

2. Example analysis notebooks



DO NOT MODIFY THE NOTEBOOKS IN THEIR NATIVE LOCATION

Always clone the repo (`git clone /sdf/data/lcls/ds/xpp/xpptut15/results/example_notebooks`) or copy the files to your area.

For many experiment, the analysis of the smalldata h5 files is not too computing intensive and the analysis team can benefit from the quick and iterative nature of the Jupyter notebooks framework over the more involved cube production.

These notebooks can be found in `/sdf/data/lcls/ds/xpp/xpptut15/results/example_notebooks`.

It is highly recommended to be able to refer to the relevant example notebook while going through this part of the documentation.

In this section, a few template notebook for common experiments are discussed.

Common features

All templates will start with a section where the experiment and run can be defined. Make sure to comment out the example file line when actual data are to be analyzed.

Define alias and filters

Sets of alias in the h5 files are defined in the following cell. This makes it convenient to change the target variable for the entire workflow, without having to change the variable name in each cell. The variables to be used for filtering (typically `i0`, timing tool fit, ...) are also defined here.

```
# helper dict to find data in file
dataDict = {'alias': 'path/in/the/h5',
            ...
            }

filters = {}
filters['alias_filt1'] = [low,high]
filters['alias_filt2'] = [low,high]
...

print('Filter selection:')
for key,value in filters.items():
    print('\t{} : {}'.format(key, value))
hist_bins = {key: 35 for key in filters.keys()}
```

Make sure to adapt the alias to the detectors names in your smalldata h5 file. The object `rr` (created from `rr = tables.File(f).root`) in the workspace allows for easy tab-exploration of the content of the h5 file to figure out variable names.

Example from `trXRD.ipynb`

While most value often remain the same for the entire experiment, the scanvar should be updated to match the axis being scanned for each run (delay, energy, azimuth, ...).

```
# helper dict to find data in file
dataDict = {
    'scanvar': 'enc/lasDelay',
    'i0': 'ipm2/sum',
    'roi': 'jungfrauM/ROI_0_area',
    'xon': 'lightStatus/xray',
    'lon': 'lightStatus/laser',
    'tt_corr': 'tt/ttCorr',
    'tt_fwhm': 'tt/FLTPOSFWHM',
    'tt_amp': 'tt/AMPL'
}

filters = {}
filters['i0'] = [200,10000]
filters['tt_amp'] = [0.02,0.2]
filters['tt_fwhm'] = [50,240]
damage = [dataDict['roi'].split('/')[0], dataDict['i0'].split('/')[0]]

print('Filter selection:')
for key,value in filters.items():
    print('\t{} : {}'.format(key, value))
hist_bins = {key: 35 for key in filters.keys()}
```

Check filter validity

The goodness of the filters be be evaluated in the next cell, where histograms of the variables are shown together with the selected boundaries..