

dpdak

directly programmable data analysis kit

Gunthard Benecke
HDRI-PNI
27.02.2012



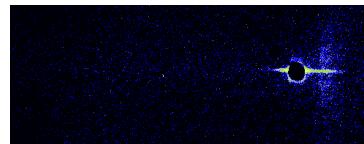
Max Planck Institute
of Colloids and Interfaces

Overview

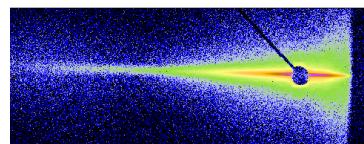
- Plugin framework for primary online analysis of SAXS/WAXS and GISAXS experiments
- Developed in a DESY and MPIKG cooperation
- Goals of the project:
 - fast analysis allowing to classify the data during the experiment
 - deeper analysis of data after experiemnt
 - extending functionality by plugins

Basic processing schema

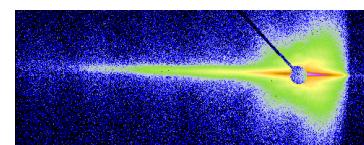
➤ Incoming Data



...

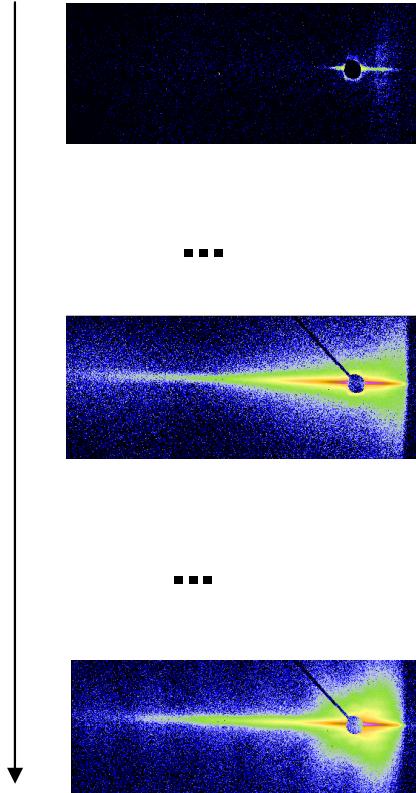


...

