – Stacks – Zimba

aXis2000 → Stacks → Analyze → Zimba

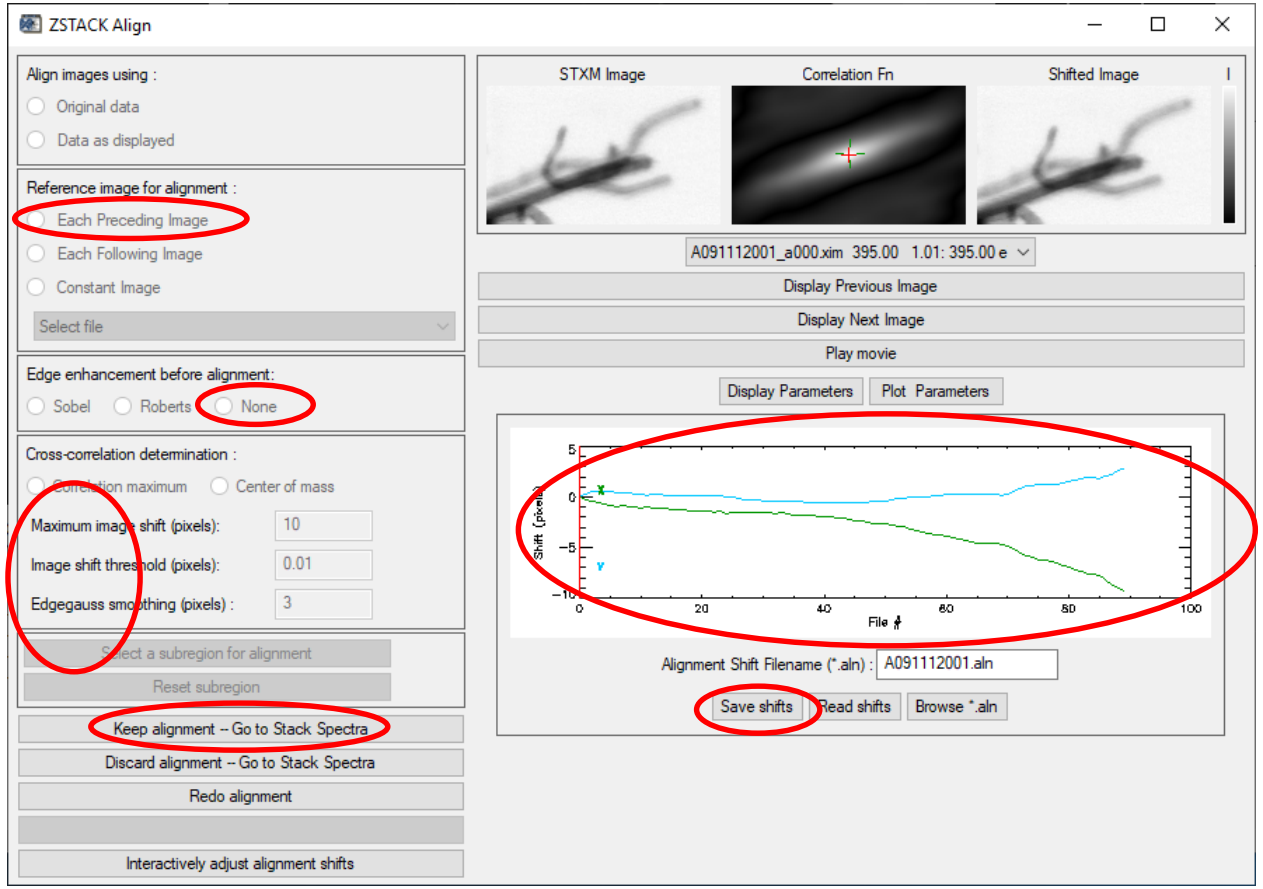


Some Useful Functions

- **Browse *.ncb**: browse and open a *.ncb stack
- **List is complete**: after loading the stack, click “List is complete” to move to the “ZSTACK Align” widget

aXis2000 – Stacks – Zimba – ZSTACK Align

aXis2000 → Stacks → Analyze → Zimba → ZSTACK Align

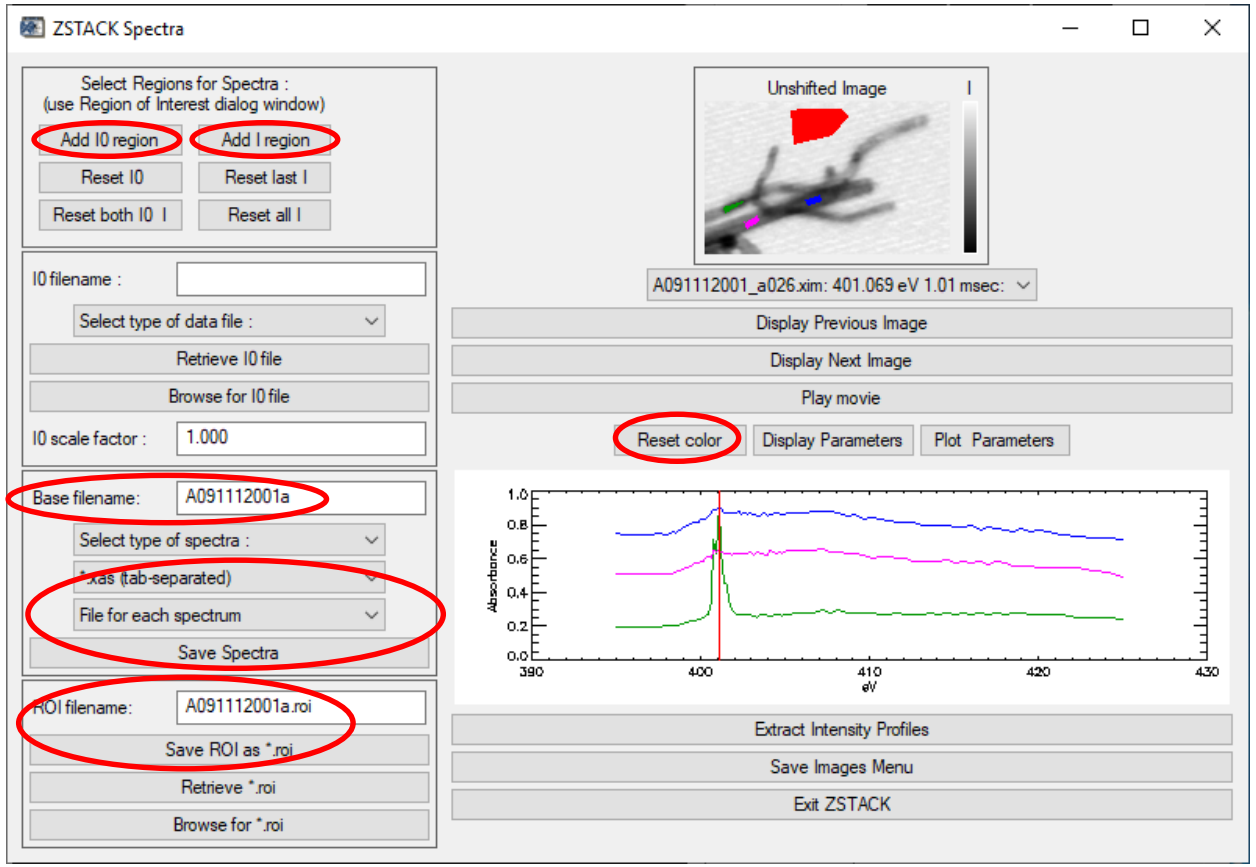


Some Useful Functions

- **Reference image for alignment**: choose “Each Preceding Image” for most cases; or choose a best contrast image for some cases; use the same reference image for all alignment iterations.
- **Edge enhancement before alignment**: choose “None” for most cases unless the image contrast is very poor.
- Use other default settings unless the alignment is very challenging.
- **Start auto-alignment**: you will see this function before alignment
- **Skip alignment**: you will see this function before alignment; click this button if alignment was done by other software
- **Keep alignment – Go to Stack Spectra**: after alignment, move on to “ZSTACK Spectra”
- **Save shifts**: after each alignment, need to save an alignment file *.aln
- Align the stack until the X-Shift and Y-Shift are zero and fully overlapped, or smaller than 1 pixel.

aXis2000 – Stacks – Zimba – ZSTACK Spectra

aXis2000 → Stacks → Analyze → Zimba → ZSTACK Align → ZSTACK Spectra



Some Useful Functions

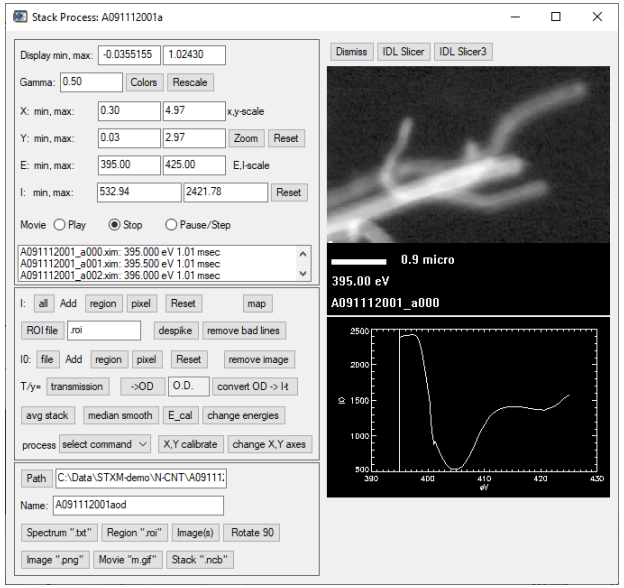
- **Add I0 region**: select an empty region without sample for Io; if the stack is already OD stack, don't need to select an Io region.
- **Add I region**: can select multiple sample regions
- **Base filename**: for convenience, just use stack file name
- Select "**xas (tab-separated)**" and "**File for each spectrum**", then click "**Save Spectra**". Note the saved spectra are transmission spectra including Io spectrum if the stack is transmission stack; and for an OD stack, the saved spectra are OD spectra.
- Give a region of interest (ROI) file name, then click "**Save ROI as *.roi**" to save the ROI file with extension name *.roi.
- **Reset color**: if there is display problem, click "Reset color" button.

aXis2000 – Stack Process – Save OD Stack, Stack Average Image, and Spectra

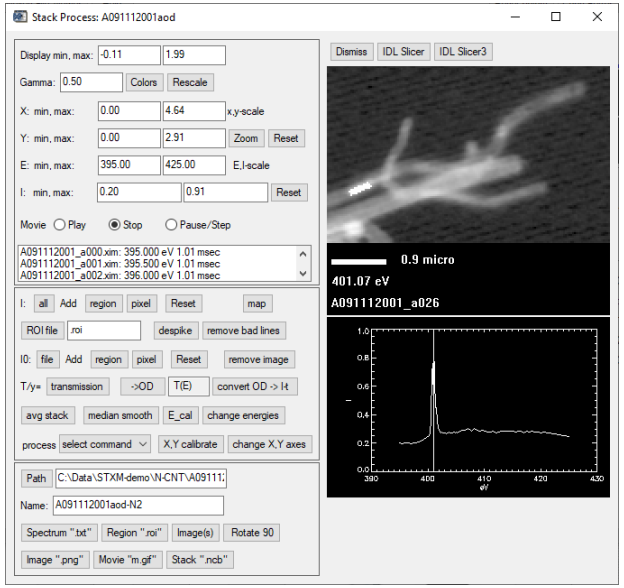
aXis2000 → Stacks → Analyze → stack process → select an aligned transmission stack → click "**No**" for alignment file → **suggested zoom** value → **I0 file** (*.*, then choose *.xas Io file) → **->OD** → save stack

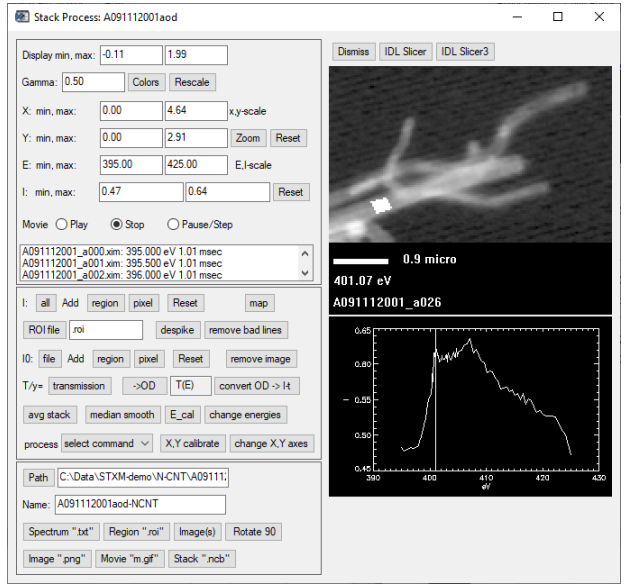
aXis2000 → Stacks → Analyze → stack process → select an aligned OD stack → click "**No**" for alignment file → **suggested zoom** value → **avg stack** → **Store average image in** an empty buffer, then save the image → select "**I: region**" → click draw a sample region on the stack → click "**ACCEPT region**" → give a file name, then hit "enter" → click "**Spectrum ".txt"**" to save the sample NEXAFS spectrum

Save OD stack, average stack image



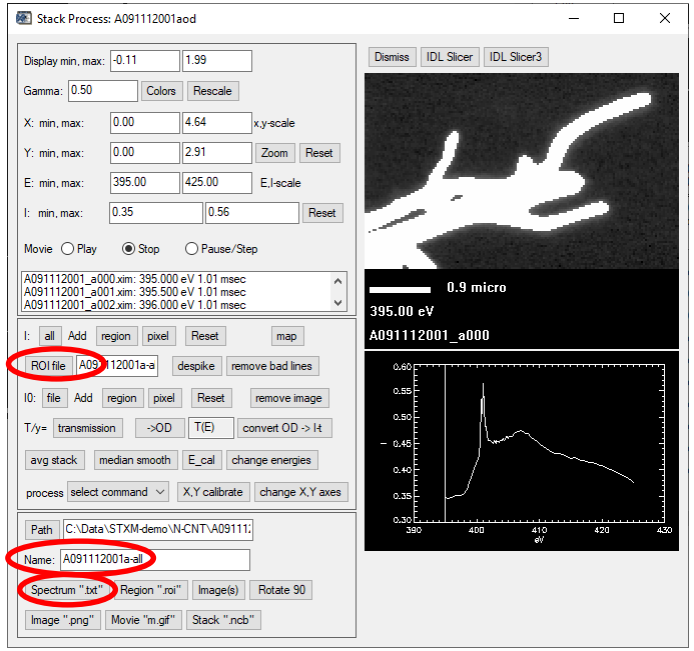
Save Sample Region NEXAFS Spectra





aXis2000 – Stack Average Image – Generate Mask and Use Mask for Spectrum

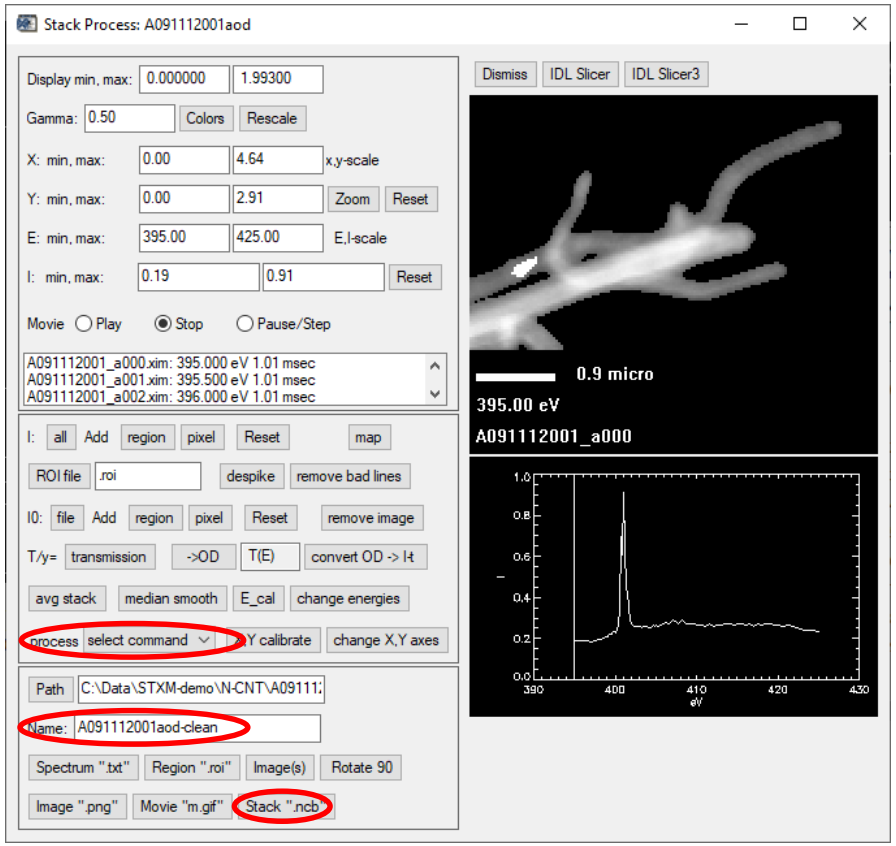
aXis2000 → Read → Images → AXIS → open a stack average OD image → **Images → generate mask → threshold → histogram** → adjust lower/upper limit to select desired sample region → **Save ROI** → (optional) edit mask → **Save mask image → Choose Buffer → Write → AXIS** → save the mask image in *.axb format → **Images → Delete region** → select regions on mask to replace with zero → repeat "**generate mask**" ……



aXis2000 → Stacks → Analyze → stack process → select an aligned OD stack → click "**No**" for alignment file → **suggested zoom** value → select "**I: ROI file**" → load the mask *.roi file → give a file name, then hit "enter" → click "**Spectrum ".txt""** to save the sample NEXAFS spectrum

aXis2000 – Stack Average Image – Generate Mask and Use Mask for Stack

aXis2000 → **Stacks** → **Analyze** → **stack process** → select an aligned OD stack → click "**No**" for alignment file → **suggested zoom** value → **process: select command** → ***image** → load a mask image in *.axb format → **weight by 1** → select an "**I region**" to activate the output menu → give a file name, then hit "enter" → click "**Stack ".ncb""** to save the background cleaned stack

