EUROPEAN
SPALLATION
SOURCE

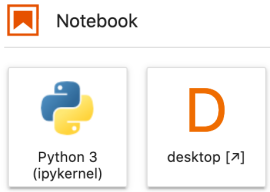
McStas and how-to generate event data for Mantid and scipp

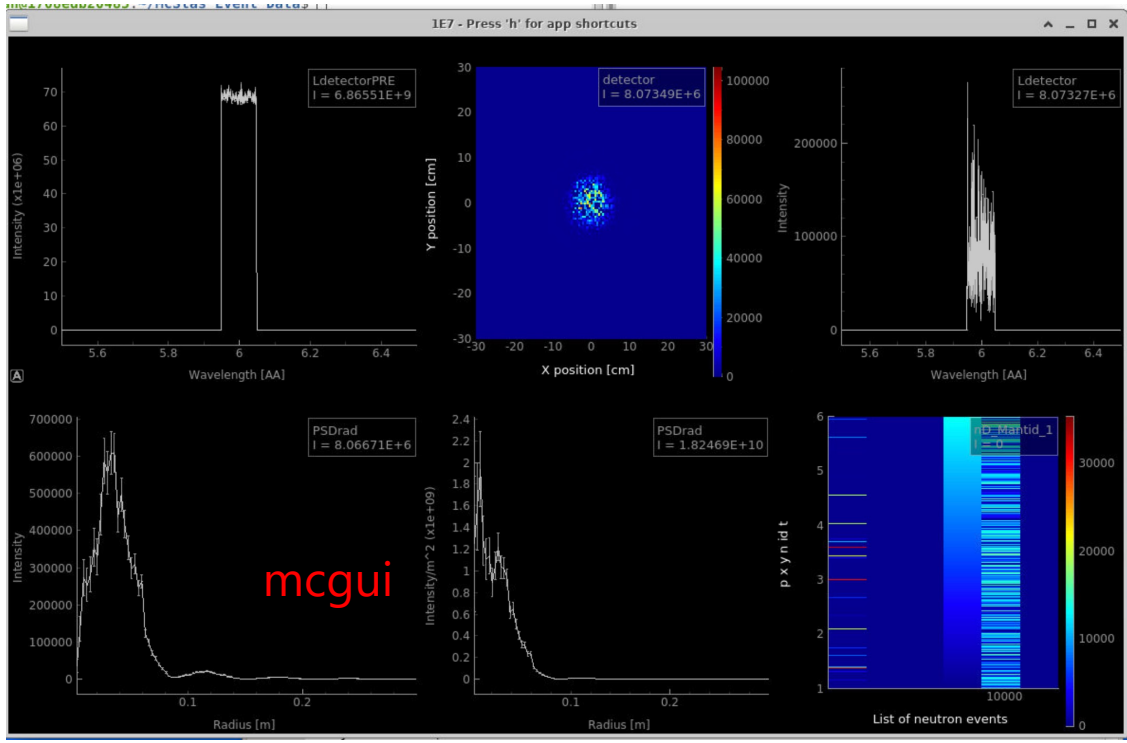
TORBEN

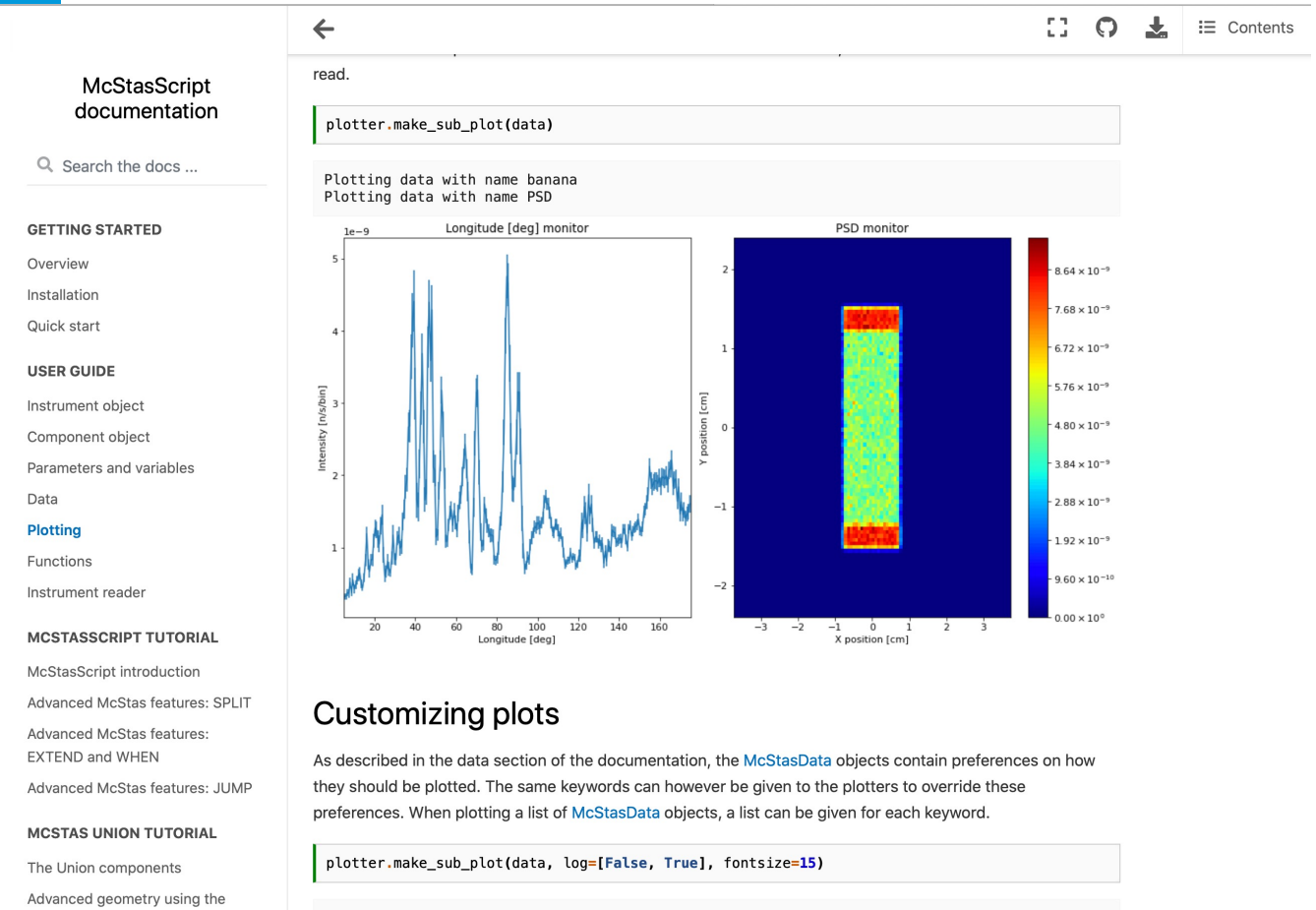
Spring 2022 – McStas workshop @ DMSC

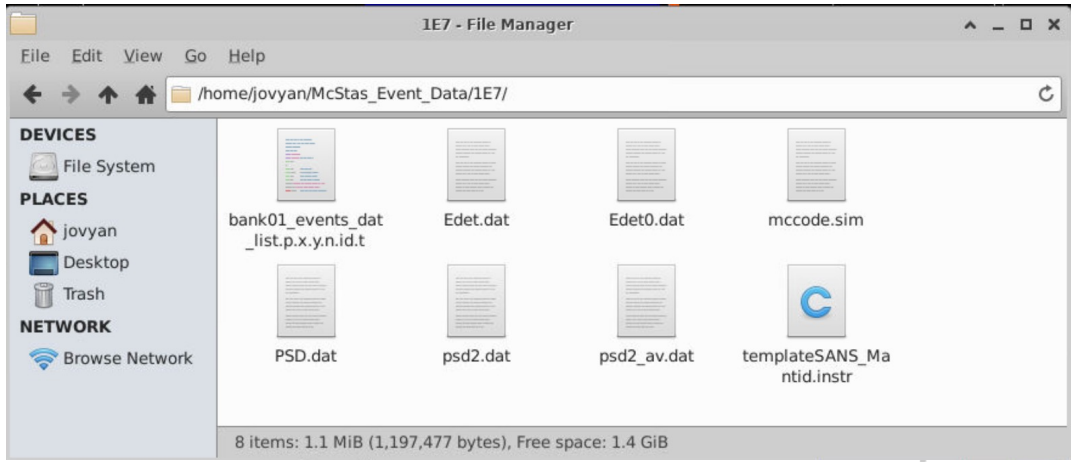
McStas and McStasScript

Histogram data / binned data / ascii data



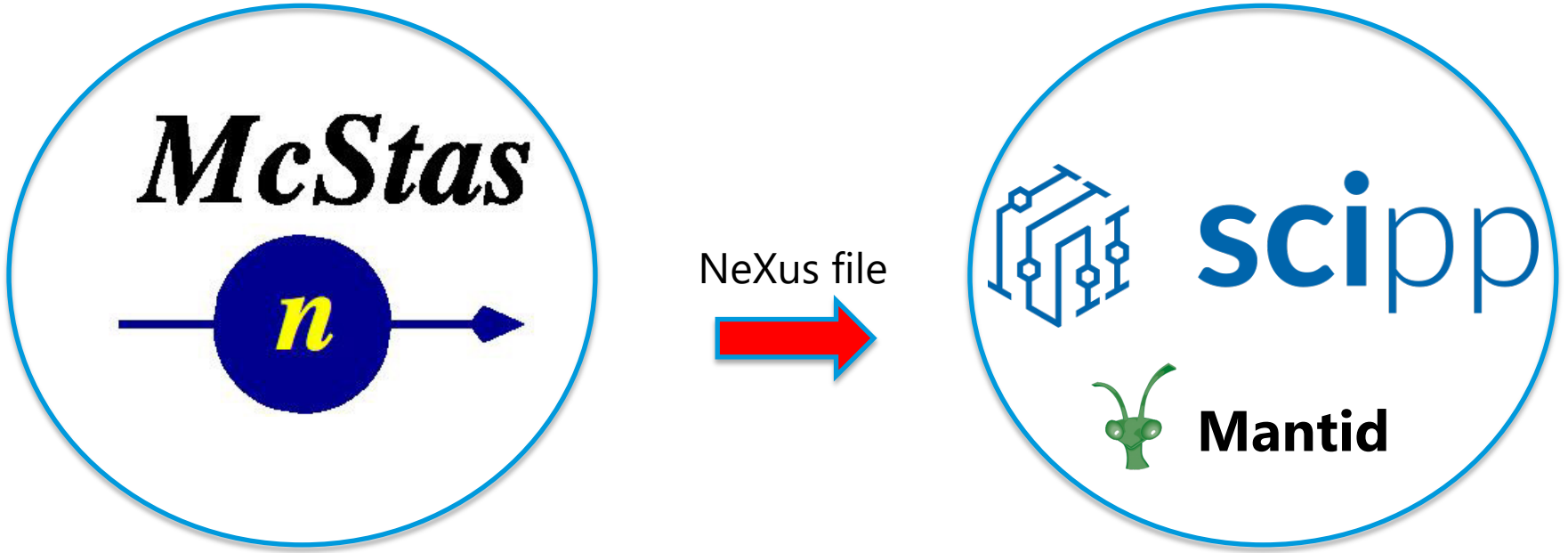