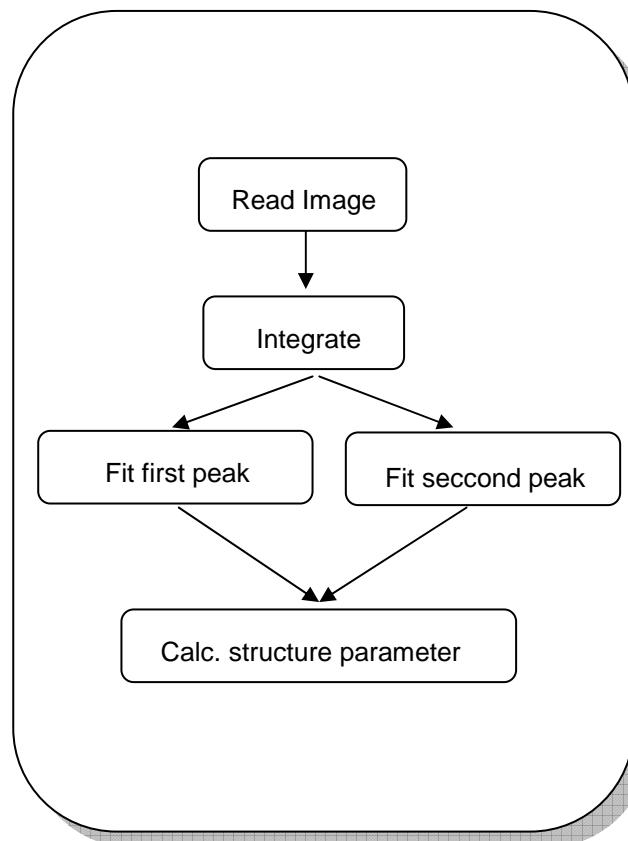


Basic processing schema

➤ Incoming Data

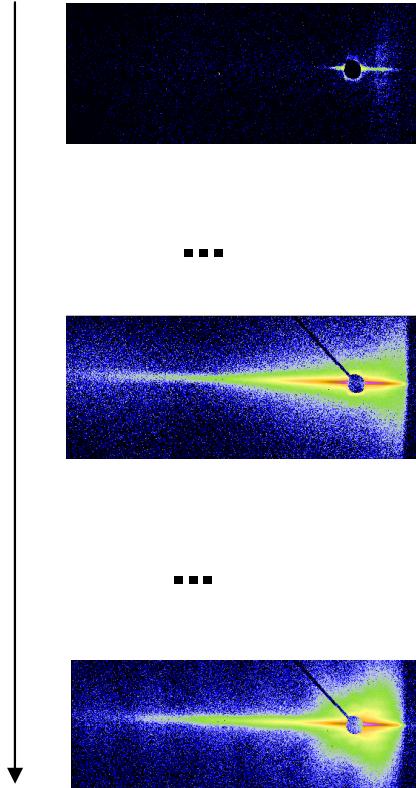


➤ Data reduction

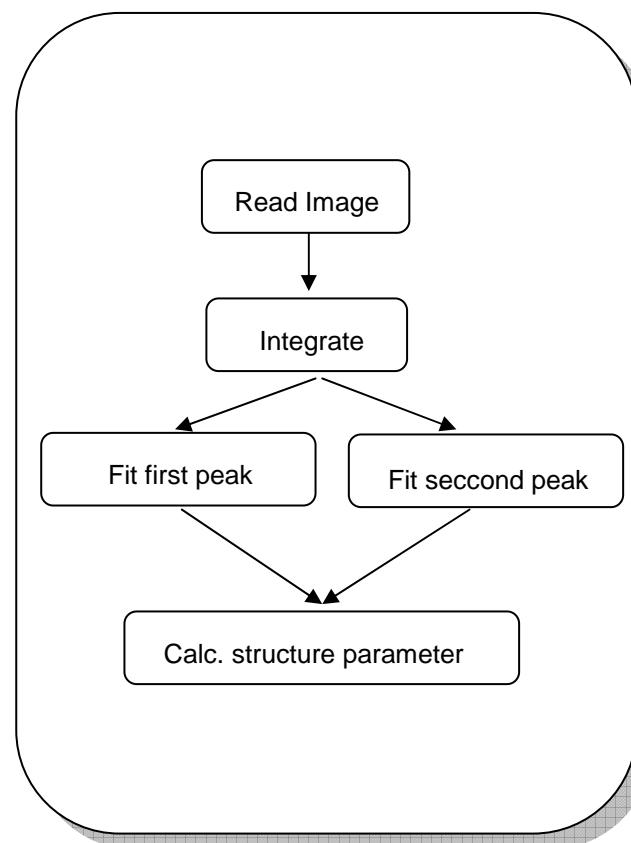


Basic processing schema

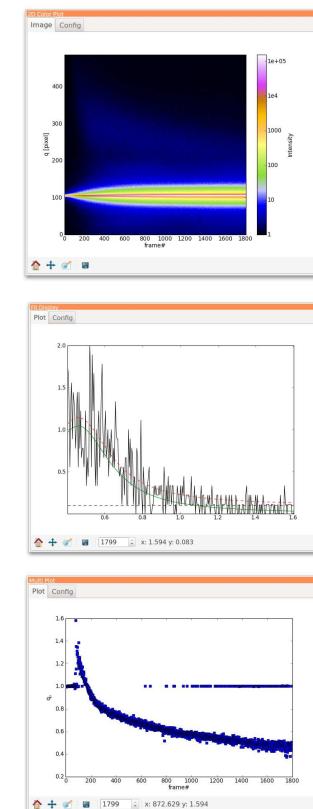
