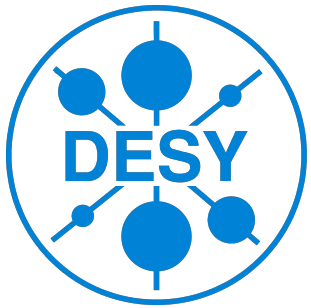
Experience with Data Analysis Programs: DAWN, DPDAK and Mantid

Gero Flucke

FS-EC



PETRA III Commissioning Meeting
April 2nd, 2014

Outline

- Introduction
- Overview of Analysis Programs:
  - DAWN
  - DPDAK
  - Mantid
- Summary

Introduction

- Data rates in modern X-ray experiments are rising.
  - $\Rightarrow$ Raw data might not be “exportable” to users’ institutes.
- Groups with little X-ray physics background less experienced with “standard” data analysis.
- Standardisation of data format envisaged: NeXus.
- How to react at DESY?
  - $\Rightarrow$ Provide central data processing?
  - $\Rightarrow$ Provide “standard” data analysis tools?

$\Rightarrow$ Investigate existing analysis tools!

Note: Within the framework of the
Photon **a**nd **N**eutron **data** infrastructure initiative.
http://pan-data.eu

Investigation of Analysis Programs

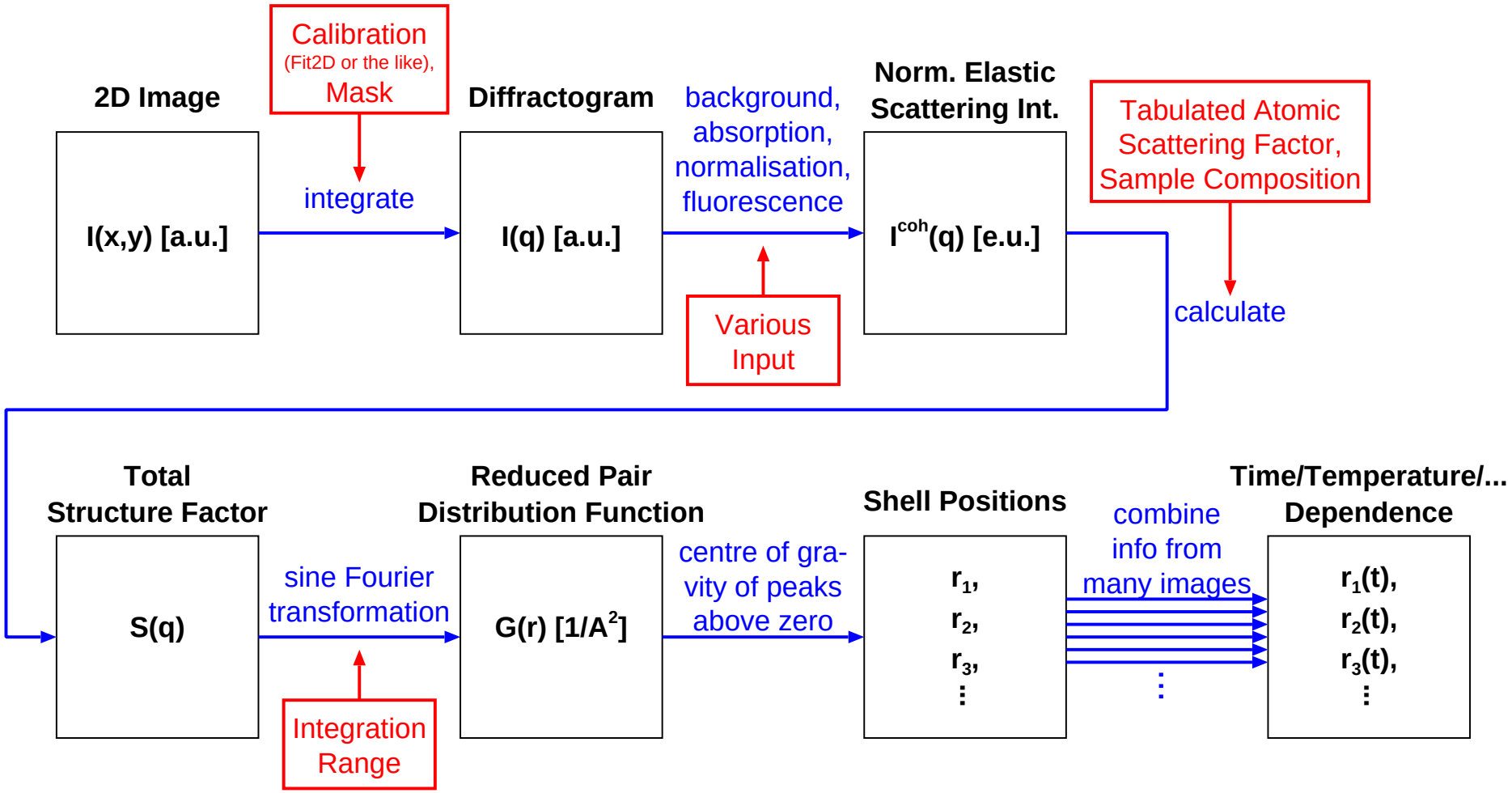
- Candidates: DAWN, DPDAK, Mantid.
- See what the programs provide,
  - but need analysis guidance,

  $\Rightarrow$ example scenario:
  Real space analysis of amorphous material,
  - thanks to J. Bednarcik (P02.1) for data and beam time!
- All programs share Python as analysis scripting language:

  $\Rightarrow$ write/(re)use tool independent code base in all programs!
- Notes:
  - My physics background is outside X-ray analysis, being familiar with ROOT (http://root.cern.ch) for analysis,
  - I spent more time on DAWN than on the other programs.

Real Space Analysis of Amorphous Material

- Changing conditions: temperature, pressure,. . .
- Capable of online monitoring.



DAWN: **D**ata **A**nalysis **W**orkbe**N**ch

- Eclipse based program $\Rightarrow$written in Java (requires Java 1.7).
- "Implements sophisticated support for:
  - Visualization of data in 1D, 2D and 3D,
  - Python script development, debugging and execution,
  - Workflows for analyzing scientific data calling Python and binary codes."
- Main developers from Diamond and ESRF.
- Open source: EPL or Apache license (getting rid of (L)GPL).
- Open to external collaborating developers.
- "By and for the synchrotron community - overlap with other communities like neutron scattering, photon science, etc."
- Supported platforms:
  Windows, Mac OS (beta version only), Linux.

http://www.dawnsci.org
http://hasyweb.desy.de/services/computing/dawnDocu

DAWN

Concept

- GUI is based on “Perspectives” and “Views”.
- View: tab in a frame of the DAWN window, e.g.
  - a plot, contents of a file/directory, a tool...
- Perspective: groups views suitable for some task, e.g.
  - data browsing, python scripting,. . .
  - user can add or remove views.
- User’s data organised in “Projects”.
- Projects and status stored in “Workspace” (default: `$HOME/workspace`).
- Many “Cheat sheets”: built-in tutorials.
- Feedback button – tell the developers
  - what does not work,
  - what would be nice to have.

DAWN

Note: mostly tested with version 1.4.0 - there is 1.4.1 and soon 1.5.0.

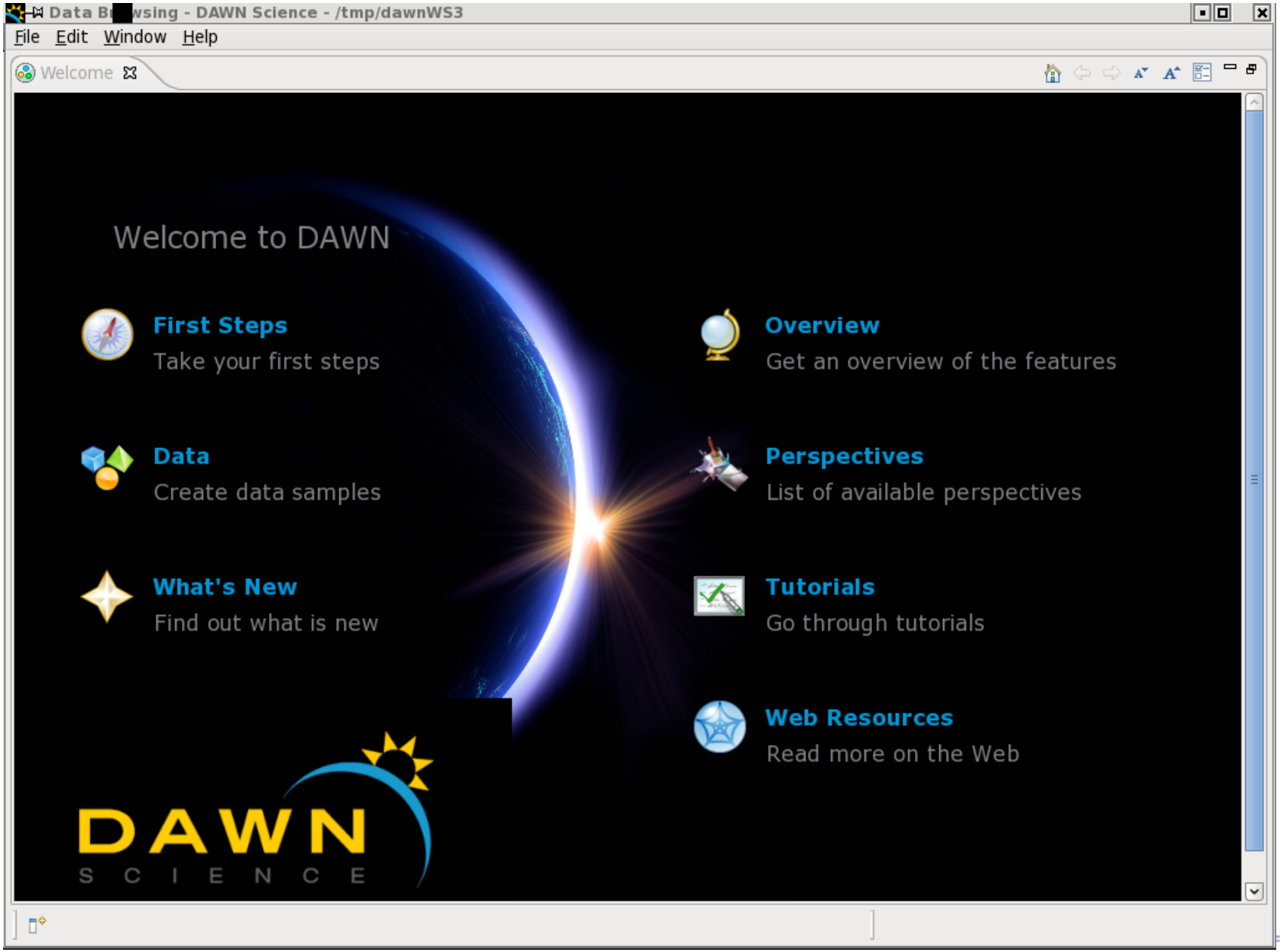
Many Interactive Features

- Browsing many data formats: Nexus/HDF5, images, .edf, ascii, ...fio (for DESY version)
- Investigate multi-D datasets:
  - 1D curves: value vs index or vs value of other dataset
    - 'value' can be expression of datasets.
  - stack of lines, stack in 3D
  - slices of multi-D datasets,
  - 2D: as surface, as image (featureful colour mapping [for non-RGB images], pixel 'averaging')
  - various nice zoom/panning features,
- Masking: creation, storage, automatic application. . .

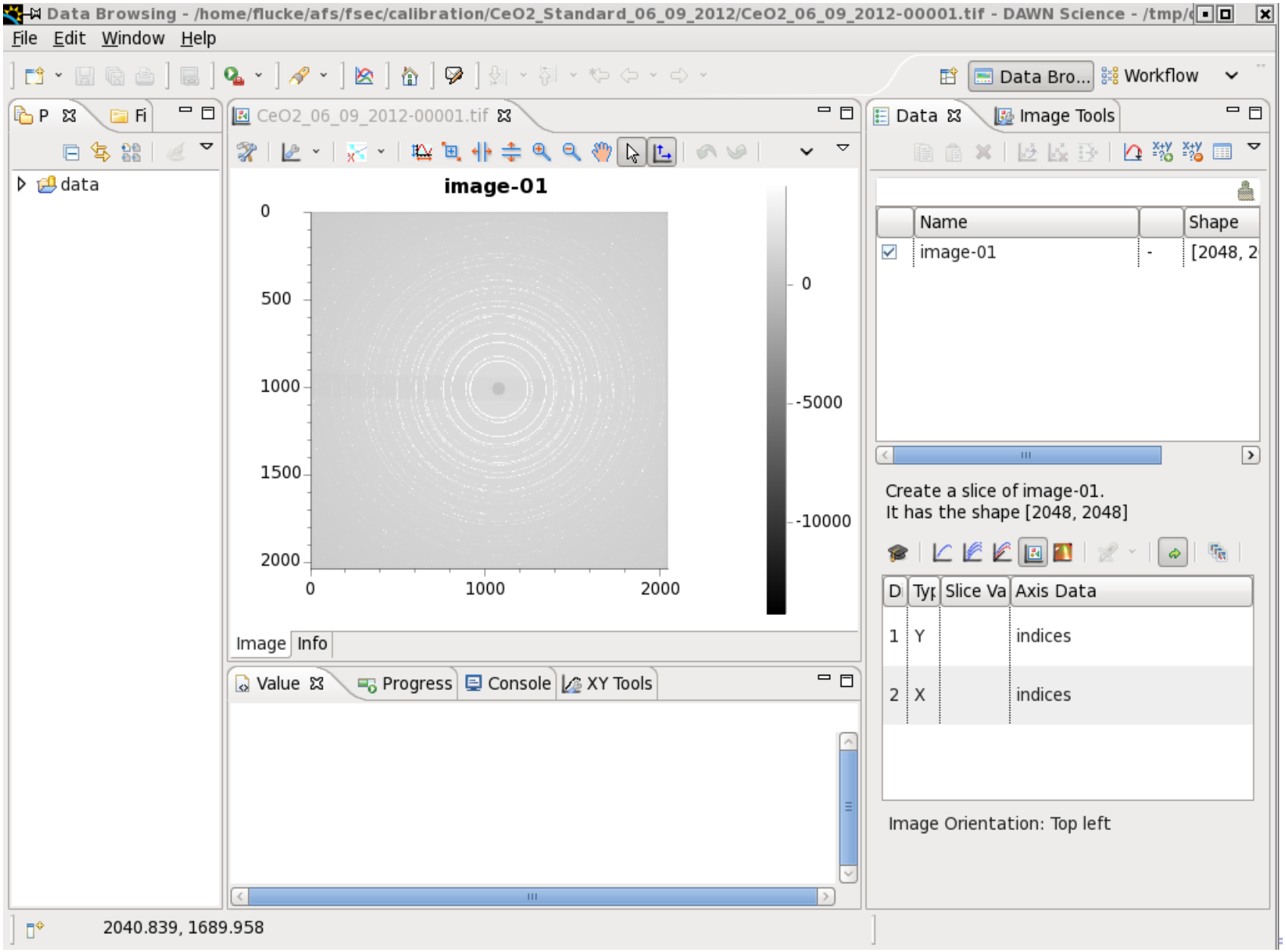
Many Interactive Features (ctd.)

- Distribution fitting
  - finding and fitting peaks,
  - fitting polynomials,
  - arbitrary function fitting not working (but will be in 1.5),
  - no fitting of 2D.
- Region of interest and profiles: line, box, sector,...
- Specific scientific methods:
  - calibration for 2D $\rightarrow$ 1D integration (Fit2D) will be in 1.5,
  - many methods go as perspectives into Diamond version (ESRF?).
- Time needed to find out how things work (as to be expected).
- Things do not always work as they should,
  - but feedback highly appreciated by development team!

