Volume 35 • Number 3 • September 2015

AN INTERNATIONAL JOURNAL

Published in association with AIRAPT and EHPRG

HIGH PRESSURE RESEARCH

Editor in Chief: Stefan Klotz

Regional Editors: Yanbin Wang and Yoshinori Katayama

Founding Editors: Marvin Ross and Jean-Michel Besson

ISSN 0895-7959

Taylor & Francis
Taylor & Francis Group

Taylor & Francis
Taylor & Francis Group

High Pressure Research

An International Journal

ISSN: 0895-7959 (Print) 1477-2299 (Online) Journal homepage: http://www.tandfonline.com/loi/ghpr20

DIOPTAS: a program for reduction of two-dimensional X-ray diffraction data and data exploration

Clemens Prescher & Vitali B. Prakapenka

High Pressure Research, 2015
Vol. 35, No. 3, 223–230, http://dx.doi.org/10.1080/08957959.2015.1059835

DIOPTAS: a program for reduction of two-dimensional X-ray diffraction data and data exploration

Clemens Prescher* and Vitali B. Prakapenka

Center for Advanced Radiation Sources, University of Chicago, Chicago, IL 60439, USA

(*Received 12 May 2015; final version received 4 June 2015*)

The amount of data collected during synchrotron X-ray diffraction (XRD) experiments is constantly increasing. Most of the time, the data are collected with image detectors, which necessitates the use of image reduction/integration routines to extract structural information from measured XRD patterns. This step turns out to be a bottleneck in the data processing procedure due to a lack of suitable software packages. In particular, fast-running synchrotron experiments require online data reduction and analysis in real time so that experimental parameters can be adjusted interactively. *Dioptas* is a *Python*-based program for on-the-fly data processing and exploration of two-dimensional X-ray diffraction area detector data, specifically designed for the large amount of data collected at XRD beamlines at synchrotrons. Its fast data reduction algorithm and graphical data exploration capabilities make it ideal for online data processing during XRD experiments and batch post-processing of large numbers of images.

Keywords: X-ray diffraction; detector calibration; data analysis software; high pressure; diamond anvil cell; synchrotron

1. Introduction

Most of the synchrotron X-ray diffraction (XRD) data are collected employing 2D area detectors and increasingly, many laboratory-based XRD instruments are equipped with image detectors. The rate with which those images are being acquired is increasing steadily with fast and more efficient detectors becoming available. The large rate of data being collected necessitates fast data reduction in order to monitor the physical and chemical processes (phase transition, dissociation, melting, etc.) occurring in the sample and being able to make meaningful adjustments in real time during the course of the experiment. This decision process requires graphical, easy-to-use data reduction and exploration software, which must be customizable for specific experimental needs. Currently, only a few software packages provide generic tools for the reduction of two-dimensional XRD data: Fit2D,[1] Datasqueeze,[2] Nika [3] and DPDAK.[4] While in principle they all work reasonably well for the specific purposes for which they were written, they fall short in speed of image integration, customizability or data exploration capabilities. For example, Fit2D [1] has been the workhorse for data processing of high pressure synchrotron XRD experiments for almost 20 years; however, its user interface is very old, not resizable and the speed of data processing is relatively slow (a few seconds for a 2048 × 2048 pixel image). Another main issue with existing software is that detector geometry calibration capabilities are

*Corresponding author. Email: clemens.prescher@gmail.com limited, which are crucial for reliable data analysis. The inability to calibrate specific detector geometries dictates the experimental design due to a lack of suitable algorithms for calibration of more sophisticated detector geometries, for example, beam center being outside of image, relatively large detector tilts, spotty diffraction pattern of calibrant, more than 15 reflections for high-energy XRD setups.

Dioptas is written with interactivity and speed in mind while still being as versatile as possible. It employs an algorithm for calibration of any possible detector geometry, features easy-to-use masking tools, and offers very fast data exploration and phase analysis capabilities.