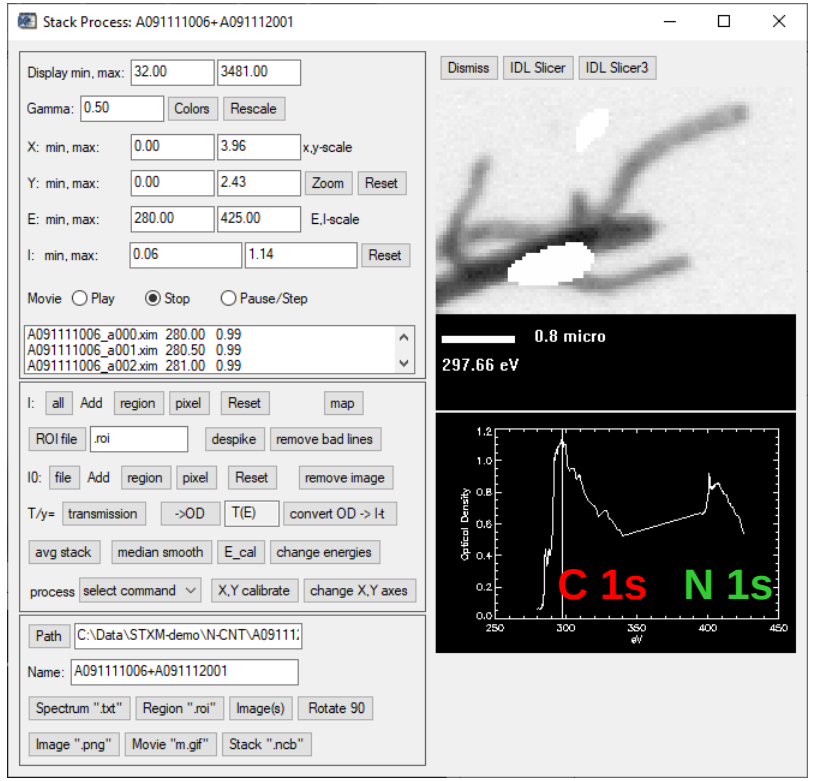


aXis2000 – Stacks – Change Mesh or Size – Append

- **Stack A091111006**: C 1s stack, 4 µm x 2.5 µm, 100 pixel x 63 pixel, step size: 40 nm
- **Stack A091112001**: N 1s stack, 5 µm x 3.0 µm, 167 pixel x 99 pixel, step size: 30 nm
- **Appending stacks procedure**
  - Cut A091112001 X-dimension 33 pixels and Y-dimension 17 pixels to have the same physical size as A091111006
  - Change mesh for the cut A091112001 to 100 pixel x 63 pixel, then append the two stacks together

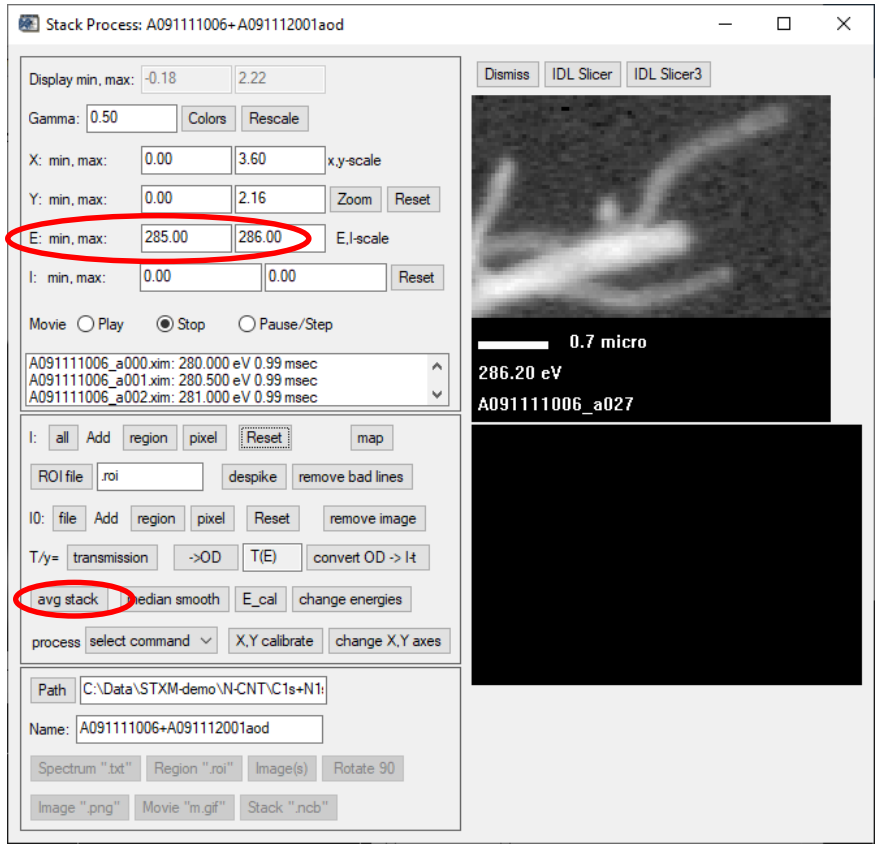
aXis2000 → Stacks → Analyze → stack process → select a raw transmission stack → click "**No**" for alignment file → **suggested zoom** value → select an "**I region**" to activate the output menu → give a file name, then hit "enter" → click "**Stack ".ncb""** → **keep columns(x)** > 33 → **keep columns(x) <** 166 → **keep rows(y)** > 17 → **keep rows(y)** < 98

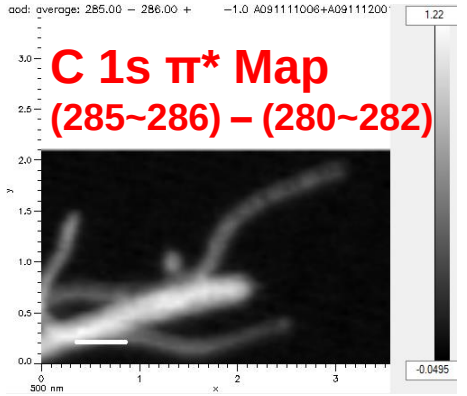
aXis2000 → Stacks → change mesh or size → click "**Yes**" for **change MESH (and keep same image size)** → input "**# of X pixels**" as 100 → input "**# of Y pixels**" as 63 → save the re-meshed stack → **Stacks → Append** → select **STACK1**: A091111006.ncb and **STACK2**: A091112001tm.ncb → save the appended stack file name as **A091111006+A091112001.ncb**

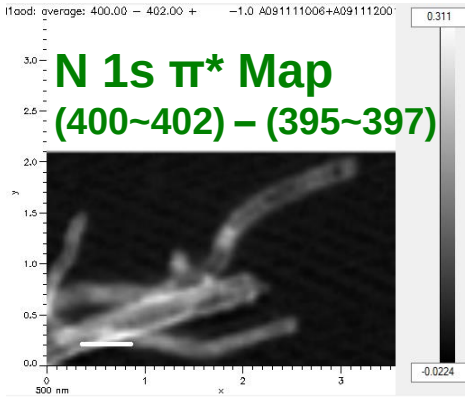


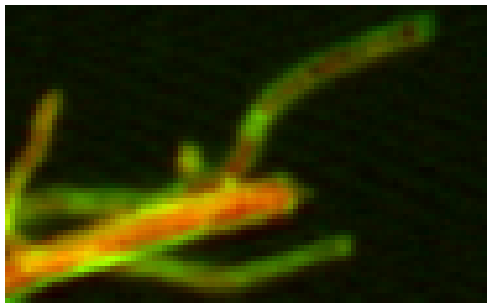
aXis2000 – Stacks – Elemental/Chemical On/Off Mapping

aXis2000 → **Stacks** → **Analyze** → **stack process** → select an aligned OD stack, e.g. A091111006+A091112001aod.ncb → click "**No**" for alignment file → **suggested zoom** value → change the "**E: min, max**" for pre-edge or post-edge → **avg stack** → **Store average image in** an empty buffer, then save the image → change the "**E: min, max**" again and generate other images → click a post-edge image → **Images** → **Add** → **Buffer** → choose the pre-edge image → **scaled by -1** → **Write** → **AXIS** → save map



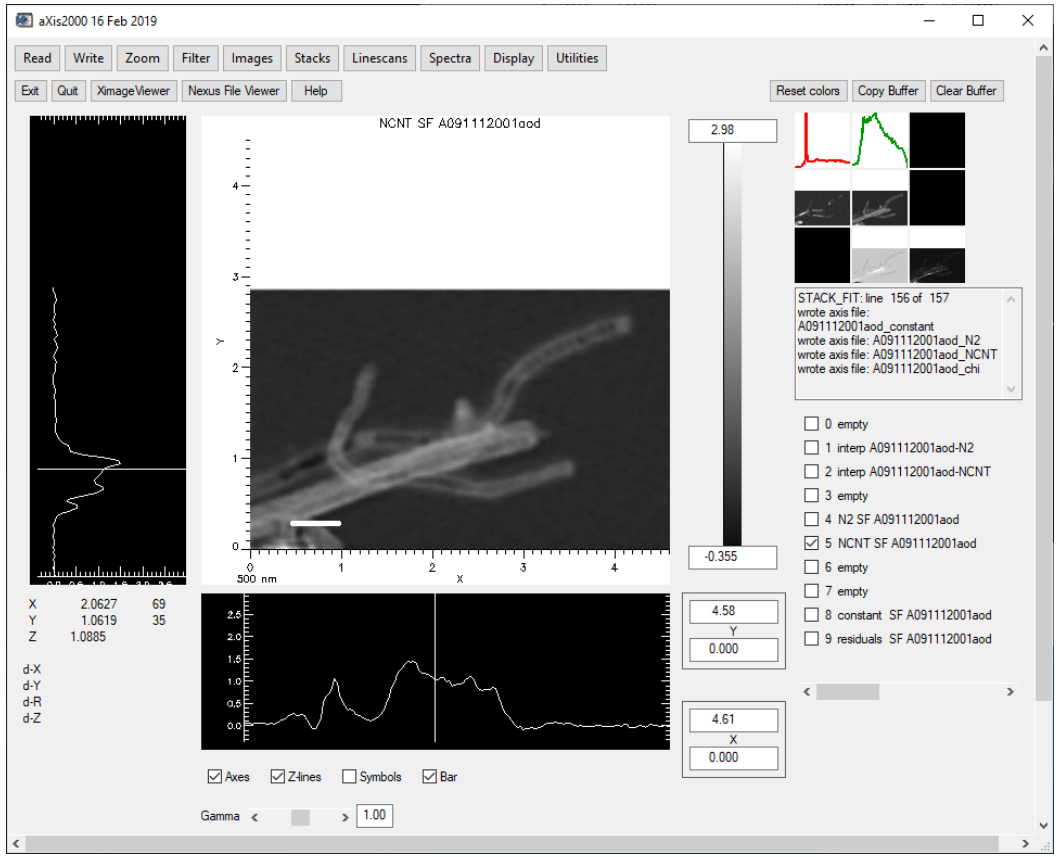






aXis2000 – Stacks – Maps – Stack Fit / SVD

aXis2000 → Stacks → maps → Stack fit or **SVD** → select an aligned OD stack → click "**No**" for parameter file → input "**# of components (1-8)**" → choose "**Spectrum of component 0**" → give a very short "**Name for component 0**" → choose the rest component(s) spectrum → give a **Name of fit parameter file** → then click "enters" for the rest default settings → result is saved and displayed in butters



Output of Stack Fit / SVD

- **Buffer 1**: reference spectrum of **Component 0**, in this case it is N2 spectrum obtained from the stack directly, i.e. internal reference
- **Buffer 2:** reference spectrum of **Component 1**, in this case it is NCNT spectrum obtained from the stack directly, i.e. internal reference
- **Buffer 4: Component 0** distribution map, in this case it is N2 distribution map
- **Buffer 5: Component 1** distribution map, in this case it is NCNT distribution map
- **Buffer 8: Constant** map (only for Stack Fit)
- **Buffer 9: Residuals** map

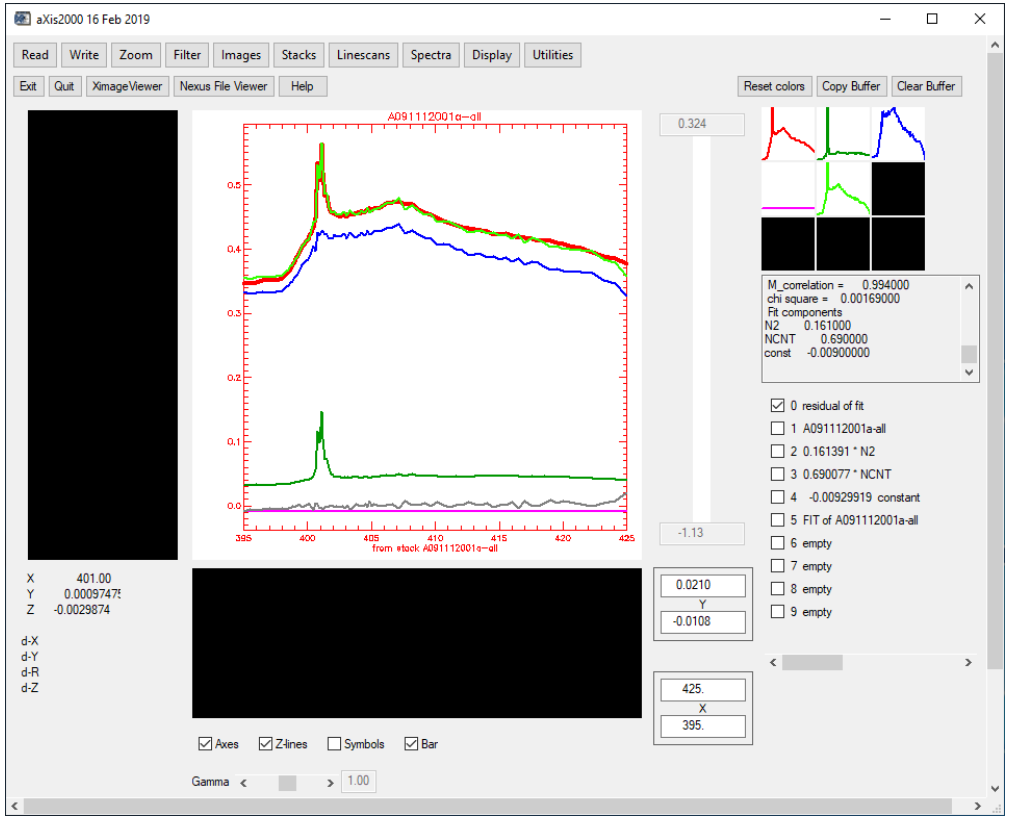
aXis2000 – Stack Fit / SVD – Color Composite Map

Click “**Component 0 map**” → **Images** → **Clip signal** → **histogram** → click “**around 0**” and “**highest visible intensity**” → **Copy Buffer**, and place it to **Buffer 1** → repeat the procedure for “**Component 1**” and place it to **Buffer 2** → in this case of only two components, a third blank image of zero intensity needs to be created → click any component image → **Images** → **Gain** → **Multiply by 0** → place the blank image to **Buffer 3** → click a blank buffer like **Buffer 7** → **Display** → **RGB composite** → select **RED**, **GREEN**, and **BLUE** images from **Buffer 1** to **3** respectively → click “**Yes**” for Autoscale each component → save the image in *.tif format

- **Buffer 0**: color composite map
- **Buffer 1**: high contrast image of **Component 0**
- **Buffer 2**: high contrast image of **Component 1**
- **Buffer 3**: blank image with zero intensity

aXis2000 – Spectra Curve Fit – Stack Fit / SVD

aXis2000 → Read → Spectra → AXIS → open a spectrum for all sample regions like A091112001aod-all.txt → click the spectrum → **Spectra → Curve fit → linear regression (stack fit)** or **SVD** → click "**No**" for parameter file → input "**Number of components**" → choose "**Spectrum of component 0**" → give a very short "**Name for component 0**" → choose the rest component(s) spectrum → give a **Name of fit parameter file** → result is displayed in buffers → save spectra by → **Write → AXIS**

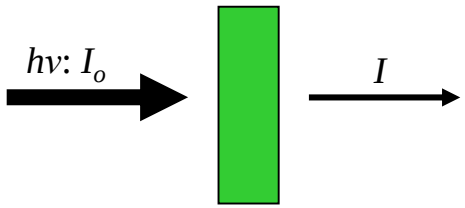


Output of Curve Stack Fit / SVD

- **Buffer 1**: the original sample spectrum
- **Buffer 2**: fitting coefficient times reference spectrum of **Component 0**, i.e. N2 spectrum
- **Buffer 3**: fitting coefficient times reference spectrum of **Component 1**, i.e. NCNT spectrum
- **Buffer 4: Constant** (only for Stack Fit)
- **Buffer 5:** simulated fit spectrum

Principles of STXM Quantitation

Beer's Law



$T = I/I_o$

OD = Abs. = ln (Io/I) = $\mu\rho l$

OD: optical density

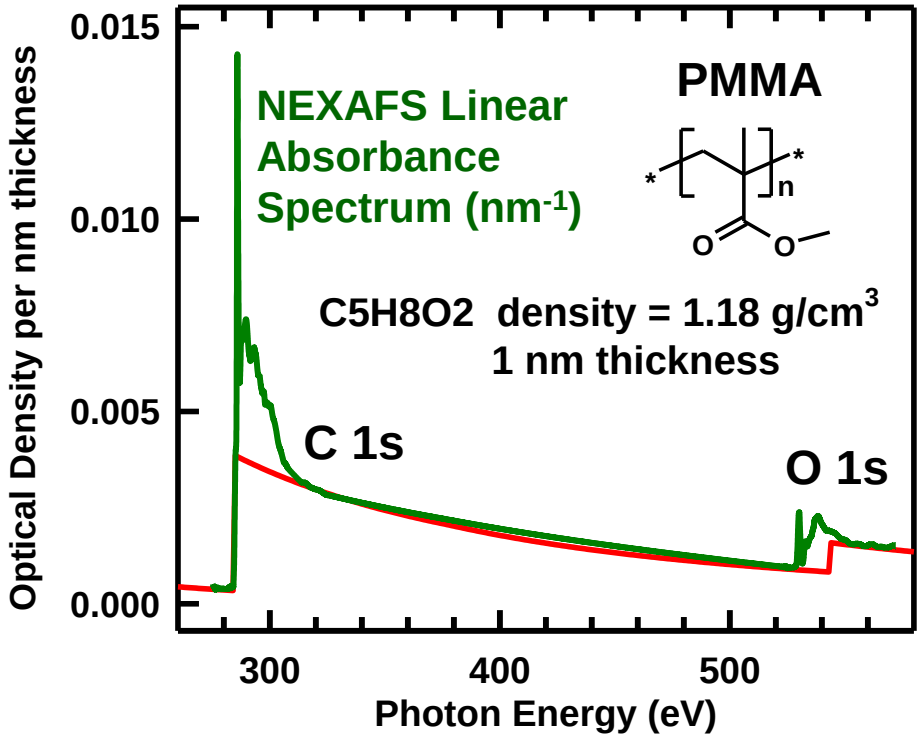
μ: mass absorption coefficient

ρ: material density

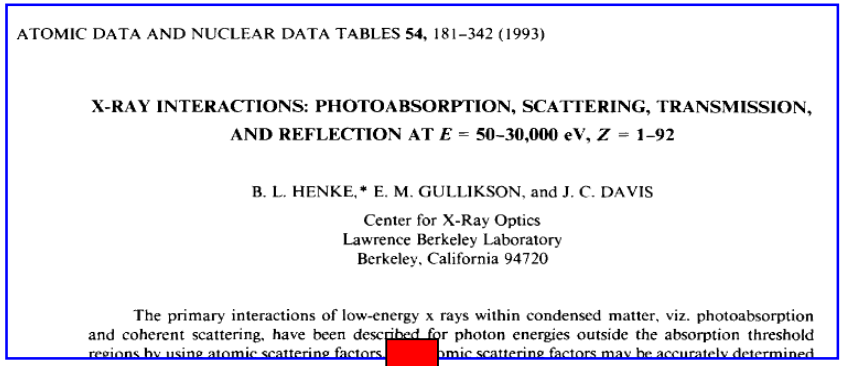
l: thickness

Optimal STXM Sample Transmission:

