**SuPerRod User Manual**

Canrong Qiu

1. Introduction of SuPerRod

SuPerRod is a nickname of GenX, which is originally developed by Matts Björck from Swedish Nuclear Waste and Management Company AB. GenX is developed for modeling surface scattering data (X-ray and neutron) using the global optimization fitting algorithm (differential evolution). SuPerRod is built on the backbone of GenX to fit X-ray surface diffraction data, including crystal truncation rod and resonant anomalous X-ray reflectivity. A lot of other functionalities have been incorporated to the based backbone of GenX in SuPerRod to allow it for dealing with a range of extra tasks, including data processing (background subtraction) in a semi-automatic way, building model files in a customized way, fitting surface models based on both local computer and super computation facilities, outputting bestfit model results for quick-checking purpose and for publication purpose.

To use SuPerRod, you need some basic knowledge of Python scripting language. Any research work in a publication, which has benefited from SuPerRod, should cite Matts Björck ‘s paper about GenX (Journal of Applied Crystallography, 2007, 40, 1174-1178) as an acknowledgement.

This tutorial will walk you through main features of SuPerRod.

1. SuPerRod installation

To launch SuPerRod, make sure you have installed Python (2.7 recommended) as well as following Python site-packages. Note the following combination of package versions is just one possible option, you can have other combinations as long as these packages are compatible to each other.

numpy: 1.13.3

matplotlib: 1.5.1

scipy: 1.0.0

wxpython: 2.8.12.0

ipython: 4.0.1

pandas: 0.22.0

The source code of SuPerRod can be found in my github website: <https://github.com/jackey-qiu/SuPerRod>, where you can download the whole package in a zip file or do ‘git clone <https://github.com/jackey-qiu/SuPerRod.git>’ in a git bash terminal window.

1. CTR/RAXR data processing

Location of the associated module: SuPerRod//accessory\_functions//ctr\_data\_processing//data\_integration.py

The background subtraction algorithm is developed by Vincent Mazet (Chemom. Intell. Lab. Syst. 76 (2), 2005). The code was originally written by Vincent Mazet based on MATLAB. I translated the scripts to Python language. Correction factors are calculated using TDL modules, developped and maintained by GSECARS 13IDC beamline at APS (Peter Eng and Joanne Stubbs are responsible persons).

1. Modify the information in the module:

Spec\_path: absolute path to the folder holding the spec file

Spec\_name: spec file name

Substrate: the name of the substrate of your sample (either ‘hematite’ or ‘muscovite’ at this moment.

Beamline: either ‘APS’ or ‘ESRF’

You may also need to fine-twick other parameters (read the documentation in the module file)

1. Run the file:

>>>Import data\_integration

>>>data=data\_integration.data\_integration()

1. Manually check the background subtracted profiles:

>>>data.integrate\_images\_twick\_mode(scan\_number=[‘feed in a list of scan number here’])

>>>use hotkeys to adjust the data integration (mostly adjust the integration window)

1. Do q correction

>>>data.q\_correction(scan\_number=?)

Interactively change the scaling factor, if you are satisfied, type SL(L is the Bragg peak L you are considering now), then move on to another L. Once you are done, type q to finish the q correction.

1. Save data

>>>data.save\_data(labels={'First\_point':[scan\_number\_start,scan\_number\_end]})

1. Pickle dump the processed data (can be restored in the future)

>>>data.dump\_data\_info()

1. Data formatting
2. Build model files
3. Model refinement
4. Output model results