Furthermore, Dioptas includes unique capabilities for processing XRD data collected at high pressures and/or high temperatures, including displaying of diffraction line positions and intensities for phases at different pressure and temperature conditions, different background subtraction possibilities and absorption corrections for the pressure vessel.

2. Program structure and dependencies

Dioptas is written using the model-view-controller architecture pattern,[5] which separates the code into subparts that are either only related to the user interface (view) or the data processing (model). The connection between those parts is achieved by having additional controller modules coupling user interface to the respective data processing algorithms. As a result of this general architecture, the code is more structured and easier to maintain. In addition, all frequently changing data models employ the observer pattern, resulting in notification of changes in the data model to previously registered observers, making the program more responsive to changes and easier to extend because objects are loosely coupled and new features can be independent of the already existent ones.

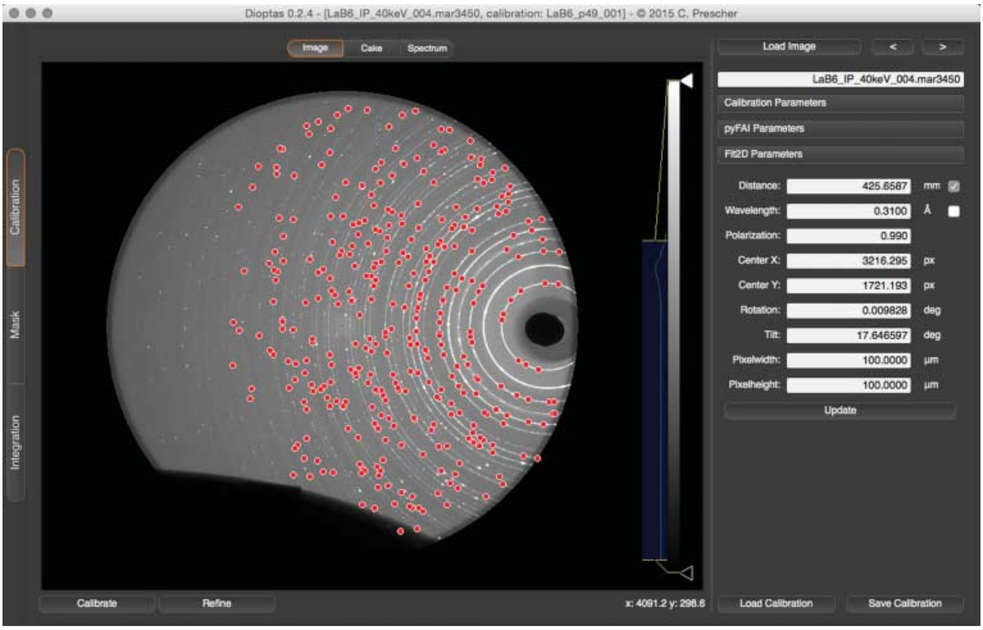
Dioptas makes heavy use of the existent scientific open source *Python* infrastructure. The graphical user interface (GUI) is written using the PyQt4 library (www.riverbankcomputing.com/software/pyqt), which provides Python bindings for the Qt4 library (www.qt.io). Instead of the typically used matplotlib library [6] for plotting images and spectra, we chose to use PyQtGraph (http://www.pyqtgraph.org) due to its superior performance, especially for displaying and manipulating images. Numerical operations and image processing are speeded up by the NumPy, SciPy [7] and Scikit-image [8] packages. Two-dimensional diffraction images are read using the fabIO library,[9] enabling support of a vast amount of data formats. For image integration and parts of the calibration process, the pyFAI [10] library is employed.

3. Distribution

Dioptas is mainly distributed via an open- source repository at http://github.com/Dioptas/Dioptas. However, since some of the required packages can be hard to install on some operating systems by non-expert end-users, we also provide executable packages via email request or via download from clemensprescher.com. Related information can be found at gsecars.uchicago.edu. Dioptas is cross-platform compatible and has been tested on Windows 7, Windows 8, Mac OS X and Linux Debian systems.

