**DaFy tutorial**

Jackey Qiu

* DaFy is Python software package short for Data Analysis Factory.
* It is designed to process data in a pipe-line pattern, which comprises of several 'Data\_Filter-->Data\_Engine' components. Filter and Engine packages are stored in seperated Python packages, which can be called in the main script. In principle, all functions during data analysis can be wrapped into either a filter-like or engine-like objects. To make the main script concise and readable, we should be forced to do that if you want to make any script contribution to DaFy package.
* Inside DaFy, you can write many main scripts for different projects to expand its functionality. To develop a project, you write first the associated filter and engine objects to be called in your main script.
* At present, I have written the CV\_XRD\_DaFy.py and CV\_XRD\_DaFy\_BKG.py projects, which are used to process CV\_XRD data collected from either P23 or ID03 beamline. CV\_XRD\_DaFy is used to process data to extract structural information, including crystallite size, strain in both horizontal and vertical directions. So far, it has been customized to process datasets collected from ESRF ID03 and DESY P23 beamlines. CV\_XRD\_DaFy\_BKG has one more functionality, which is to do background subtraction for diffraction images. In principle, these two programs can be used to process CV\_XRD, potential step and rod scan data.

**Instruction of running CV\_XRD\_DaFy**

\*\*Dependency\*\*

scripy, numpy, PyMca5, matplotlib

[mpi4py] is optional, only reqired if you want to run the script on multi-processors

\*\*Project overview\*\*

The main script take in spectial configuration files to reconize which data format to process (P23 or ID03). You should edit the configuration file (edit the config\_p23\_template.ini or config\_id03\_template.ini for either p23 data or id03 datasets) accordingly for data processing.

The main script will loop through all frames to extract following informations in the reciprocal space: peak position, peak width.

You can easily customize the peak fitting by editing the configuration file accordingly.

\*\*Step by Step running procedure\*\*

1. Edit the config file (config\_id03\_template.ini or config\_p23\_template.ini). Most important items to be changed include 'spec\_file', 'img\_path',

'scan\_no','l\_scan', 'cut\_offset'. Read the comments inside the config file to get some sense of each item. Most are self explanatory.

2. Change the config\_file in the main script (line 51 of CV\_XRD\_DaFy.py)

3. Set debugging par

\*line 22 debug = 1 for debug mode, under which only one frame will be processed. It offers you a quick eye-check of the fit quality.

Based on what you see, you may need to change the items inside the config file, eg the width of cut.

\*If you are satisfied with the fit, then set the debug to 0, which means you want to process all frames for the scan. Then you can either

set the live\_image to 1 to show live-image during data analysis or to 0 without showing live image for fast analysis.

4. Then run this script in a terminal: python CV\_XRD\_DaFy.py. Sit back to get a coffee while waiting.

5. pitfalls: sometimes you need to play with the mask threshold values (line 148 in CV\_XRD\_DaFy.py), which is used to exclude hot pixels. If this value is set too high, some hot pixels are not screened out. Then the peak locating algorithm could fail due to the influence of these hot pixels. On the other hand, if is value is too load, you will screen out pixels inside your targeted peak, which is also not good. So play with this value. Center pix position is very important in the script, so you must double check this value is right.