McStas event data and The NeXus file

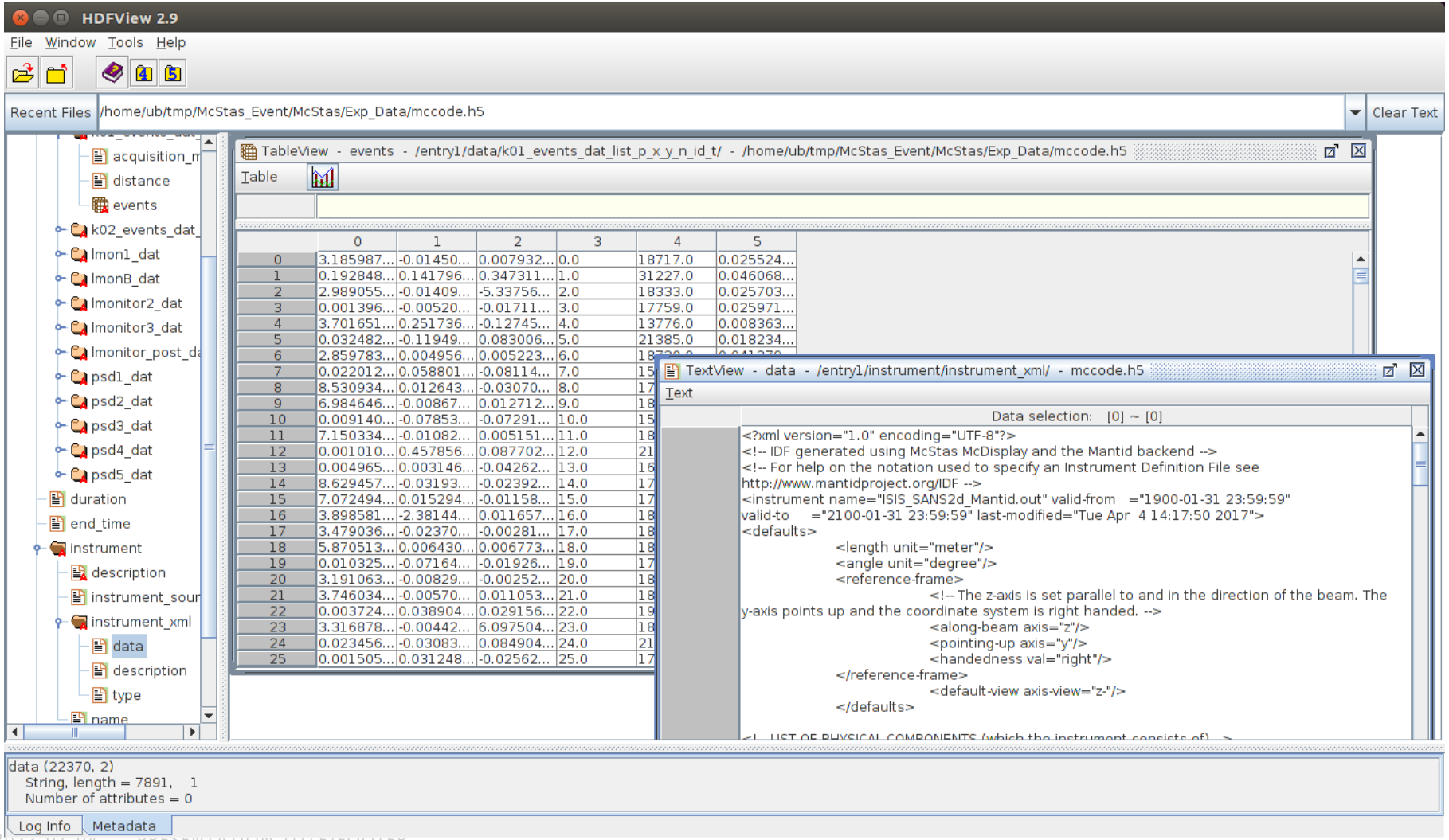
Process McStas event data as you would do with recorded data from a beamline



- https://www.nexusformat.org
- The McStas Nexus file must contain:
- Event data, i.e. each neutron has a pixel id and a time stamp
- An IDF McStas monitor_nD gives pixle id & time for each event
- *mcdisplay can auto-generate an IDF*