4. Capabilities and workflow description

The GUI of Dioptas is divided into three general parts, the calibration module, the mask module and the integration/analysis module, which can be accessed by the tab indicators on the left side of the GUI (*e.g.* Figure 1).



Dioptas 0.2.4 - [LaB6_IP_40keV_004.mar3450, calibration: LaB6_p49_001] - © 2015 C. Prescher

LaB6_IP_40keV_004.mar3450

Distance: 425.6587 mm ☑

Wavelength: 0.3100 Å □

Polarization: 0.990

Center X: 3216.295 px

Center Y: 1721.193 px

Rotation: 0.009828 deg

Tilt: 17.646597 deg

Pixelwidth: 100.0000 µm

Pixelheight: 100.0000 µm

x: 4091.2 y: 298.6

Figure 1. Dioptas calibration window after calibration of a tilted detector setup with the primary beam being off-center. Red points on the image are the automatically found positions on the individual diffraction rings after initializing the calibration procedure with points on the first diffraction ring. All of these points are used for the detector geometry calibration. The panel on the right shows the calibrated parameters in Fit2D geometry.

4.1. *The calibration procedure*

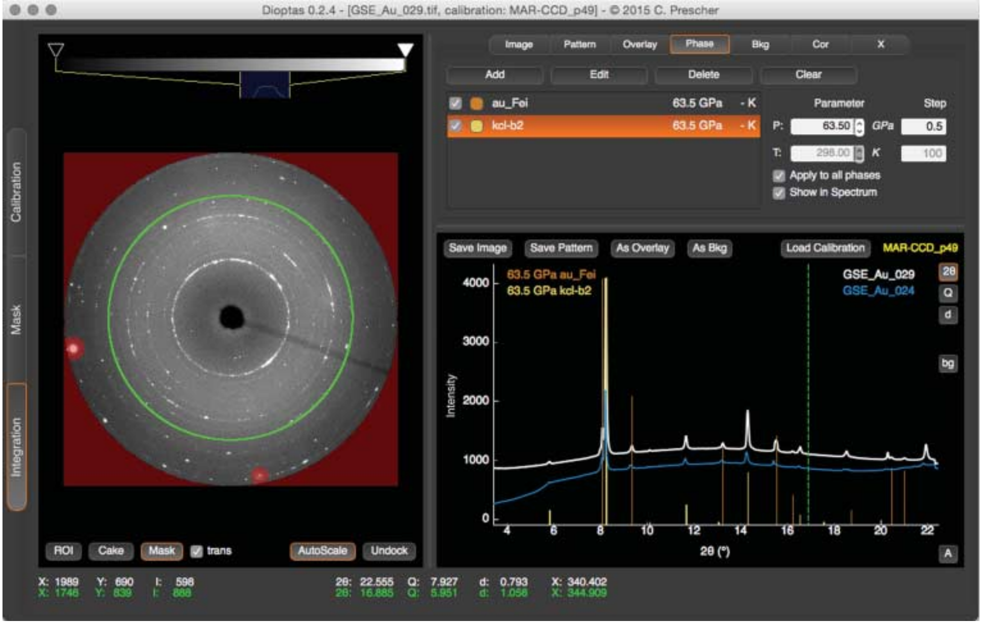
The GUI for the detector geometry calibration is shown in Figure 1. The calibration procedure is implemented as simply as possible while still being flexible enough to calibrate even the most complex detector orientation and position. The user provides an initial set of starting parameters: approximate sample-detector distance, wavelength, pixel size and selects the calibrant used in the calibration image. In the simplest case, when the XRD line intensities are uniform with high signal/background ratio, which is typical for synchrotron XRD measurements, the peak detection routine will find peaks on the innermost diffraction ring automatically after clicking close to it on the image. In any other case, there is the option to manually select and add individual peaks, if the automatic peak search failed. Furthermore, calibration does not need to start from the first diffraction ring, which enables calibration of detector geometries with the beam center being far outside of the area of the detector. After the initial peak selections, Dioptas calibrates the detector in an iterative process. The selected peaks are used for obtaining initial calibration parameters, then peaks on the first two rings are searched and the calibration parameters are recalculated, after that the algorithm iteratively adds new points for each successive ring and refines the calibration parameters for all peak positions for all found rings. The geometric refinement process has several tunable parameters for the peak search algorithm during the procedure, which enables more detailed control and handling of even difficult image qualities, for example, very spotty data or low signal intensity (Figure 1). Specific detector areas (oversaturated, shadows, non-sample related scattering, etc.) can be excluded from the refinement process by employing a mask. The tunable parameters for the peak search algorithm and the masking capability make Dioptas the most reliable software for calibration of all kinds of 2D detector geometries and signal intensities.