0.1~1~2 = OD = ln (Io/I) = -ln (T) = -ln (90%~37%~14%)



Mass Absorption Coefficient for the Elements



ATOMIC DATA AND NUCLEAR DATA TABLES 54, 181–342 (1993)

X-RAY INTERACTIONS: PHOTOABSORPTION, SCATTERING, TRANSMISSION, AND REFLECTION AT E = 50–30,000 eV, Z = 1–92

B. L. HENKE,* E. M. GULLIKSON, and J. C. DAVIS

Center for X-Ray Optics
Lawrence Berkeley Laboratory
Berkeley, California 94720

The primary interactions of low-energy x rays within condensed matter, viz. photoabsorption and coherent scattering, have been described for photon energies outside the absorption threshold regions by using atomic scattering factors. [illegible]omic scattering factors may be accurately determined

Calculated Elemental Mass Absorption Coefficient for Compounds

$$\mu = \frac{N_A}{M}\sum_i x_i \sigma_i$$

N_A: Avogadro's number

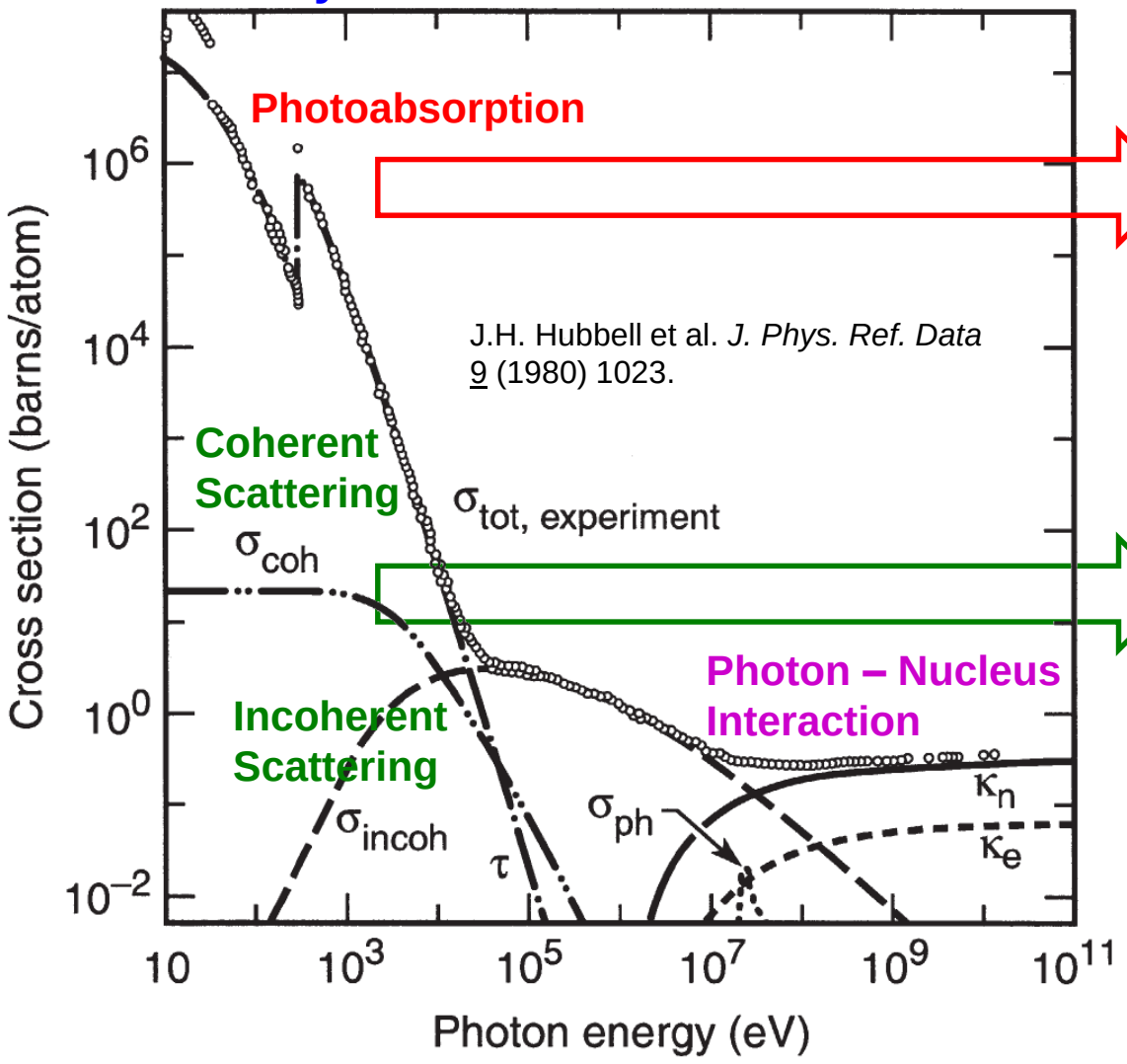
M: molecular weight

x_i: number of atom i

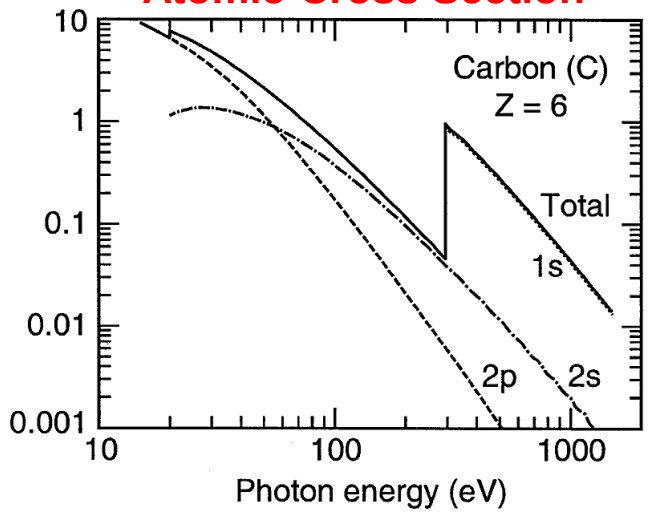
σ_i: atomic photoabsorption cross section

X-ray – Matter (Atomic) Interaction

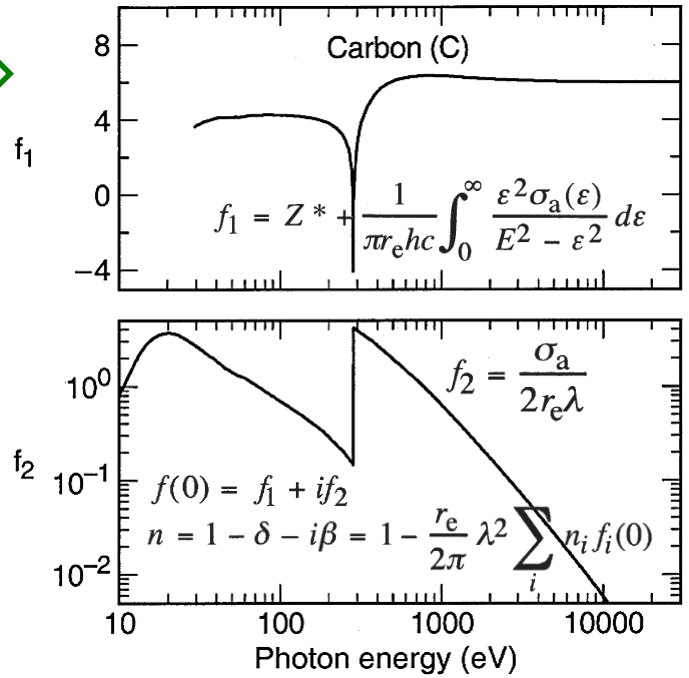
X-ray Interaction with Carbon Atoms



Atomic Cross Section

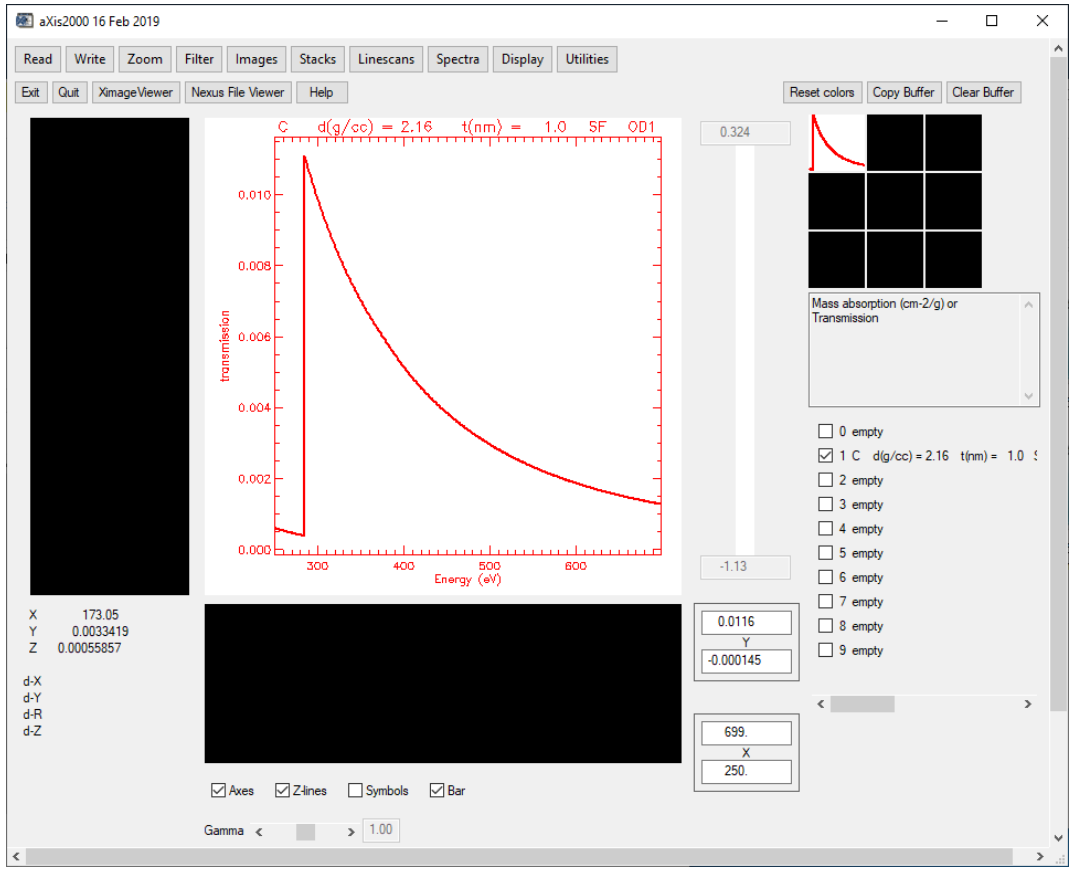


Atomic Scattering Factors



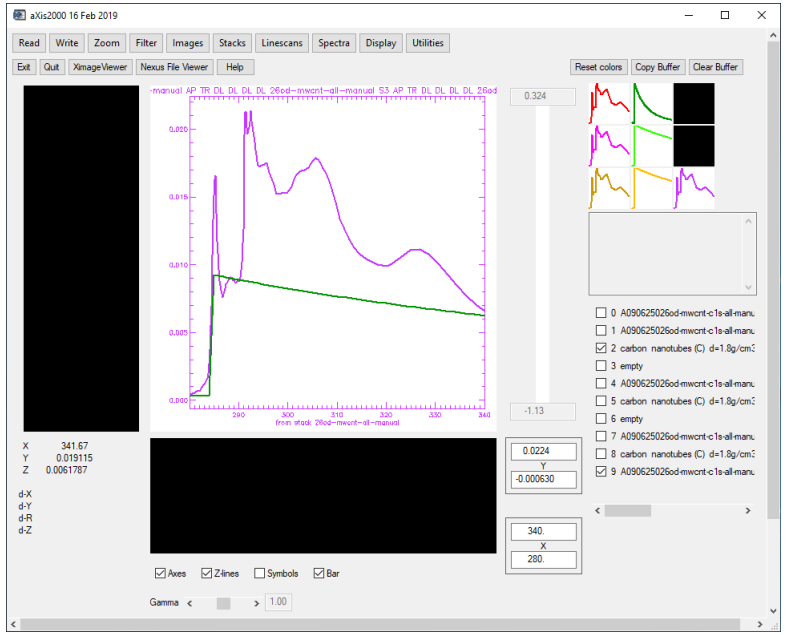
aXis2000 – Utilities – Calculate X-ray Parameters (SF)

aXis2000 → Utilities → Calculate X-ray parameters (SF) → Formula, e.g. C → **minimum energy → maximum energy → transmission → density (g/cc)**, e.g. 2.16 (graphite) → **thickness (μm): 0.001** → click "**Yes**" to "**Convert to OD**" → accept or update "**Header for output file**" → **Write → AXIS** → give a file name and save file



aXis2000 – Quantitative Scaling Reference Spectra

aXis2000 → Read → Spectra → AXIS → open a pure sample spectrum, i.e. **MWCNT** at **Buffer 1**, and a **sf** file for MWCNT at **Buffer 2** → click **Buffer 1 → Spectra → Calibrate → Y → 1 point** → click pre-edge baseline, and set "**New Y**" to zero → **Copy Buffer**, and place it at **Buffer 4** → click **Buffer 2** and change the X display limits to **280 – 340** eV → **Spectra → Truncate**, and click 280 and 340 eV positions → **Copy Buffer**, and place it to **Buffer 5** → calibrate pre-edge to zero and place it to **Buffer 8** → click **Buffer 4 → Spectra → Gain → divide by 14 → Display → Over Plot → No Rescale**, and choose **Buffer 8**, until the spectrum and the sf file overlap in both pre-edge and post-edge → place the scaled spectrum to **Buffer 7** → check **Buffer 5** pre-edge **Y value**, and use this value to calibrate **Buffer 8** pre-edge Y value → place the calibrated and scaled spectrum to **Buffer 9 → Write → AXIS**

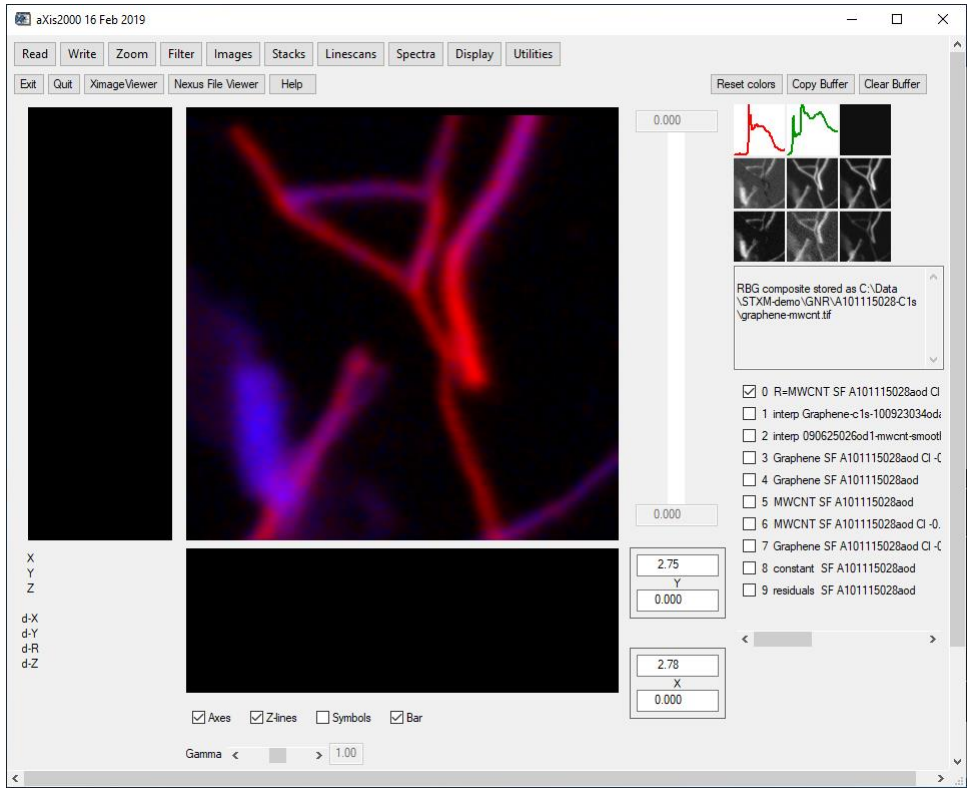


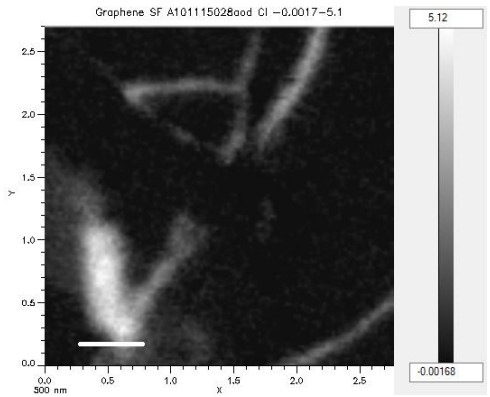
Process of Quantitative Scaling Reference Spectra

- **Buffer 1**: the original MWCNT sample spectrum
- **Buffer 2**: sf file of MWCNT, i.e. 1 nm thick elemental X-ray absorption profile
- **Buffer 4**: pre-edge zeroed MWCNT sample spectrum
- **Buffer 5**: truncated sf file, i.e. 280-340 eV
- **Buffer 7**: pre-edge zeroed and scaled MWCNT spectrum
- **Buffer 8**: pre-edge zeroed and truncated sf file
- **Buffer 9**: pre-edge calibrated and scaled MWCNT spectrum, i.e. 1 nm thick MWCNT NEXAFS spectrum

aXis2000 – Stacks – Quantitative Stack Fit / SVD

aXis2000 → Stacks → maps → Stack fit or **SVD** → select an aligned OD stack → click "**No**" for parameter file → input "**# of components (1-8)**" → choose "**Spectrum of component 0**", i.e. 1 nm thick graphene spectrum → give a very short "**Name for component 0**" → choose the rest component(s) spectrum, i.e. 1 nm thick MWCNT spectrum → give a **Name of fit parameter file** → then click "enters" for the rest default settings → result is saved and displayed in butters