➤ Incoming Data



➤ Data reduction



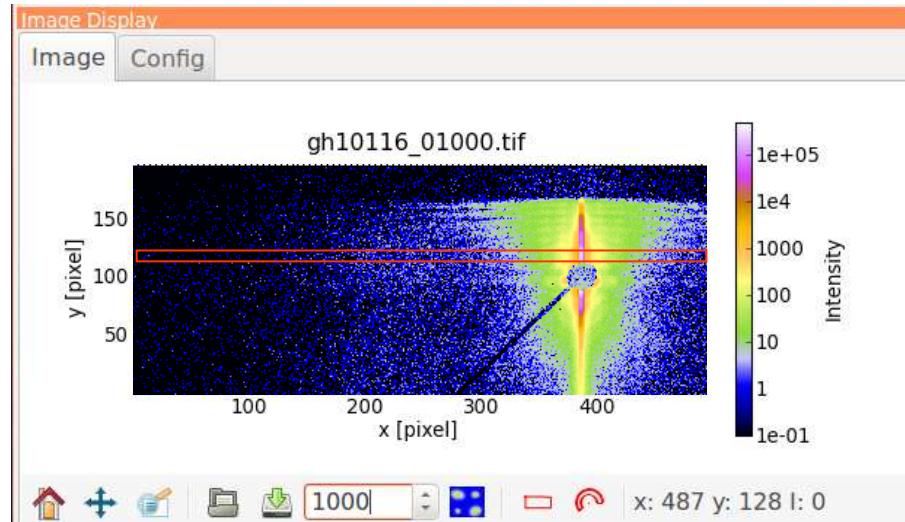
➤ Display results



Use Case I: Sputter Experiments

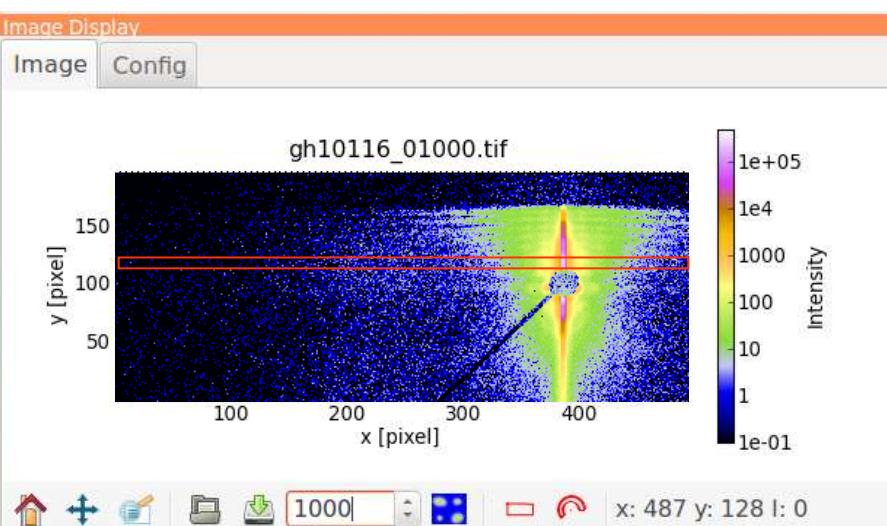
- Online (up to a frame every 20ms) plot of cuts show lateral structure