After the calibration procedure, Dioptas automatically creates and shows the calculated cake (integrated image along the azimuthal lines) and integrated pattern with the corresponding calibrant reflection positions as vertical lines for verification of the correctness of the calibration. The refined calibration parameters can be saved and loaded for later reuse as *.poni files with the pyFAI [10] geometry. Dioptas also displays its calibration parameters in the Fit2D geometry for cross-referencing or if some specific feature is more accessible in Fit2D.[1]

4.2. *The integration process*

The GUI for the integration module is shown in Figure 2. The integration window shows the original diffraction image and integrated diffraction pattern at the same time. Both are interconnected; clicking in either one will show the corresponding cursor position on the other. The integration window has controls for browsing image and pattern files by file number or creation time, optional automatic processing of new image files written in a specific folder and optional auto-saving of all integrated patterns. Due to the use of the speed-optimized pyFAI [10] library, browsing through the data is very fast (∼ 100 ms integration time for pattern and 120 ms for cake, on a modern PC and an image size of 2048 × 2048 pixel) even though the integration is performed on each loaded image file. Furthermore, it is possible to perform full image cake integration by just a single click or even to browse through image cakes.

The integration window provides detailed controls for the integration process. The image areas for integration can be chosen either by using a mask previously defined in the mask module (Figure 3) or by choosing a rectangular region of interest. By selecting a different output