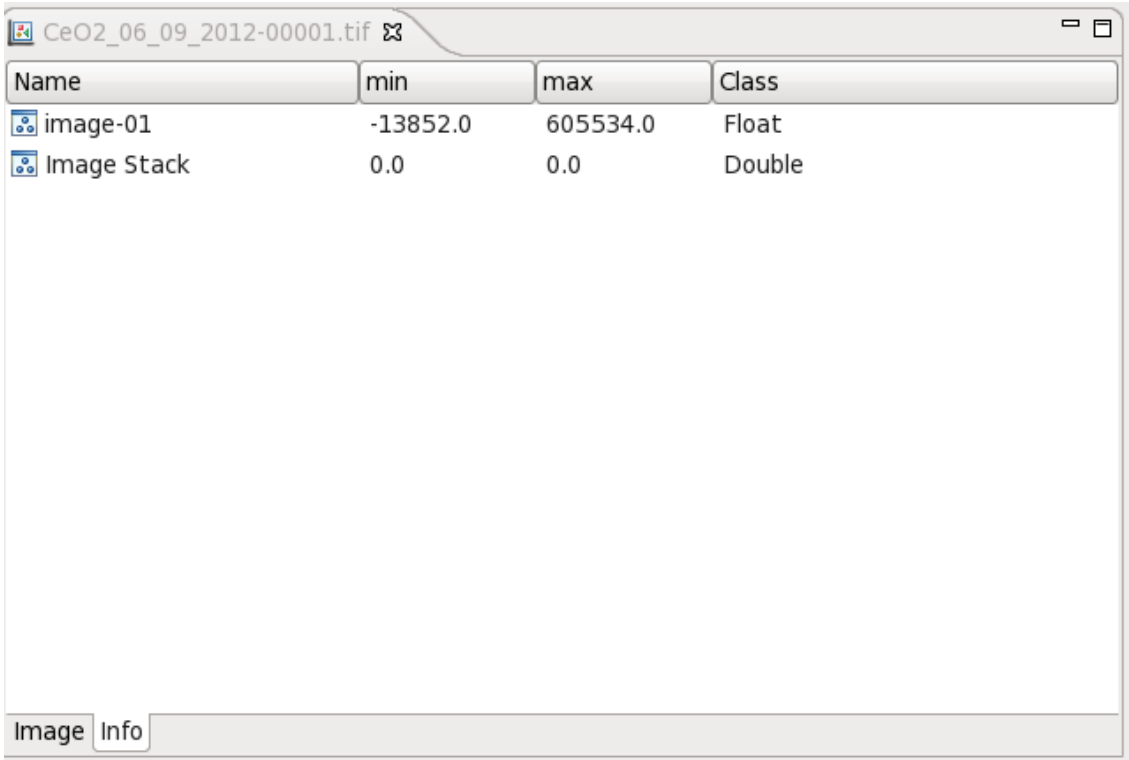
Guided Tour through DAWN: First Screen



DAWN: Image Browsing

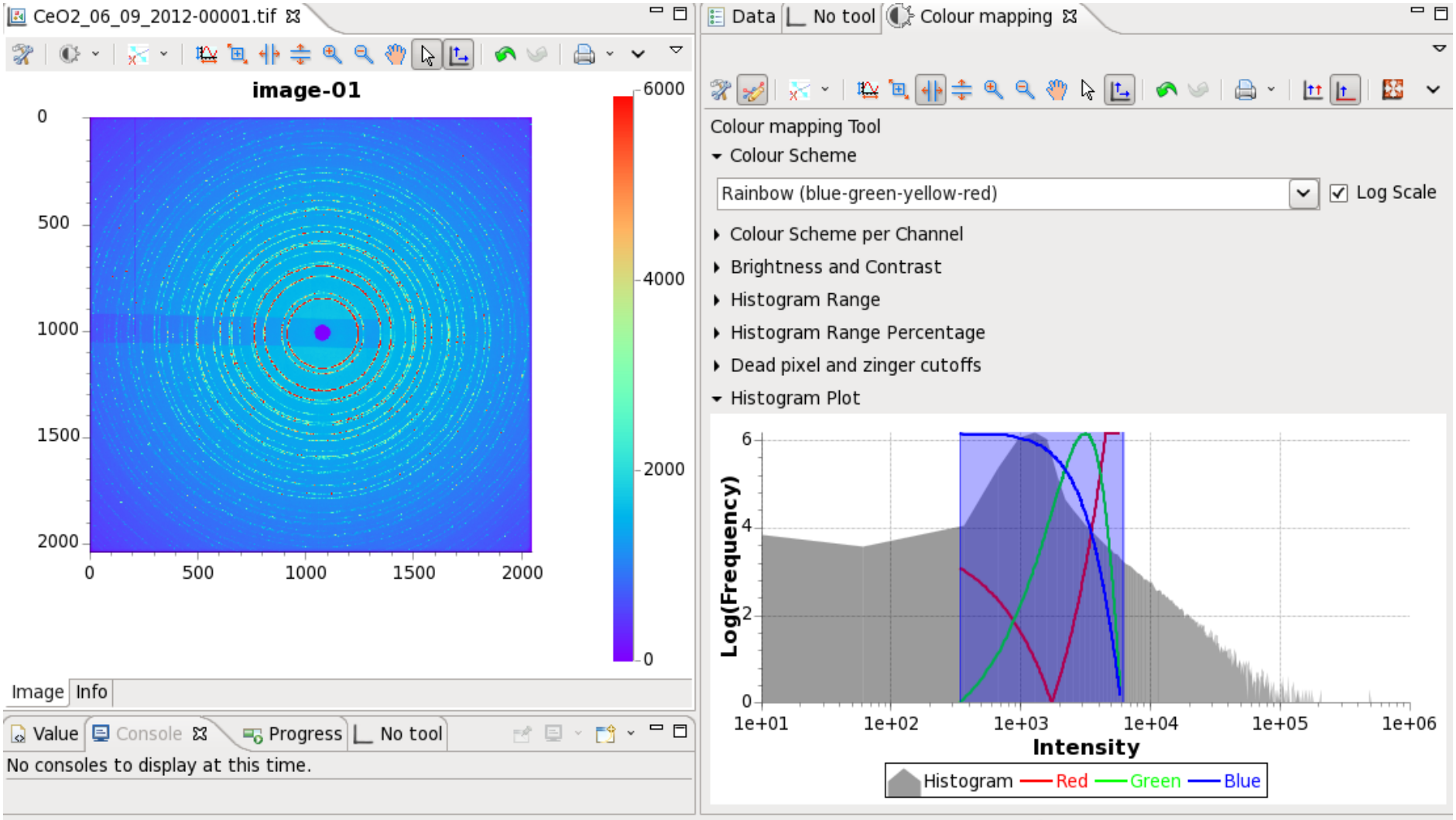


DAWN: Image Browsing



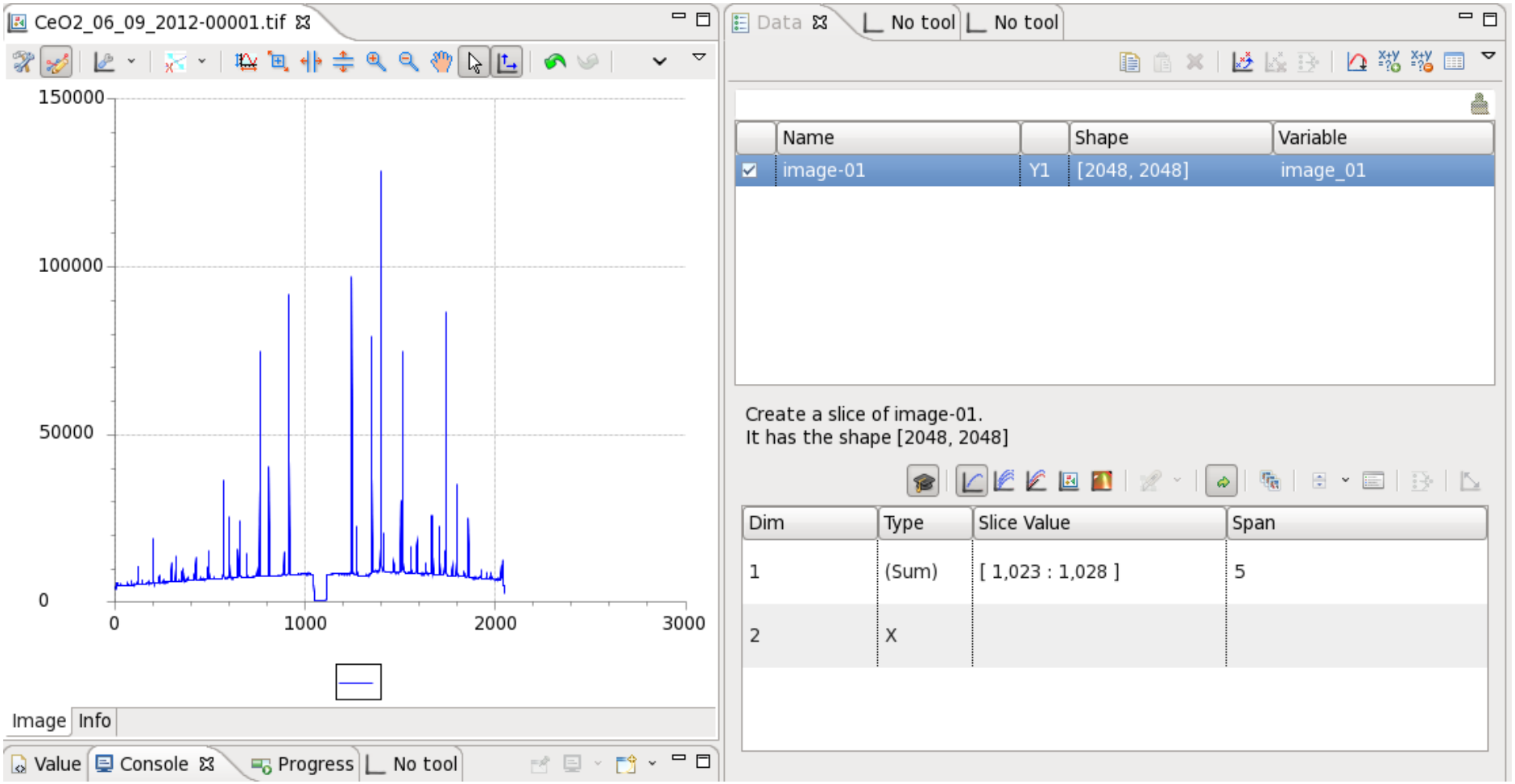
Easily check range of values in (here: image) dataset.

DAWN: Colour Scale



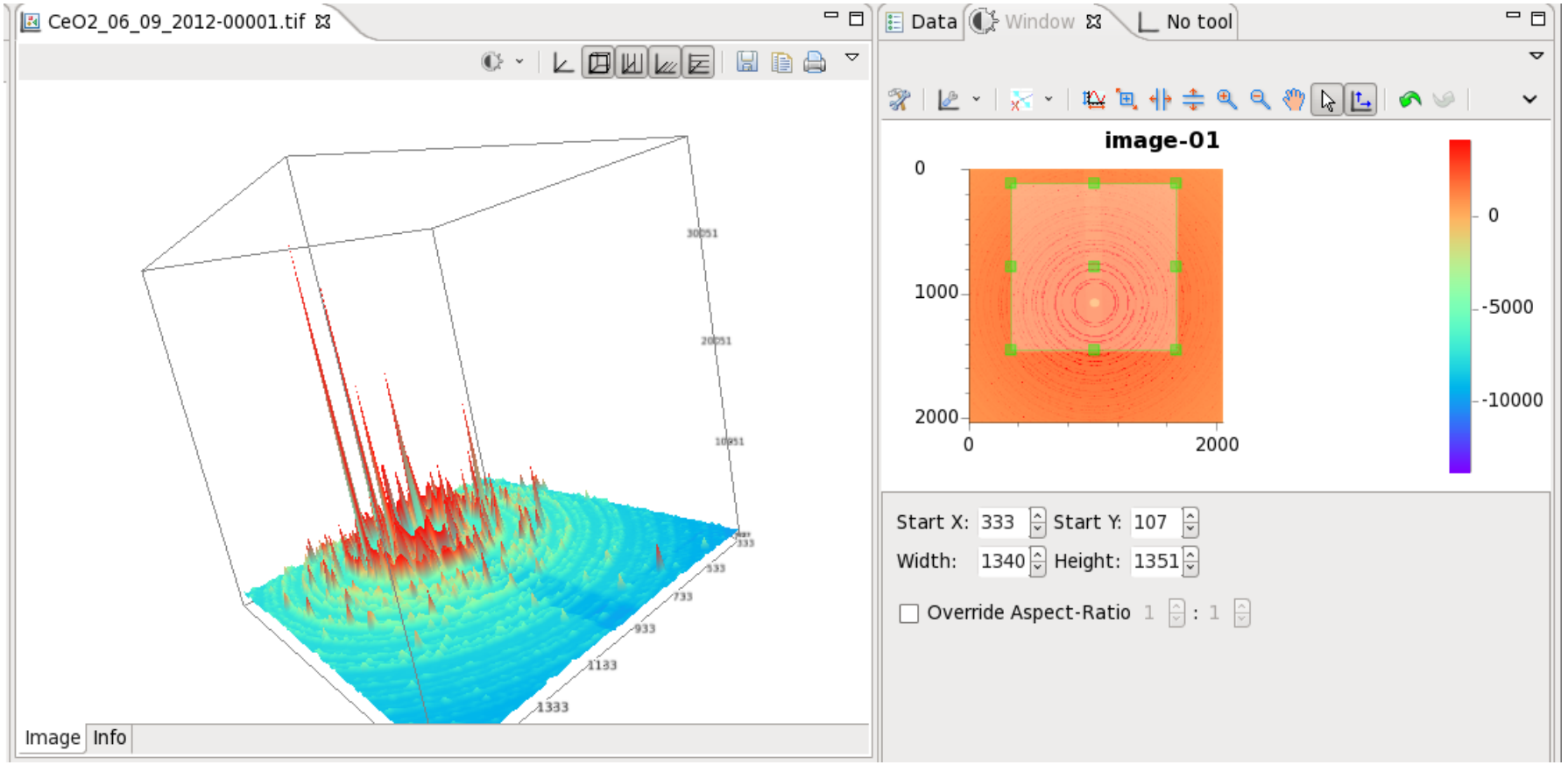
Full interactive control of colour mapping and outliers to ignore.

DAWN: Slice Datasets



Here: Sum of 5 consecutive slices.

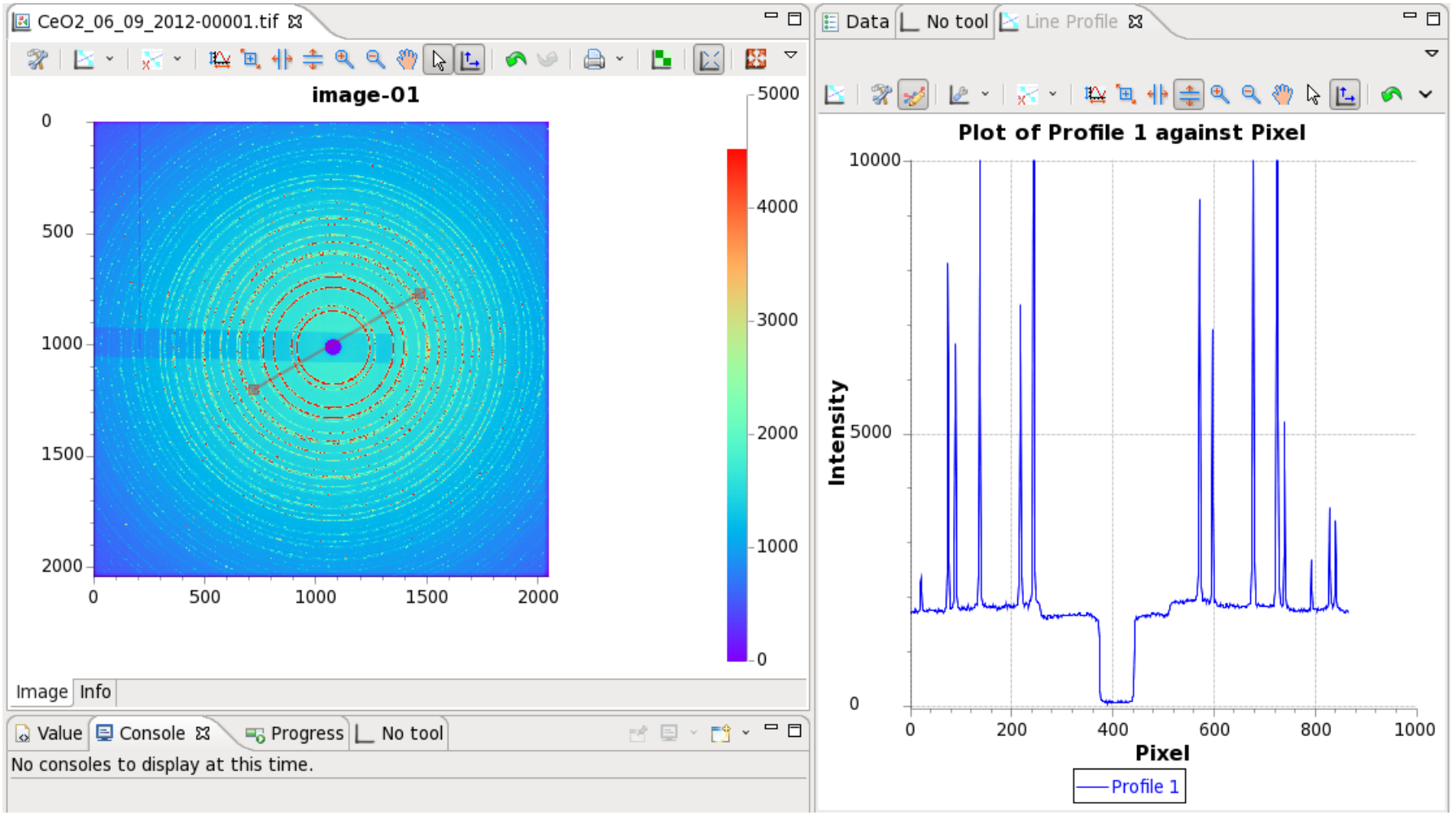
DAWN: Surface Plot



Live 3D view (instantly updated):