McStas NeXus file

Mantid/scipp compatible

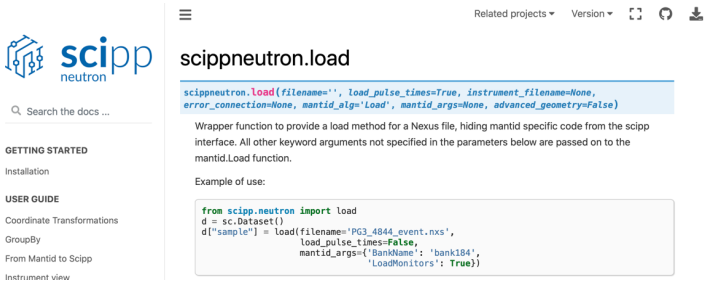


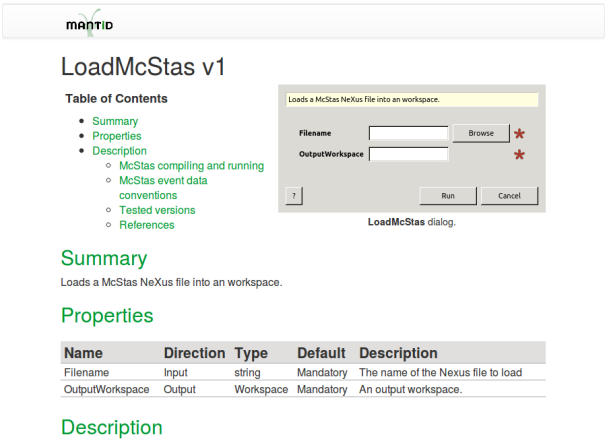
How to use: Online documentation

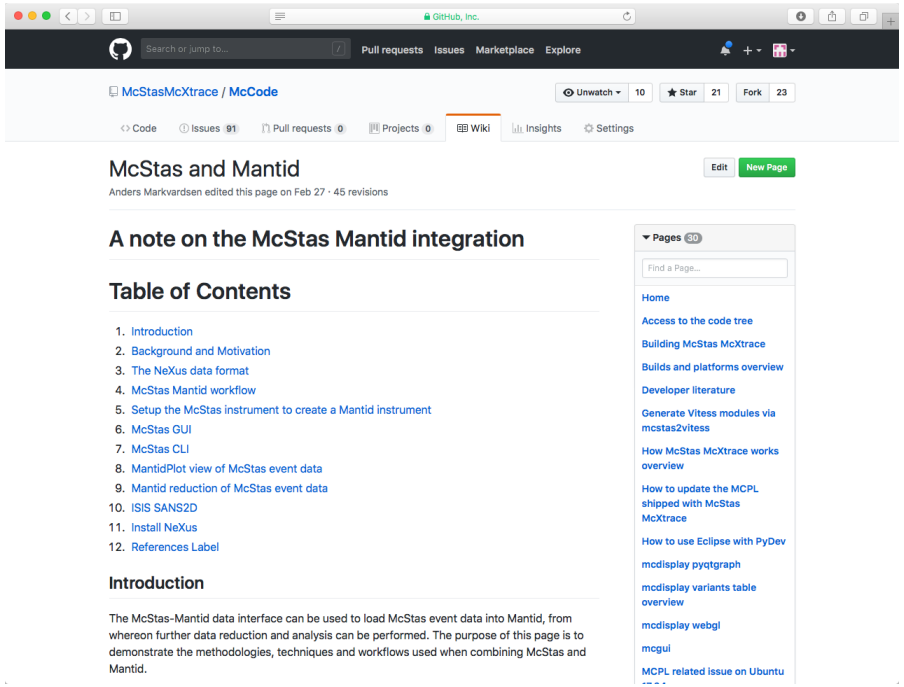
Github McStas wiki pages

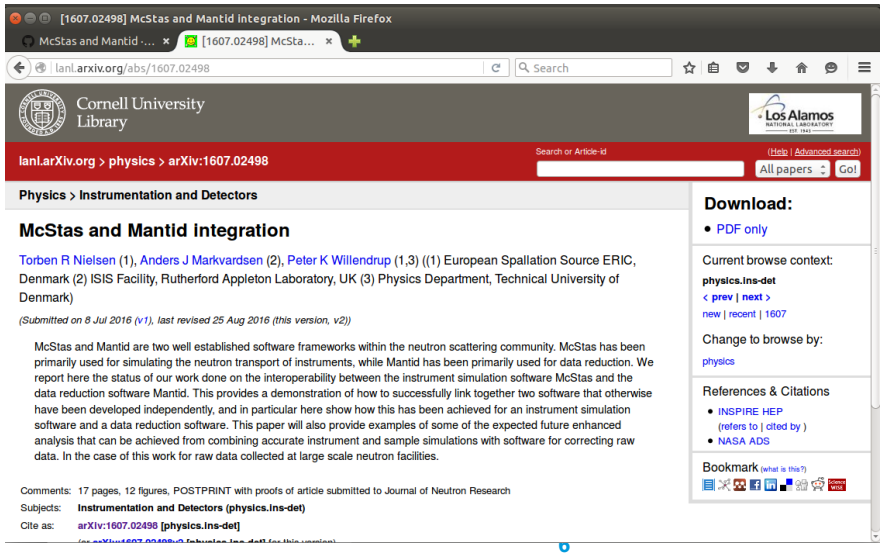
Archive - lanl.arXiv.org

Docs in Mantid/scipp









IDF xml data, TOF and pixel ID's

Mantid's IDF stores geometry information used in TOF analysis

This implies parsing information about:

1. where the neutron source is located,
2. where the sample is located,
3. where each individual detector pixel is located.

McStas event KEYWORDS

McStas instrument file name and the McStas defined name of the instrument must be the same:

E.g. **templateSANS_Mantid.instr** and "**DEFINE INSTRUMENT templateSANS_Mantid**(....)"

In the McStas instrument file the source must be named "**sourceMantid**"

E.g. "**COMPONENT sourceMantid = Source_simple**(....)"

In the McStas instrument file the sample must be named "**sampleMantid**"

E.g. "**COMPONENT sampleMantid = Sans_spheres**(....)"

In the McStas instrument file the event monitors must be named "**nD_Mantid_#**"

E.g. "**COMPONENT nD_Mantid_1 = Monitor_nD**(...)"