Dioptas 0.2.4 - [GSE_Au_029.tif, calibration: MAR-CCD_p49] - © 2015 C. Prescher

☑	🟧	au_Fei	63.5 GPa	- K
☑	🟨	kcl-b2	63.5 GPa	- K

	Parameter		Step
P:	63.50	GPa	0.5
T:	298.00	K	100

☑ Apply to all phases
☑ Show in Spectrum

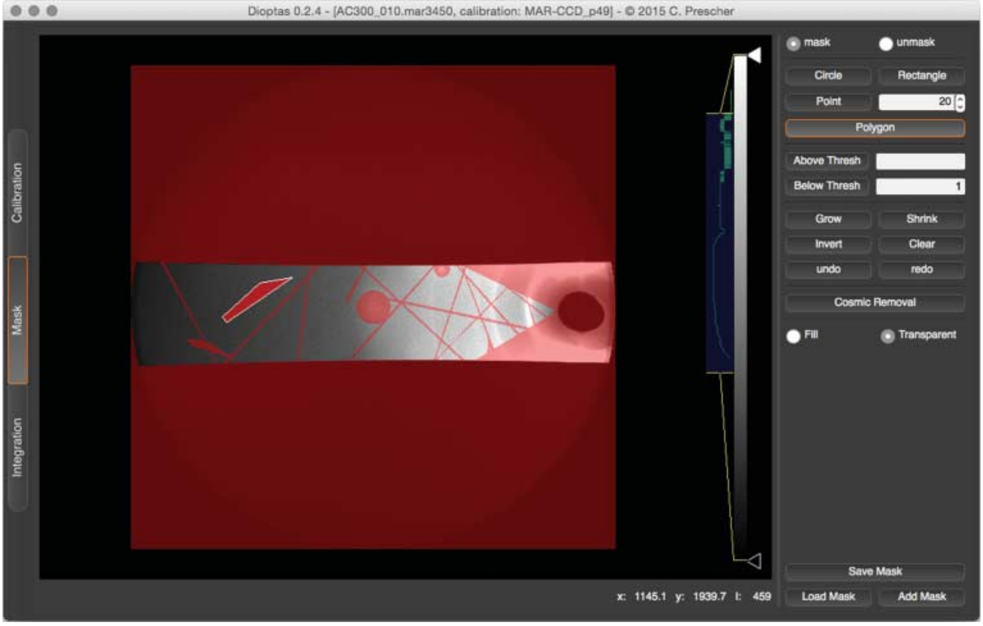
MAR-CCD_p49

☑ trans

X: 1989 Y: 690 I: 598
X: 1746 Y: 839 I: 888

2θ: 22.555 Q: 7.927 d: 0.793 X: 340.402
2θ: 16.885 Q: 5.951 d: 1.056 X: 344.909

Figure 2. Dioptas integration window. The image is displayed on the left and the corresponding integrated diffraction pattern is shown on the right as white line. The red shaded area represents the integration mask, specifying an area that is excluded from the integration procedure. The position of the green circle in the image widget corresponds to the position of the green vertical line in the pattern widget. The other vertical lines in the pattern widget correspond to the peak positions of gold (orange) and KCl (yellow) at 63.5 GPa calculated from known Birch Murnaghan equations of state given in the literature.[11,12] The blue pattern on the right is an overlay of a measurement of the same sample with laser heating mirrors inside the X-ray beam path.



Point 20

Below Thresh 1

Figure 3. Dioptas mask window showing an ongoing masking procedure of collected data that which occupies only a part of the detector and includes many artifacts not related to the sample. All available masking geometries can be created by simple clicks and double clicks with immediate visibility of which features are masked.

bin number or using supersampling, the integration process itself is customized. Supersampling divides each original pixel into 2^n equal area pixel with equal intensity, whereby n is the supersampling amount, prior to the integration process. For perfect powder XRD patterns (continuous diffraction rings in the images), this results in much smaller peak widths especially noticeable for detectors with relatively large pixel size. However, some of the effects might be artificial. Particularly for spotty or single crystal diffraction patterns, supersampling leads to erroneous, artificial results. By default, Dioptas calculates the number of integration bins to be 2 per pixel size diagonal and uses no supersampling.

Dioptas has the additional option to employ image intensity corrections. Those include: (1) self-absorption correction based on different sample shapes (plate, sphere, cylinder), (2) sample signal absorption by high pressure vessel, for example, diamond and cBN seat correction for diamond anvil cell measurements, (3) correction for change in path length in the phosphor of the detector due to different incidence angles of the diffracted photons. All corrections are calculated per pixel based on the detector geometry prior to integration and can be combined as needed.

4.3. *Data exploration*

Dioptas includes controls for file browsing based on file numbers or creation time and has the option to auto-process any newly created image file in a watched folder. Dioptas automatically integrates each loaded image file with the previously defined parameters, including mask, calibration, binning, and so on. The integrated spectra can be visually compared to each other (*e.g.* for variable temperature or pressure, two-dimensional scans) by using series of overlays, which have a scaling and offset parameter for better comparison. Dioptas offers several background subtraction options. It is possible to define a user-provided background either as a scalable image prior to