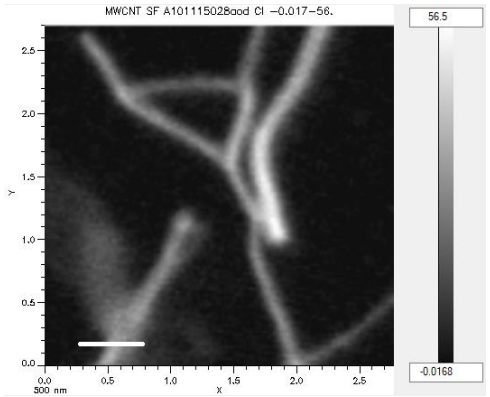




5.1 nm

Graphene

0



56.5 nm

MWCNT

0

Summary: Typical Measure & Analysis of a STXM Stack

MEASURE **Typical Steps:**

1. FIND A SUITABLE AREA
2. Check it has suitable properties (based on prior knowledge)
3. Measure **stack map**s (few images ➔ few components) or **stacks** (good to look for surprises)
4. Check for **damage**
5. make sure you have a **valid Io** - **measure at SAME TIME (2nd area if needed)**
6. Check **energy calibration**

ANALYSIS - STACKS

- Convert from raw data to a binary stack
- Align - Jacobsen stack_analyze or Zimba
- Convert to OD
  (best to use built-in Io; otherwise, measure Io (point or stack) just before or after)
- Inspect → Zimba
- Identify suitable reference spectra - external or internal
- Convert ref. spectra to OD1
- FIT - Singular Value Decomposition (SVD) versus Stack Fit (SF)
- Clean component maps (remove outliers)
- Display - non-rescale versus rescale RGB
- CHECK critical aspects: residuals, residual stacks; extract spectra of component-map-masked regions & fit to the reference spectra & inspect QUALITY of FIT
- Perform Multivariate Statistical Analysis (PCA_GUI, Mantis) & compare

Courtesy A.P. Hitchcock

Cryo-STXM

Cryo-STXM Data Format

HDF5 File: e.g. **C190126021.hdf5**,
Scans, STXM, and Beamline settings

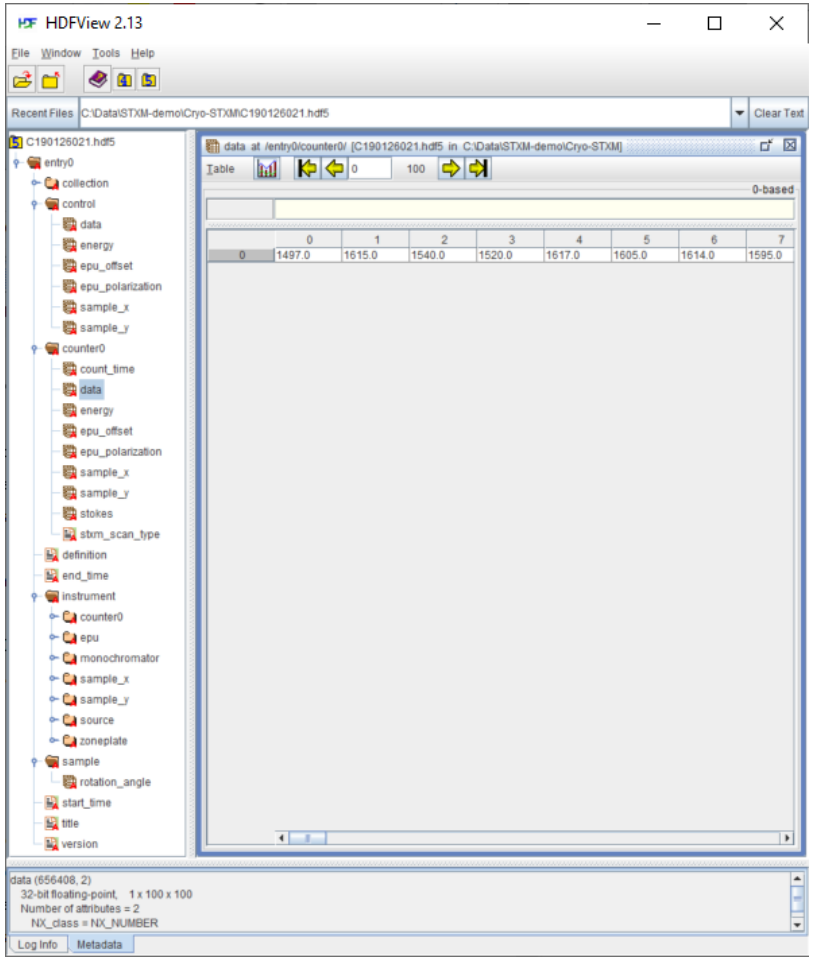
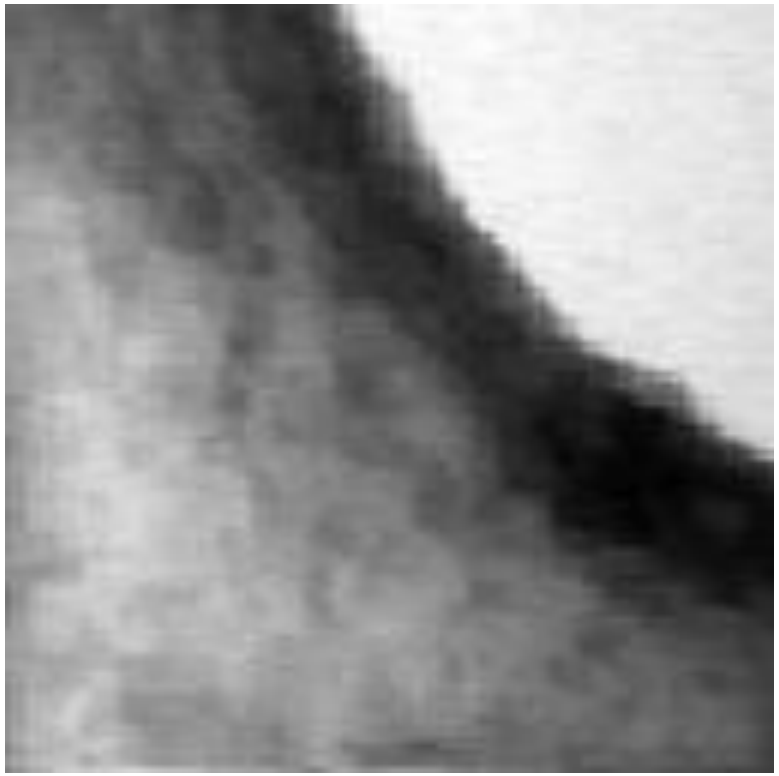
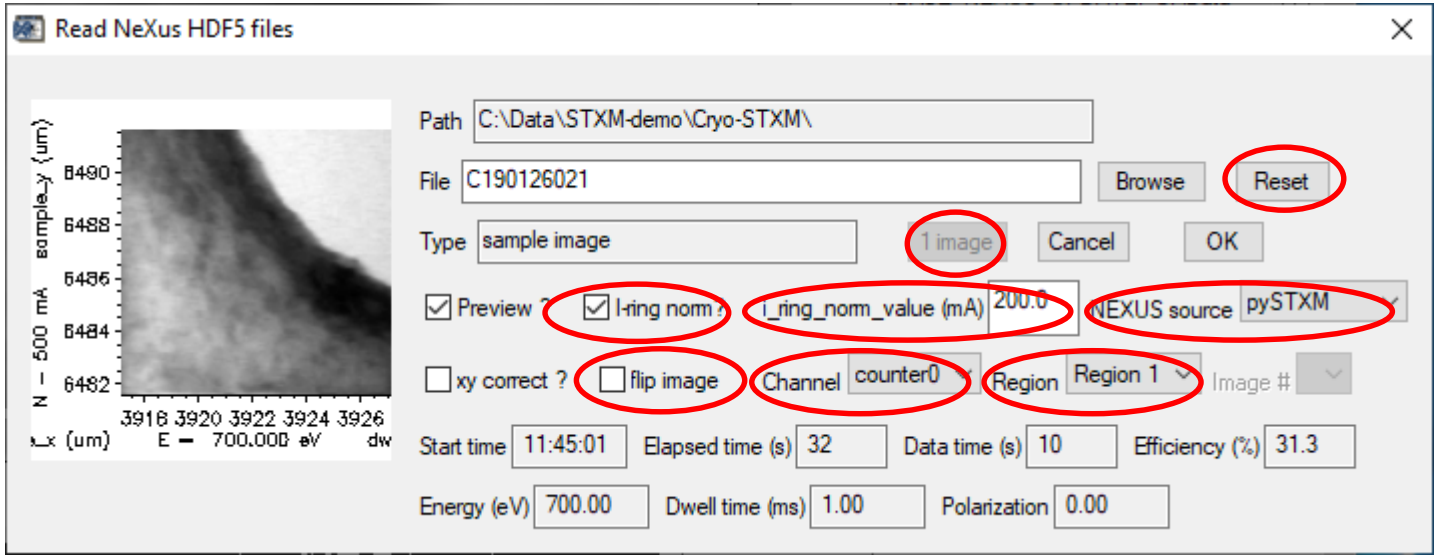


Image Preview File:
e.g. **C190126021.jpg**

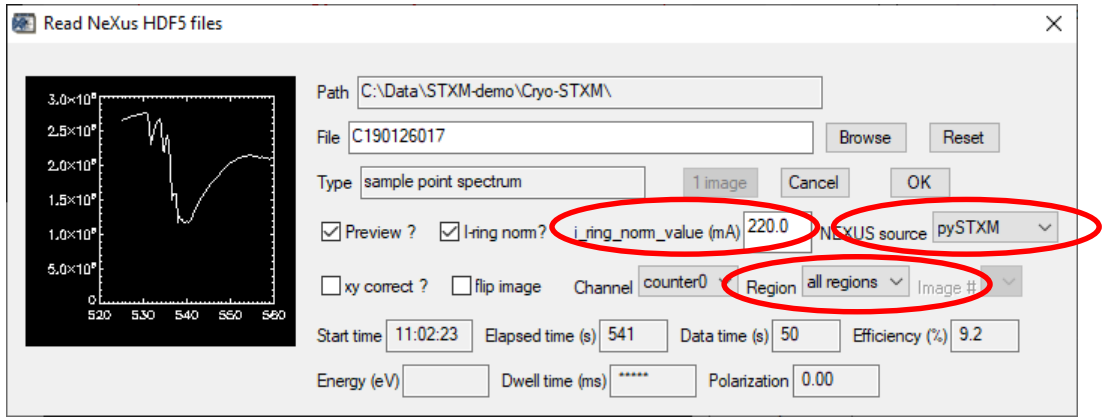


aXis2000 – Read Cryo-STXM Images

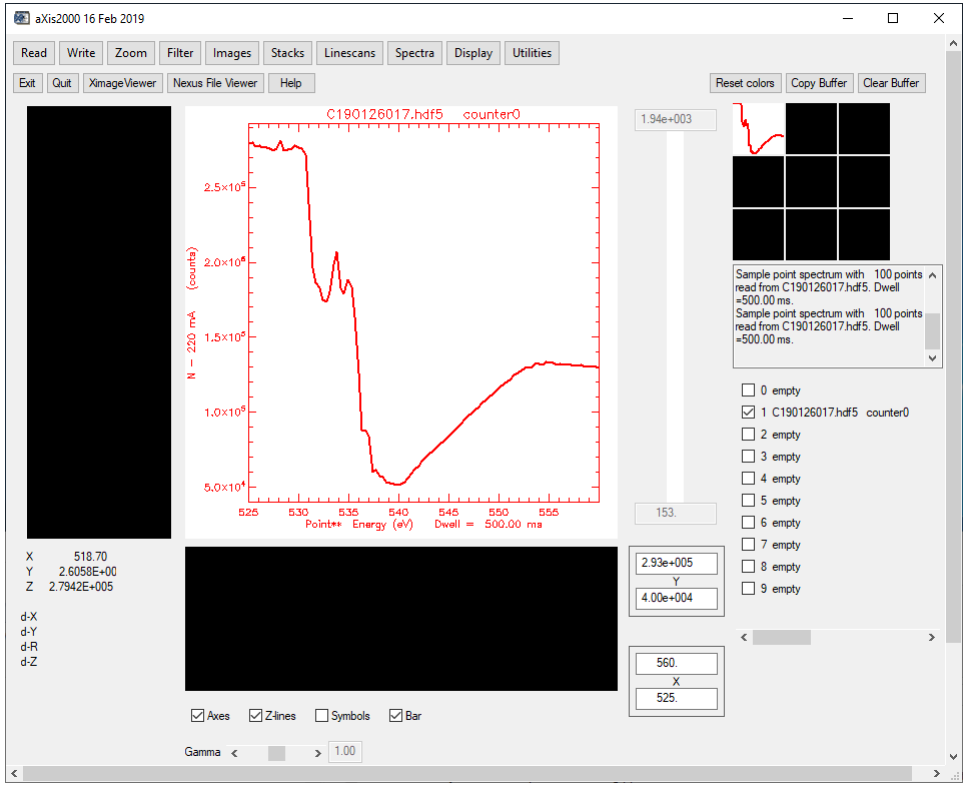


- **Reset**: clear loaded image and information
- **1 image**: read an image from a selected photon energy for a stack
- **I-ring norm?**: normalization to I-ring
- **I_ring_norm_value**: CLS 220 mA
- **NEXUS source**: CLS pySTXM
- **Flip image**: flip image up and down
- **Channel**: select data channel if more than one detector is used
- **Region**: select sample region if more than one image region is defined

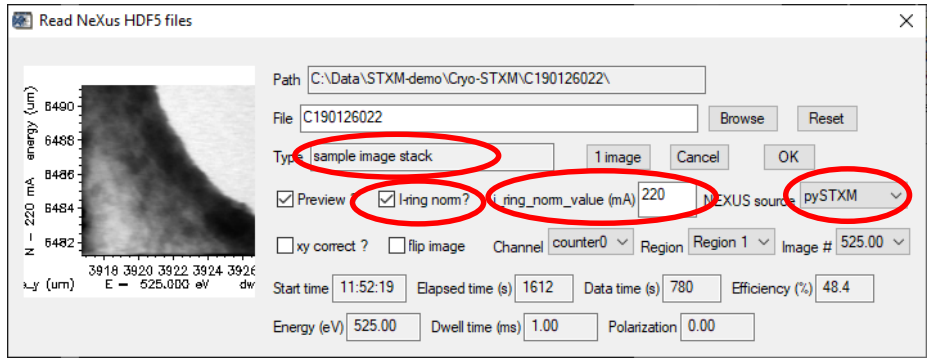
aXis2000 – Read Cryo-STXM Point Spectra



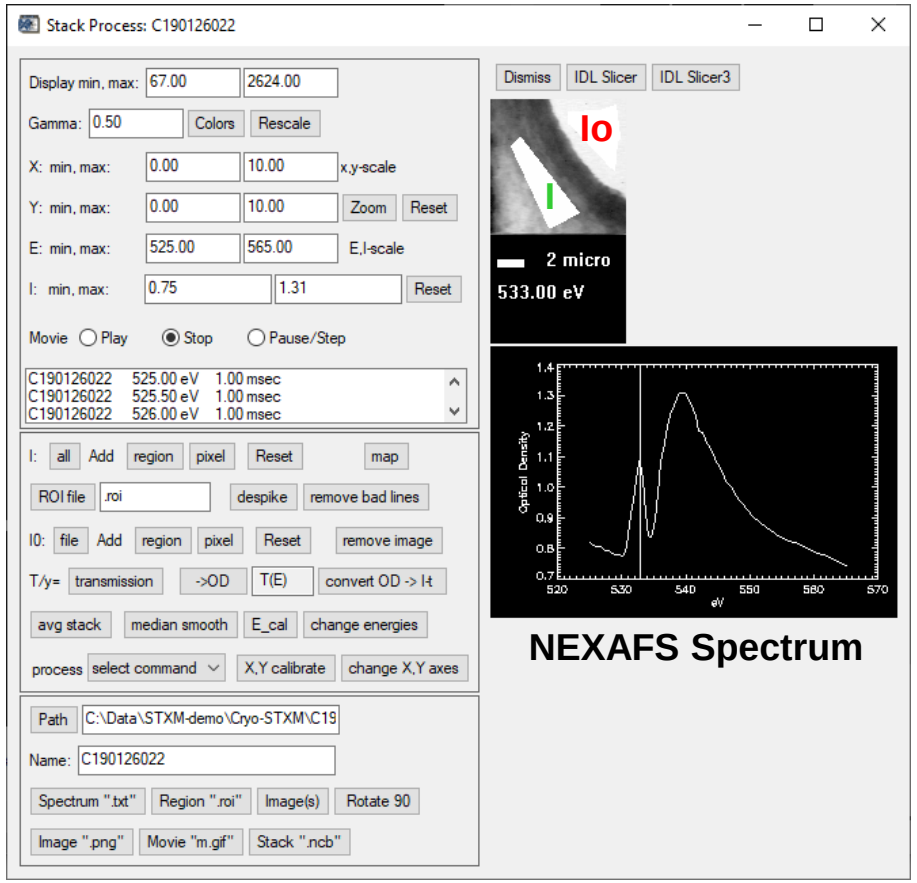
- **Note**: even select all regions, only one spectrum can be loaded into aXis2000



aXis2000 – Read Cryo-STXM Stacks



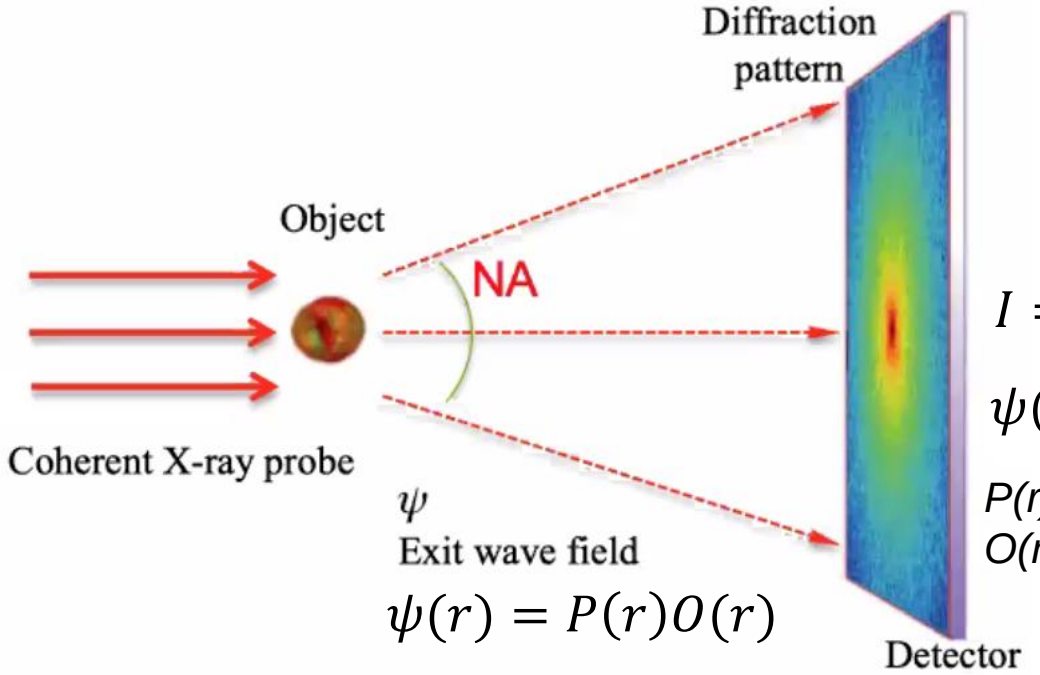
- Compile raw Cryo-STXM stack data: aXis2000 → Read → STXM (NeXus)
- Other steps are the same as Ambient-STXM