templateSANS.inst vs templateSANS_Mantid.inst

templateSANS.instr — UNREGISTERED

templateSANS.instr | templateSANS_Mantid.instr

```
1  DEFINE INSTRUMENT templateSANS(lambda=6, dlambda=0.05, r=100, PHI=1e-
   3, Delta_Rho=0.6, sigma_abs=0.5)
2
3  TRACE
4
5  COMPONENT a1 = Progress_bar()
6    AT (0,0,0) ABSOLUTE
7
8  COMPONENT arm = Arm()
9    AT (0, 0, 0) ABSOLUTE
10
11 COMPONENT source = Source_simple(
12       radius = 0.02, dist = 3, focus_xw = 0.01, focus_yh = 0.01,
13       lambda0 = lambda, dlambda = dlambda, flux = 1e16)
14   AT (0, 0, 0) RELATIVE arm
15
16 COMPONENT coll1 = Slit(
17       radius = 0.005)
18   AT (0, 0, 3) RELATIVE arm
19
20 COMPONENT coll2 = Slit(
21       radius = 0.005)
22   AT (0, 0, 6) RELATIVE arm
23
24
25
26
27
28
29
30 SPLIT COMPONENT sample = Sans_spheres(
31   R=r, Phi=PHI, Delta_rho=Delta_Rho, sigma_abs=sigma_abs,
32   xwidth=0.01, yheight=0.01, zdepth=0.005, focus_xw=0.6, focus_yh=0.
     6, target_index=2)
33    AT (0,0,0.2) RELATIVE coll2
34
35 COMPONENT STOP = Beamstop(
36       radius = 0.02)
37   AT (0, 0, 2.9) RELATIVE sample
```

Line 1, Column 1 — Spaces: 2 — Plain Text

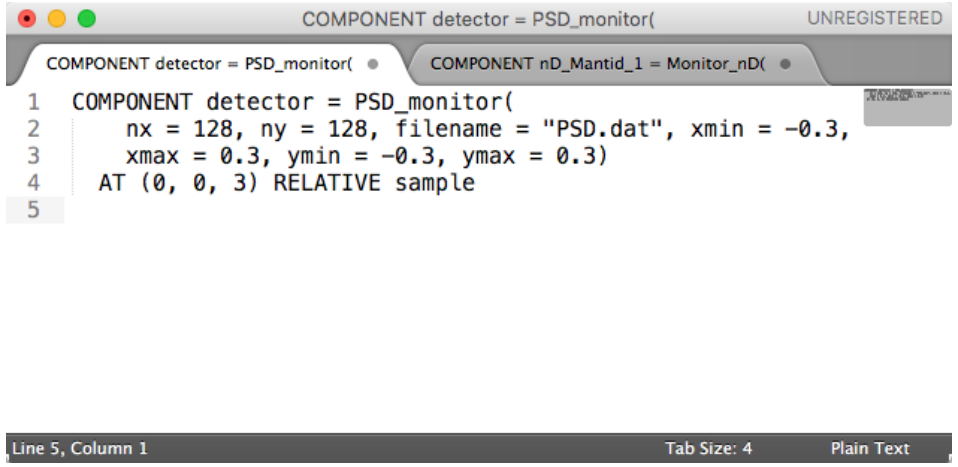
templateSANS_Mantid.instr — UNREGISTERED