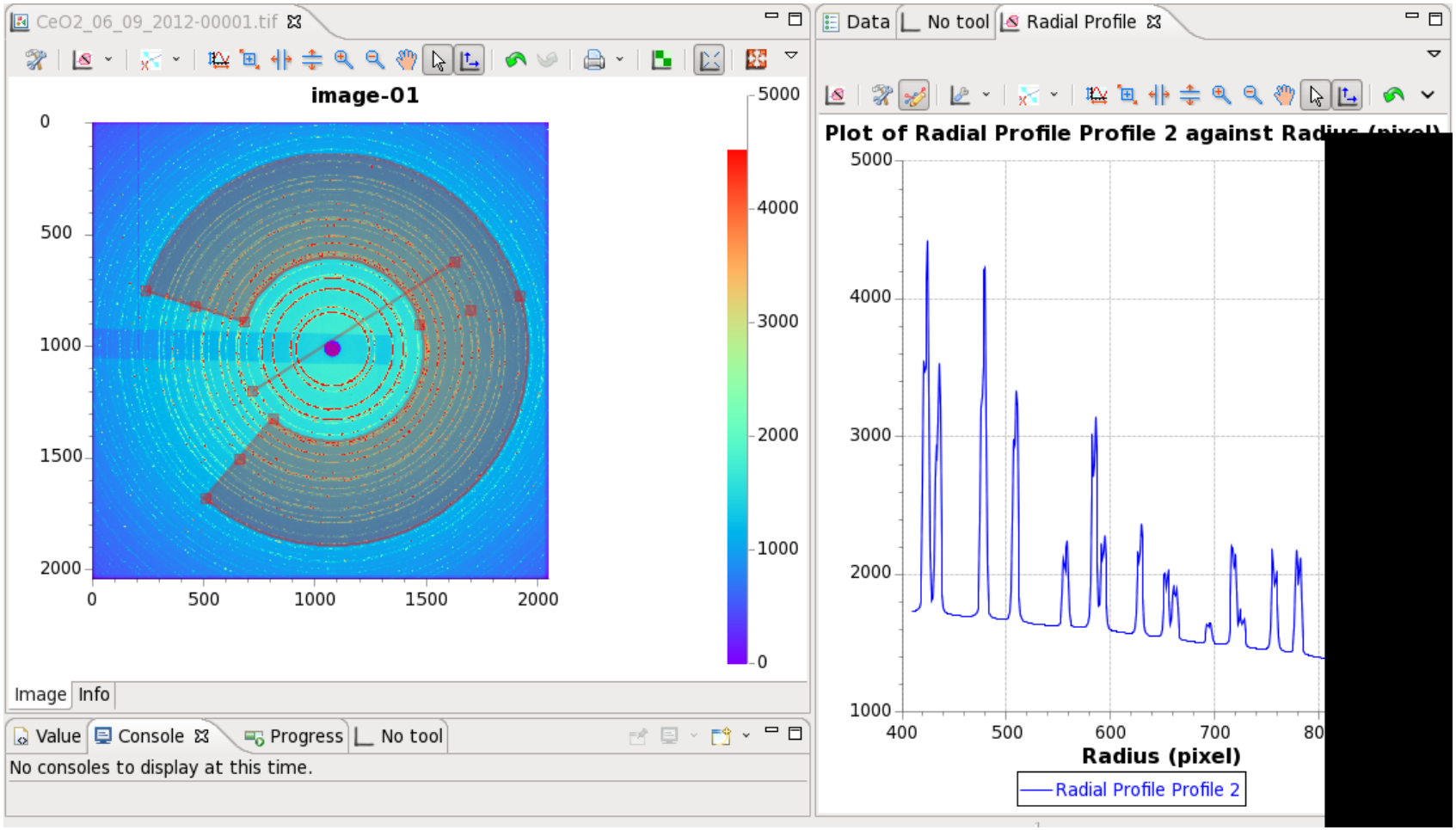
- Rotate as you want.
- Select a box

DAWN: Line Profile



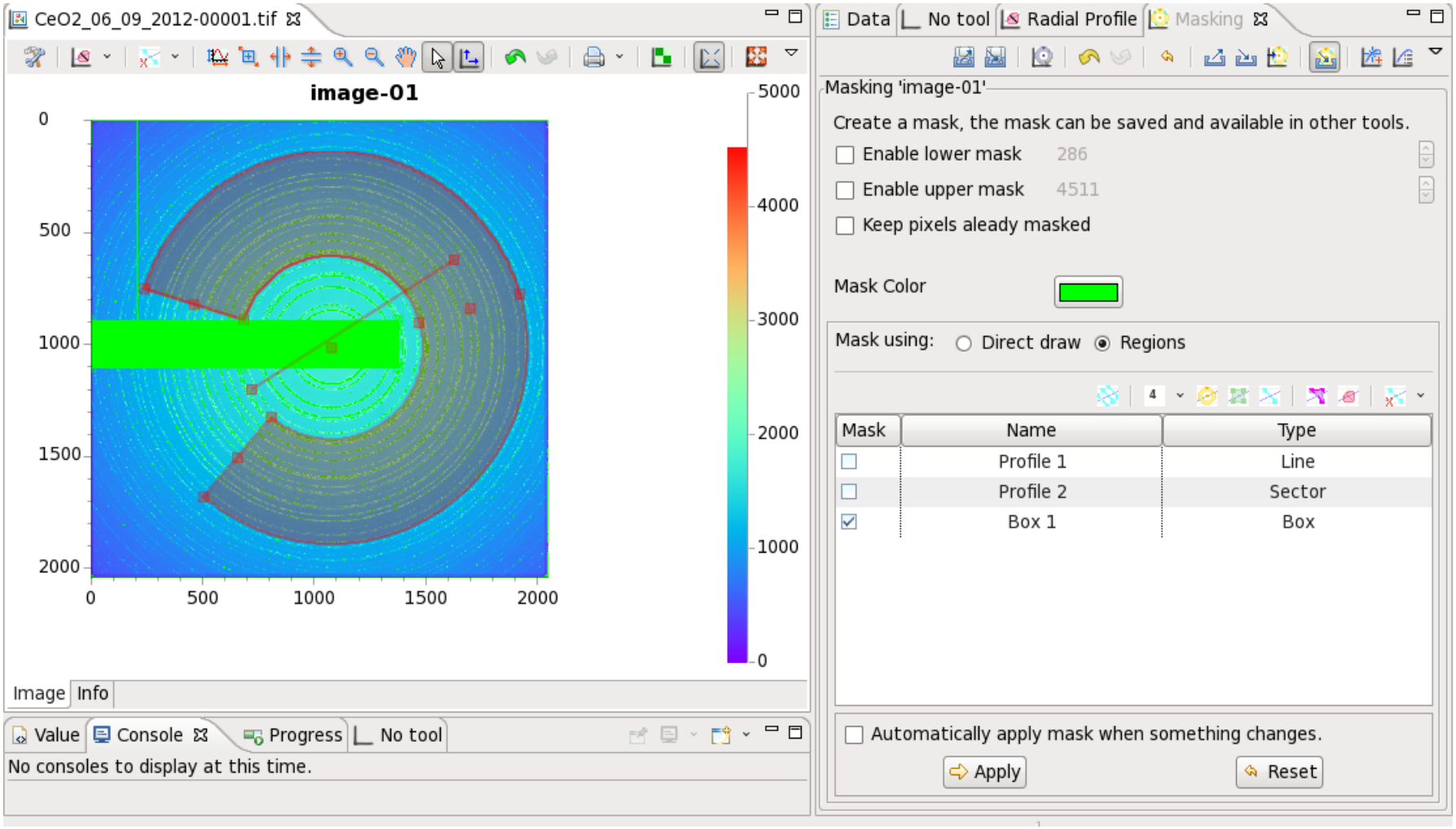
1D plot updated instantly if mouse moves line.

DAWN: Radial Profile



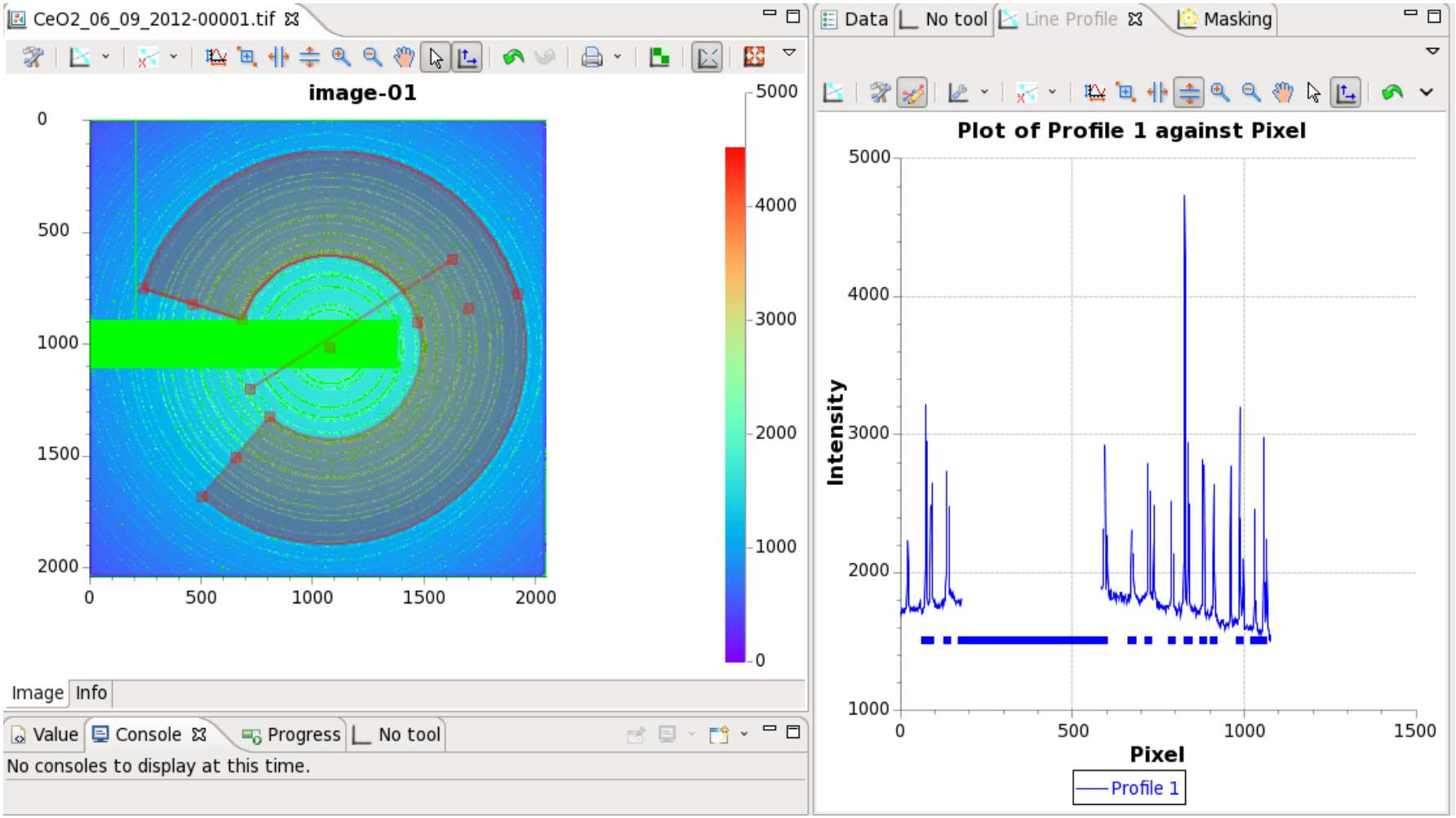
Other profiles available as well, e.g. azimuthal profile.

DAWN: Masking



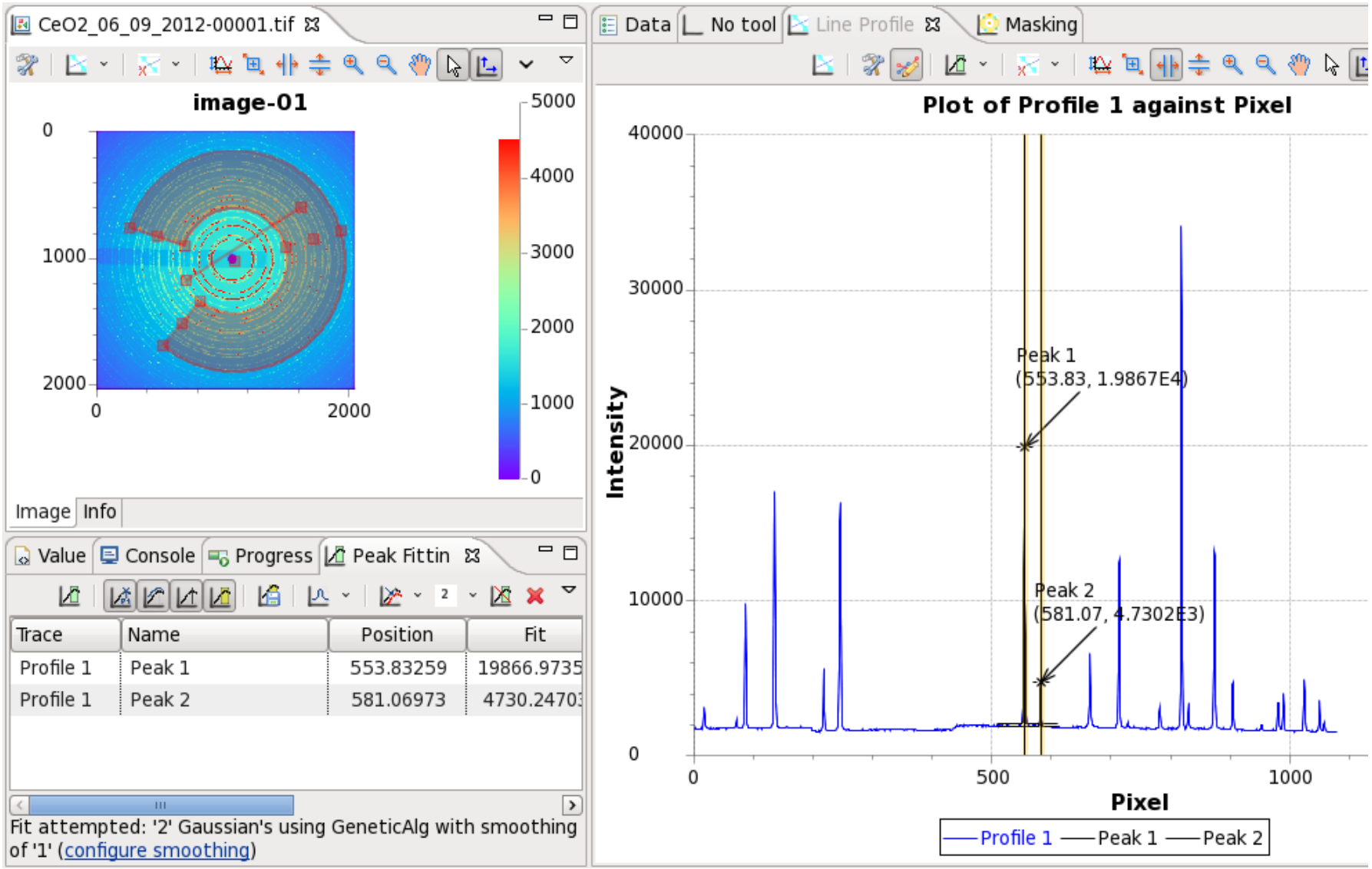
Choose regions, upper/lower masks, 'by hand',...

DAWN: Line Profile Masked



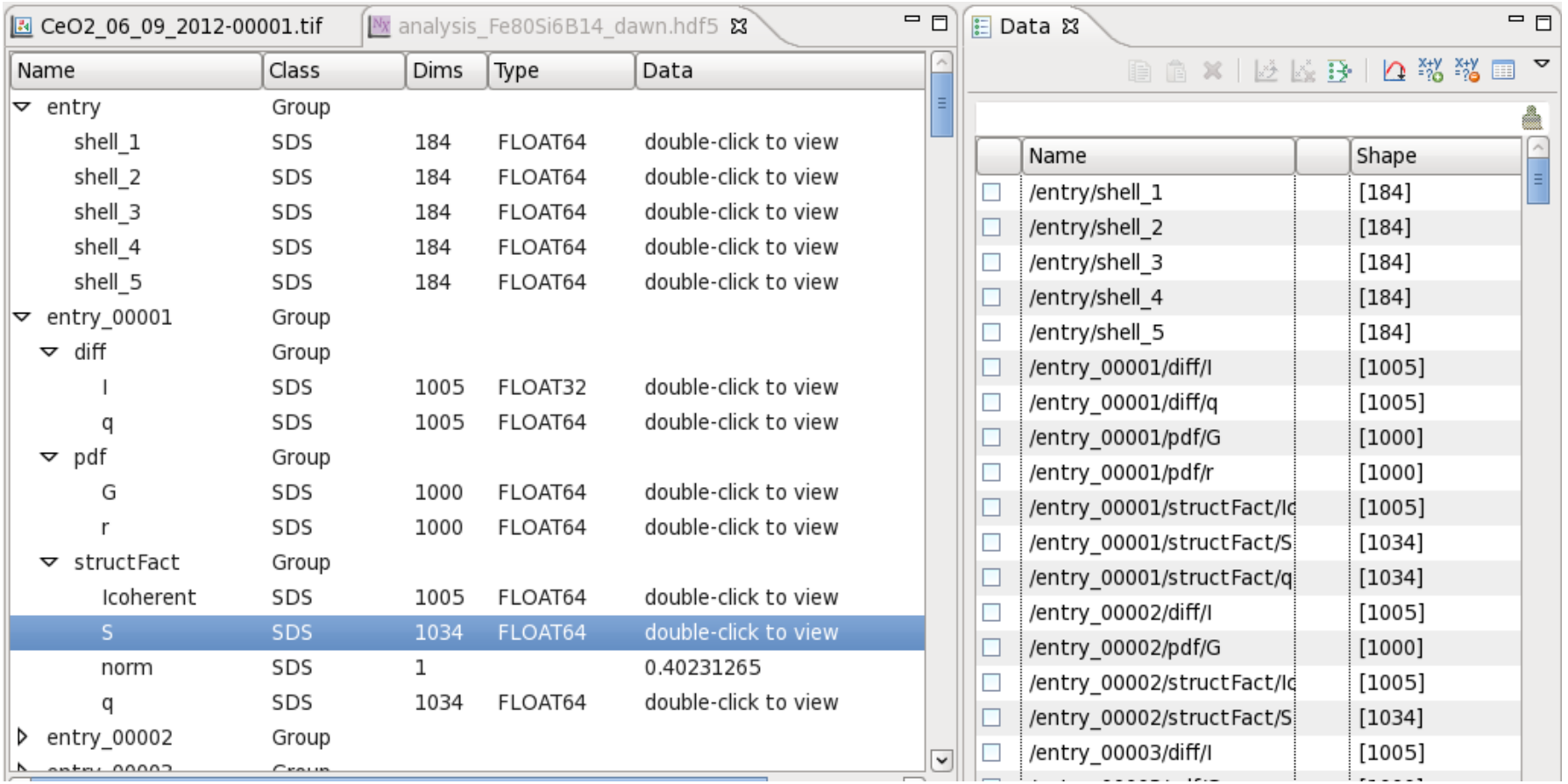
Mask taken into account in profiles.