- Detector image with cut tool



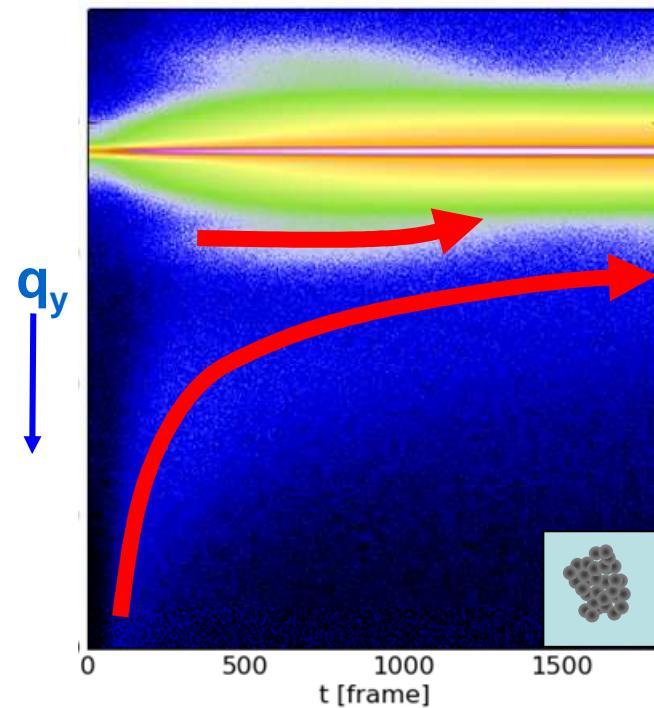
Use Case I: Sputter Experiments

- Online (up to a frame every 20ms) plot of cuts show lateral structure



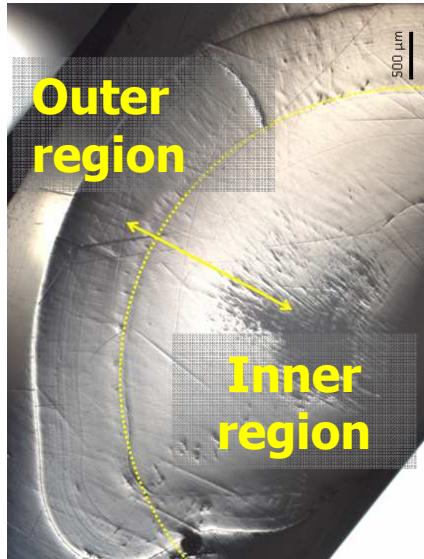
- Detector image with cut tool

- Broad structure at ~30nm
- Co nanoparticles, R(Co) growing



Use Case II: Crayfish Gastrolith Nanostructure

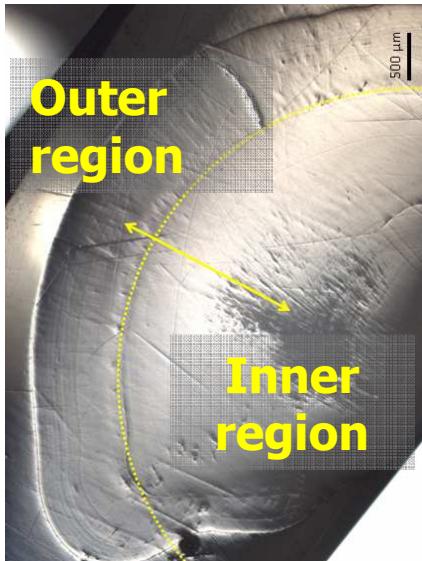
➤ Sample Data



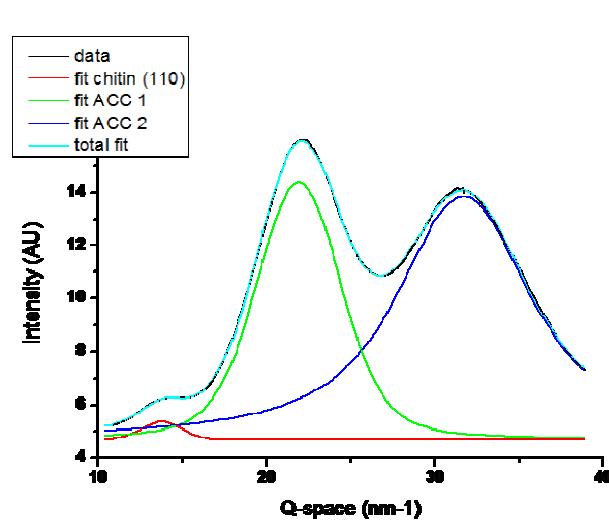
- 2D scan of 150 mccd detector files
- 10 samples scanned
- 20GB data

Use Case II: Crayfish Gastrolith Nanostructure

> Sample Data



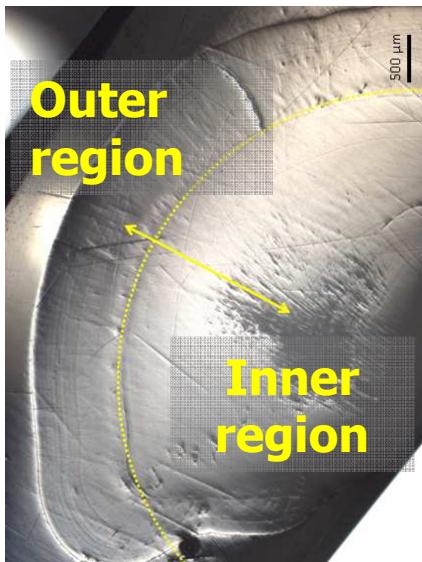
> Processing



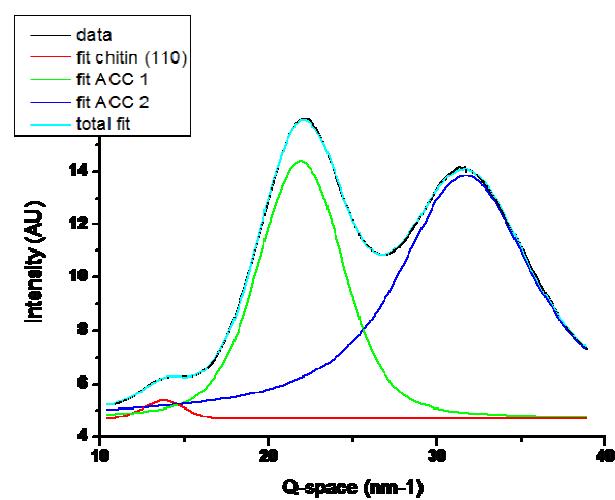
- 2D scan of 150 mccd detector files
- 10 samples scanned
- 20GB data
- integrating 2D data
- fitting chitin and ACC peaks
- calculate area of peaks and ratios