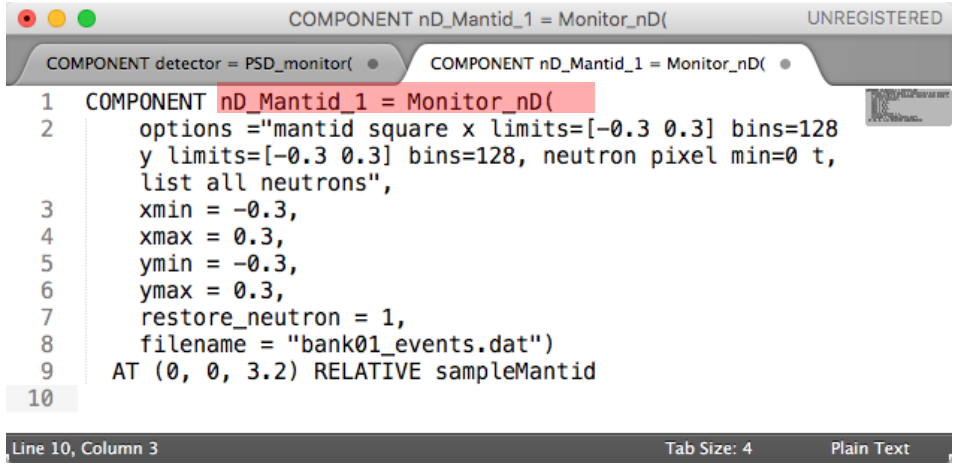
templateSANS.instr | templateSANS_Mantid.instr

```
1  DEFINE INSTRUMENT templateSANS_Mantid(lambda=6, dlambda=0.05, r=150,
   PHI=1e-3, Delta_Rho=0.6, sigma_abs=0.0)
2
3  TRACE
4
5  COMPONENT a1 = Progress_bar()
6    AT (0,0,0) ABSOLUTE
7
8  COMPONENT arm = Arm()
9    AT (0, 0, 0) ABSOLUTE
10
11 COMPONENT sourceMantid = Source_simple(
12       radius = 0.02, dist = 3, focus_xw = 0.01, focus_yh = 0.01,
13       lambda0 = lambda, dlambda = dlambda, flux = 1e16)
14   AT (0, 0, 0) RELATIVE arm
15
16 COMPONENT coll1 = Slit(
17       radius = 0.005)
18   AT (0, 0, 3) RELATIVE arm
19
20 COMPONENT coll2 = Slit(
21       radius = 0.005)
22   AT (0, 0, 6) RELATIVE arm
23
24 COMPONENT LdetectorPRE = L_monitor(
25       nL = 1000, filename = "Edet0.dat", xmin = -0.3,
26       xmax = 0.3, ymin = -0.3, ymax = 0.3, Lmin = 5.5,
27       Lmax = 6.5)
28 AT (0,0,0.05) RELATIVE coll2
29
30 SPLIT COMPONENT sampleMantid = Sans_spheres(
31   R=r, Phi=PHI, Delta_rho=Delta_Rho, sigma_abs=sigma_abs,
32   xwidth=0.01, yheight=0.01, zdepth=0.005)
33    AT (0,0,0.2) RELATIVE coll2
34 EXTEND %{
35         if (!SCATTERED) ABSORB;
36 %}
```

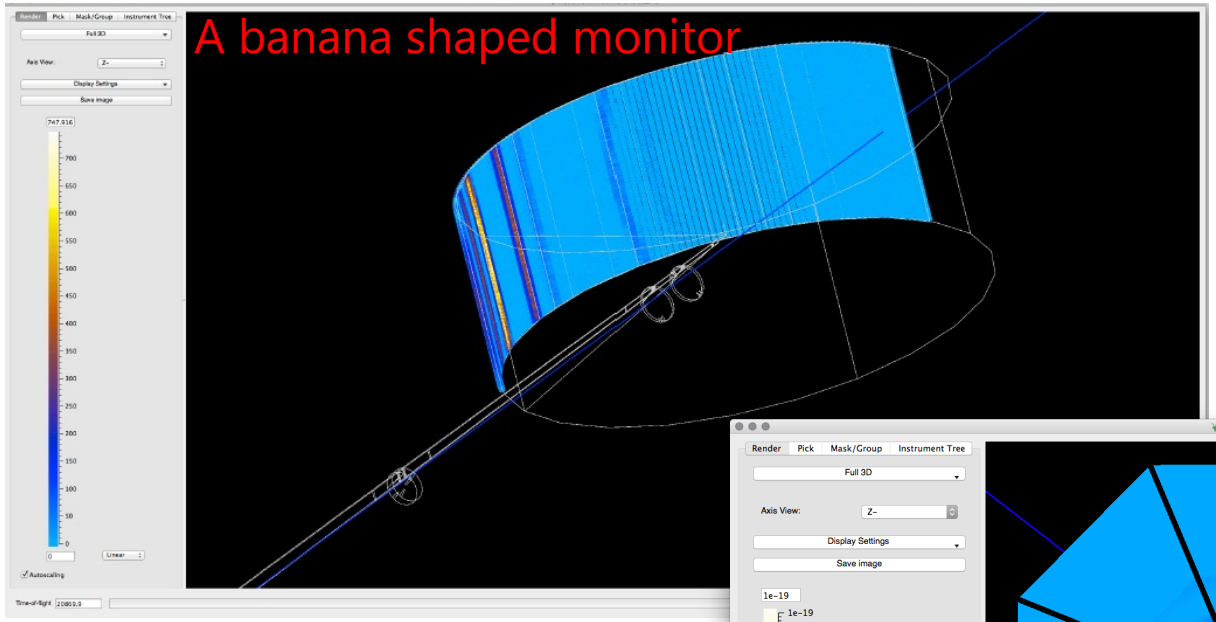
Line 1, Column 1 — Spaces: 2 — Plain Text

