1. Generation of the small data hdf5 files

The small data generation takes advantage of the local development of the PSANA framework. It can be customized and run both against the xtc files while they are being written on the ffb system (ongoing experiment only) as well as against the .xtc files on the offline system,. Parallel computing has been built-in. For a typical experiment your instruments ECS engineer will set up the code in the directory `/reg/d/psdm/<hutch>/<expname>/results/smalldata_tools` for the offline running or `/cds/data/drpsrcf/<hutch>/<experiment>/scratch/smalldata_tools` when running in the new ffb.

A "driver" python file (typically called `smd_producer.py` in the `producers` subdirectory) can then be edited to, e.g., choose and optimize a different ROI on area detectors, define beam center for radial integration, define delay time range and bin sizes, etc. How to set up this "userData" or reduced data / feature extracted data is described in more detail in A. Configuring smalldata production.
The output will be saved to a directory that can be specified, by default this will be:
`/reg/d/psdm/<hutch>/<expname>/hdf5/smalldata` or `/cds/data/drpsrcf/<hutch>/<experiment>/scratch/hdf5/smalldata`
The filenames will be of the form: <expname>_Run<runnr>.h5

Setup and automatic production

During the experiment we will produce the smallData files automatically. Since Run 18, we are using the Automatic Run Processing (ARP) for that. During experimental setup, we usually take test runs of the appropriate length to set up production to finish close to the end of run. Jobs can be rerun and stopped. The job will print out where is it. We can set up a second job, started when the hdf5 production is done that can either make data quality plots or produce the binned data.

While the processing is generally done from the elog (ARP), if you would like to run an interactive test job with only a few events, you can use:
`./arp_scripts/submit_smd.sh -r <#> -e <experiment_name> --nevents <#> --interactive`

The full list of options is here:

```
(ana-4.0.30-py3) -bash-4.2$ ./arp_scripts/submit_smd.sh -h
submit_smd.sh:
        Script to launch a smalldata_tools run analysis

        OPTIONS:
                -h|--help
                        Definition of options
                -e|--experiment
                        Experiment name (i.e. cxilr6716)
                -r|--run
                        Run Number
                -d|--directory
                        Full path to directory for output file
                -n|--nevents
                        Number of events to analyze
                -q|--queue
                        Queue to use on SLURM
                -c|--cores
                        Number of cores to be utilized
                -f|--full
                        If specified, translate everything
                -D|--default
                        If specified, translate only smalldata
                -i|--image
                        If specified, translate everything & save area detectors as images
                --norecorder
                        If specified, don't use recorder data
                --nparallel
                        Number of processes per node
                --postTrigger
                        Post that primary processing done to elog to seconndary jobs can start
                --interactive
                        Run the process live w/o batch system
```

Debug the h5 production / job logs

General comments

It is generally a good idea to test outside the producer that the functions that return the arguments for the area detector work as intended. Copy it to a Jupyter notebook and give it a try with a few run numbers, making sure it returns the keyword arguments you expect.

Log files

While the logs are full of pretty cryptic text, there are a few key messages one can look at.

Access the logs

The job logs can be accessed from the workflow/controls tab, on the same line where the job can be launched. Click on the four bars in the **Actions** column and a (often rather long) text file will open.

Did my job run properly?

If a job runs as expected, the last part of the log should look like (minus some garbage encoding text):

```
########## JOB TIME: 5.580481 minutes ###########
posting to the run tables.
URL: https://pswww.slac.stanford.edu/ws-auth/lgbk//run_control/xpplw8419/ws/add_run_params
Closing remaining open files:/cds/data/drpsrcf/XPP/xpplw8419/scratch/hdf5/smalldata/xpplw8419_Run0123.h5...
done
```

If the job did not run properly, here is how to spot common errors in the logs:

Syntax errors

These are the easiest to spot, as the script won't even start, and the log will only contain that error and point you at the problematic line.

Example:

```
File "/cds/data/psdm/xpp/xppx47019/results/smalldata_tools/producers/smd_producer.py", line 53
    elif run>19 and run<=65
                        ^
```

Wrong or bad argument for the area detector functions

When the argument for an area detector function is wrongly defined, the code will fail at the detector instantiation. This can, for example, be, a typo in the argument name or a wrong shape / type.