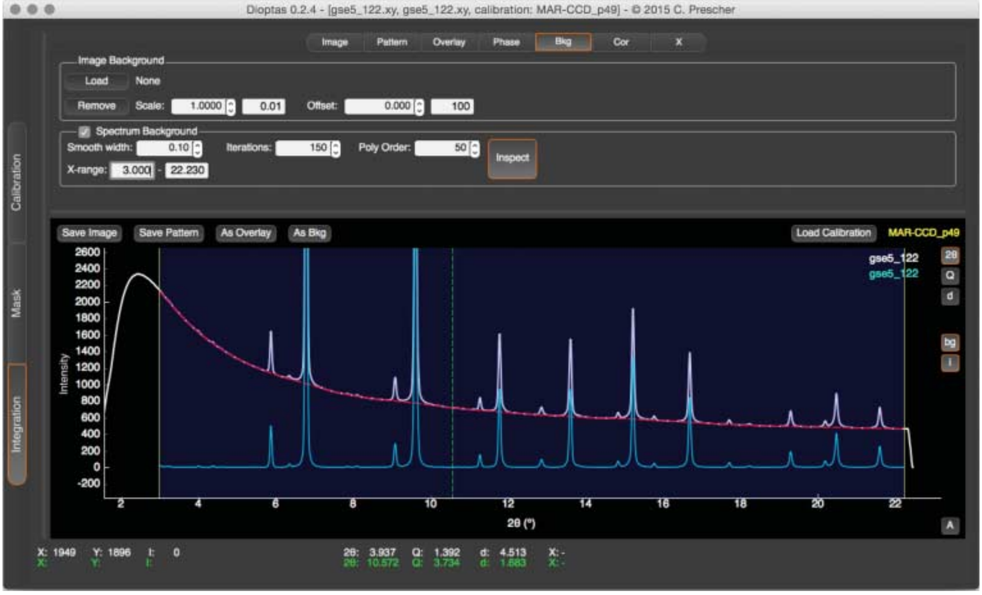

Dioptas 0.2.4 - [gse5_122.xy, gse5_122.xy, calibration: MAR-CCD_p49] - © 2015 C. Prescher

Image Background: None

Scale: 1.0000 0.01 Offset: 0.000 100

☑ Spectrum Background

Smooth width: 0.10 Iterations: 150 Poly Order: 50

X-range: 3.000 - 22.230

MAR-CCD_p49

X: 1949 Y: 1896 I: 0 2θ: 3.937 Q: 1.392 d: 4.513 X: -

X: Y: I: 2θ: 10.572 Q: 3.734 d: 1.683 X: -

Figure 4. Dioptas integration window showing the controls for automatic background subtraction. The white, dashed red and blue line in the plot widget represent the original diffraction pattern, estimated background and the background subtracted diffraction pattern, respectively. The region for background subtraction can be visually adjusted by resizing the blue shaded area using the draggable yellow lines and the background estimation is customizable by adjusting the smooth width, iterations and polynomial order used for the algorithm.

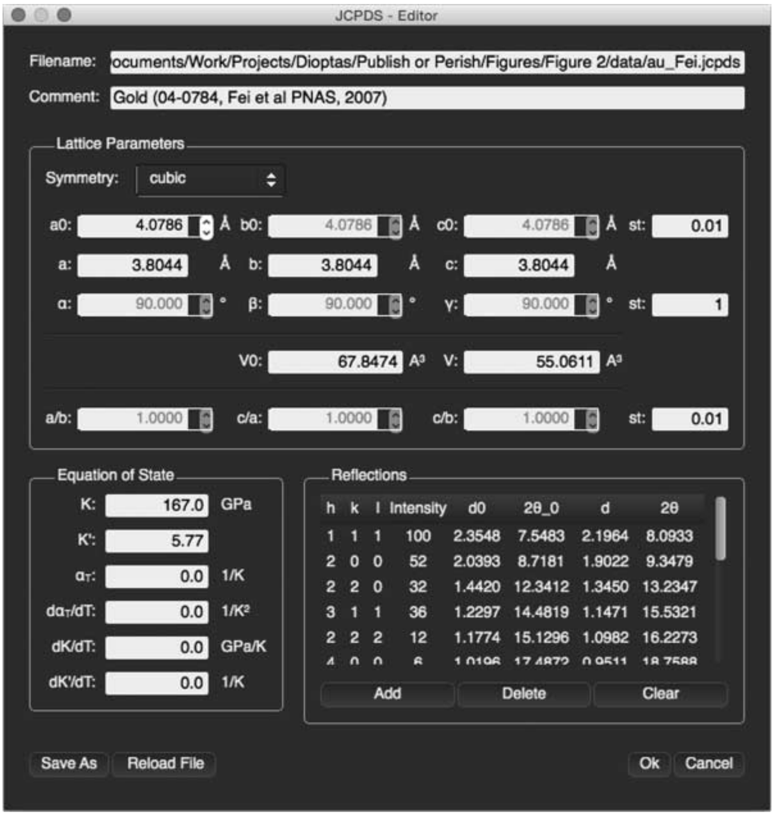
the integration process or after as scalable diffraction pattern (overlay) and a parameterized automatic background subtraction algorithm can be used if no background measurement is available (Figure 4).