Use Case II: Crayfish Gastrolith Nanostructure

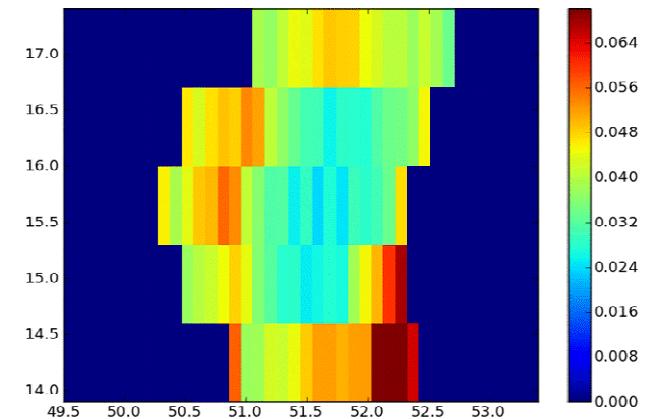
> Sample Data



> Processing



> Result

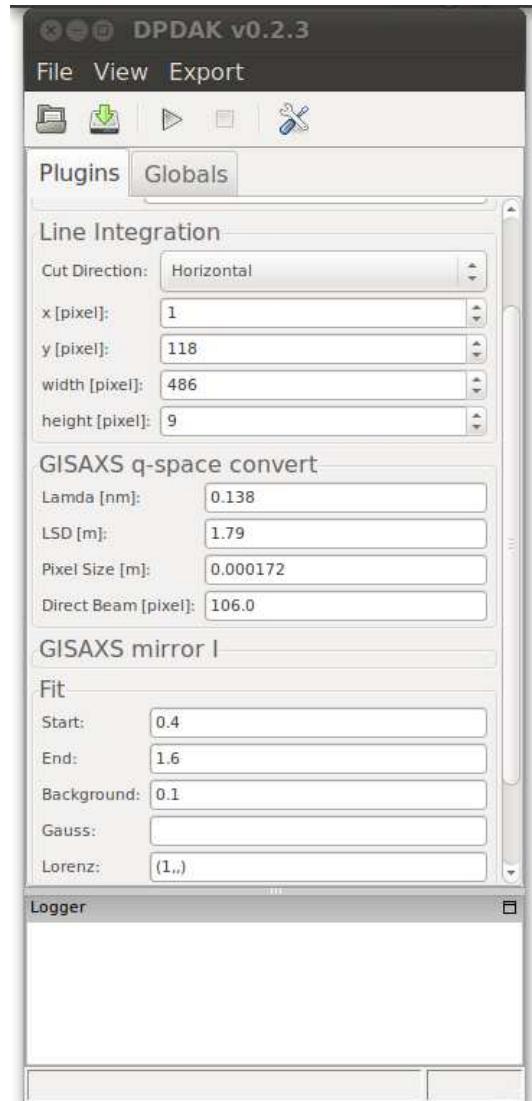


- 2D scan of 150 mccd detector files
- 10 samples scanned
- 20GB data

- integrating 2D data
- fitting chitin and ACC peaks
- calculate area of peaks and ratios

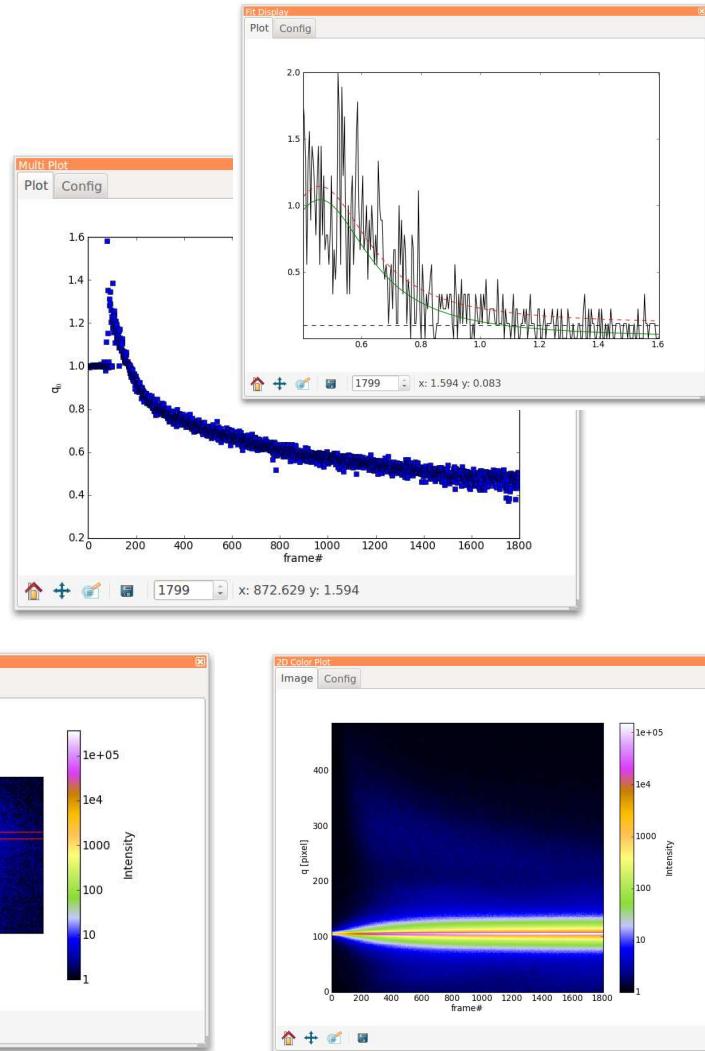
- 2D plots of ratios
- created online during the scan

