Stanford Synchrotron Radiation Lightsource

Scattering and In-situ Data Processing Workshop
July 18, 2018
Notes and Analysis Notebooks updated Feb 2020

Donata Passarello, Natalie Geise*

* E-mail: donata@slac.stanford.edu, natalie.geise@gmail.com

I. INTRODUCTION

This workshop will cover how to calibrate and process scattering and diffraction data from in-situ and operando measurements at SSRL. It will discuss the types of files produced during your beamtime and how to interpret them. In the tutorial section, we use LaB6 calibrant image to demonstrate how to use pyfai-calib to calibrate the detector with a single image. We then use a silicon calibrant to calibrate the set-up with a series of detector images from a data set at 7-2. Calibration of the detector and sample orientation should be done for each beamtime and/or setup. Typically, you'd calibrate at the beginning of each beamtime, even if you have the "same setup" because the detector's location and rotation are not exactly the same as your previous visit. You'd also calibrate again during a beamtime if you switch set ups - for example moving the detector back at beamline 11-3. The calibration is used to process all diffraction images and scans taken from a sample with that set up. The last section uses a processed series of intensity vs q vector curves from an in-situ measurement to create waterfall and time-correlated plots.

II. DATA FILES

1. Detector Images
   The detector images are saved as .tif files for Mar detectors or .raw files for the pilatus detectors. Both can be opened with ImageJ, which is on all diffraction beamline computers and available for free download online.
   - opening .raw files in python and PyFAI: the images cannot be opened as straightforwardly as the ones in the pyFAI documentation. The read raw function in the tutorial addresses this by reformatting the data based on the detector dimensions.
2. Configuration Files
   Here we start from the "largest" file structure and moving down to the individual images. Some of the information is saved in multiple locations among the filetypes.
   - spec files: For each filename (initiated through for example `newfile battery1`), there is a textfile created, with details about all scans taken with that filename. It retains relevant information for that set of scans, e.g. scan type, time stamp for each scan, diffractometer motor positions,.
   - csv files: For each scan (initiated through for example `ascan th 5 30 50 5`), the csv files tabulate relevant information for each image in the scan (ie for all 50 scans in the example above), including diffractometer motor positions, intensities of user-designated regions of interest (ROIs), ion chamber and monitor readings.
   - pdi files: For each image taken, the image itself is saved and this pdi file retains relevant information for the respective Pilatus image, e.g. diffractometer motor positions, regions of interest.

III. PYFAI

PyFAI is a python library for azimuthal integration of 2D diffraction data, with relatively easy-to-load detector, mask, and calibration settings developed at ESRF. It is well documented on-line at http://www.silx.org/doc/pyFAI/dev. If using for published data, please cite their publication: The fast azimuthal integration Python library: pyFAI. J. Kieffer, G. Ashiotis, A. Deschildre, Z. Nawaz, J. P. Wright, D. Karkoulis, F. E. Picca Journal of Applied Crystallography (2015) 48 (2), 510-519.

Some useful functions are:

- calibration:
  pyFAI-calib2, which is a GUI, for single image calibration and mask making. Area diffraction patterns at SSRL from the Pilatus detectors are saved as raw images, which must be converted to .tif form to be opened with the pyFAI-calib tools.

- azimuthal integration: the pyFAI-integrate tools perform the actual integration processes to create 1D and 2D representations of the diffraction data (q vs chi, intensity vs chi, intensity vs q) both in single and multi-geometry scans. There is additional pyFAI functionality for GIWAXS and SAXS which we will not cover here.

Hands-On Analysis

- We will start with calibrating a single image like would be obtained at SSRL's 11-3. Notebook: "multi-geometry calibration Si"
- Now, we will calibrate a set of images of a calibrant, like would be obtained at 7-2 or 2-1. This is data from from a scan of Si from the command `ascan  tth 10 40  50 1`.
- We will now use that calibration to process a scan obtained with that same geometry in a scan of the sample from the command `ascan tth 5 75  35 5`.
- We will now use a set of processed 1D (intensity vs q) scans to make some waterfall-type plots and represent our operando data sets.

IV. HELP

PyFAI is well documented online, but note that the ReadTheDocs site is no longer being updated while the silx site is. There is a pyFAI email list that can help with specific questions about pyFAI.

Comments or suggestions about this workshop/tutorial are welcome and can be sent to Natalie.