Simulated XRD reflection lines of selected phases with line intensities normalized to the current integrated, visible pattern can be displayed in the pattern widget (Figure 2), whereby lattice parameters of the phases are manually adjustable and if the equation of state is provided, lattice parameters are changed automatically according to selected pressure and/or temperature conditions. The file format used to save the phase information and equation of state is jcpds. Dioptas includes an editor to create, edit and save those files for later reuse (Figure 5).

All data visualizations are highly customizable and interactive. Image color table and scaling can be changed by simple clicks. Overlay and phase line colors can be adjusted easily to better discriminate between them if the automatic color selection is not appropriate. The images and spectra contents can be easily explored by zooming and dragging with simple mouse interactions.

4.4. *Masking tools*

The GUI for masking is shown in Figure 3. A mask excludes specific areas from the integration process, which enables the removal of unwanted features such as, for example, diamond reflections in diamond anvil cell experiments, saturated peaks or detector blemishes. Available geometries for creating masks are circles, rectangles and polygons. Additionally, pixels can be masked based on threshold levels. Dioptas also includes an algorithm for finding cosmic X-ray peaks, which can arise from very long collection times and might be tedious to remove manually. The masking module has the capability to undo/redo every performed step, which is commonly needed in the masking process. Individual masks can be saved as compressed .tif files for later reuse.



Filename: ocuments/Work/Projects/Dioptas/Publish or Perish/Figures/Figure 2/data/au_Fei.jcpds

Comment: Gold (04-0784, Fei et al PNAS, 2007)

Symmetry: cubic

a0: 4.0786 Å b0: 4.0786 Å c0: 4.0786 Å st: 0.01

a: 3.8044 Å b: 3.8044 Å c: 3.8044 Å

α: 90.000 ° β: 90.000 ° γ: 90.000 ° st: 1

V0: 67.8474 A³ V: 55.0611 A³

a/b: 1.0000 c/a: 1.0000 c/b: 1.0000 st: 0.01

K: 167.0 GPa

K': 5.77

α_T: 0.0 1/K

$d\alpha_T$/dT: 0.0 1/K²

dK/dT: 0.0 GPa/K

dK'/dT: 0.0 1/K

h	k	l	Intensity	d0	2θ_0	d	2θ
1	1	1	100	2.3548	7.5483	2.1964	8.0933
2	0	0	52	2.0393	8.7181	1.9022	9.3479
2	2	0	32	1.4420	12.3412	1.3450	13.2347
3	1	1	36	1.2297	14.4819	1.1471	15.5321
2	2	2	12	1.1774	15.1296	1.0982	16.2273
4	0	0	6	1.0196	17.4872	0.9511	18.7588

Figure 5. Jcpds editor displaying the jcpds file of Gold, extracted from Fei et al. [11], used in Figure 1. All lattice parameters and d-spacings of HKL (Miller indices) reflections are shown for ambient pressure and temperature conditions in addition to the calculated values for the specifically selected pressure and temperature conditions (here 63.5 GPa). All parameters can be changed, which automatically triggers a recalculation of all derived values and also updates in the pattern widget.

