Indexing chess data to make large reciprocal space maps. The Indexing workflow is described in the google doc CHESS Data indexing

Scripts for Stacking and Indexing CHESS data written in python. Some parts of indexing originally written by J.P. Castellan and J. Ruff, modified by A. Singer and J. Gollwitzer. All other codes written by J. Gollwitzer

1) Stacking.py: make a stack of metadata by reading in detector .cbf images and metadata from the spec file

2) Peaklist\_from\_Images.py: find the peaks to use for orienting the crystal

3) UBmatrix\_finder.ipynb: orient the crystal by matching the peaks, geometry and crystal structure. From random matrices, finds the minimum orientation matrix so that the peaks are placed on integer numbers of reciprocal space lattice H, K, and L.

4) Indexing.py: apply the UB matrix to the detector images to obtained the indexed object. you can define the HKL resolution.

5) Visualize.ipynb: visualize the indexed object, define a matrix which tweaks the UB matrix so the peaks in the HK plane are aligned