DAWN: Peak Fitting



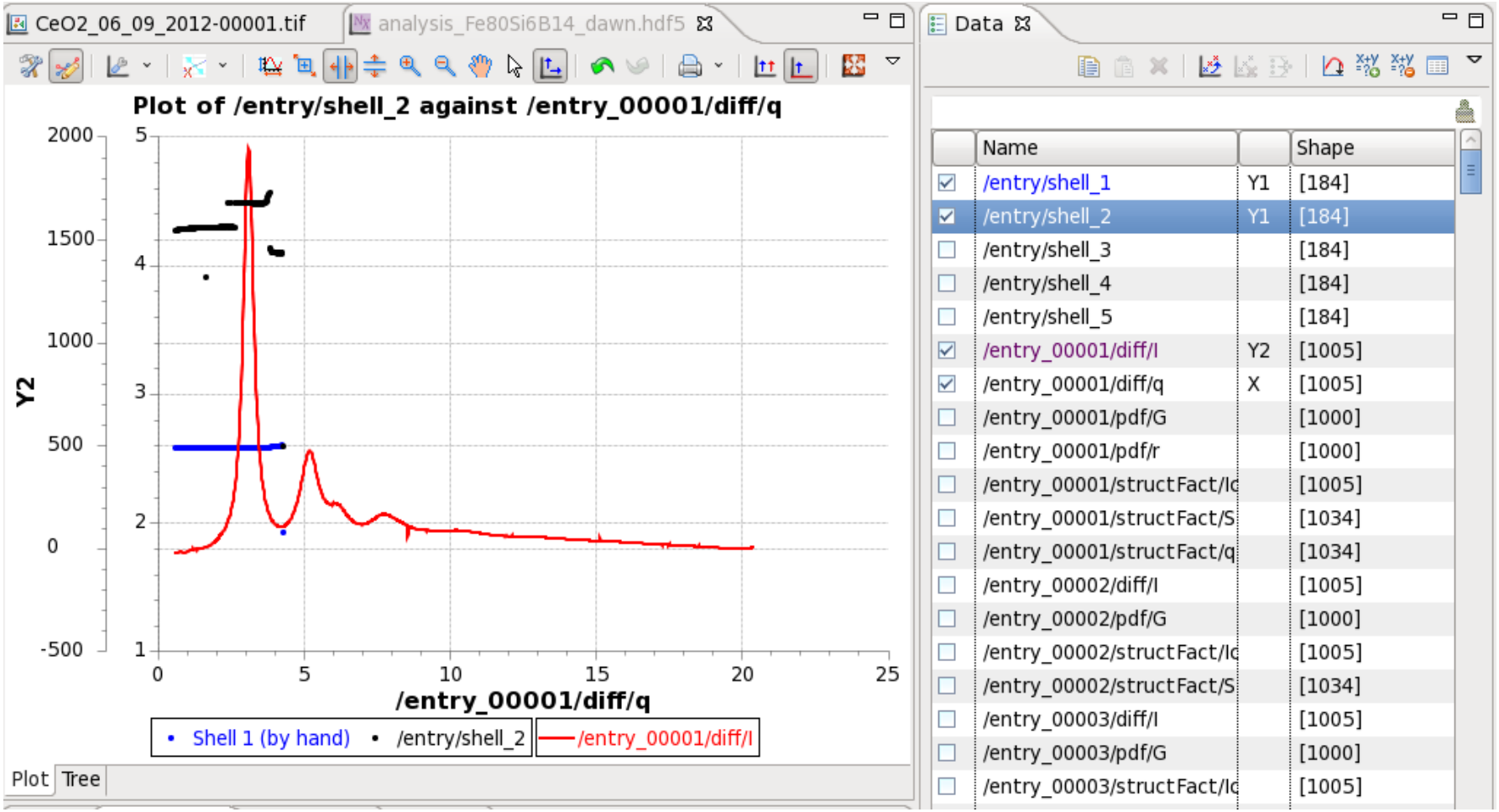
Interactively define range and maximum number of peaks to be fitted (can be many!).

DAWN: HDF5/NeXus Browsing



Easy browsing of HDF5/NeXus file content.
(But it does not tell you what is a link. . .)

DAWN: HDF5/NeXus Browsing



Simple selection what to draw as X, Y1 and Y2.
(Non-default line styles a bit more complicated to achieve.)

DAWN: Python - Scripting Interface for Users

- Interactive prompt (using IPython).
- Nice script development environment (Eclipse and PyDev)!
- (C)Python: usual Python ("reference implementation"), implemented in C.
- Jython: Same language, implemented in Java.

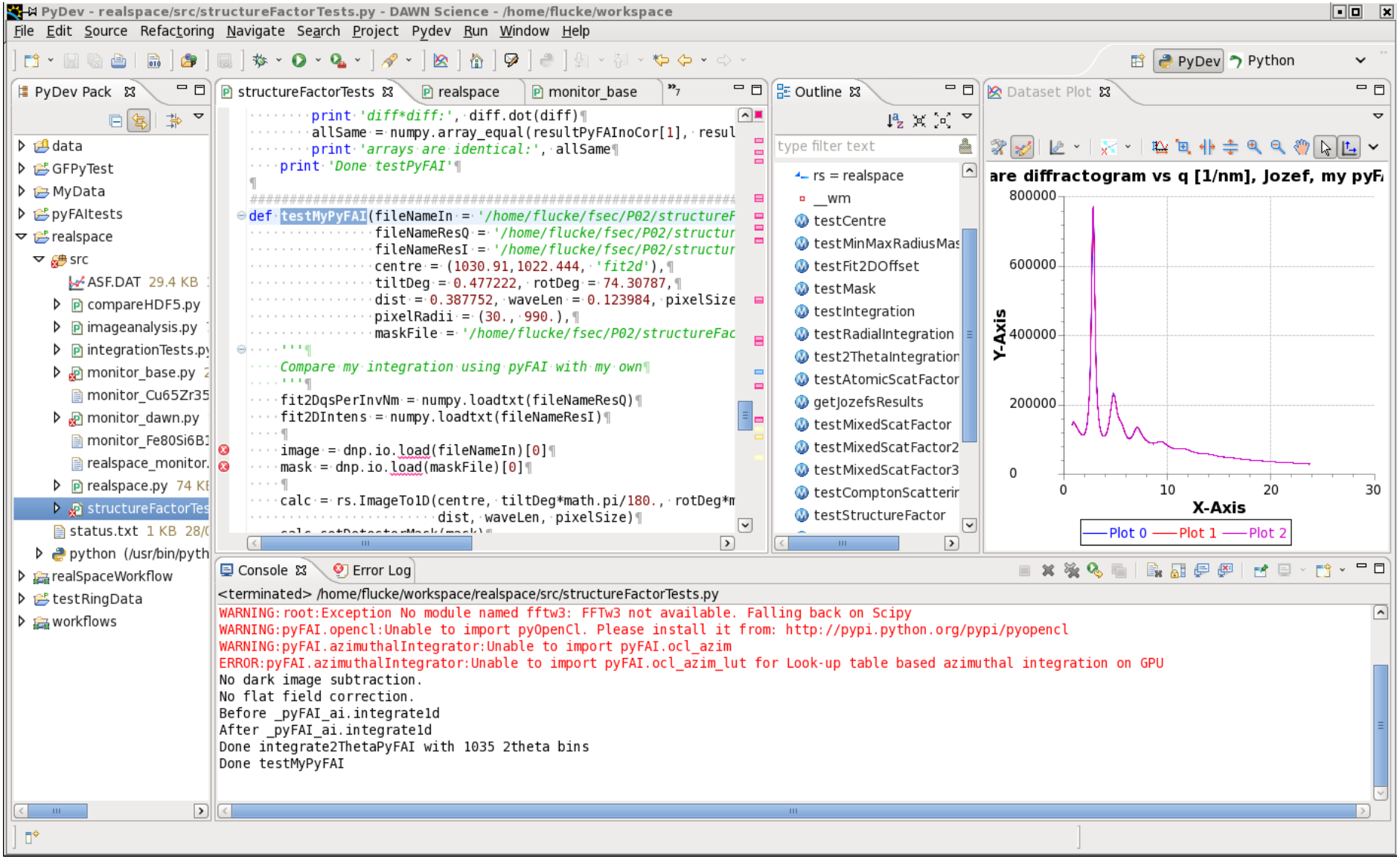
(C)Python in DAWN

- Access to plotting, file access for various formats.
- Use standard python tools - e.g. for function fitting.
- Access to DAWN Java code possible via `py4j` (expert action).

Jython in DAWN

- Access to plotting, file access for various formats, fitting.
- No standard (C)Python tools available!
- Simple to use any DAWN Java code?

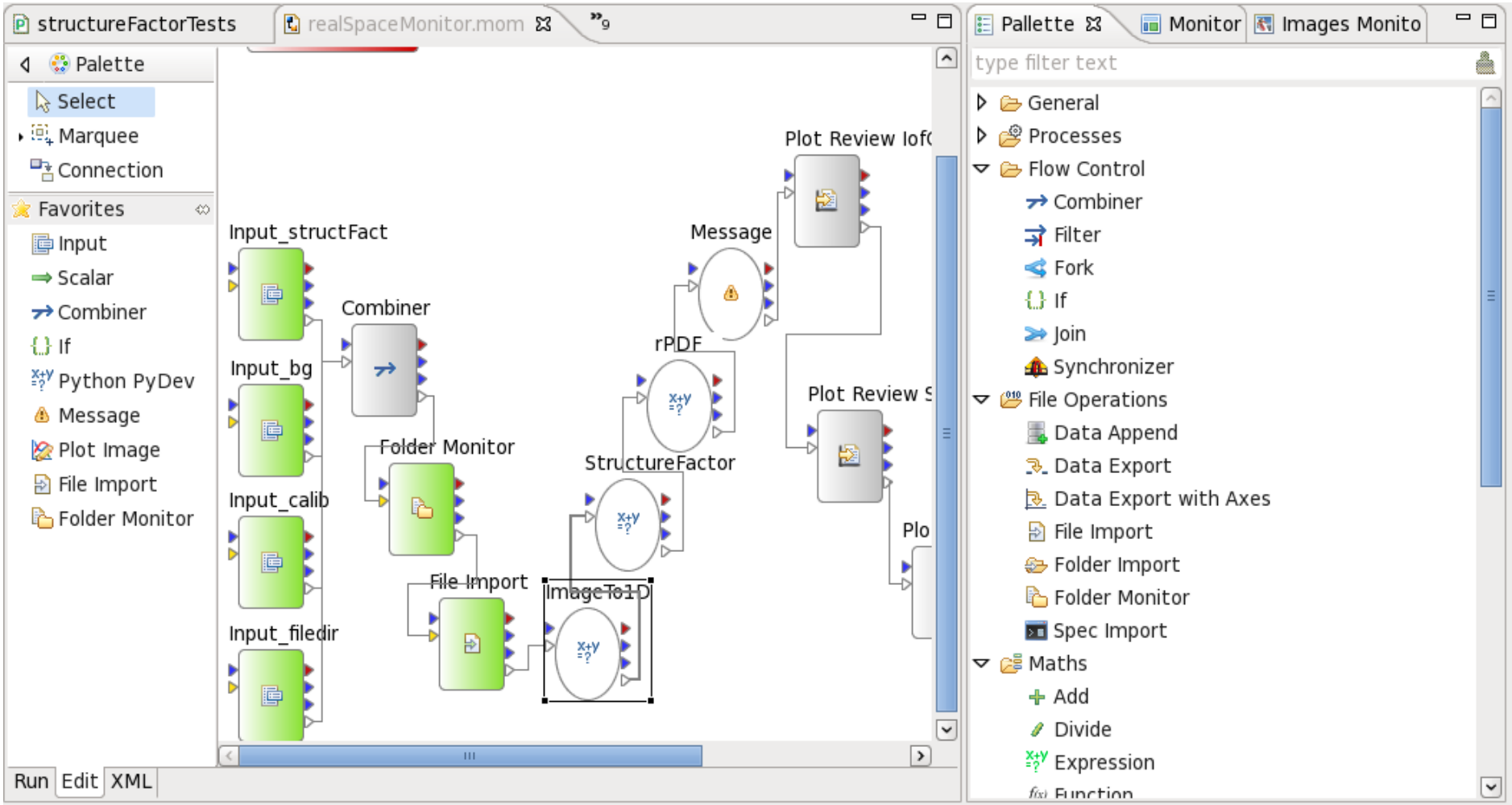
DAWN: Python Development



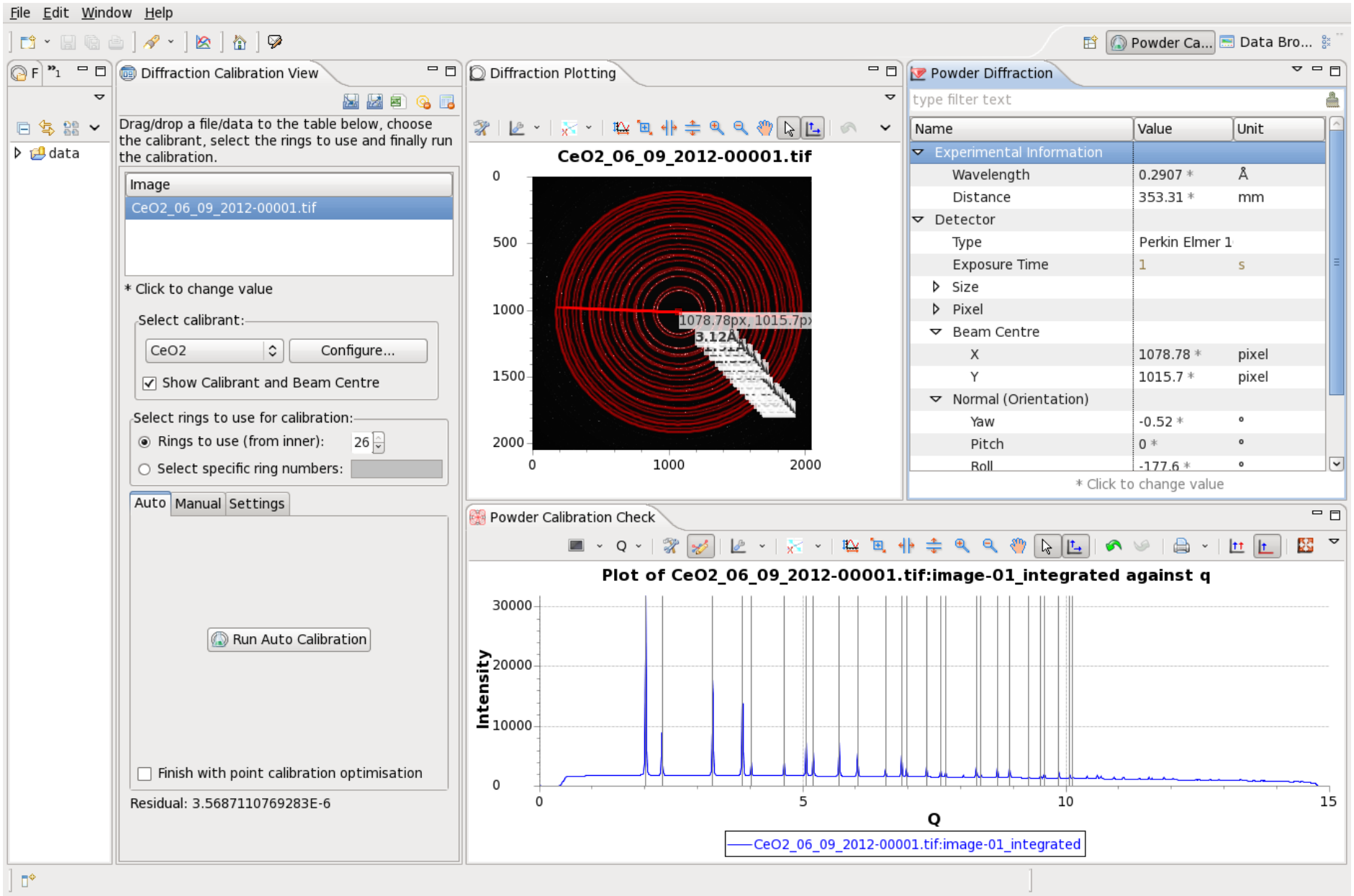
DAWN: Workflows

- Idea:
  - graphically compose standard components (“actors”),
  - to form a new analysis/data reduction procedure.
- Actors for
  - request user input,
  - open files or monitor directories,
  - flow control,
  - mathematics operations or execute python code,
  - plot or review plots,
  - export to files,...
- Can be run in batch mode, parallelised using threads.
- Interesting to use for monitoring procedure,
  - but I failed: little docu, not (yet) flexible enough, bugs,...

DAWN: Workflow Example



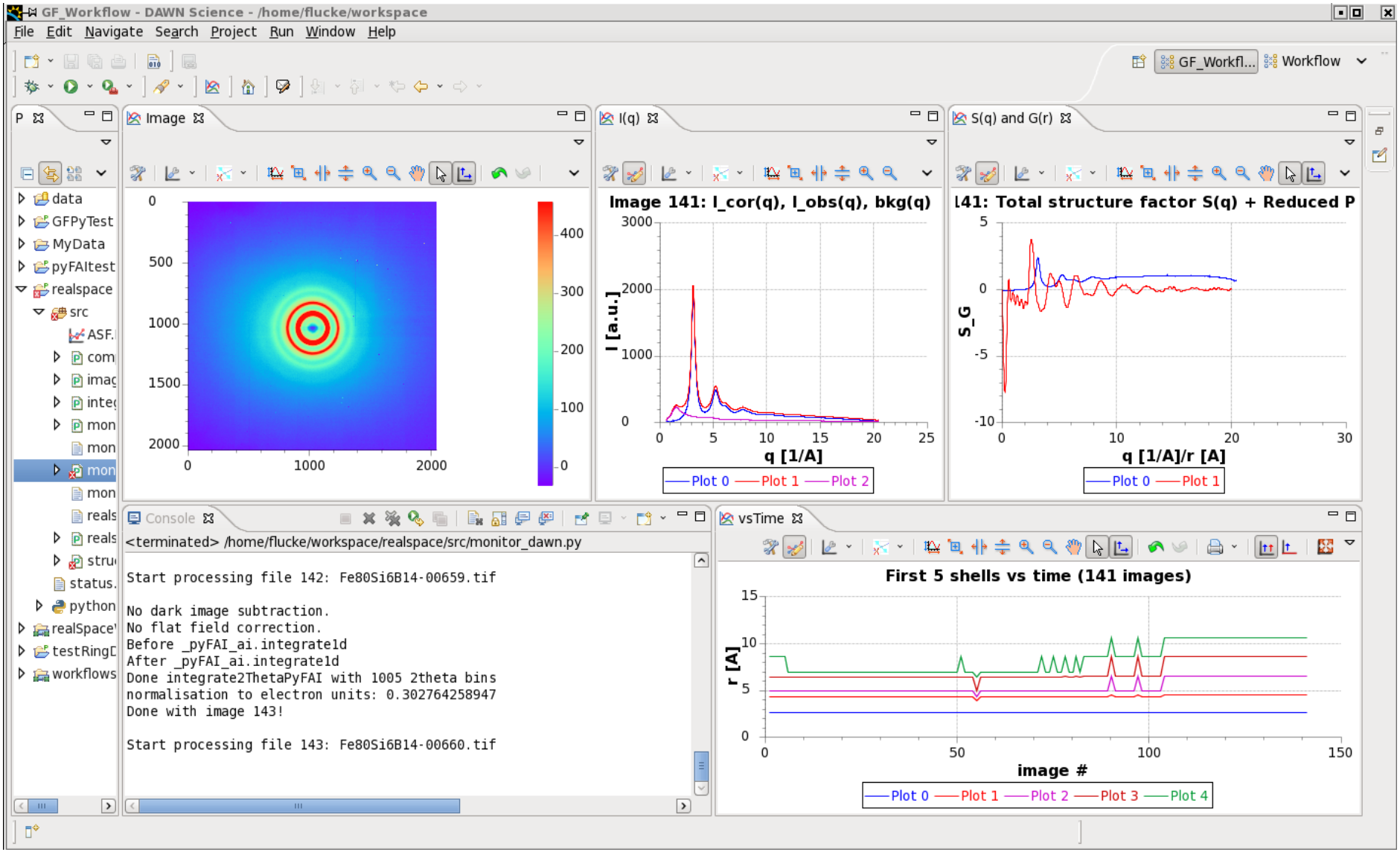
DAWN (1.5.0 beta): Diffraction Calibration



DAWN: Real Space Analysis/Monitoring

- Use DAWN basically only for
  - Python code development,
  - graphics output.
- Configuration using configuration file.
- Monitoring directory for new images:
  “borrow” DPDAK’s concept.
- Storing results on HDF5 file using `h5py`.
- Arrangement of “Views” stored as new “Perspective”.
- Miss mechanism to “softly” stop procedure,
  - needed to close HDF5 file properly.
- Control of plot appearance via Python very limited:
  - cannot change colour/style of lines, titles, legend,...
  - no way (yet) to set proper legend entries
- Plotting acceptably fast (required fix from DAWN developers).