Screenshot of a analysis session



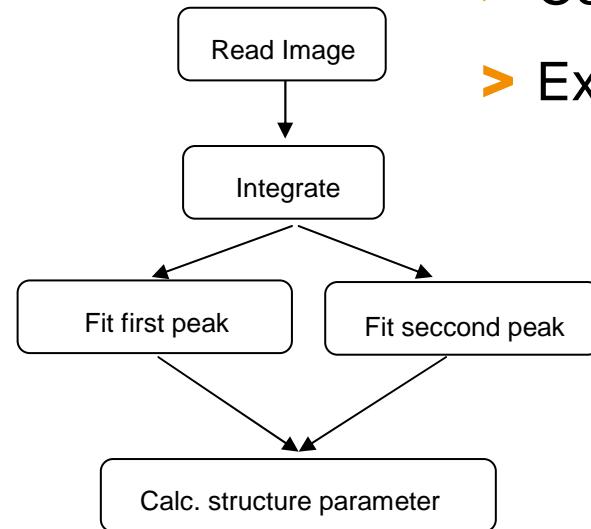
> Configuration window for plugins

> Displays for analysis results



Plugin Framework

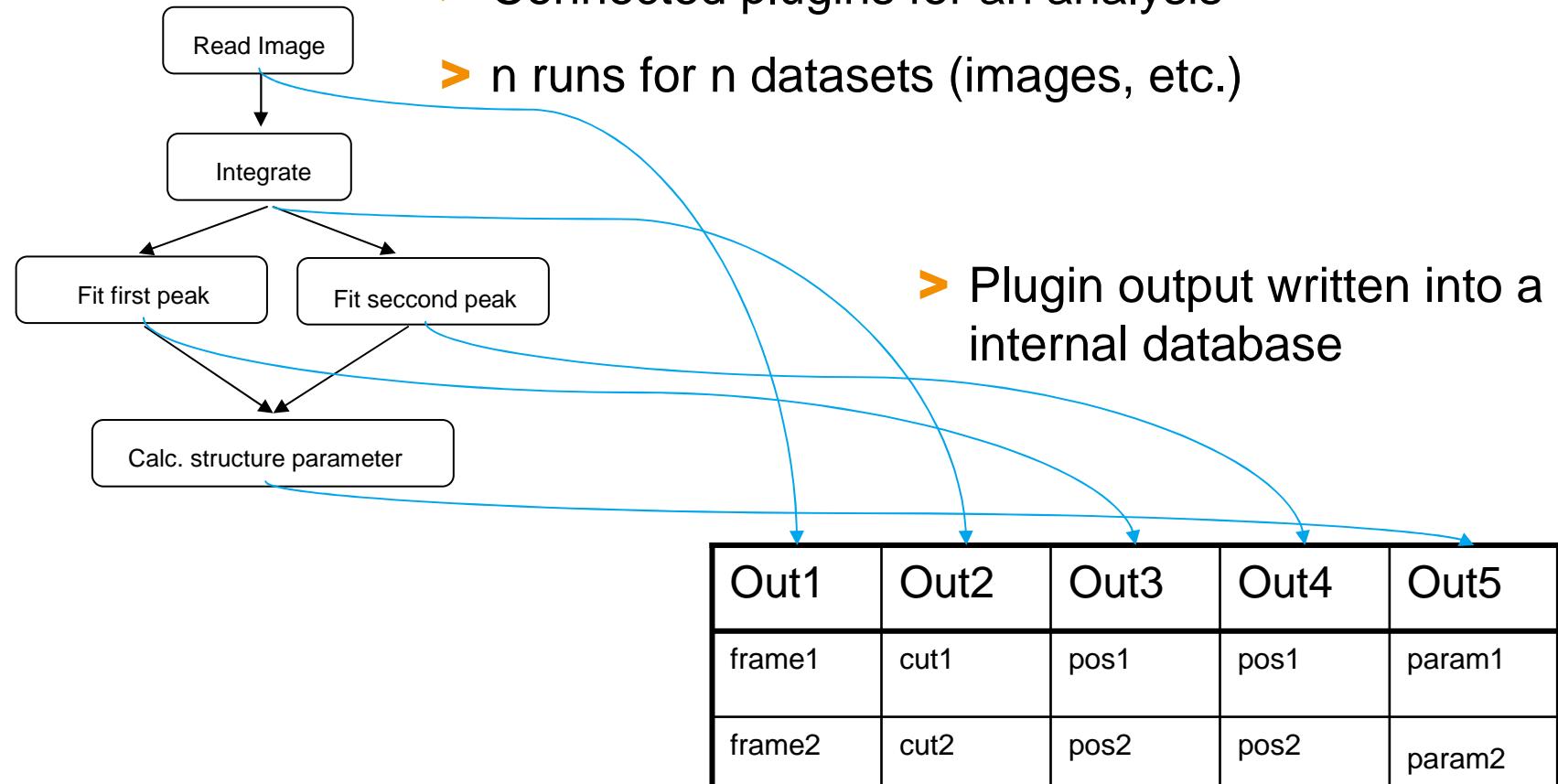
- Connected plugins for an analysis
- Execute analysis for each data point in the dataset