NEXAFS Spectrum

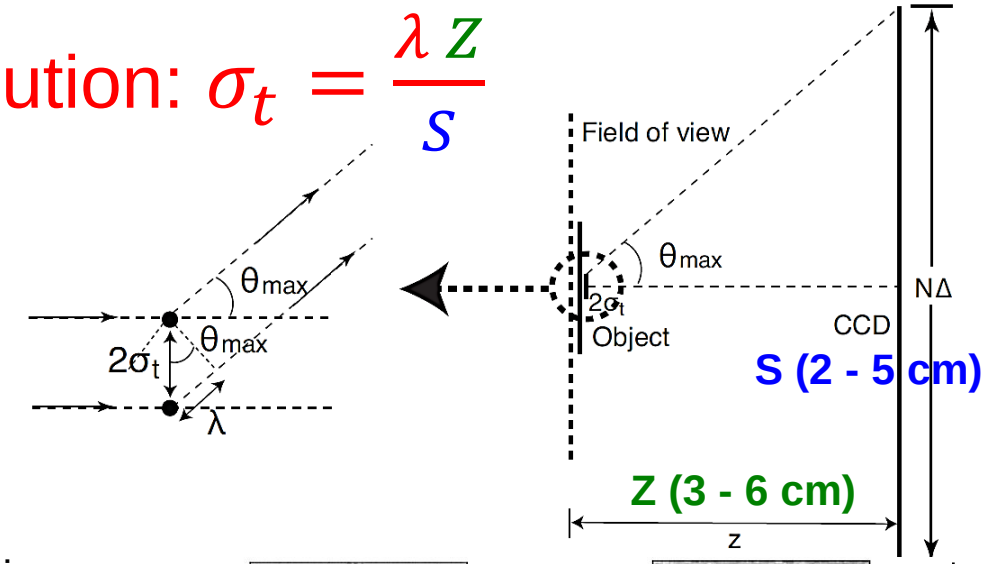
STXM-Ptychography

Coherent Diffractive Imaging (CDI)



$\psi(r) = P(r)O(r)$

Resolution: $\sigma_t = \frac{\lambda Z}{S}$



$I = |F(\psi)|^2$

$\psi(r) = P(r)O(r)$

P(r): Probe function
O(r): Complex object function

SEM

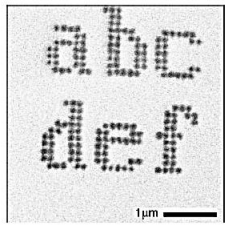


Figure 1 A scanning electron microscope image of the specimen. The specimen was fabricated by depositing gold dots, each ~100 nm in diameter and 80 nm thick, on a silicon nitride membrane.

OM

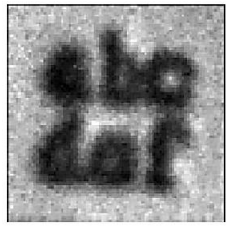


Figure 3 An optical microscope image of the specimen.

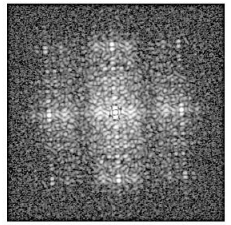


Figure 2 A diffraction pattern of the specimen (using a logarithmic intensity scale). The central 15-pixel-radius circular area is supplied by the squared magnitude of the Fourier transform of the optical microscope image (Fig. 3).

CDI

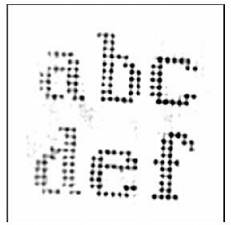


Figure 4 The specimen image as reconstructed from the diffraction pattern of Fig. 2.

Complex Refractive Index: $n = 1 - \delta - i\beta = 1 - \alpha\lambda^2(f_1 + if_2)$

Complex Wavefield of Light: $P(r) = Ae^{ikt}$

Complex Object Function: $O(r) = e^{iknt} = e^{ik(1-\delta-i\beta)t} = e^{k\beta t} \times e^{-ik\delta t}$

Absorption: $|O(r)| = e^{-k\beta t}$ Phase: $Arg\{O(r)\} = -k\delta t$

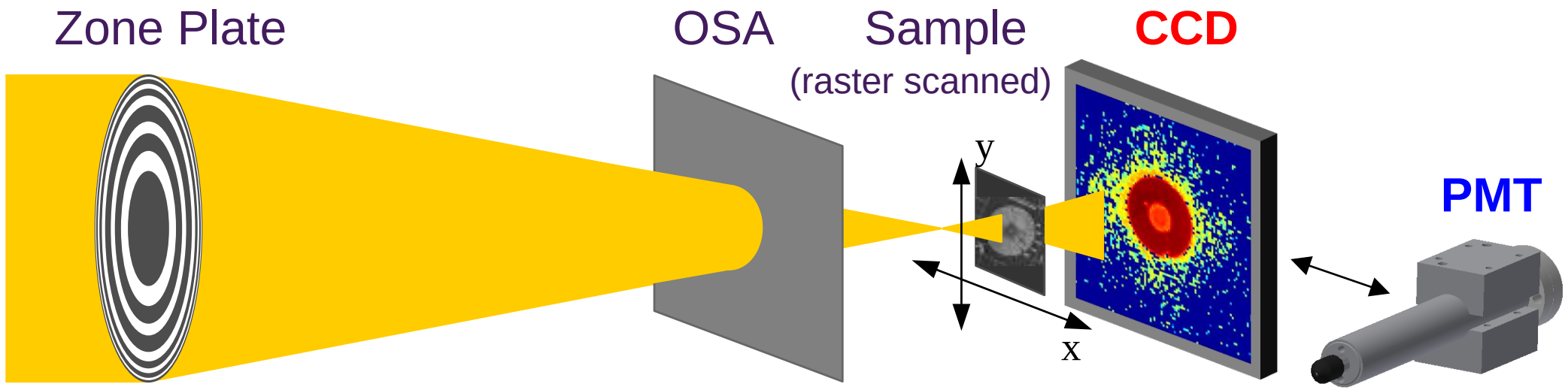
Diffraction Pattern: Fourier Transform of Product of Probe and Object. Inverse Fourier Transform and Phase Retrieval to solve Object.

Canadian Light Source | Centre canadien de rayonnement synchrotron

THE BRIGHTEST LIGHT IN CANADA | lightsource.ca

J. Miao et al. *Nature* 400 (1999) 342.

From Conventional STXM to STXM-Ptychography



Conventional STXM

- Sample in-focus
- Point (0D) detector: Scintillator+PMT or PD
- Real-space images
- Diffraction/spot size limited spatial resolution ($1.22*\Delta r_n$: ~30 nm)

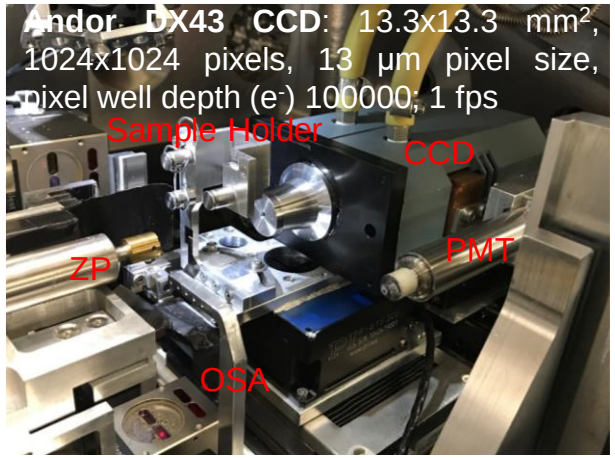
STXM-Ptychography

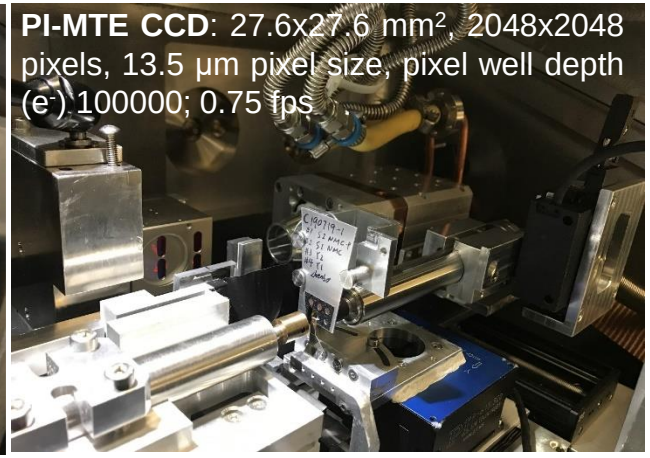
- Sample in-focus or out-of-focus
- 2D detector: X-ray CCD
- Reciprocal-space images
- Wavelength limited spatial resolution (1 - 2 nm by soft X-rays)
- Large computation in data process

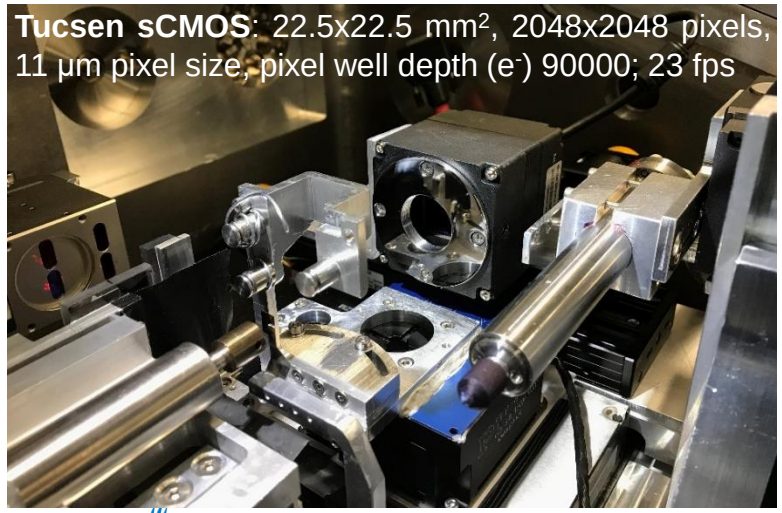
CLS-SM STXM Instrumentation Development Highlights – STXM-Ptychography

Ambient-STXM

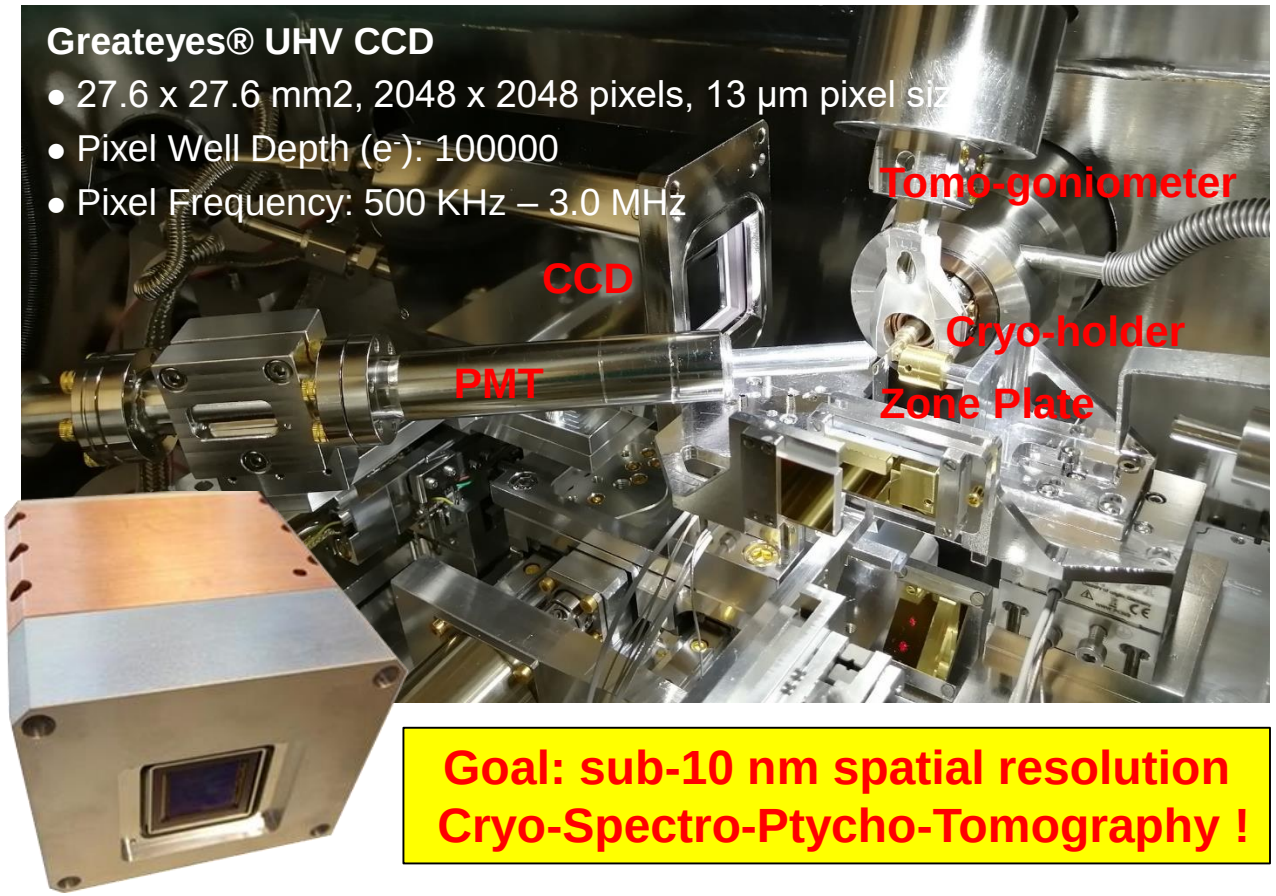
Cryo-STXM







Diffraction from Au/Pd nanoparticles captured by sCMOS



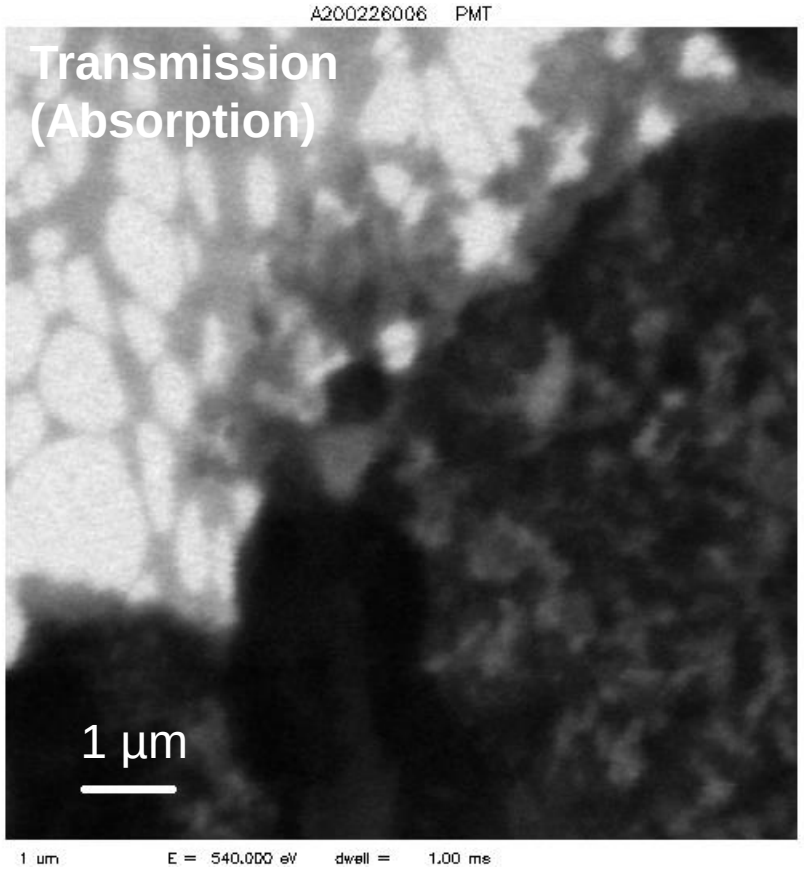
Future AXIS-SXR: 4 MPixel Soft X-ray sCMOS camera

Goal: sub-10 nm spatial resolution Cryo-Spectro-Ptycho-Tomography !