DAWN: Real Space Analysis/Monitoring



DAWN: Pros and Cons

- Flexible multi-dimensional data display and interactive analysis,
  - slicing, masking, profiling, fast 3D plots.
- Browsing file content, e.g. HDF5.
- Python as scripting language
  (built-in development environment).
- Large support from Diamond and ESRF.
- OSGI plugin-based design is modular and thus flexible.

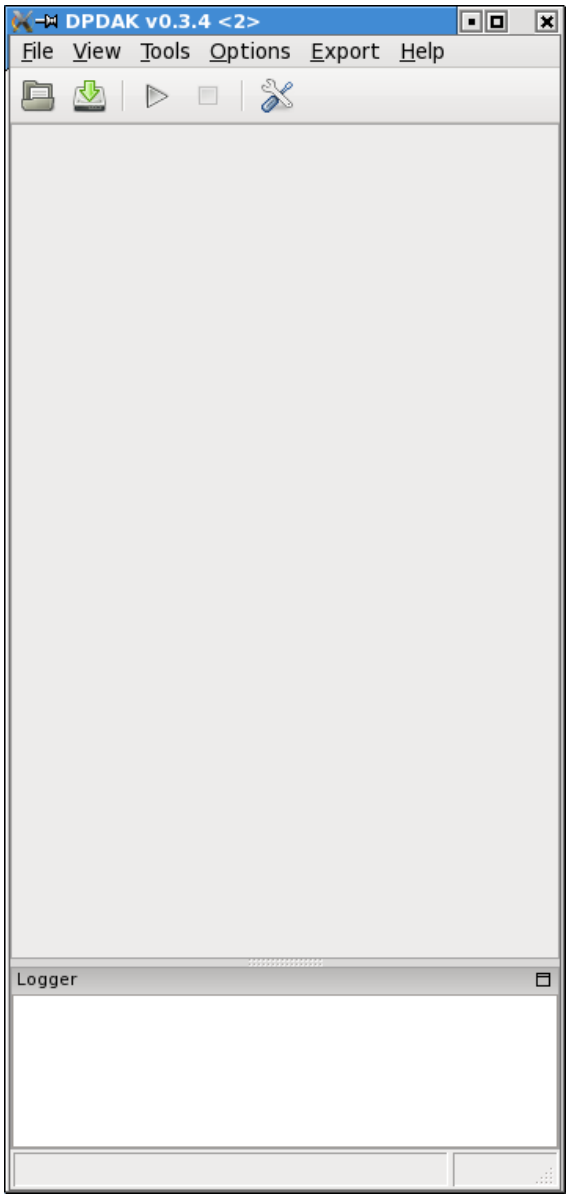
- Java: difficult to import “external” algorithms.
- Fitting arbitrary functions not working yet.
- Still bugs hitting the user.
- Limited interface from Python to core functionalities
  (besides expert way via `py4j`).
- OSGI difficult to pick up until you understand it.

DPDAK:
Directly **P**rogrammable **D**ata **A**nalysis **K**it

- Pure Python program using ‘standard’ Python packages:
  - NumPy, SciPy, matplotlib, fabio, wxPython.
- Developed in cooperation between DESY and MPIKG,
  - for “(online) analysis of 2D scattering data”,
  - standard tool at P03 (S. Roth) and for its users,
  - author G. Benecke left end of January.
- Supported platforms: Windows, Linux.
- Open Source, GNU GPL.

https://dpdak.desy.de

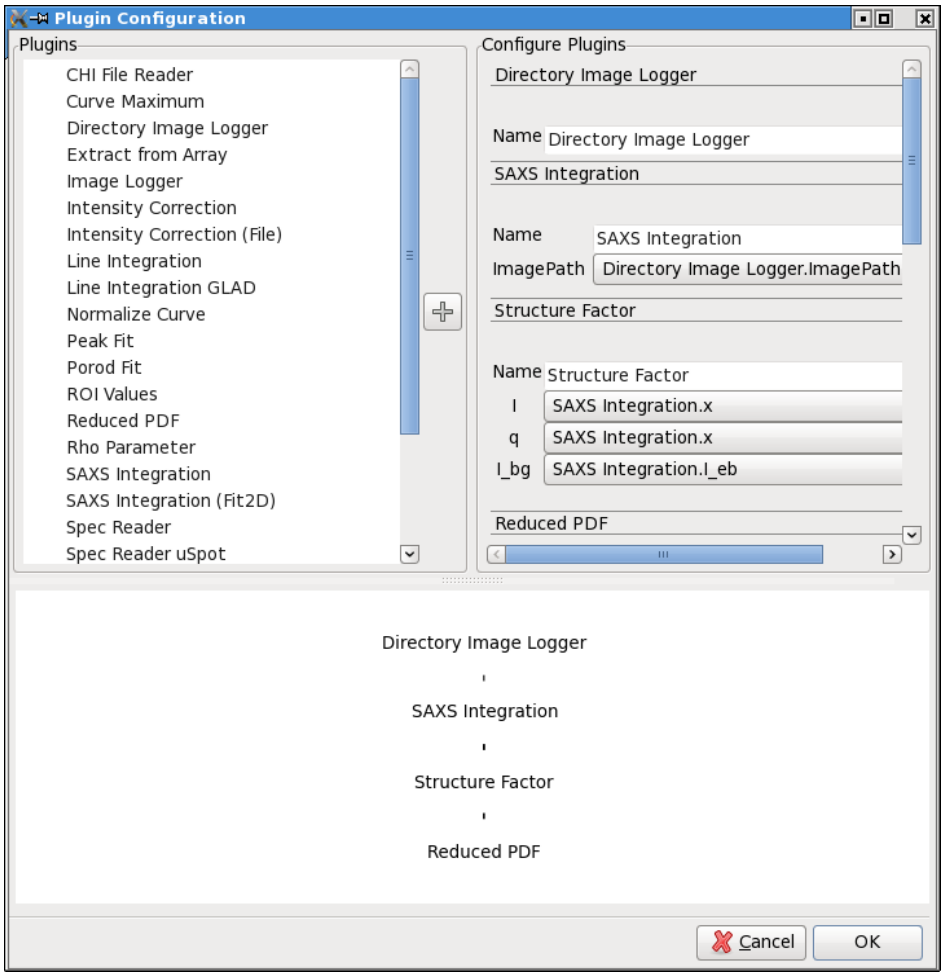
DPDAK: Concept



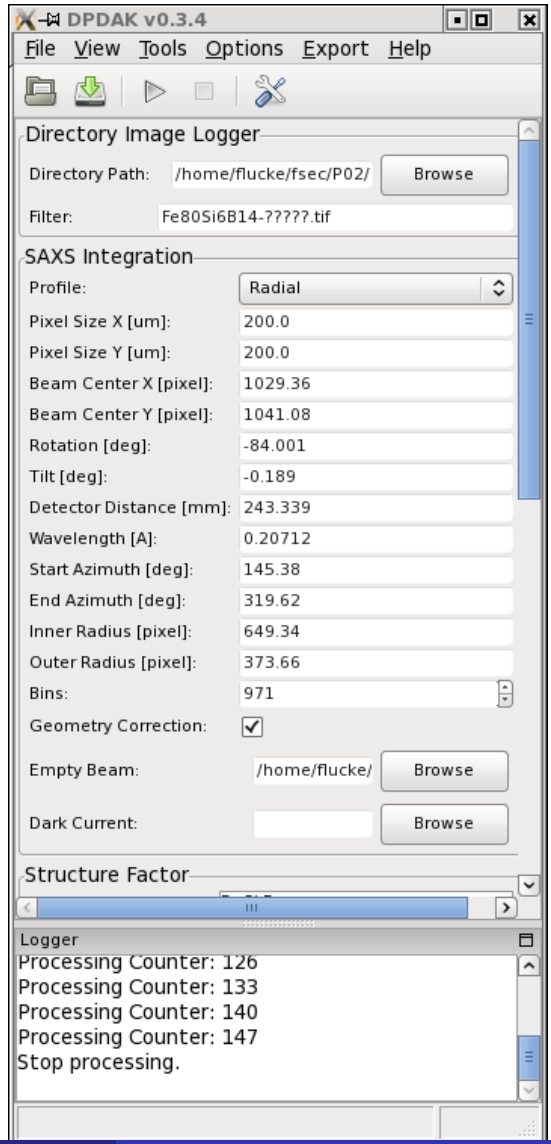
- Minimalistic start-up GUI.
- Select and configure 'plugins':
  - data processing steps,
  - tools (data display etc.),
  - data export.
- Interfaces of processing plugins:
  - input and output types, parameters,
  - types: scalars, 1D arrays, strings, file/directory paths.
- Processed data stored in 'database':
  - no check if data always identical,
  - for images just their paths,
  - interactive analysis via tools.
- General image options:
  - rotation, axis flipping, background.

DPDAK: Configure Data Processing Plugins

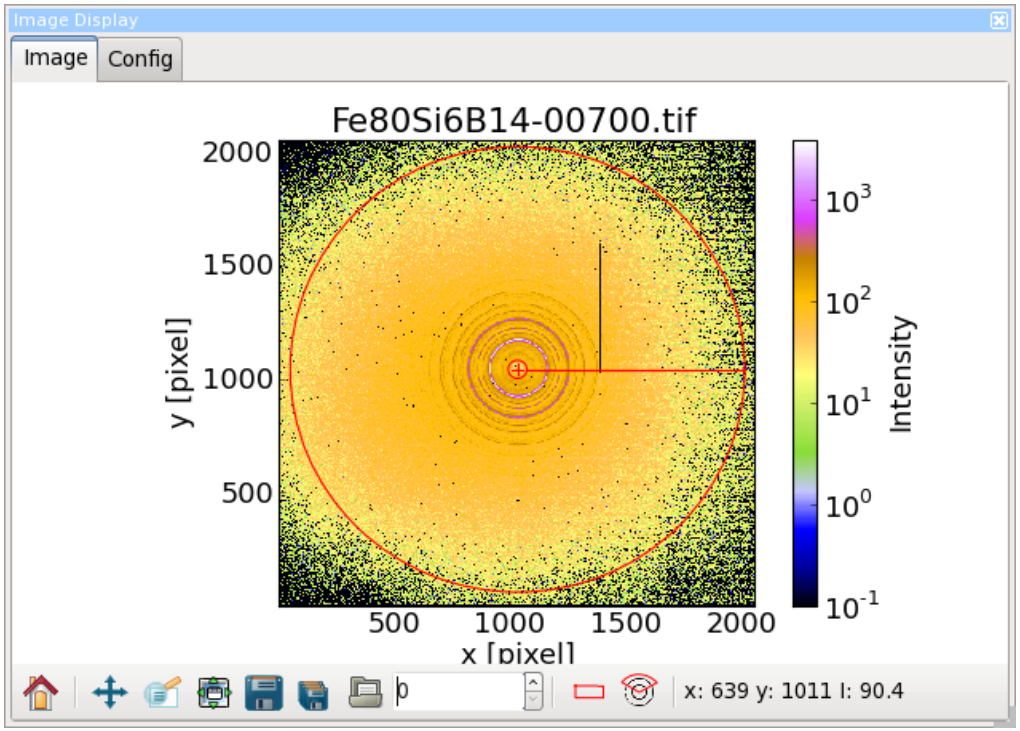
- Select plugins from list.
- Select (matching) input.

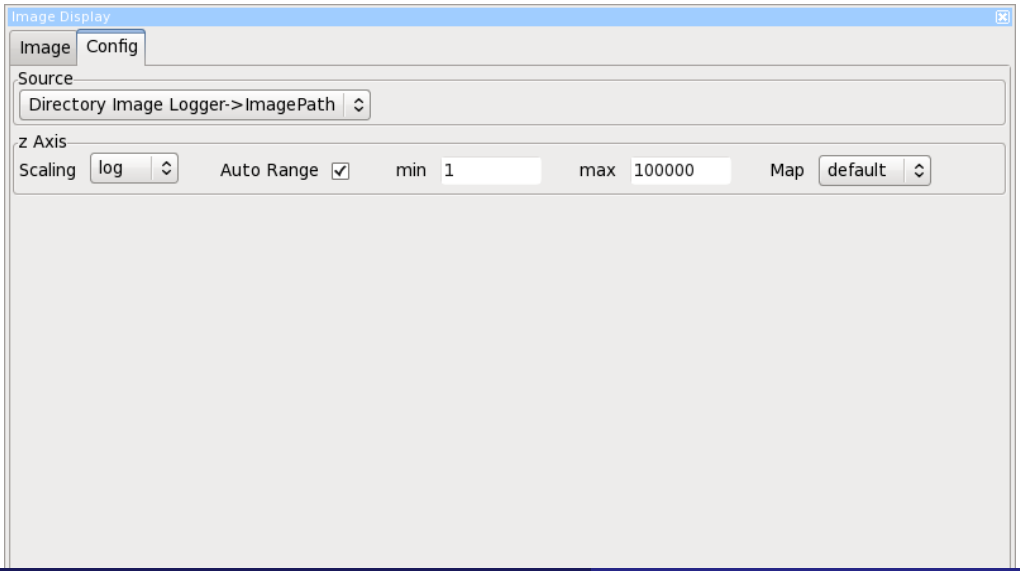


- Set parameters.



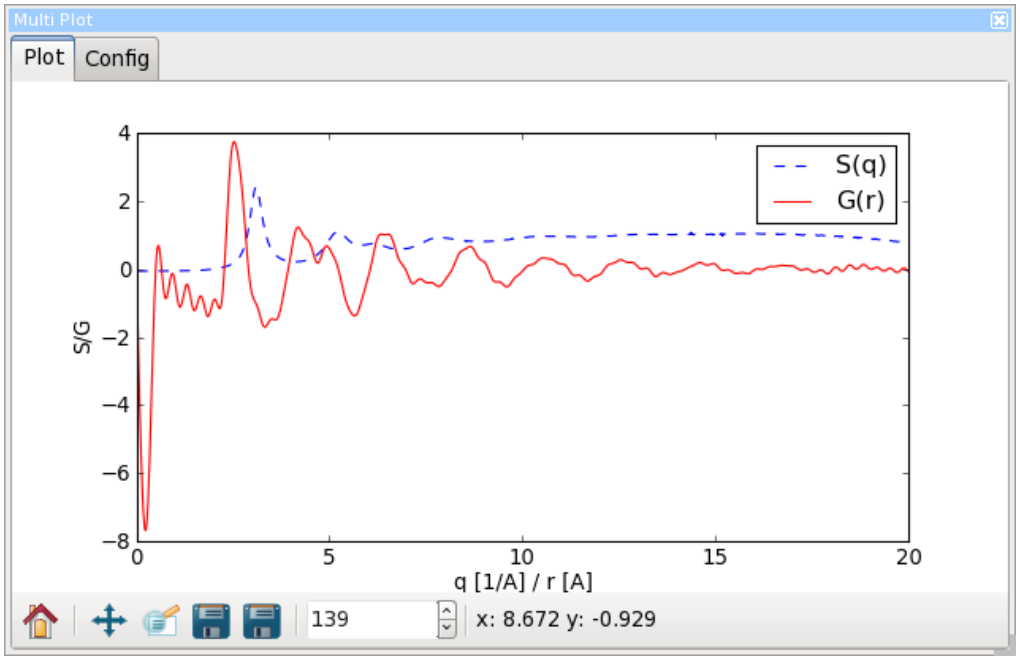
DPDAK: Image Display

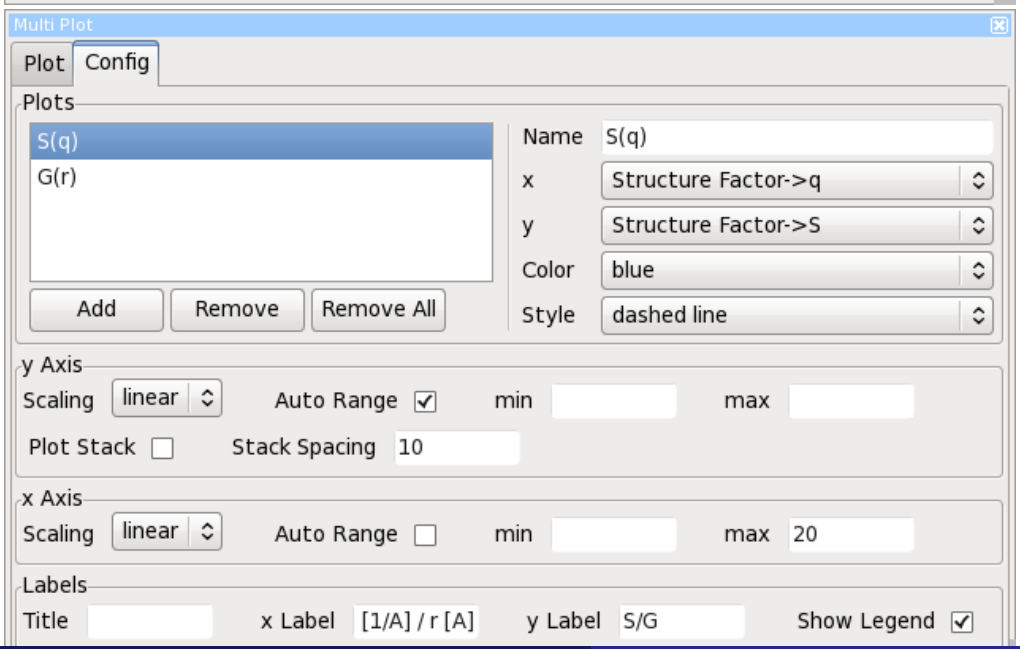




- Choose from database of processed images (broken in v0.3.4, showing always the last?).
- Directly select file.
- Range of 2D $\rightarrow$ 1D integration shown - and editable!