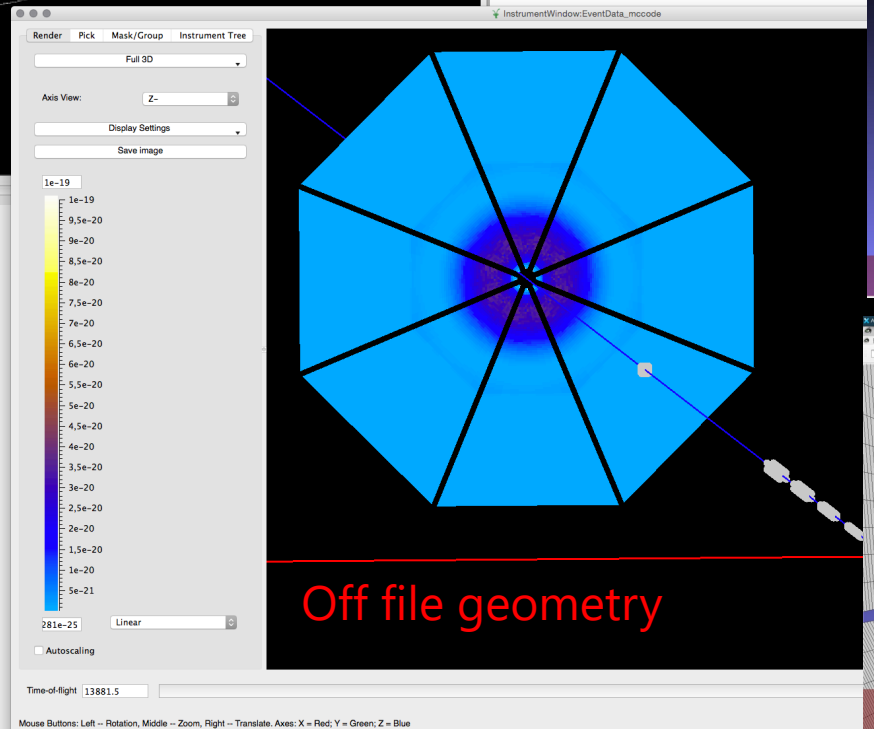
templateSANS.inst vs templateSANS_Mantid.inst

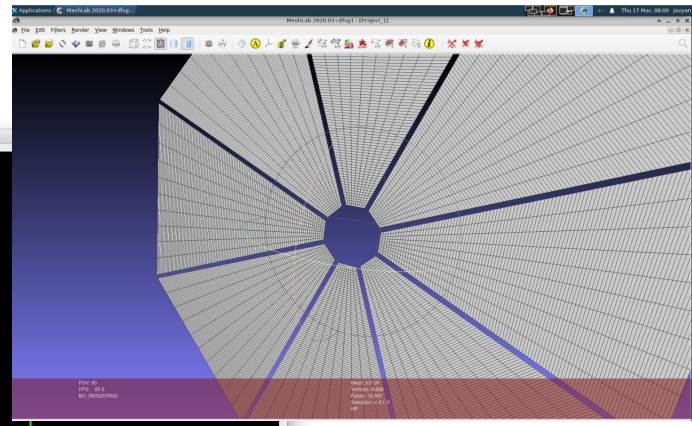


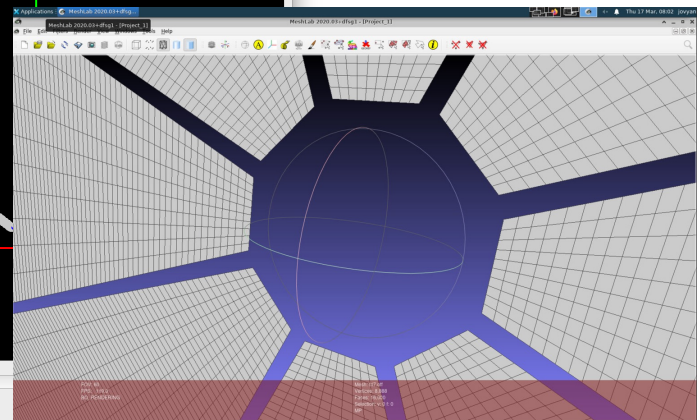


Examples of detectors: IN5 (ILL) and LoKI (ESS)