Plugin Framework

➤ Connected plugins for an analysis

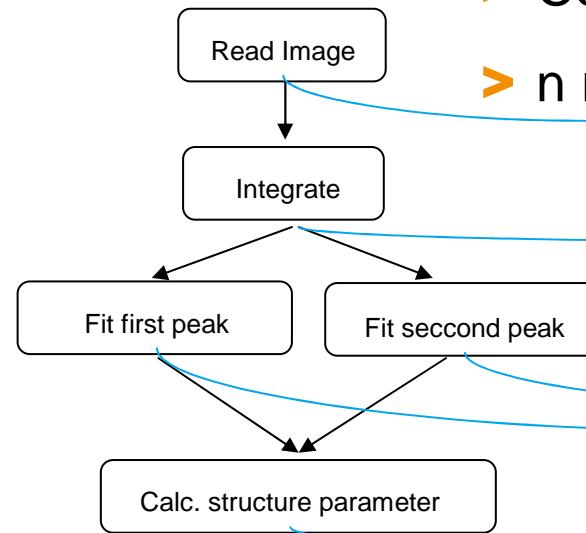
➤ n runs for n datasets (images, etc.)



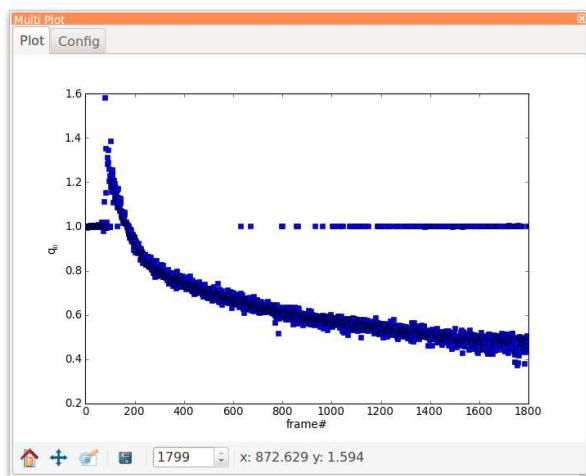
Plugin Framework

➤ Connected plugins for an analysis

➤ n runs for n datasets (images, etc.)