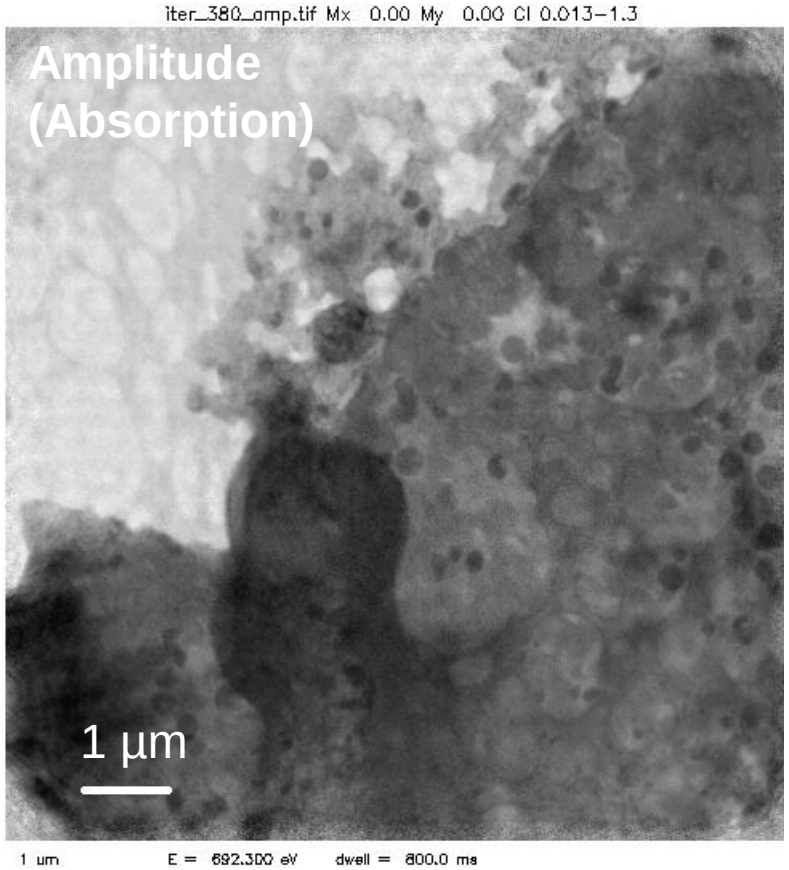
Canadian Light Source | Centre canadien de rayonnement synchrotron

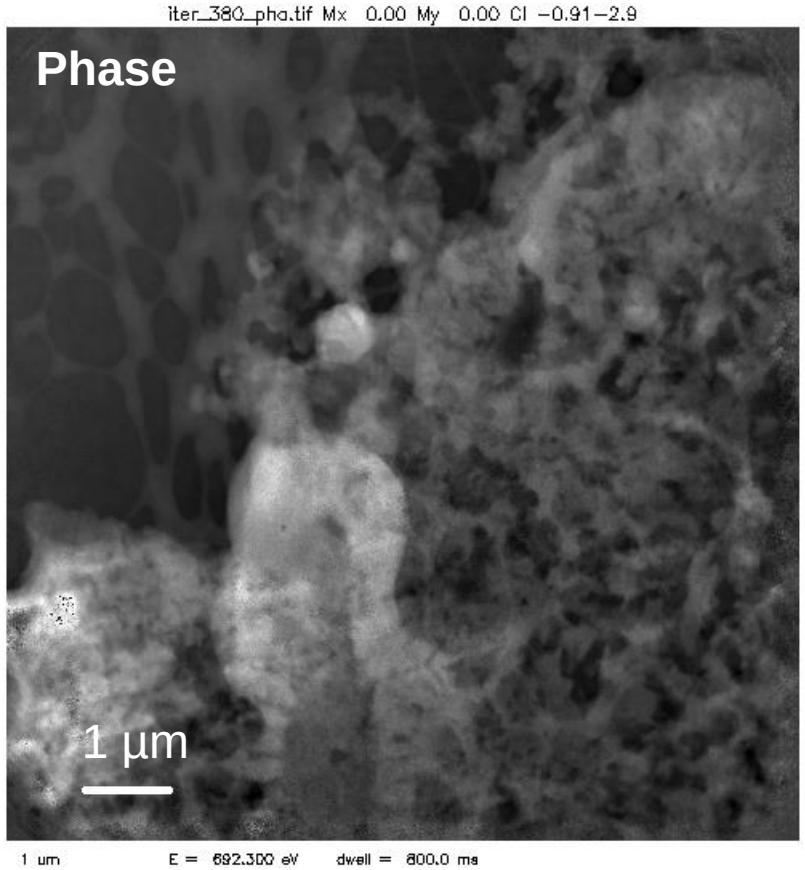
Recent STXM Spectro-Ptychography of LIB Cathode

Conventional STXM



STXM-Ptychography

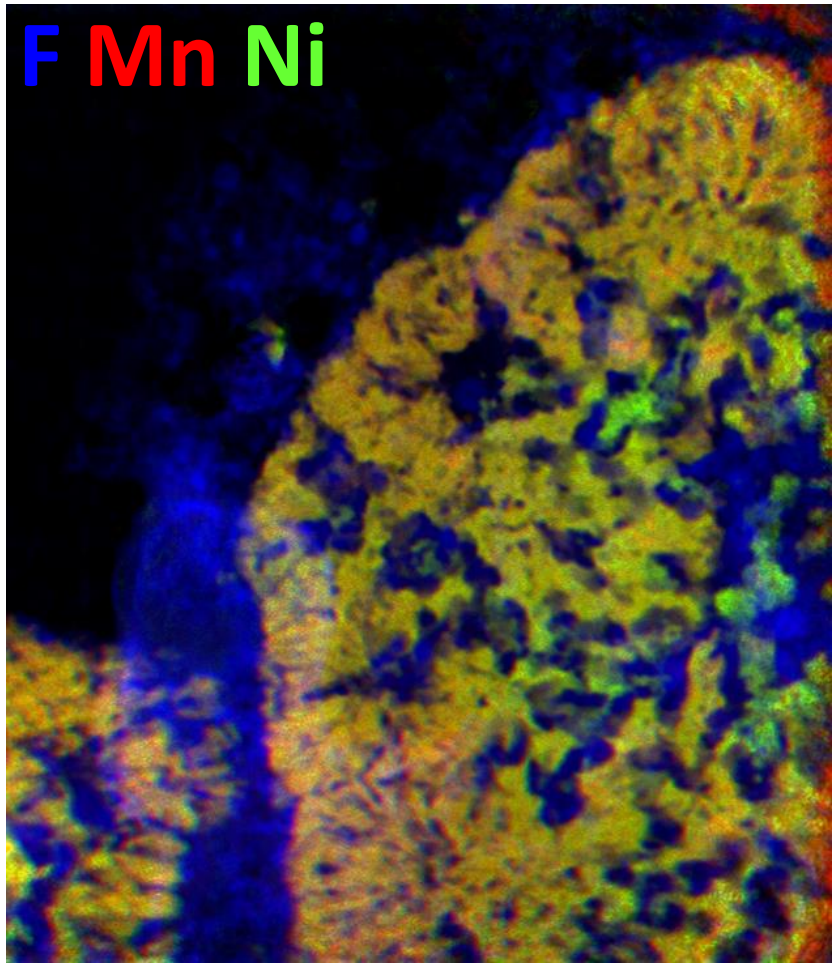


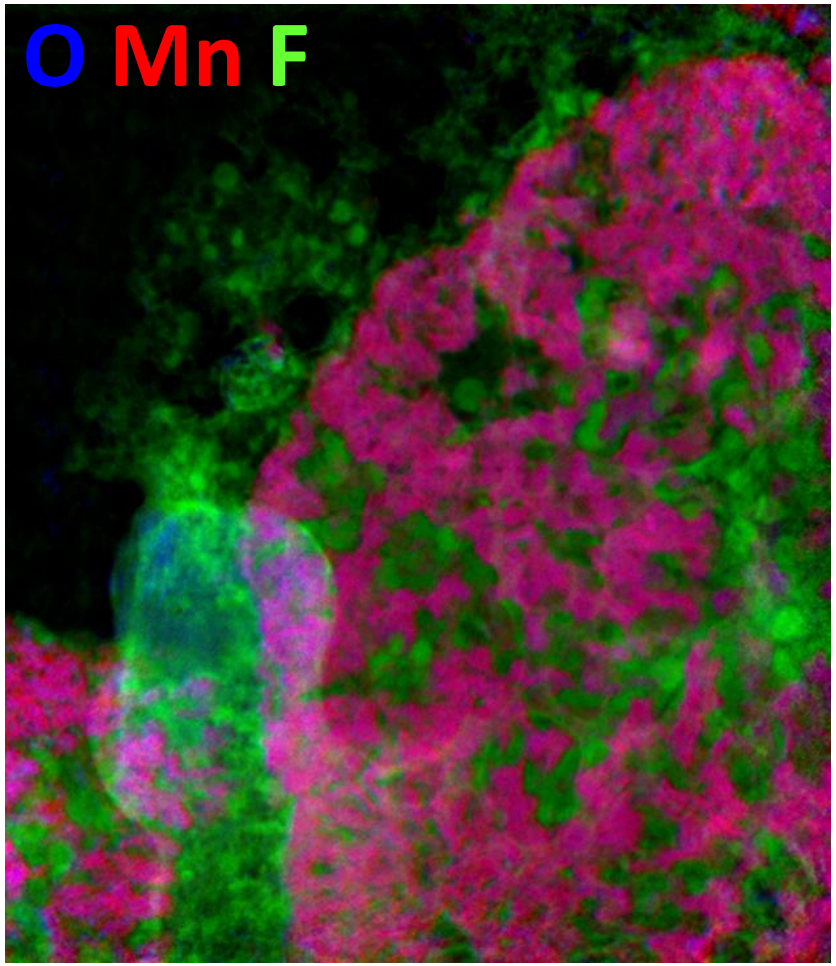


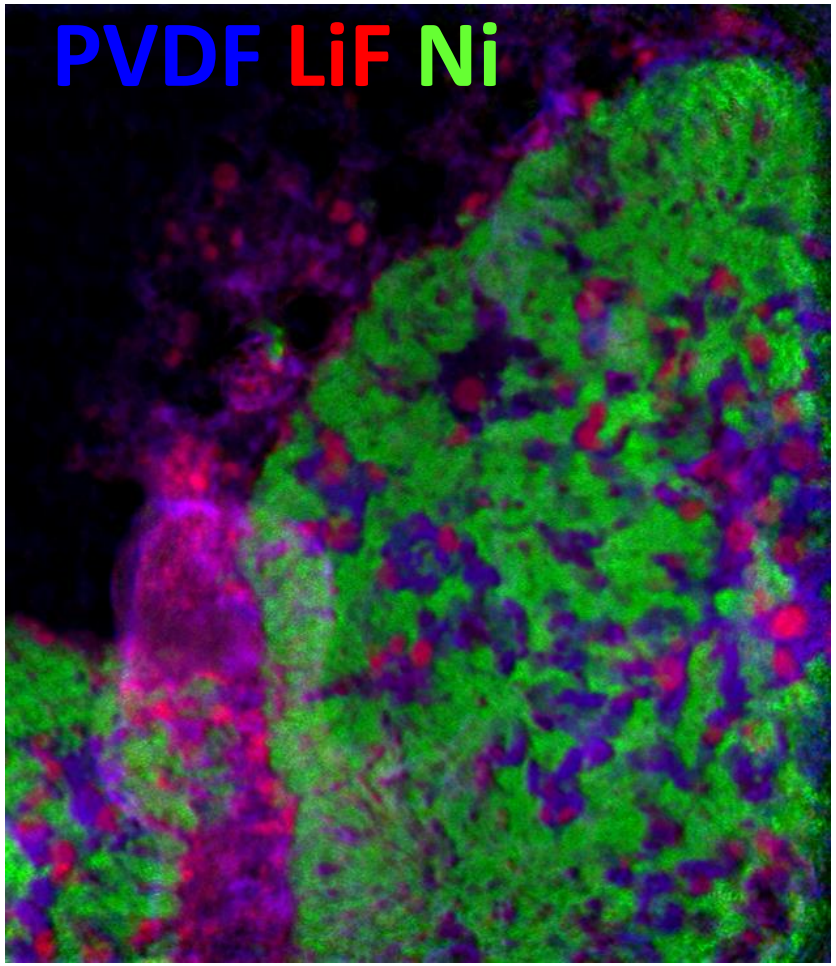
Pixels: 300 x 300
Resolution: ~40 nm

Pixels: 2426 x 2426
Resolution: ~8 nm

Recent STXM Spectro-Ptychography of LIB Cathode

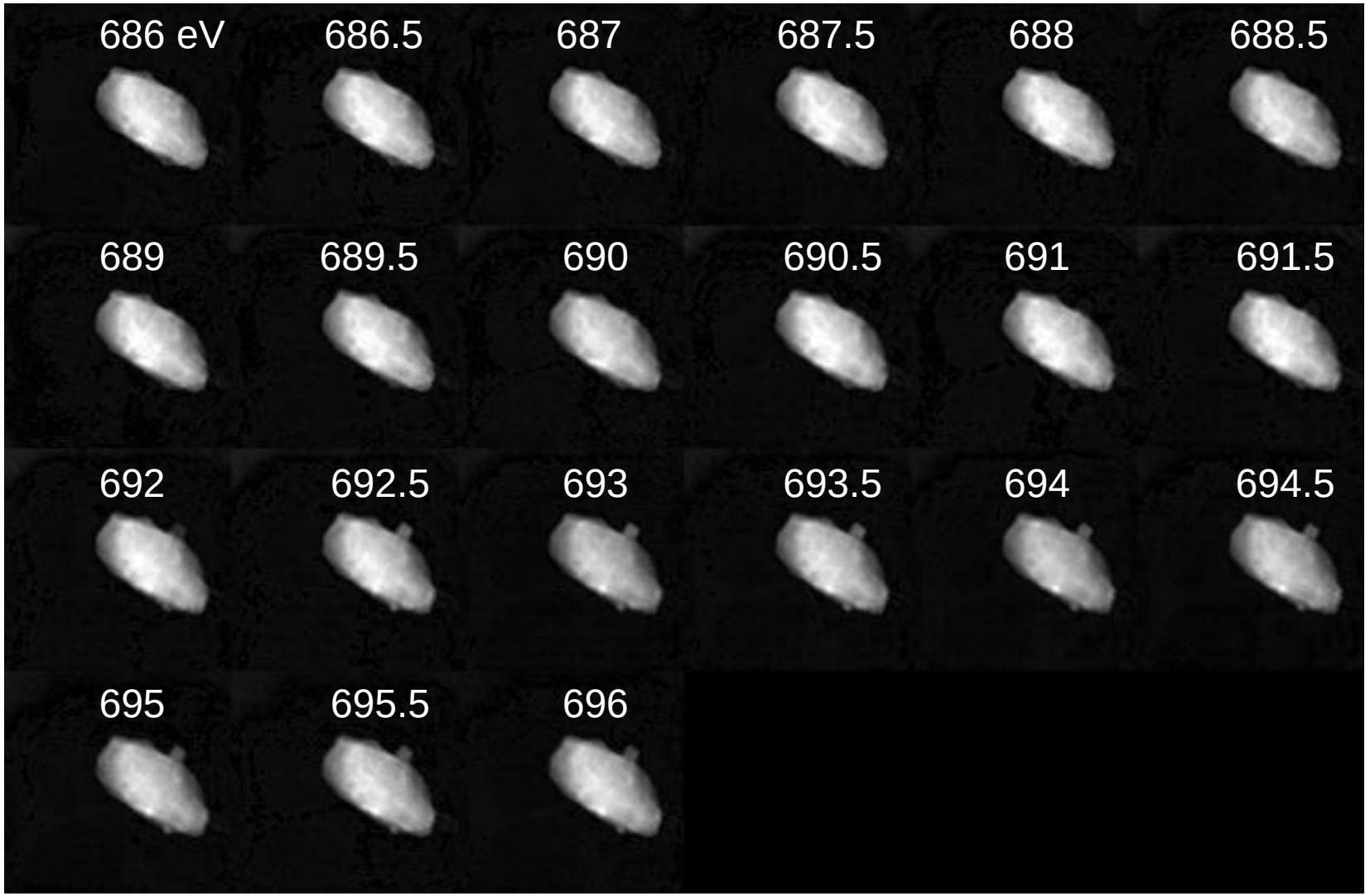


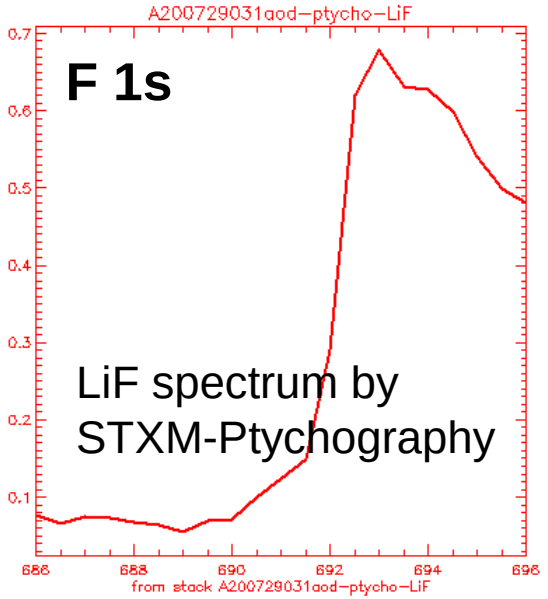


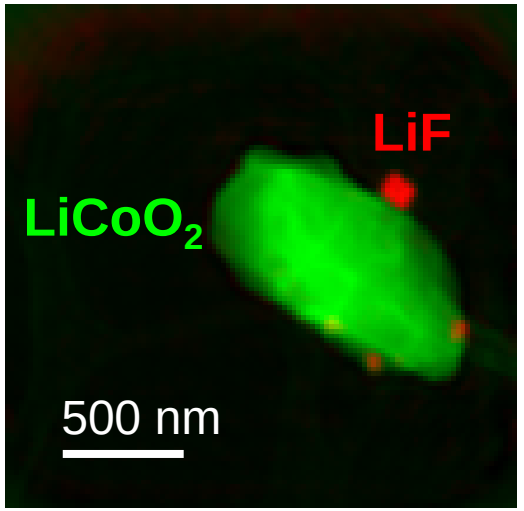


Spatial Resolution: ~8 nm

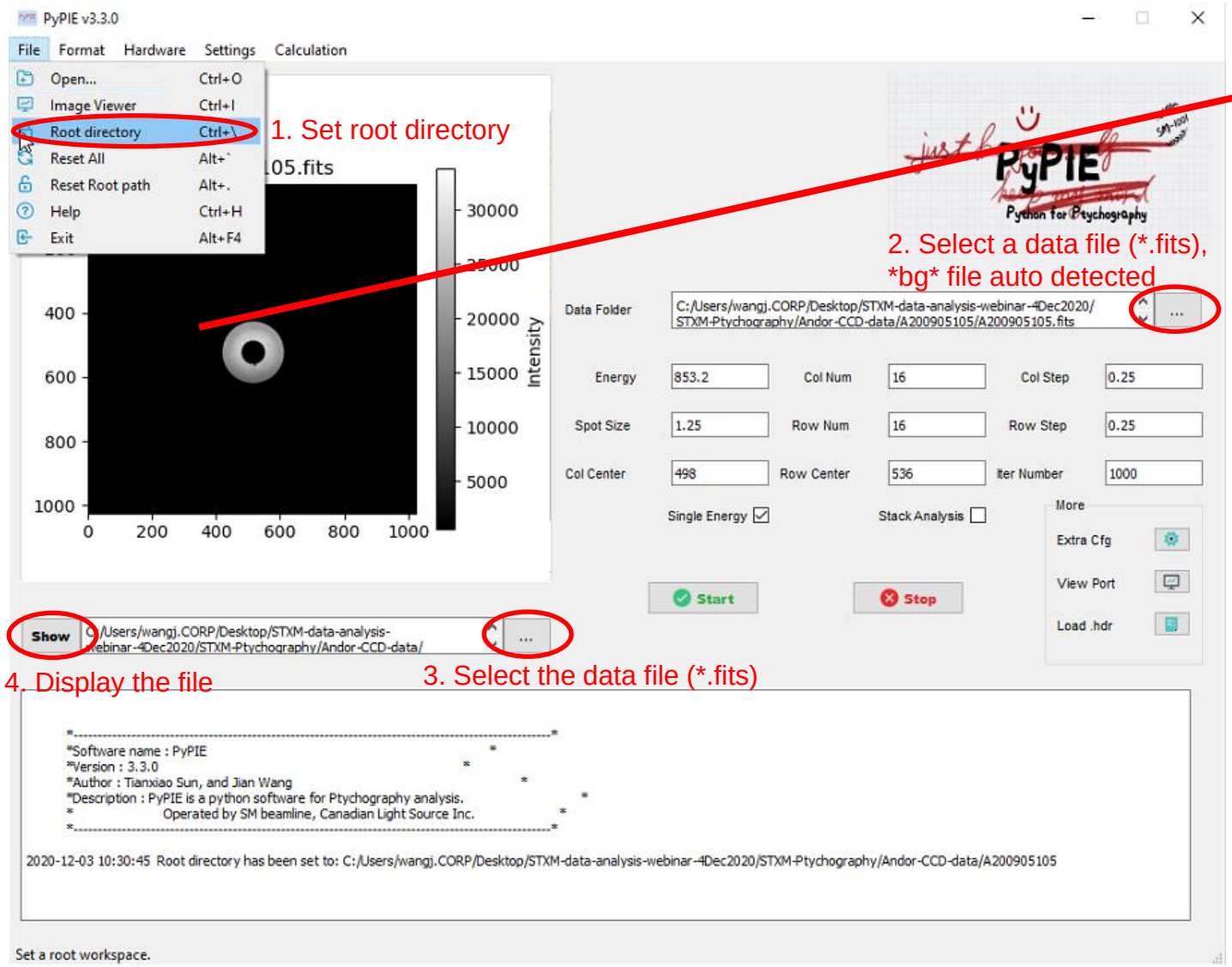
Recent STXM Spectro-Ptychography of LIB Cathode