5. Further development

The Dioptas features described here are just a current snapshot of the development process. We plan to further develop Dioptas to be a fully fledged analysis program of X-ray diffraction experiments. While it is now mainly used for the integration process and the integrated spectra are then analyzed with other analysis software, we plan to include extraction of pair distribution functions, peak fitting and structure refinement in the future.

6. Conclusion

The user-friendly intuitive graphical interface with its highly customizable calibration procedure and integration process make Dioptas an efficient and powerful tool for on-the-fly data processing and analysis for all kinds of experiments, either online at Large Scale Facilities or home laboratories. Dioptas has a very fast-growing user base and is currently employed for online data processing and post experiment data analysis at CARS (Sectors 13–15, APS), HPCAT (Sector 16, APS) and ID27, ESRF. By the time of this writing, already more than 60 scientists are registered Dioptas users.

Acknowledgements

CP thanks Matt Newville and Mark Rivers for all the answers to CPs endless Python and general programing-related questions. Furthermore, the authors want to thank Dan Shim for sharing his implementation of the automatic background subtraction algorithm and we thank Joanne Stubbs for proofreading of the article.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was performed at GeoSoilEnviroCARS (Sector 13), Advanced Photon Source (APS), Argonne National Laboratory. GeoSoilEnviroCARS is supported by the National Science Foundation – Earth Sciences (EAR-1128799) and Department of Energy – GeoSciences (DE-FG02-94ER14466). This research used resources of the Advanced Photon Source, a US Department of Energy (DOE) Office of Science User Facility operated for the DOE Office of Science by Argonne National Laboratory under Contract No. DE-AC02-06CH11357.

References

[1] Hammersley AP, Svensson SO, Hanfland M, Fitch AN, Hausermann D. Two-dimensional detector software: from real detector to idealised image or two-theta scan. High Pressure Res. 1996;14:235–248.

[2] Heiney PA. Datasqueeze: a software tool for powder and small-angle X-ray diffraction analysis. Comm Powder Diffr Newsletter. 2005;32:9–11.

[3] Ilavsky J. Nika: software for two-dimensional data reduction. J Appl Crystallogr. 2012;45:324–328.

[4] Benecke G, Wagermaier W, Li C, et al. A customizable software for fast reduction and analysis of large X-ray scattering data sets: applications of the new DPDAK package to small-angle X-ray scattering and grazing-incidence small-angle X-ray scattering. J Appl Crystallogr. 2014;47:1797–1803.

[5] Krasner GE, Pope ST. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. J Object Oriented Program. 1988;1:41.

[6] Hunter JD. Matplotlib: a 2D graphic environment. Comput Sci Eng. 2007;9:90–95.

[7] Jones E, Oliphant T, Peterson P, et al. {SciPy}: open source scientific tools for {Python}. Available from: http://www.scipy.org/.

[8] Van der Walt S, Schönberger JL, Nunez-Iglesias J, et al. Scikit-image: image processing in Python. Peer J 2014;2:e453.

[9] Knudsen EB, Sørensen HO, Wright JP, Goret G, Kieffer J. FabIO: easy access to two-dimensional X-ray detector images in Python. J Appl Crystallogr. 2013;46:537–539.

[10] Ashiotis G, Deschildre A, Nawaz Z, et al. The fast azimuthal integration Python library: pyFAI. J Appl Crystallogr. 2015;48:510–519.

[11] Fei Y, Ricolleau A, Frank M, Mibe K, Shen G, Prakapenka V. Toward an internally consistent pressure scale. P Natl Acad Sci USA. 2007;104:9182–9186.

[12] Campbell AJ, Heinz DL. Compression of KCl in the B2 structure to 56 GPa. J Phys Chem Solids. 1991;52:495–499.