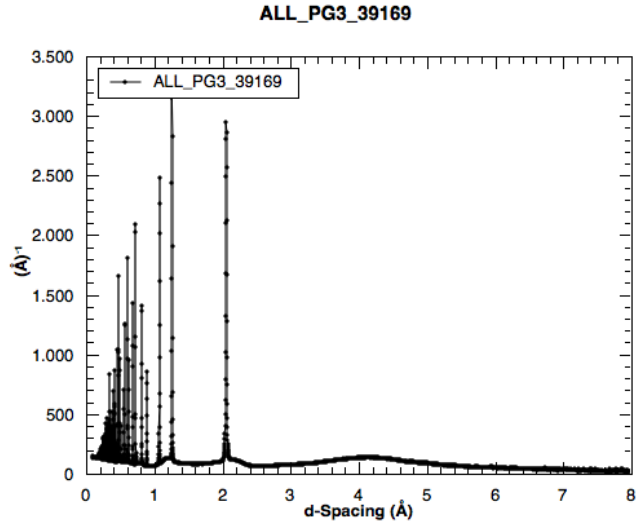


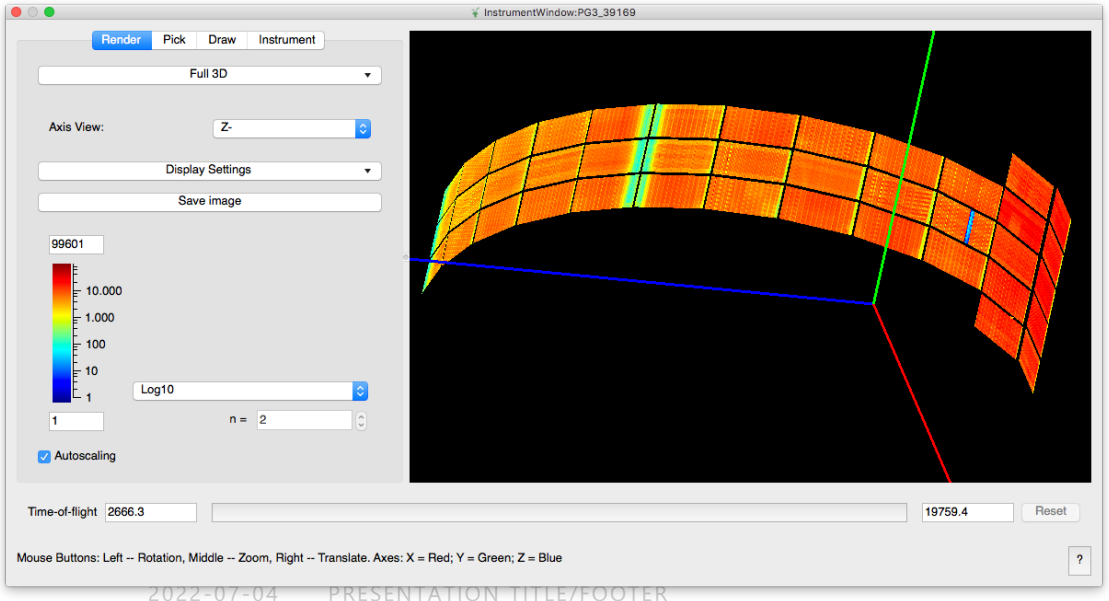


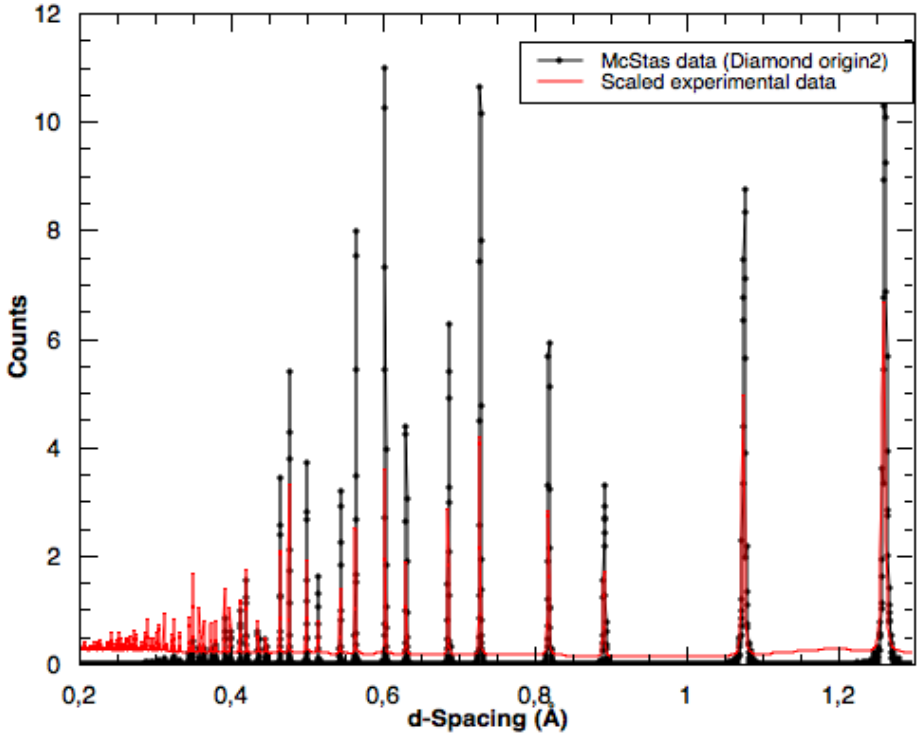




McStas and experimental data: POWGEN







By CelineD @ ESS + SNS collaboration

How to use: How to run simulation

Github wiki pages

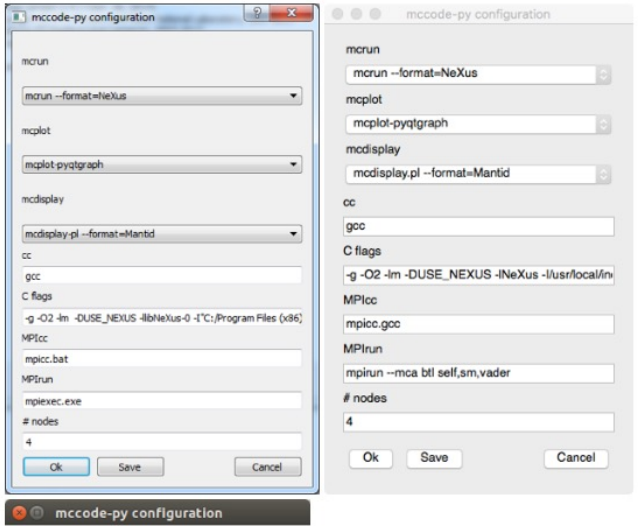
GitHub, Inc.

McStas GUI

Generating McStas event data for Mantid can be achieved from the McStas GUI `mcgui`. Below we show how to setup the simulation on Windows 7, OSX 10.12, and Ubuntu 16.04. For McStas we use version 2.4.1. For Mantid use version 3.4 or later.

Windows users need to download the latest version of mccode-r.c in order to generate IDF data in the mccode.h5 file which Mantid can read. Copy the downloaded file to the directory where your McStas instrument file is located.

1. Open the McStas configuration file. In `mcgui` go to: File -> Configuration
2. Change the setting as shown in figures below:
   - In the section `mcrun` select `mcrun --format=NeXus`
   - In the section `mcplot` select `mcplot-pyqtgraph`
   - In the section `mcdisplay` select `mcdisplay-pl --format=Mantid` (Windows) or `mcdisplay.pl --format=Mantid` (OSX or Ubuntu)
   - In the section `C flags` select this line depending on your OS
     - Windows: `-g -O2 -lm -DUSE_NEXUS -llibNeXus-0 -I "C:/Program Files (x86)/NeXus Data Format/include/nexus" -L "C:/Program Files (x86)/NeXus Data Format/lib/nexus"`
     - OS X: `-g -O2 -lm -DUSE_NEXUS -lNeXus -I/usr/local/include/nexus`
     - Ubuntu: `-g -O2 -lm -DUSE_NEXUS -lNeXus`



McStas CLI

Generating McStas event data for Mantid can also be controlled from the terminal, using this command line interface (CLI). *Below we show how to setup the simulation for OSX 10.12 and Ubuntu 14.04. The procedure for Windows is similar.* For McStas we use version 2.4.1.