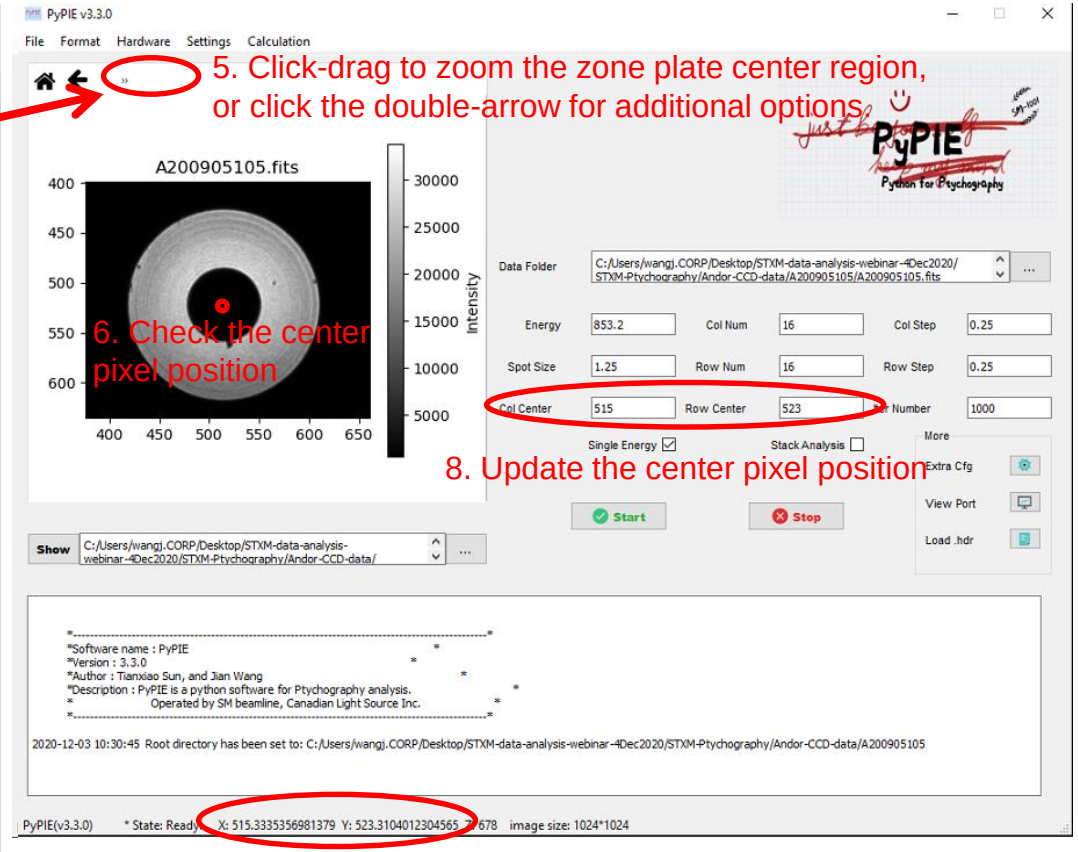




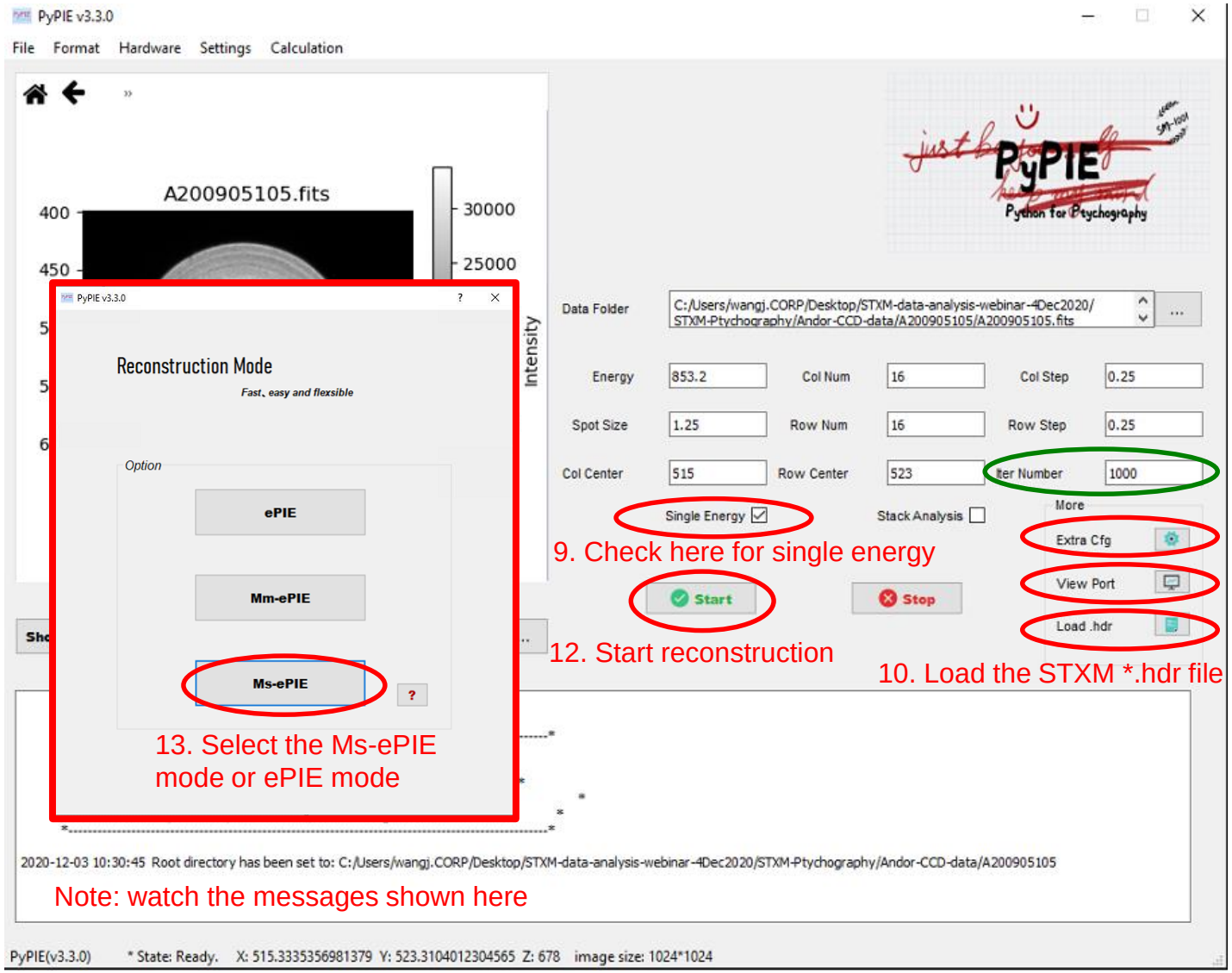


PyPIE – Single Energy Data Reconstruction





PyPIE – Single Energy Data Reconstruction



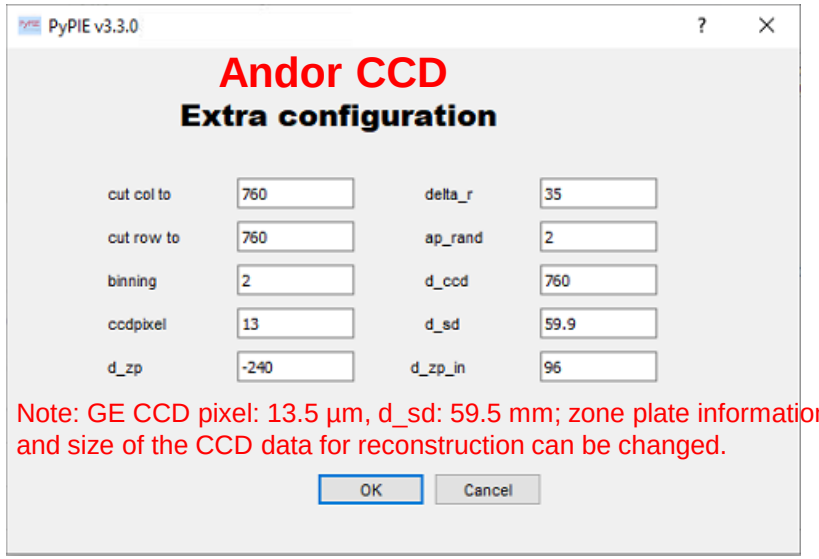
9. Check here for single energy

12. Start reconstruction

10. Load the STXM *.hdr file

13. Select the Ms-ePIE mode or ePIE mode

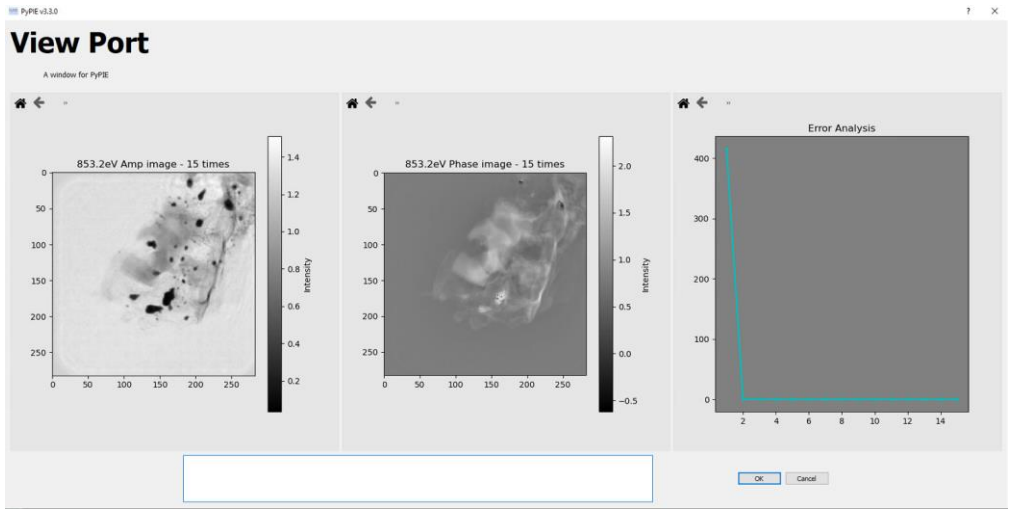
Note: watch the messages shown here



Note: GE CCD pixel: 13.5 μm, d_sd: 59.5 mm; zone plate information and size of the CCD data for reconstruction can be changed.

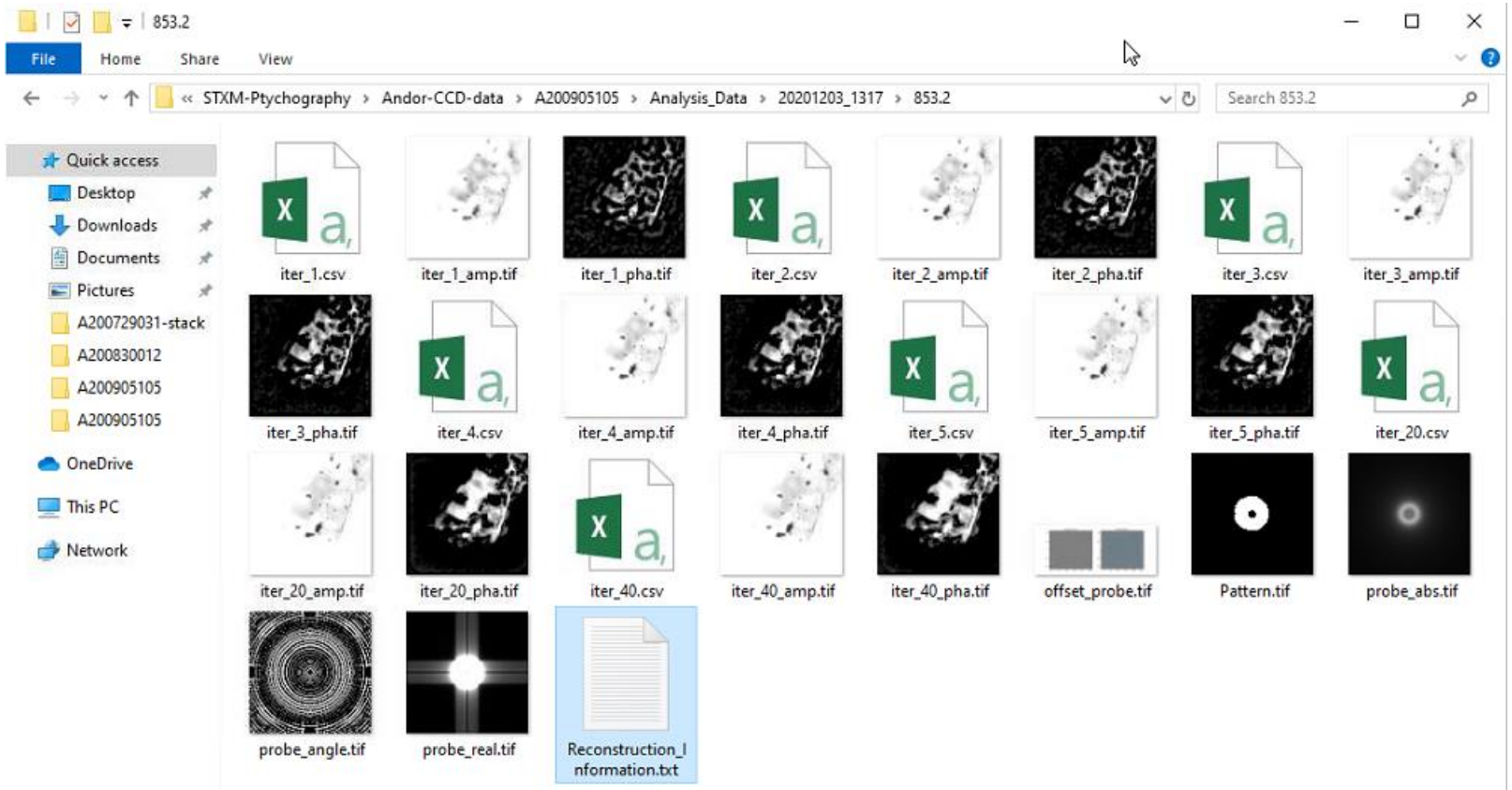
11. Check the extra configuration file

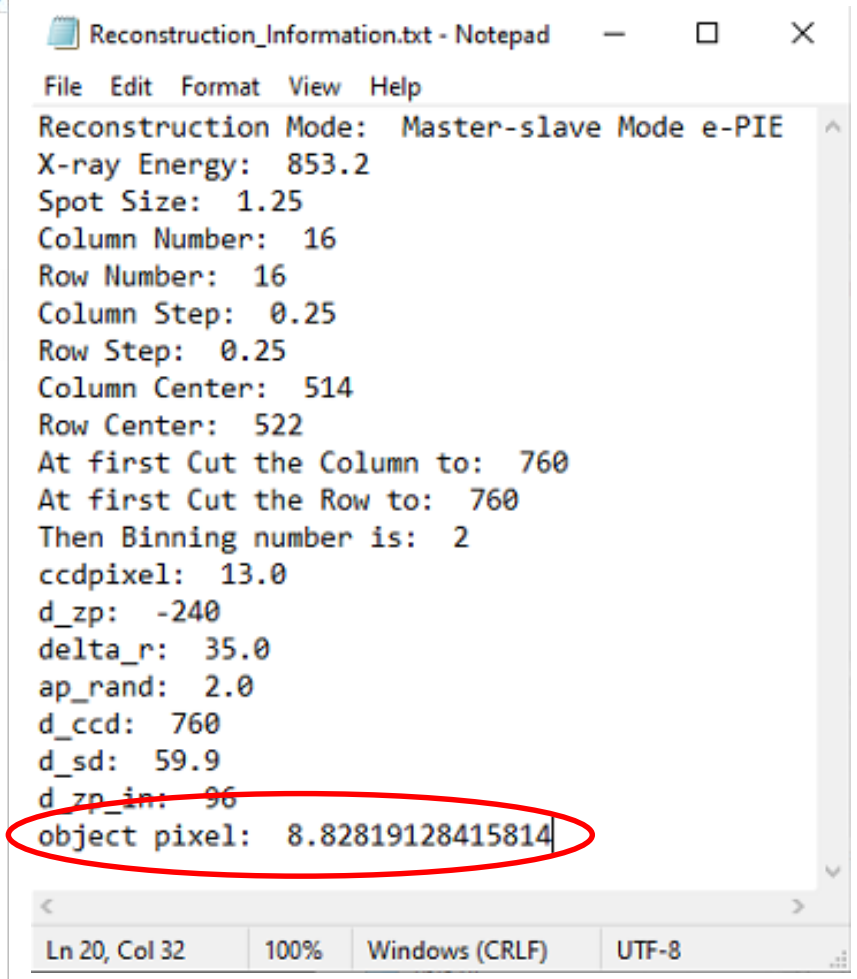
14. Click the View Port to check reconstruction result; reconstruction finishes at the set **iteration number, default 1000**.