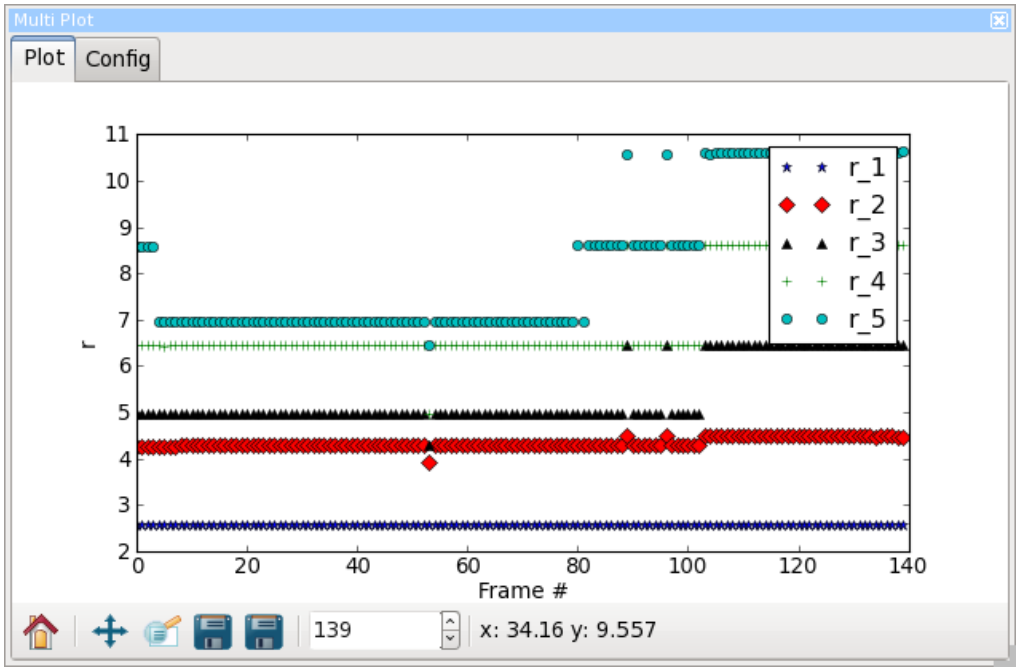
DPDAK: 1D Plots

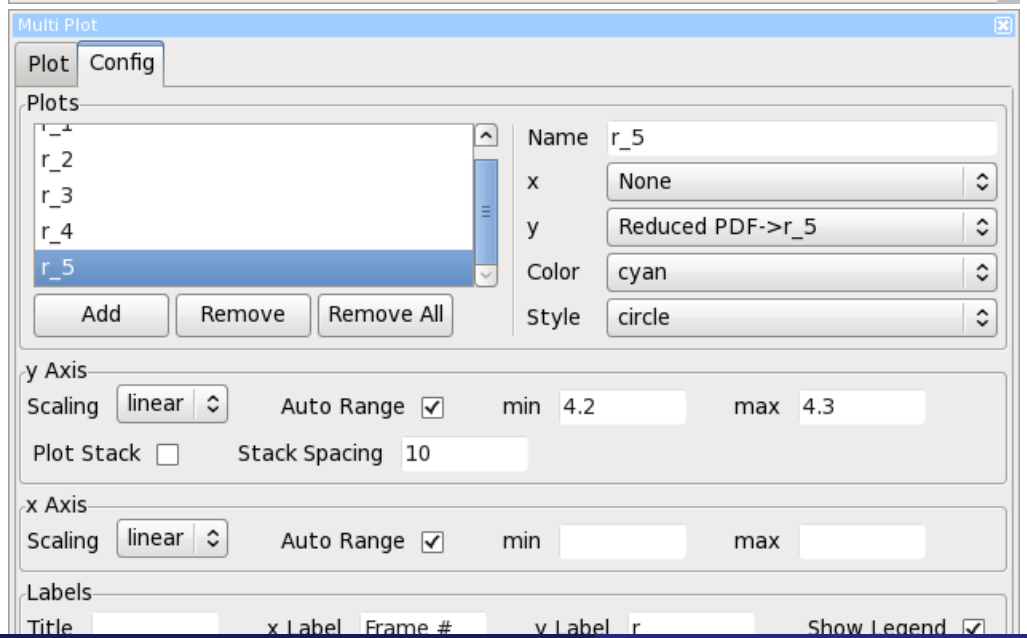




- Select x and y from output of plugins.
- Many lines can be overlayed.
- Choose result of which processed image to be shown.
- Stack of all lines also possible
  (broken in recent version 0.3.4?).

DPDAK: 1D Plots (ctd.)



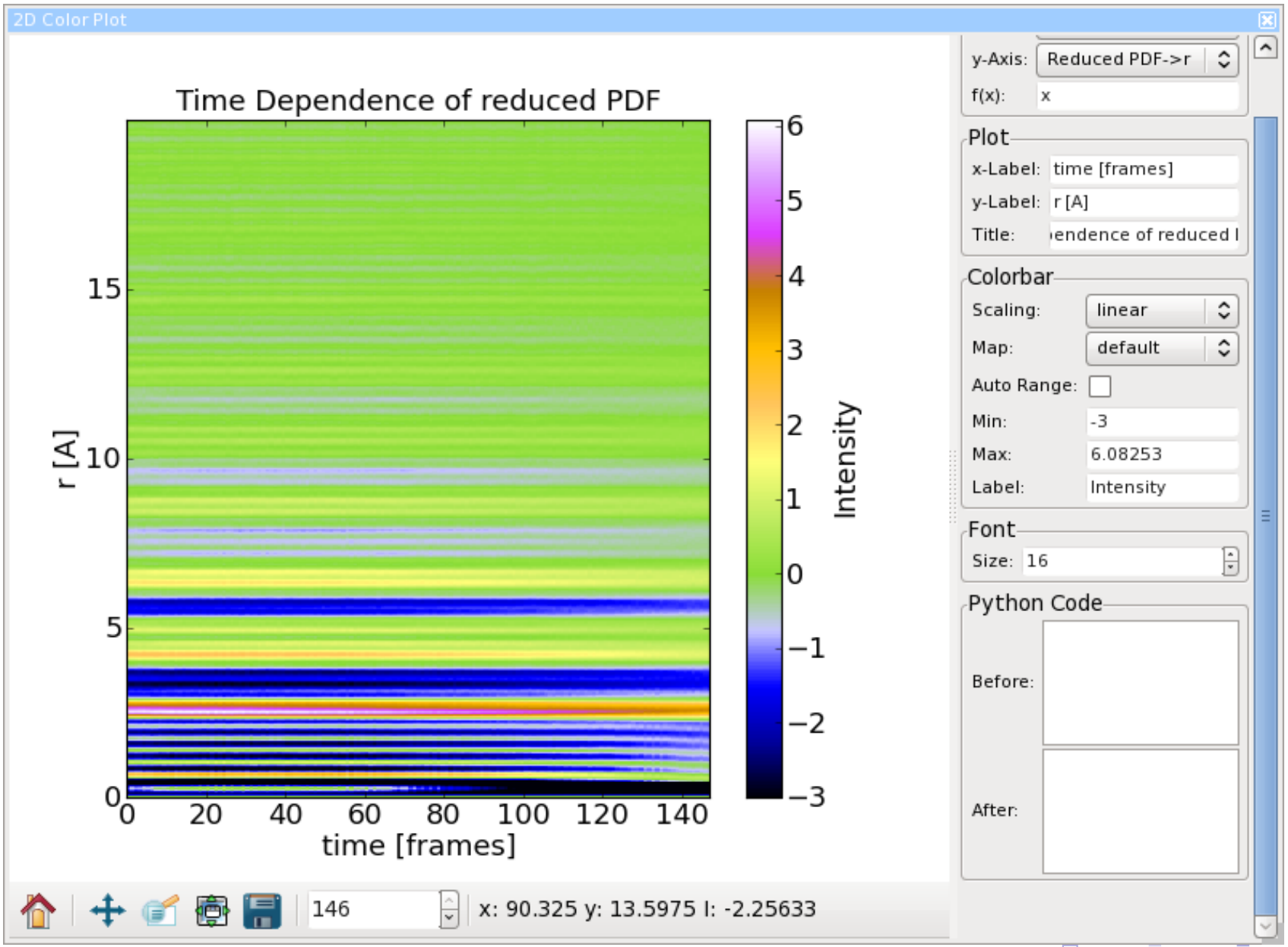


No x axis chosen:

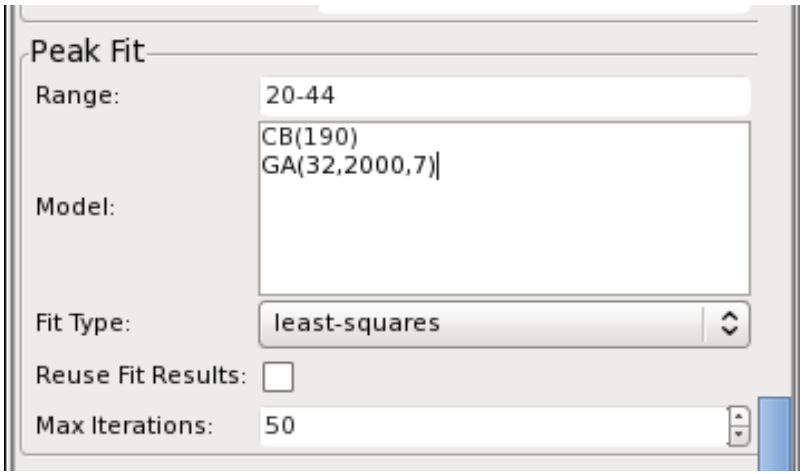
- y data is 1D array: array plotted vs its index.
- y data is scalar: scalar plotted vs frame number (can choose up to which frame number).

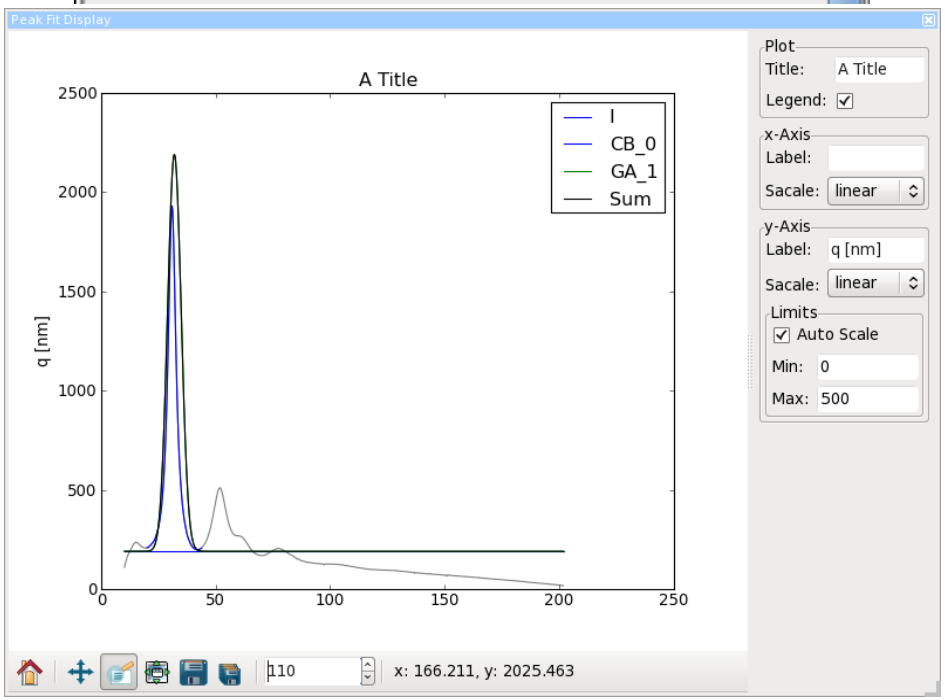
- Legend position static (but can be switched off).

DPDAK: 2D Plot of Distribution vs Time



DPDAK: Fitting





Processing Plugin to Fit

- (sum of) predef. functions, backgr.: constant, linear, peaks: Gauss, Lorentz, Pseudo-Voigt,
- restrict fit range,
- (possibly fixed) start values.

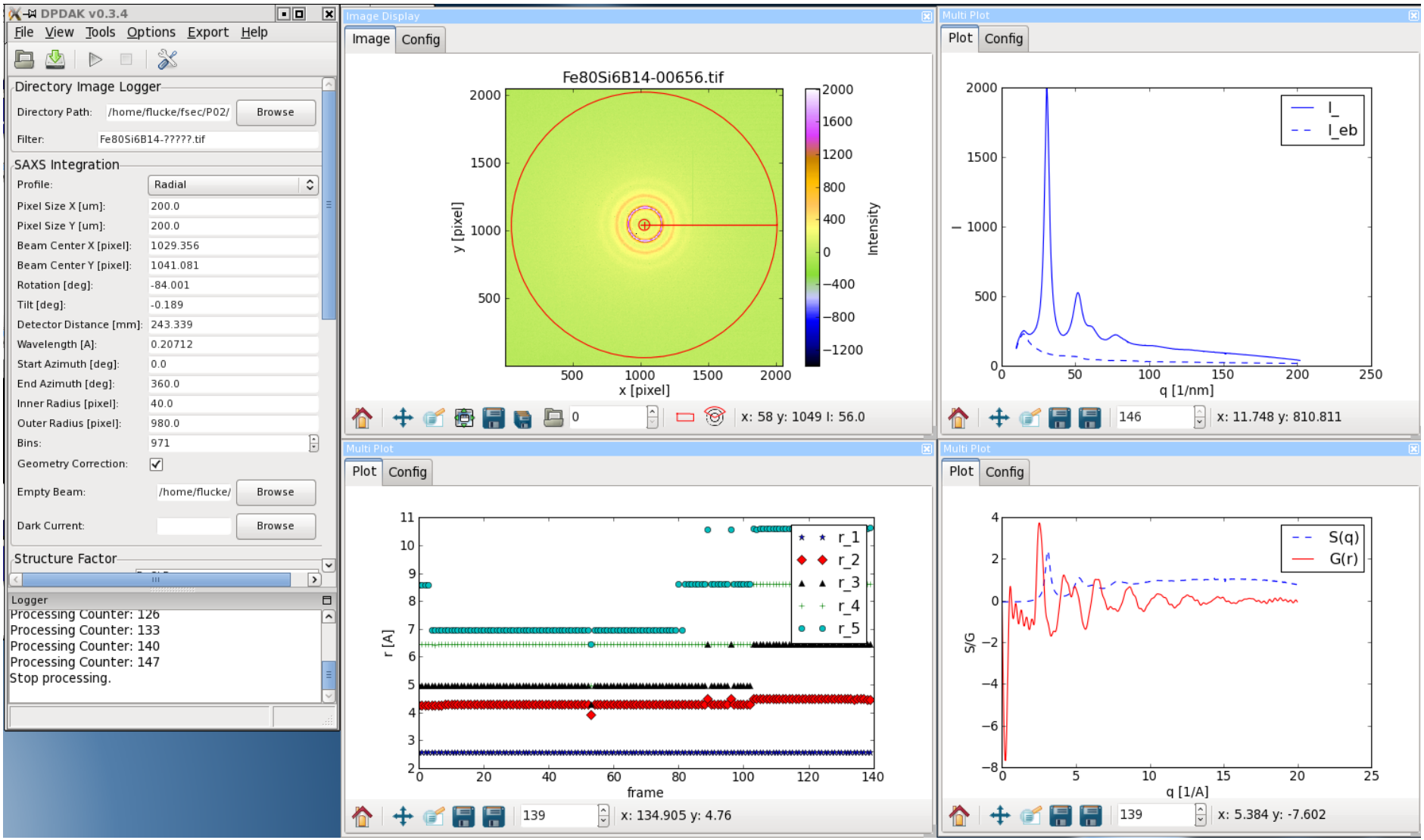
Peak Fit Display Plugin

- Original distribution (highlight fit region).
- Fitted curve and its components.

DPDAK: Real Space Analysis/Monitoring

- DPDAK designed for that purpose:
  - add data processing plugins,
  - configure (including display plugins),
  - run!
- Already there:
  - directory monitoring,
  - 2D $\rightarrow$ 1D integration,
  - storage of results as DPDAK 'database'
    (can be exported to ascii - completely or per plugin).

DPDAK: Real Space Analysis/Monitoring



DPDAK: Pros and Cons

- Built to be extendible by user,
  - and fulfils that promise (within boundaries. . .)!
- Rather simple interface for new plugins.

- How to handle multi-dimensional datasets?
- Does internal database scale with 10 thousands of images?
- Some intransparent shortcuts.
- Display plugins do not store configuration (could be added).
- Limited core support now that G. Benecke left.

Mantid

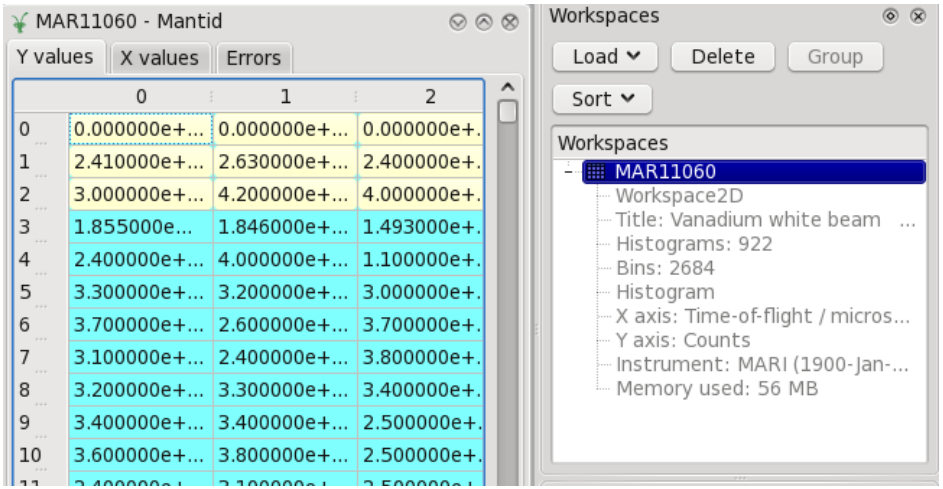
- Framework for “high-performance computing and visualisation of scientific data”.
- To “manipulate and analyse Neutron and Muon scattering data, but could be applied to many other techniques.”
- Main developers from ISIS and SNS.
- Written in C, C++ and Python
  (using scientific visualisation software “QtiPlot”).
- Open source: GNU General Public License.
- Each release can be cited with a digital object identifier.
- Supported platforms: Windows, Linux, Mac OS.

http://www.mantidproject.org

Mantid: Concepts

- Data processing via **algorithms** working on **workspaces**.
- Once data loaded into a workspace
  - investigate interactively: 1D, 2D, 3D plots, 1D fits, slices
  - or call algorithms.
- Store status (workspace, plots,. . .) as 'project'.
- Smooth integration of Python as scripting language:
  - access algorithms,
  - graphics control,
  - extend Mantid by Python algorithms and fit functions,
  - workspaces keep track of their history - as Python script,
  - all but graphics available also outside MantidPlot GUI.
- Visualisation of instruments.
- Good documentation for beginners (GUI plus Python).