➤ Plugin output written into a internal database

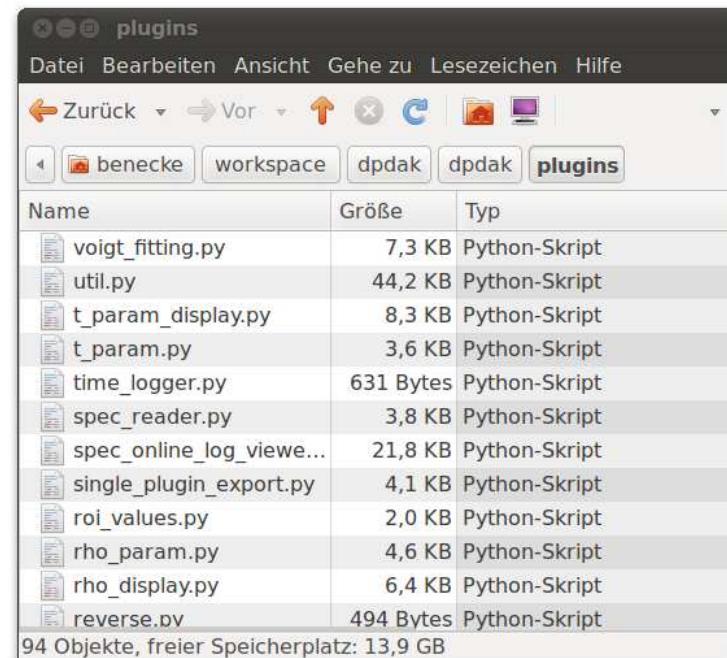


Out1	Out2	Out3	Out4	Out5
frame1	cut1	pos1	pos1	param1
frame2	cut2	pos2	pos2	param2

➤ Display plugins read the database and visualize the results

Plugins

- Plugins are python scripts
- Framework import plugin scripts on runtime
- Plugin interfaces for:
 - Data input & analysis (base plugins)
 - Visualisation (display plugins)
 - Export to different formats (export plugins)
- Base plugins are well defined:
 - Typed parameter
 - Typed inputs
 - Typed outputs



The screenshot shows a file browser window with the title bar 'plugins'. The menu bar includes 'Datei', 'Bearbeiten', 'Ansicht', 'Gehe zu', 'Lesezeichen', and 'Hilfe'. Below the menu is a toolbar with icons for back, forward, up, and other file operations. A tab bar at the top right shows tabs for 'benecke', 'workspace', 'dpdak', 'dpdak', and 'plugins', with 'plugins' being the active tab. The main area is a table listing files:

Name	Größe	Typ
voigt_fitting.py	7,3 KB	Python-Skript
util.py	44,2 KB	Python-Skript
t_param_display.py	8,3 KB	Python-Skript
t_param.py	3,6 KB	Python-Skript
time_logger.py	631 Bytes	Python-Skript
spec_reader.py	3,8 KB	Python-Skript
spec_online_log_viewe...	21,8 KB	Python-Skript
single_plugin_export.py	4,1 KB	Python-Skript
roi_values.py	2,0 KB	Python-Skript
rho_param.py	4,6 KB	Python-Skript
rho_display.py	6,4 KB	Python-Skript
reverse.py	494 Bytes	Python-Skript

94 Objekte, freier Speicherplatz: 13,9 GB

Plugin Example

```
1 import numpy
2
3 import dpdak
4
5 class ROIValues(dpdak.BasePlugin):
6
7     def get_info(self):
8         info = dpdak.PluginInfo()
9         info.set_info('ROI Values', 'Gunthard Benecke', '1.0')
10
11         info.add_input('Image', dpdak.IMAGE)
12
13         info.add_output('Sum', dpdak.SCALAR)
14         info.add_output('Mean', dpdak.SCALAR)
15         info.add_output('Min', dpdak.SCALAR)
16         info.add_output('Max', dpdak.SCALAR)
17
18         info.add_parameter('x', dpdak.INTEGER, 0)
19         info.add_parameter('y', dpdak.INTEGER, 0)
20         info.add_parameter('width', dpdak.INTEGER, 10)
21         info.add_parameter('height', dpdak.INTEGER, 10)
22
23     return info
24
25 def process(self, counter, parameter, input):
26     x, y = parameter['x'], parameter['y']
27     width, height = parameter['width'], parameter['height']
28
29     image = input['Image'][y:y+height, x:x+width]
30     data = {}
31     data['Sum'] = image.sum()
32     data['Mean'] = sum / (image.shape[0] * image.shape[1])
33     data['Min'] = image.min()
34     data['Max'] = image.max()
35
36     return (dpdak.STATUS_OK, data)
```

Features

> Software

- Python + numpy, scipy, wx, matplotlib, PIL, h5py
- Windows (XP/Vista/7), Linux
- GNU GPL licence

> Formats

- Detector images (tif, cbf, mccd, edf, ima, txt, ...)
- SPEC & Hasylab Online log file → **a standart data format would improve usability**
- Spreadsheet text files

> Analysis

- Integration (azimuthal/radial) & cuts
- Basic fitting
- Structure parameter (Rho/T/L)

> Visualisation of detector images, 2D maps of cuts and plots

Contact

➤ Web:

- <http://www.desy.de/~benecke/dpdak/>
- <http://dpdak.desy.de> (wiki, March 2012)

➤ SVN:

- <https://svnsrv.desy.de/viewvc/dpdak/>
- open hoster (sourceforge, github, google google)