Canadian Light Source / Centre canadien de rayonnement synchrotron

PyPIE – Reconstruction Output

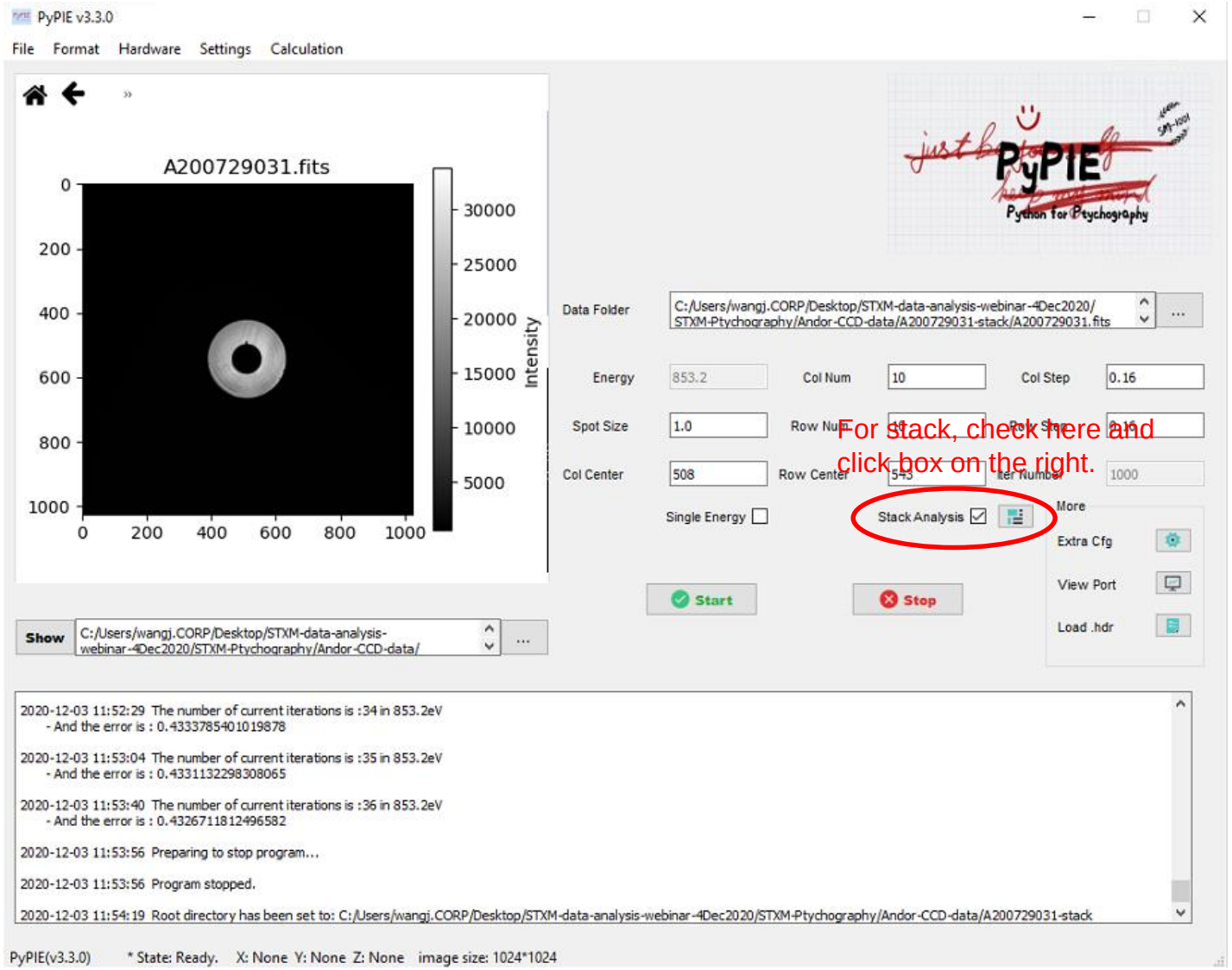




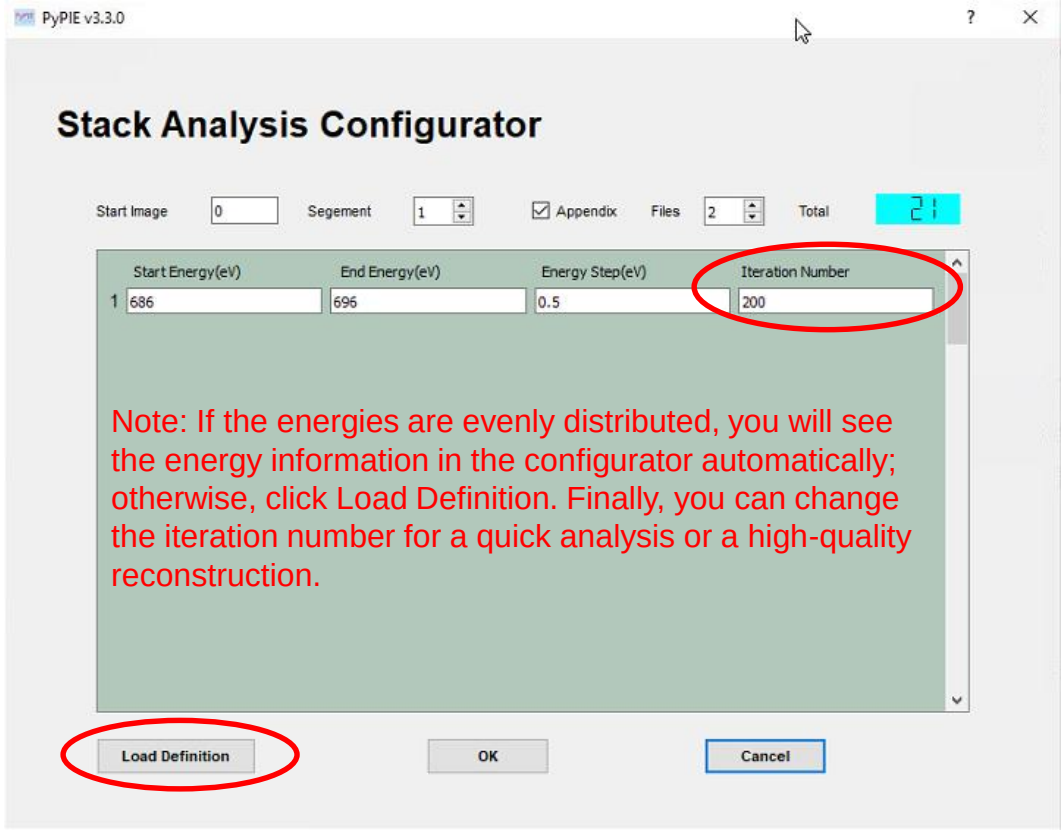
Note: reconstruction images and other files are saved at the raw data directory under the **Analysis_Data folder** and its sub **time-tag folders** and **energy folders**.

Note: object pixel size times total reconstructed pixels in each dimension to produce the actual physical size. But sometimes the object pixel size is half of the actual size due to data binning.

PyPIE – Stack Data Reconstruction



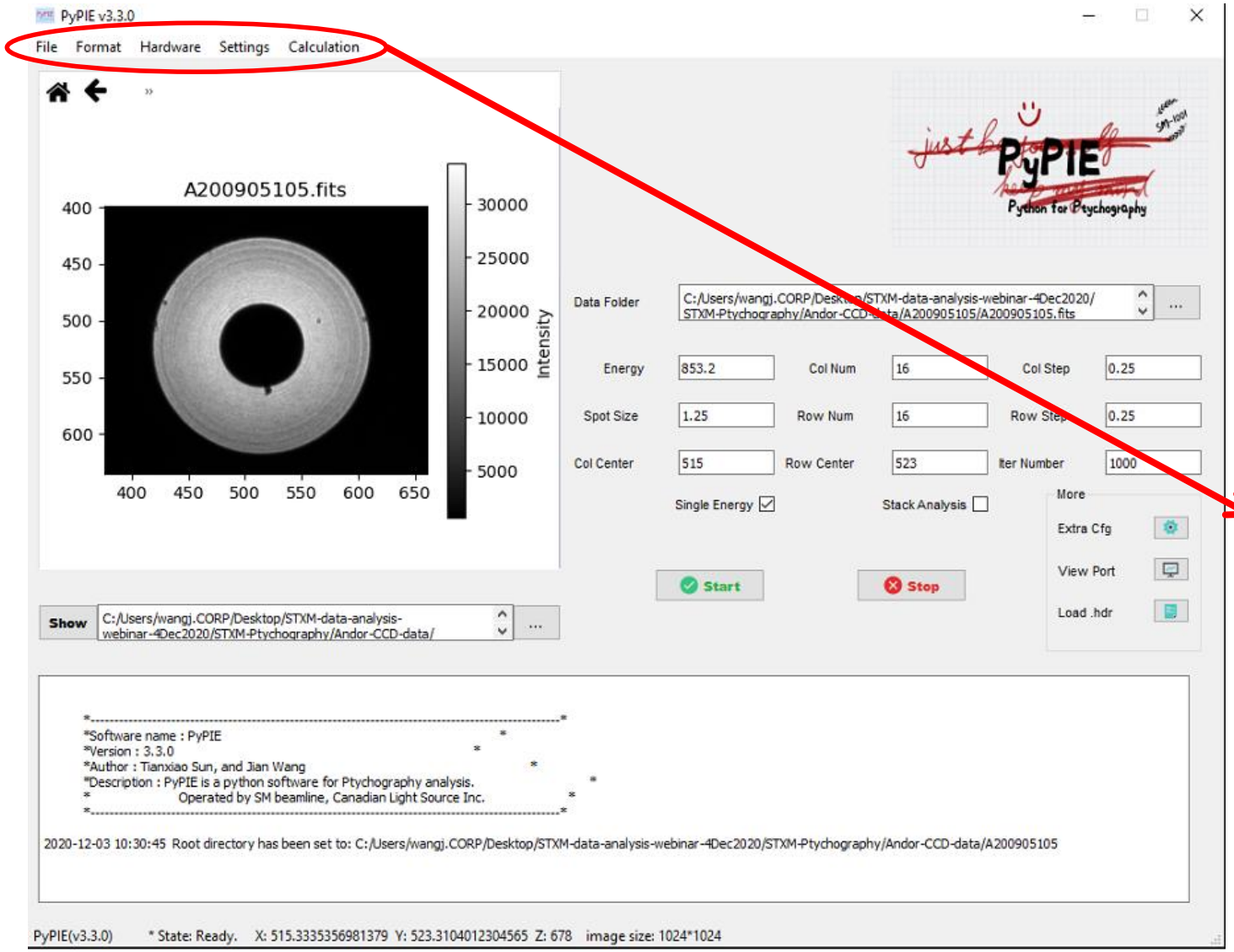
For stack, check here and click box on the right.



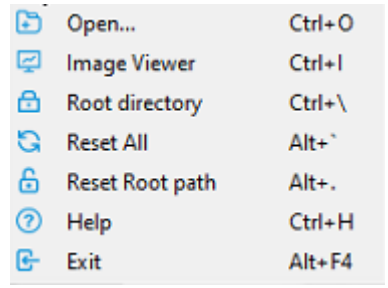
Note: If the energies are evenly distributed, you will see the energy information in the configurator automatically; otherwise, click Load Definition. Finally, you can change the iteration number for a quick analysis or a high-quality reconstruction.

Note: reconstruction images and other information files are saved at the raw data directory under the **Analysis_Data folder** and its sub **time-tag folders** and **energy folders**.

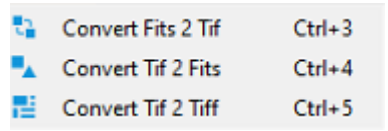
PyPIE – Other Useful Functions



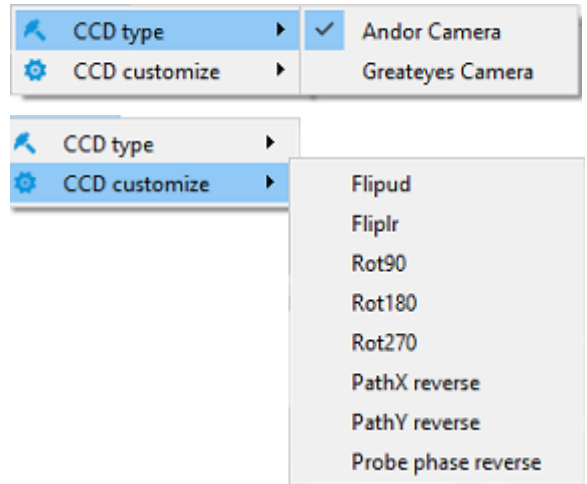
File



Format

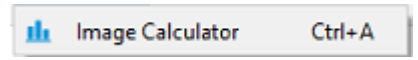


Hardware

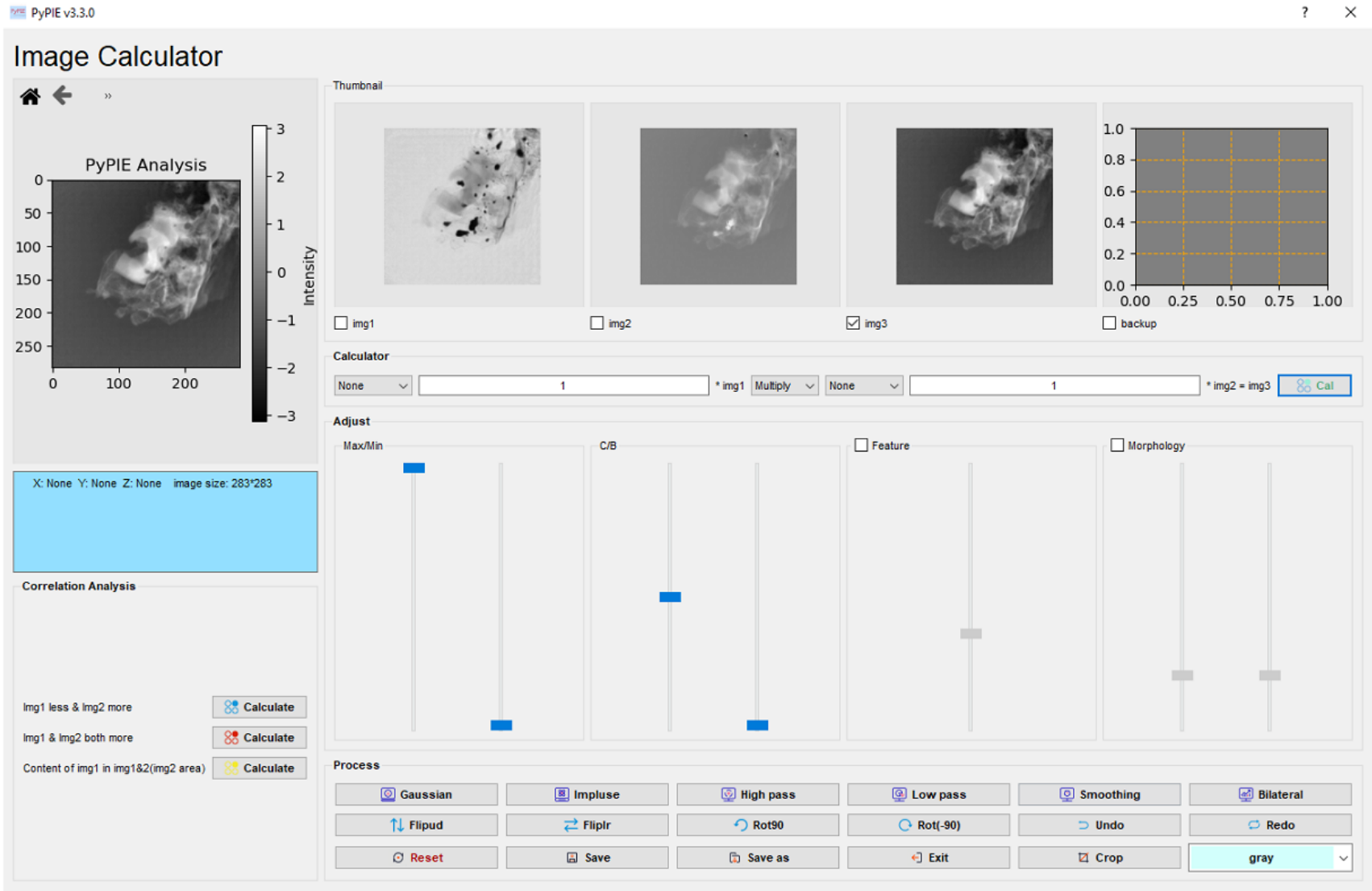


Settings

Calculation



PyPIE – Other Useful Functions

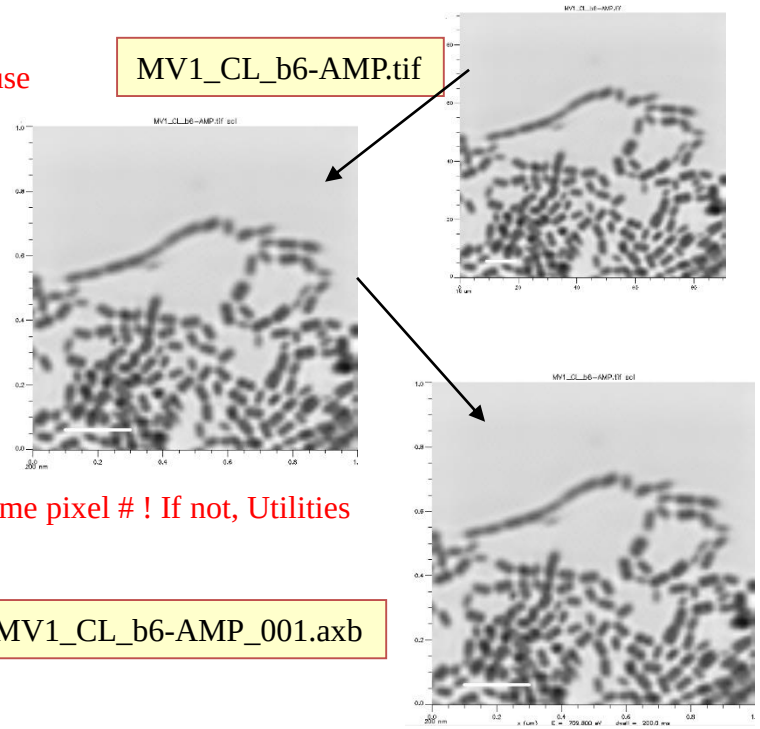


aXis2000 – Generating Ptycho Stacks

Convert a set of tif files to a stack starting from ptycho images reconstructed by PyPIE or SHARP (extracted from from the *.cgi file using e.g. recon_result_grey.py, on CLS-SM-ptycho workstation)

1. Read ~ Images ~ Graphics ~ tiff ~ data
(aXis200~Zoom, Numerical cut)

2. Images~set xy scale
(Use Images ~ Modify X,Y aXes, use object pixel to scale to the image default pixel size of 1µm/pixel)



3. Utilities ~ change energy

4. Utilities ~ change dwell

Make sure all images have the same pixel # ! If not, Utilities ~ Change mesh

5 Write~axis

6. Do the above for each tif file in the stack

7. Prepare a *.sl (stack list file)

```
E:\data\XRM\Soleil\2019\19-06\06-17\30\
SS30_1.axb
SS30_2.axb
SS30_3.axb
SS30_4.axb
SS30_5.axb
SS30_6.axb
SS30_7.axb
SS30_8.axb
SS30_9.axb
SS30_10.axb
SS30_11.axb
```

8. Stacks ~ Analyze ~ stack list input

9. Write out as a stack (*.dat, *.ncb)

10. Process as required (alignment, convert to OD, etc, etc)