Mantid: Workspaces



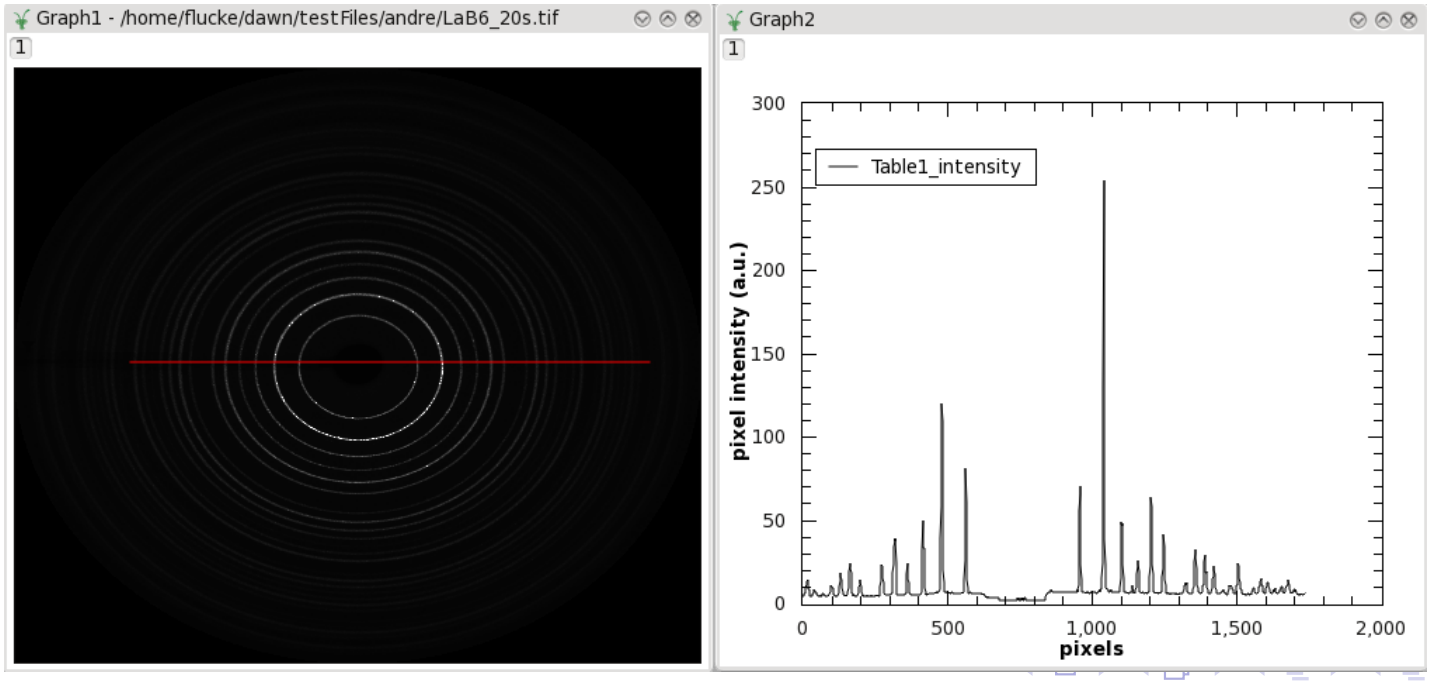
Most Common: Workspace2D

- Vertically: many “Spectra”.
- Horizontally: histogram/ curve, tabs choosing
  - signal “Y” values,
  - their “Errors”,
  - “X” positions/bin borders.

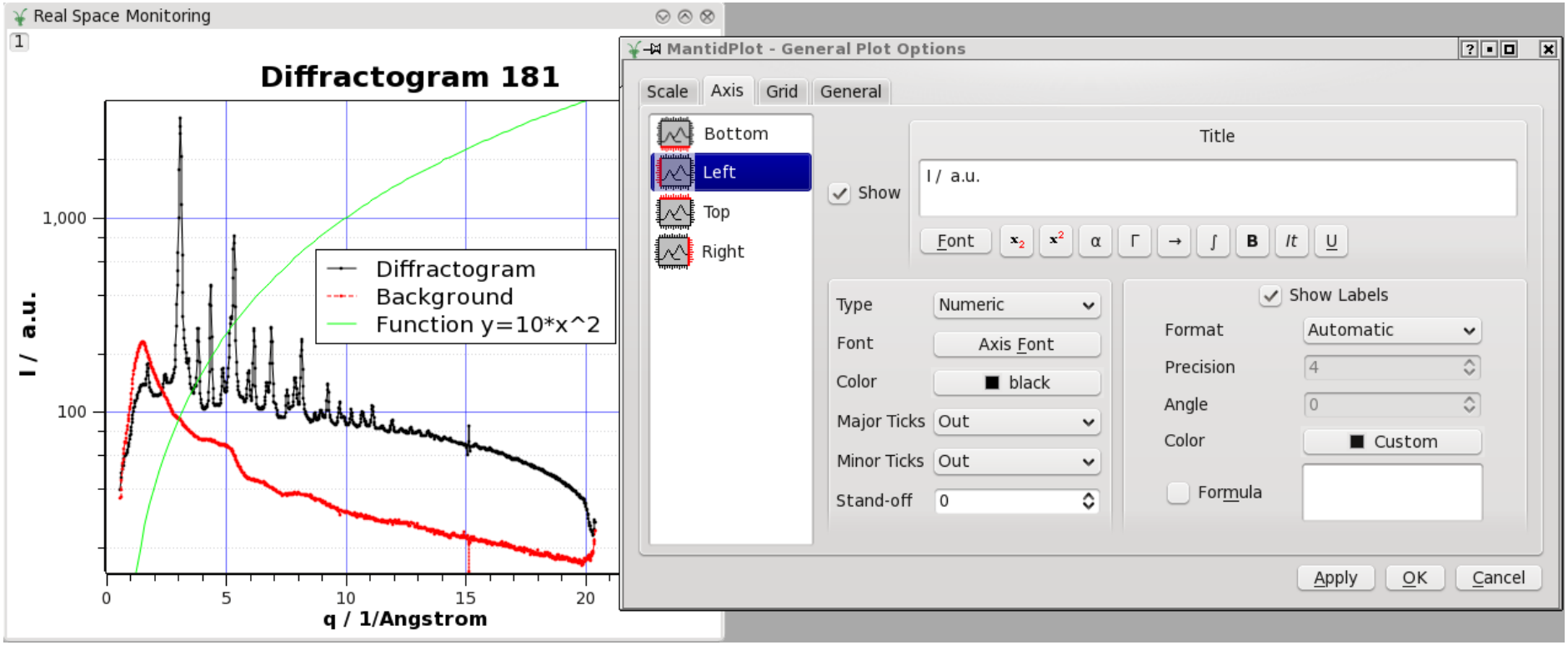
- EventWorkspace:
  as Workspace2D, keeping data unbinned.
- MDWorkspaces:
  Signal and error for coordinates of 1–9 dimensions.
- TableWorkspace:
  rows of columns of particular types (text, integer, . . .).
- Several variants, e.g. file-backed, special purpose, ...

Mantid: Data Formats

- Generic “Load” algorithm chooses proper sub-algorithm.
- No generic loader to browse arbitrary HDF5 files.
- No loader for image files.
  - MantidPlot *can* open image files.
  - All I found: make a line profile.
  - Nothing to create a workspace from an image.

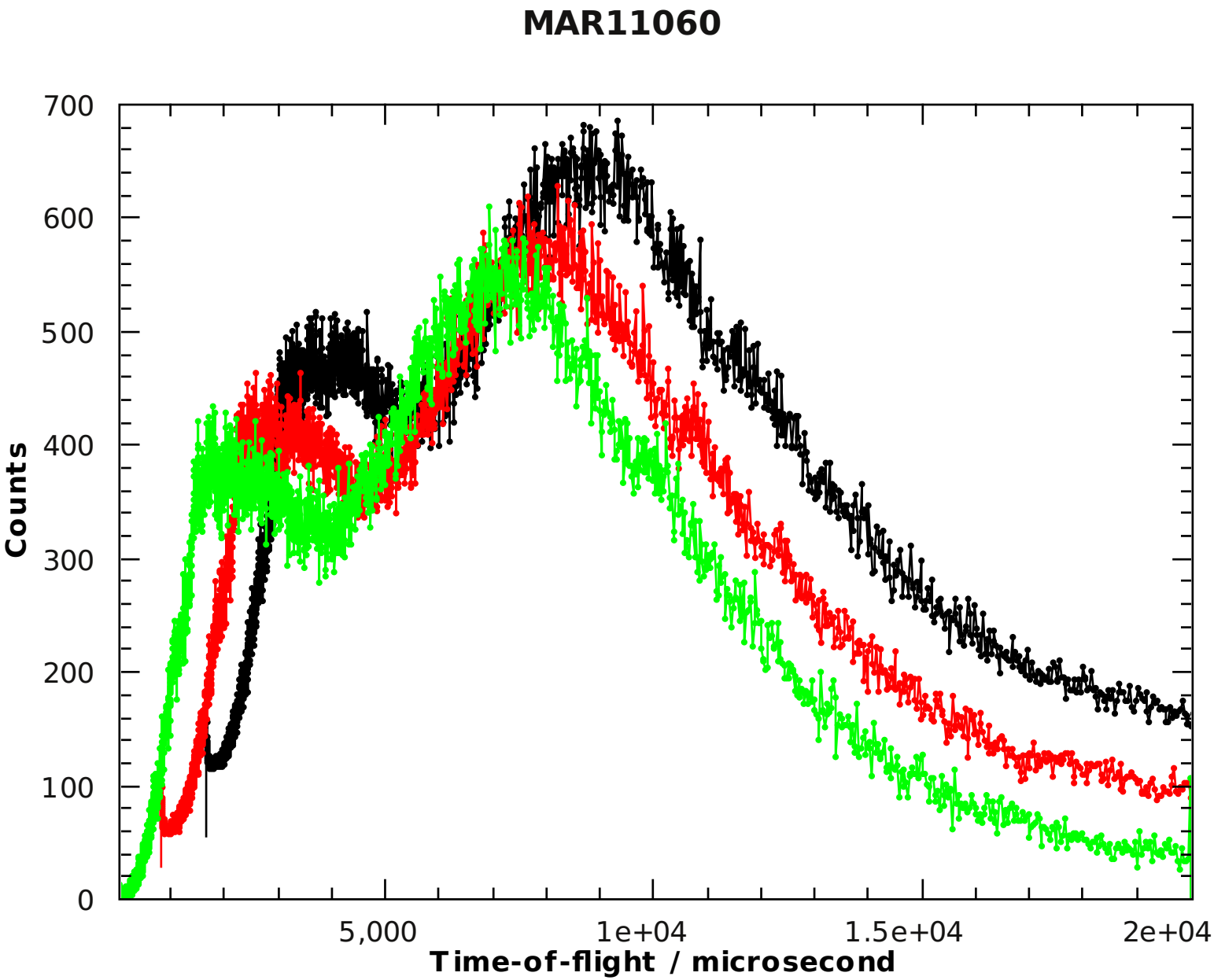


Mantid: 1D Plots

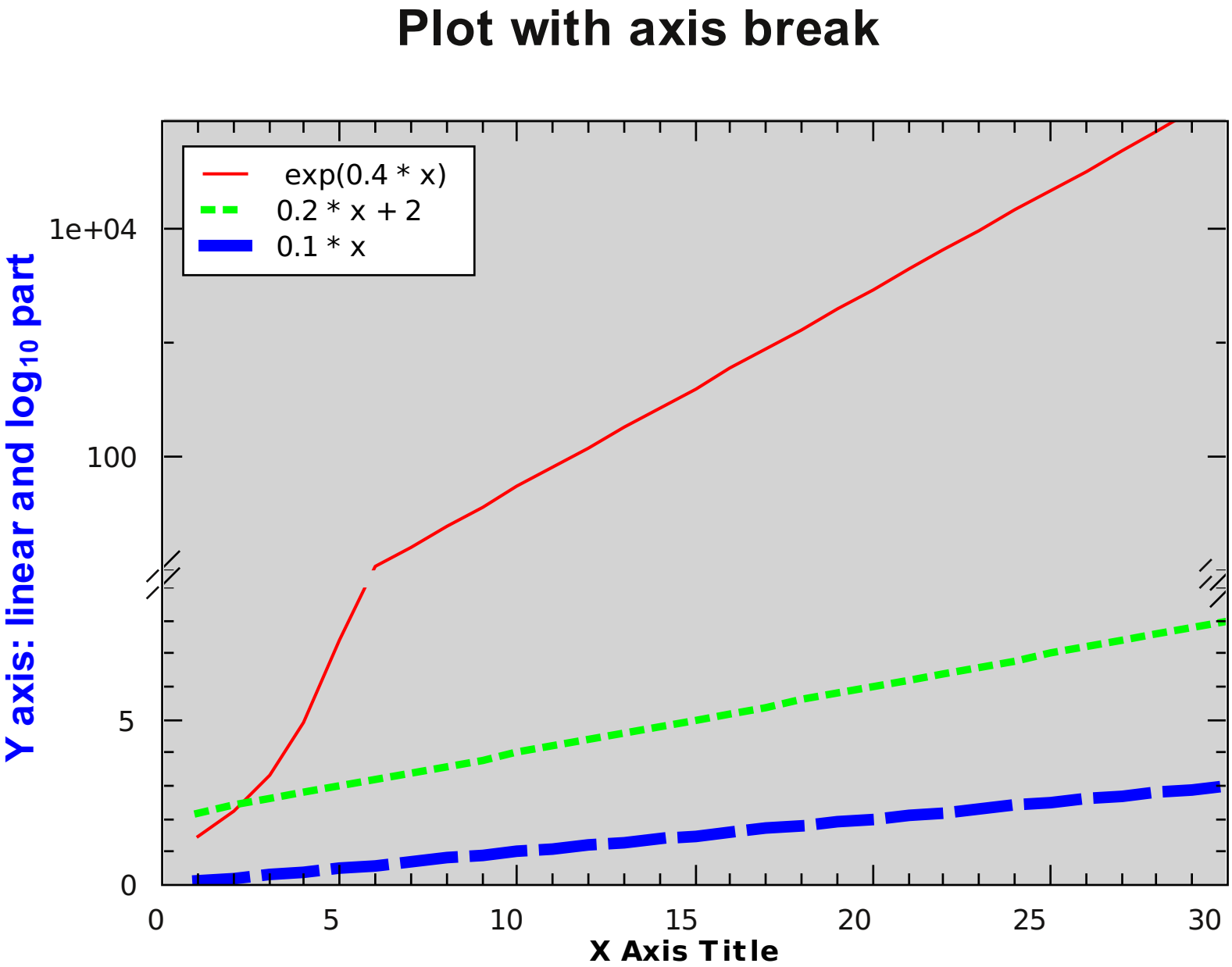


- Plotting needs input workspaces.
- Interactive control of plot appearance.
- Add (parametric) functions.
- May place plots next to each other in same window.