Mac OS X and Ubuntu

1. Compile McStas instrument file

```
$ mcstas templateSANS_Mantid.instr --trace
```

2. Generate the IDF for Mantid

```
$ mcdisplay.pl templateSANS_Mantid.instr --format=Mantid -n0
```

3. Generate the executable for Mac OS X

```
$ gcc -o templateSANS_Mantid.out templateSANS_Mantid.c -lm -DUSE_NEXUS -lNeXus -I/usr/local/include/nexus/
```

And for Ubuntu

```
$ gcc -o templateSANS_Mantid.out templateSANS_Mantid.c -lm -DUSE_NEXUS -lNeXus
```

4. Run the executable and use the default parameters

```
$ ./templateSANS_Mantid.out --format=NeXus -d 1E8
```

How to use CLI: Generate the IDF

- How-to generate McStas event data for Mantid/scipp

Step 1. Generate c file:

```
mcstas templateSANS_Mantid.instr --trace
```

Step 2. Generate IDF:

```
mcdisplay-mantid.pl templateSANS_Mantid.instr -n0
```

A new xml IDF file is then generated on disk

How to use CLI: Make NeXus file for Mantid/scipp

- How-to generate McStas event data for Mantid/scipp

Step 1. Compile c code:

```
gcc -o templateSANS_Mantid.out templateSANS_Mantid.c   -DUSE_NEXUS –lNeXus –lm  -I/usr/include/nexus
```

Step 2. Run simulation:

```
templateSANS_Mantid.out --format=Nexus
```

Now IDF is embedded in the NeXus file to be read by Mantid/scipp

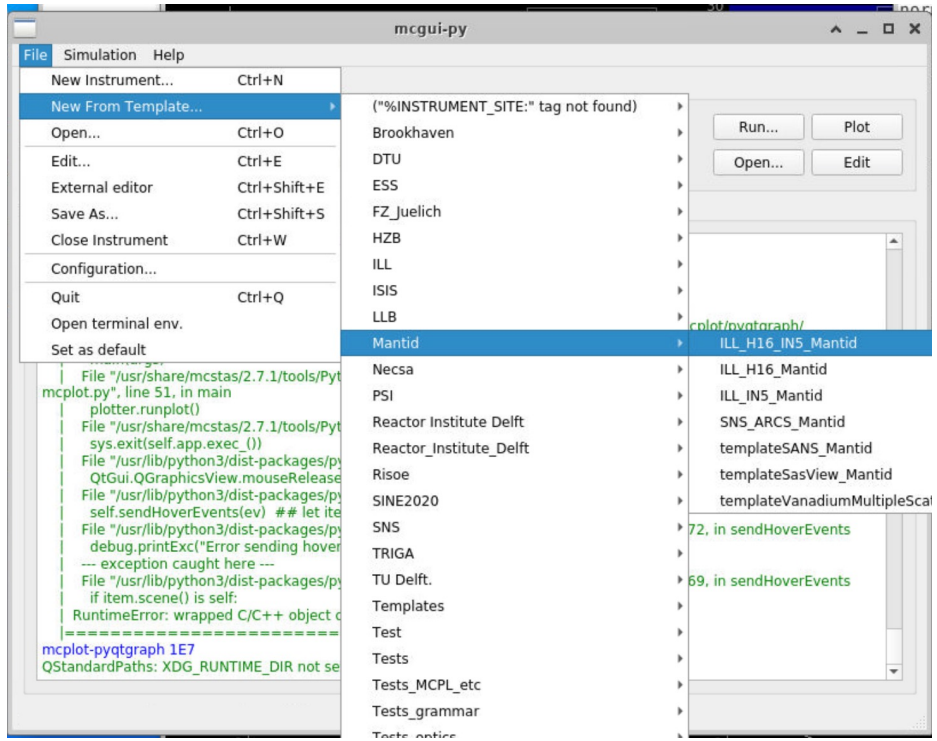
Demo

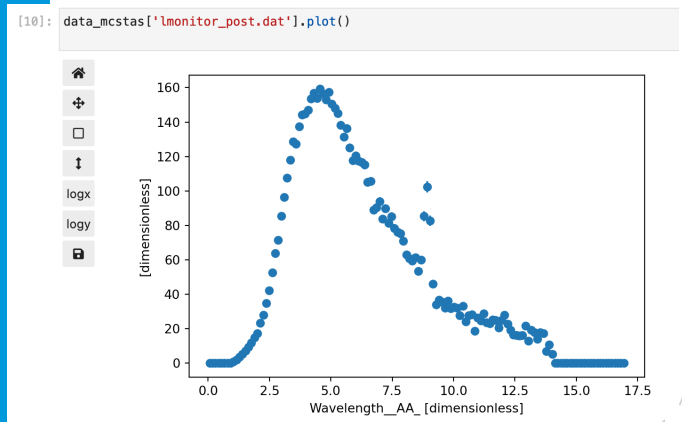
Let’s try to do a test . . . .

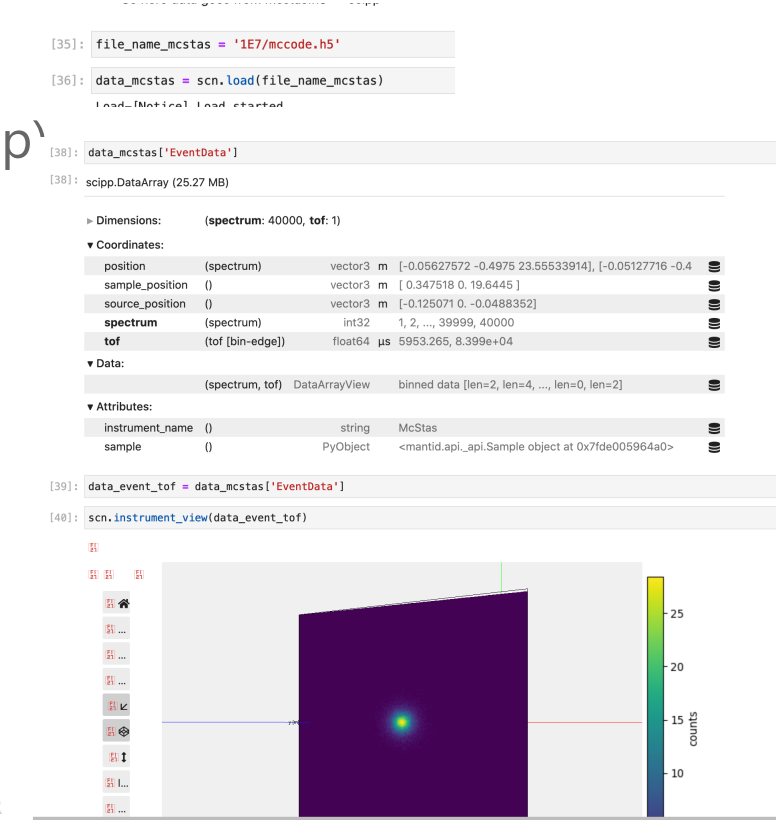
Exercise

Compile a Mantid/scipp instrument and generate event data

* choose your instrument
* compile and run (use 2.7.1)
* load into scipp
* view instrument
* plot monitors (in scipp)







Finish presentation