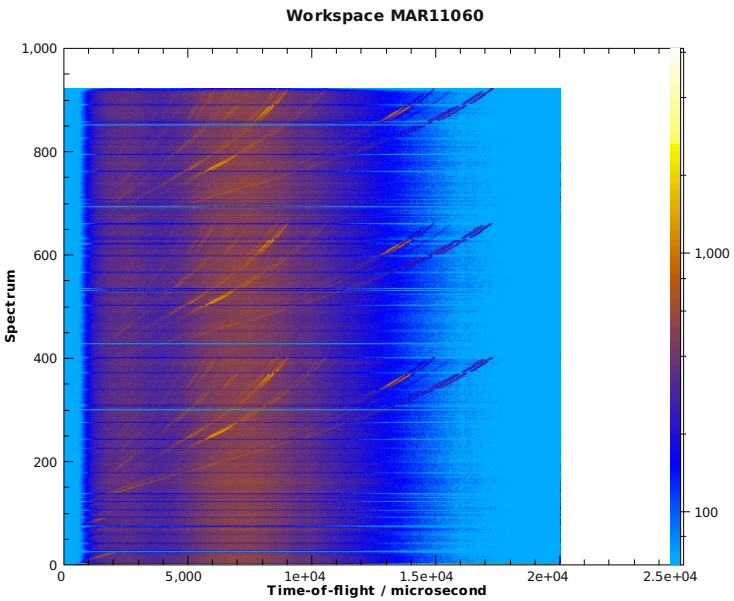
Mantid: 1D Waterfall

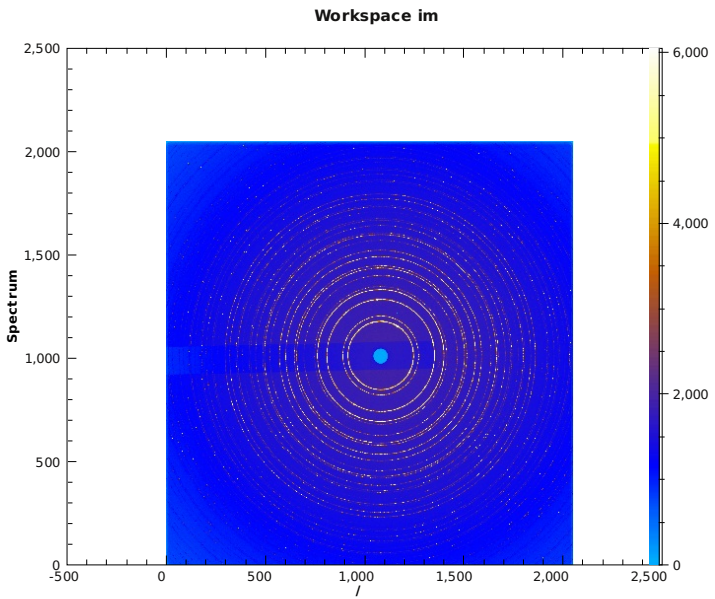


Mantid: Axis with Break



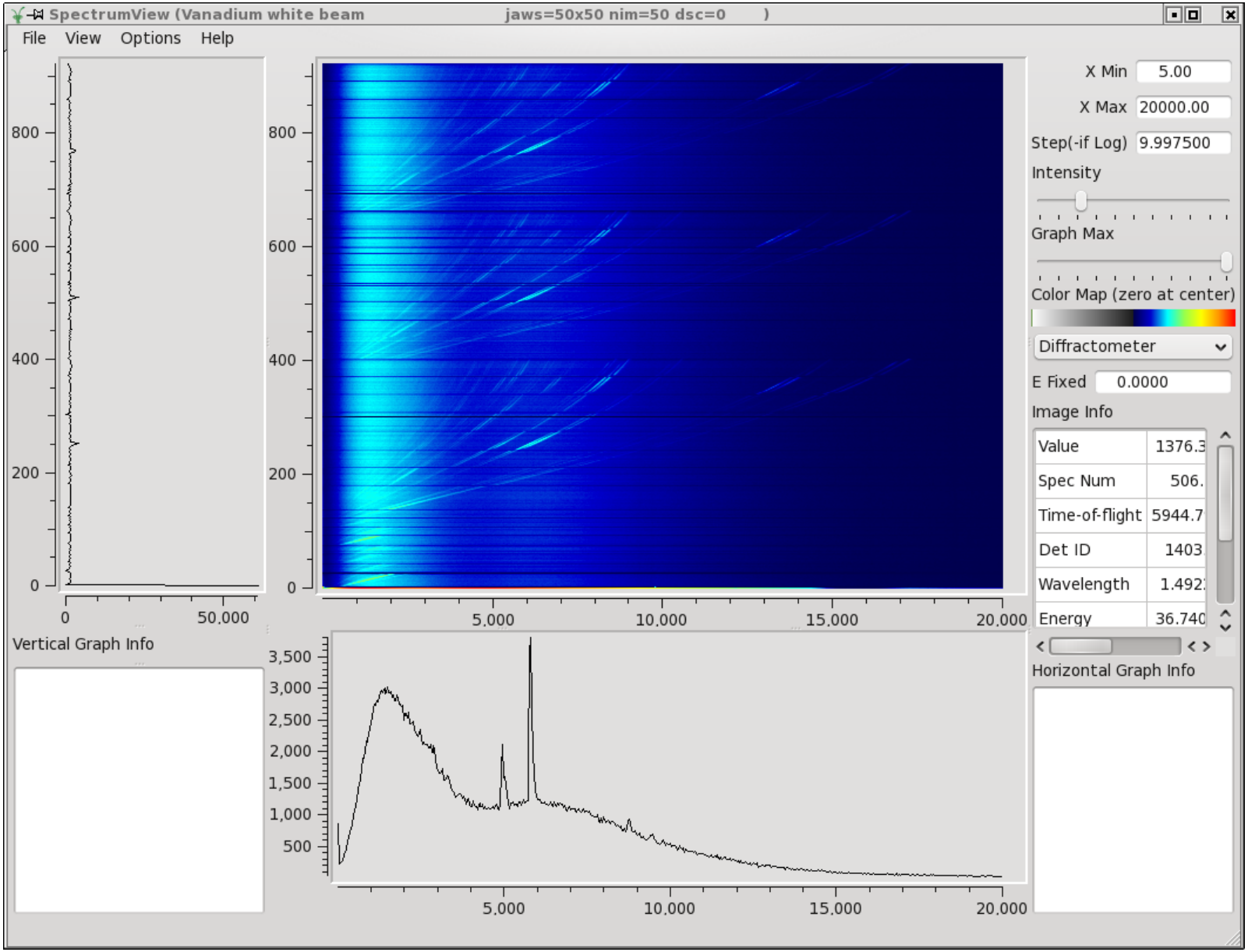
Mantid: “2D Colour Fill”





- Default colour scale does not exclude outliers.
- Also 3D views available (first display slow).

Mantid: Spectrum Viewer



Mantid: 1D Fitting

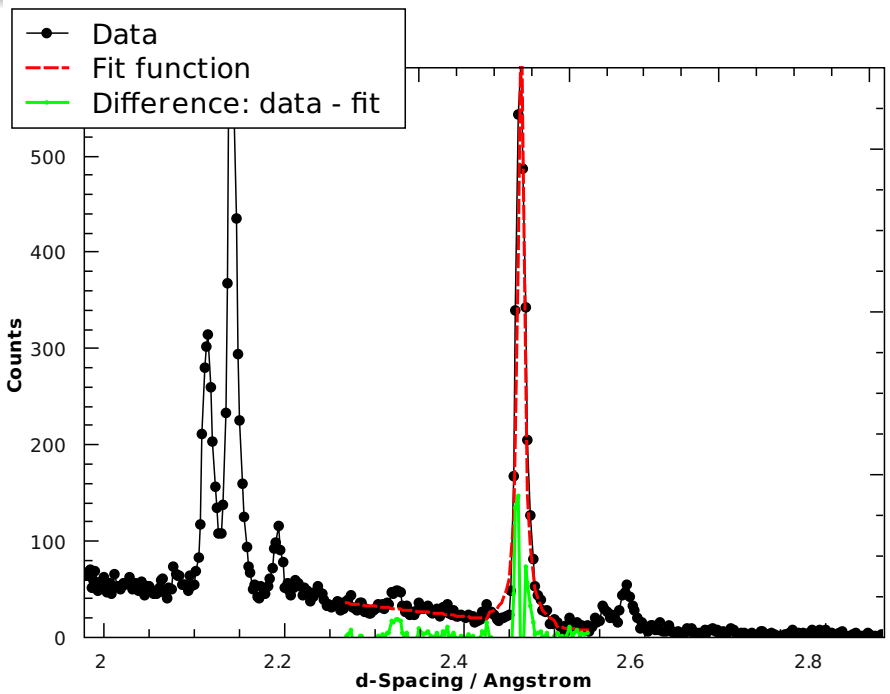
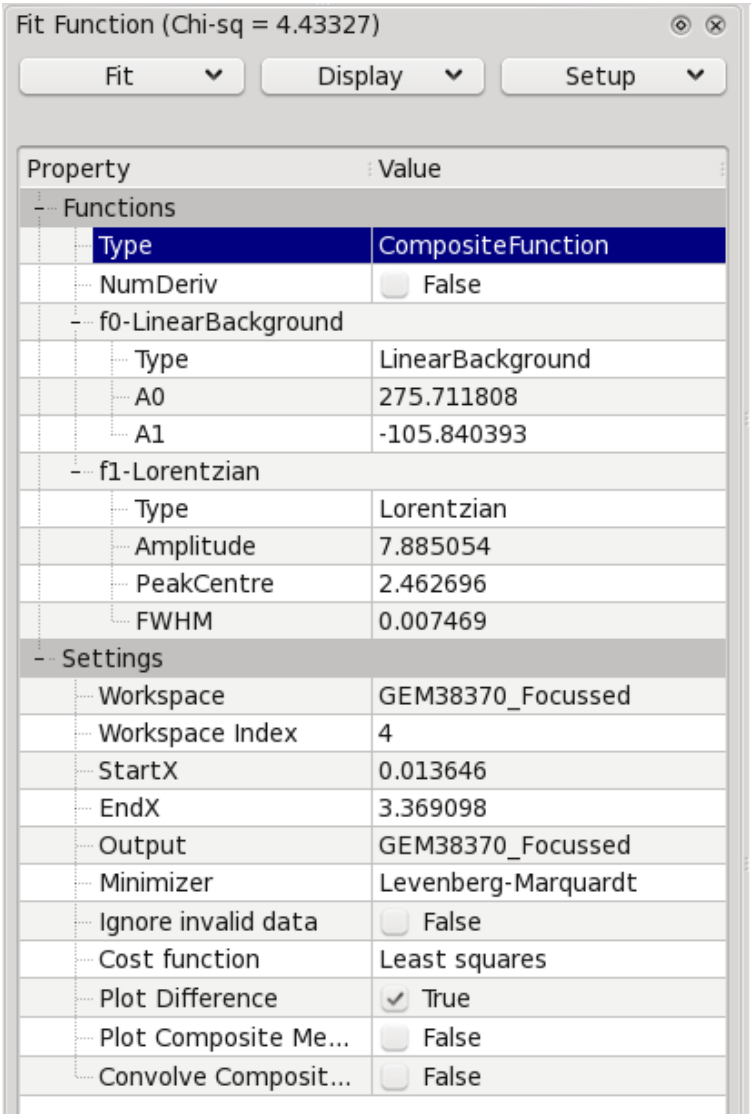


Table-1 - GEM38370_Focussed_NormalisedCovarianceMatrix

	Name[L]	f0.A0[Y]	f0.A1[Y]	f1.Height[Y]	f1.PeakCentre[	f1.Sigma[
0	f0.A0	100	-99.9113	0.00387227	0.775206	4.36445
1	f0.A1	-99.9113	100	0.0112753	-0.744489	-4.85306
2	f1.Height	0.00387227	0.0112753	100	-9.86734	-60.0745
3	f1.PeakCentre	0.775206	-0.744489	-9.86734	100	13.4125
4	f1.Sigma	4.36445	-4.85306	-60.0745	13.4125	100

- Graphically choose fit range.
- Compose fit function: various background and peak types.
- Graphically choose start values of peaks.
- Graphical result includes difference “data minus fit” (and possibly individual components).
- Fit result and errors (incl. normalised covariance matrix) available as workspace.

Mantid: 1D Fitting (ctd.)



Many options

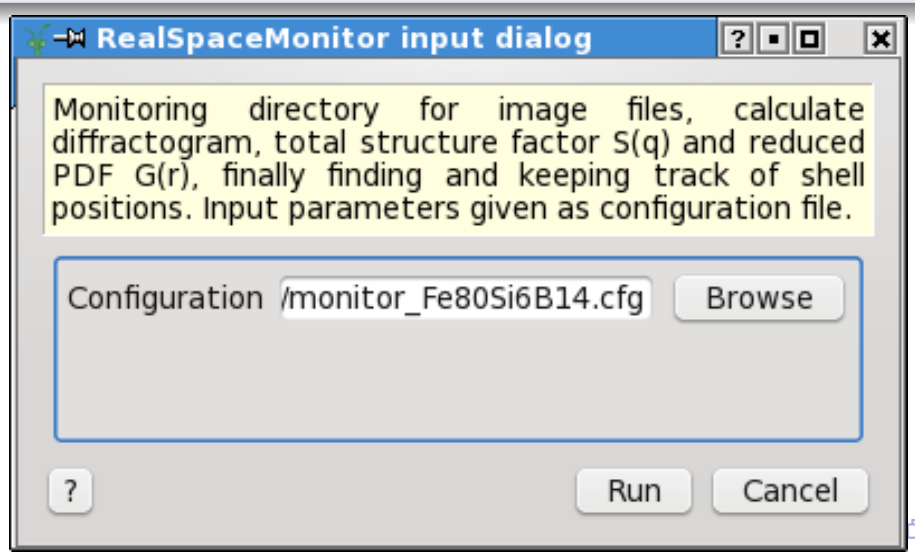
- Fit function composed of arbitrary amount of components.
- Rich set of predefined functions.
- Define your own function (e.g. implemented in Python).
- Tie parameter values to functions of other parameters.
- Fix parameters.
- ...

Mantid: Miscellaneous

- Masking? There is something I did not investigate.
- Poor Python editor (but can use external one).
- Python algorithms are not really object-oriented - problem:
  - CPU-intense initialisation,
  - two different configurations in one data processing chain.
- Cannot reset python terminal once I messed it up. . . (?)

Mantid: Real Space Analysis/Monitoring

- Make use of algorithm/workspace concept.
- Write algorithms in Python:
  - LoadImage, using `fabio`,
  - other using “common code base”,
  - Fourier transformation from $S(q)$ to $G(r)$ already in Mantid.
- (Ab)use Workspace2D even for images
  (waste of space: errors, “X” values).
- One master algorithm “RealSpaceMonitor”:
  - Configured via configuration file (not really Mantid style).

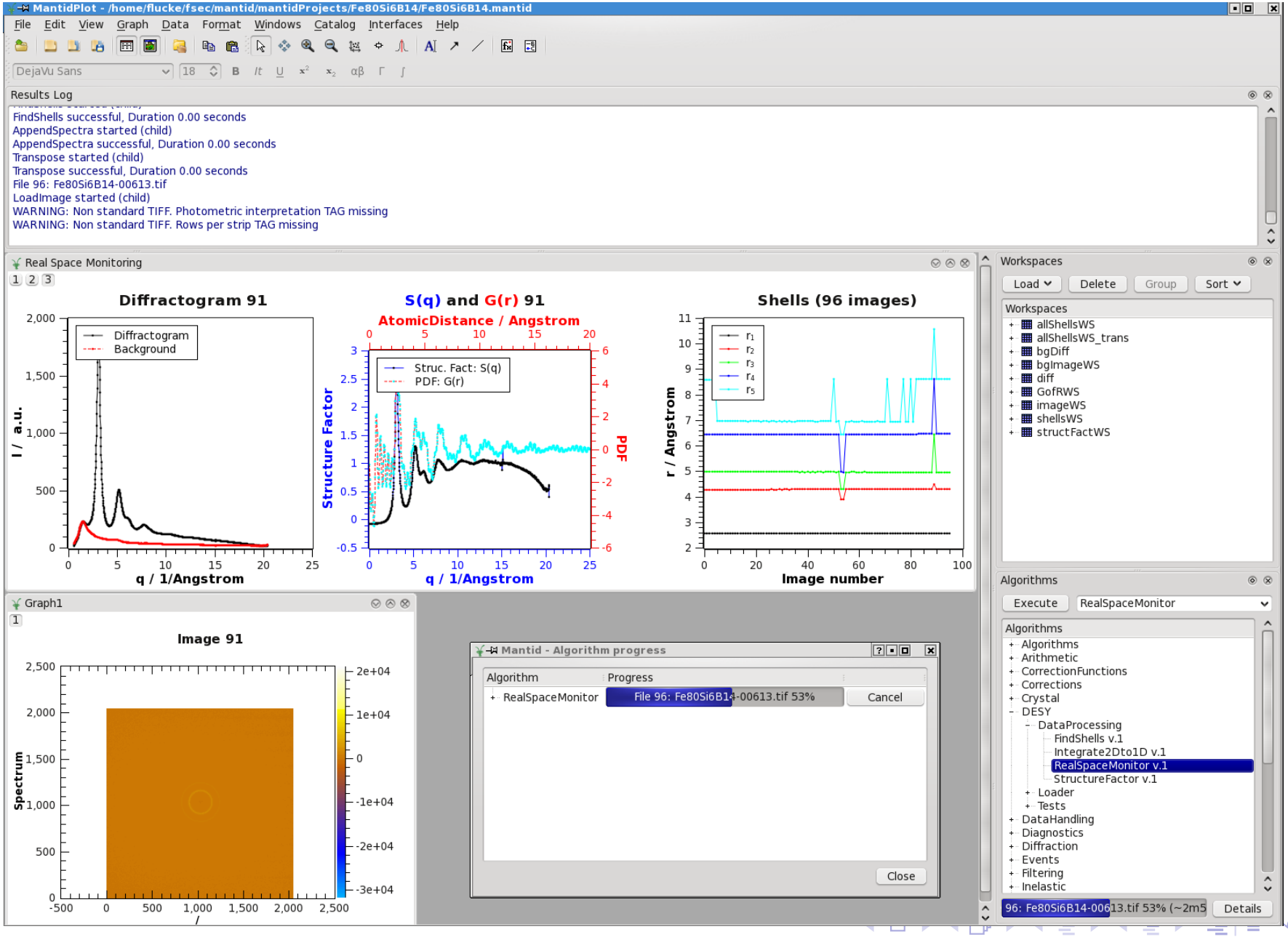


Mantid: Real Space Analysis/Monitoring

Master Algorithm RealSpaceMonitor

- Watching directory ("borrow" DPDAK's logic again).
- Calling other algorithms.
- Write HDF5 output file (using `h5py`).
- Report progress (and catch cancellation to flush HDF5 file).
- Doing graphics output.
- Large control of line plots:
  common window, axes, legend, ...
- 2D image plot in extra window:
  - no control of appearance from Python,
  - slow.

Mantid: Real Space Analysis/Monitoring



Mantid: Pros and Cons

- Deep integration of Python as scripting language:
  - can extend MantidPlot!
- Lots of graphics control (interactively and via Python).
- Flexible function fitting with error propagation.
- Version control via doi.

- No built-in image handling.
- Workspace concept may not be flexible enough,
  - but did not investigate too much the multi-dimension version.
- Neutron and muon scattering driven:
  - set of available algorithms,
  - workspaces with signal (counts...) and error,
  - wording.

Summary

- All three programs have their own value.
- **DAWN** has the potential to cover standard needs of X-ray data analysis tasks, but suffers from limited extendability by users and (still) little bugs.
- **DPDAK** has proven itself in praxis and is extendible by design, but is not 'complete' and might fail for multi-dimensional data sets.
- **Mantid** provides great integration of user (Python) extensions, good control of plotting, and a professional fitting interface, but is currently driven by neutron/muon scattering needs.

Outlook

- Plan to exercise a SAXS analysis procedure as well.

What to Do to Test on Your Own?

DAWN, DESY Version (i.e. .fio Format Aware)

- Login on workgroup server `p3-wgs12`.
- Run

  `/scratch/DawnVanilla-1.4.1.v20131212-1003-linux64/startDawn.sh`.
- Note that it redirects the workspace to `/scratch/$USER/dawn/workspace` ($USER is your login name).
- Or try a more recent, but probably less stable version that supports reading of .fio files even from CPython:

  `/scratch/DawnVanilla-1.5.0.v20140326-1353-linux64/startDawn.sh`.

Mantid

- A bit more complicated - please contact me. . .

Please contact me (gero.flucke@desy.de) if you need help.

What to Do to Test on Your Own? (ctd.)

DPDAK - Windows

- Follow https://dpdak.desy.de/index.php/Install_dpdak_v0.3.2#Binary_Package.
- I did not test that...

DPDAK - Linux

- Login on workgroup server `p3-wgs12`.
- Download 'Source/Linux' from https://dpdak.desy.de.
- Put the `*.tar.gz` file into SOME/DIRECTORY.
- `cd SOME/DIRECTORY`
- `tar xzf dpdak-dpdak.tar.gz`
- `python dpdak/main.py`

Please contact me (gero.flucke@desy.de) if you need help.