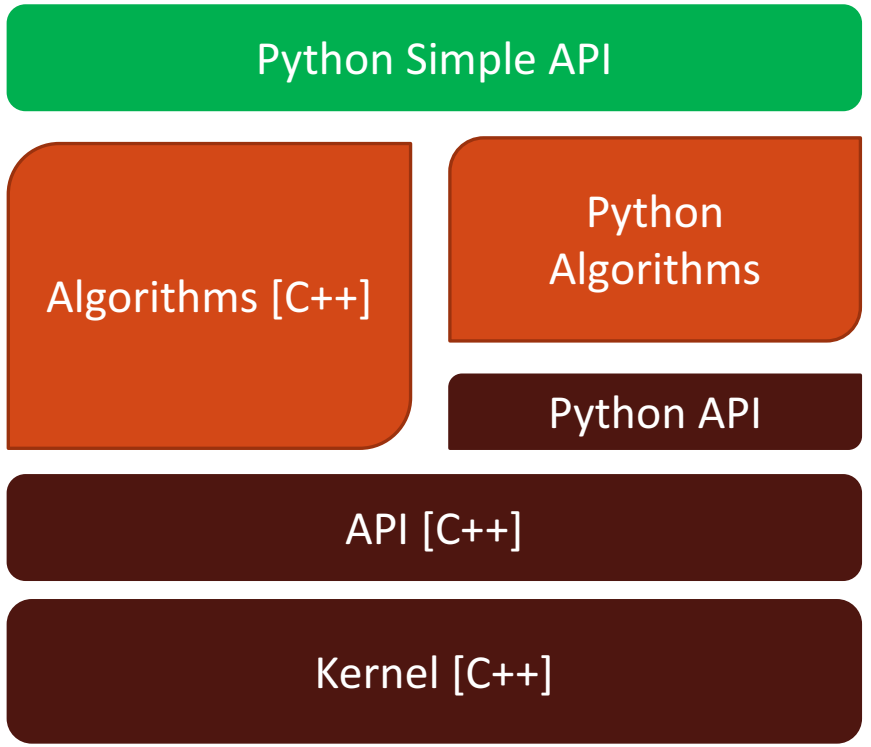
Introduction to Mantid

31.03.17, ILL

Mantid framework

- High performance data treatment and visualisation solution tailored for neutron scattering experiments
- Core implemented in C++, API exposed to python
- Easily extendible due to plug-in architecture
  - Shipped with O(800) existing algorithms
  - New algorithms can be created and plugged in
  - Python algorithms plugged in at run time
- Open source, cross-platform (Linux, OS X, Windows)
- Main tool at ISIS, SNS, ESS and soon at ILL
- Data manipulation is performed using key concepts as **workspaces** and **algorithms**



Workspaces

- Are the main data containers, analogue concept exists in LAMP
- Several types of workspaces exist; Matrix, Events, Peaks, Table, MD etc.
- Most common is the MatrixWorkspace which is a list of 1D histograms

Detector data

single_0.0_red - Mantid

Y values | X values | Errors

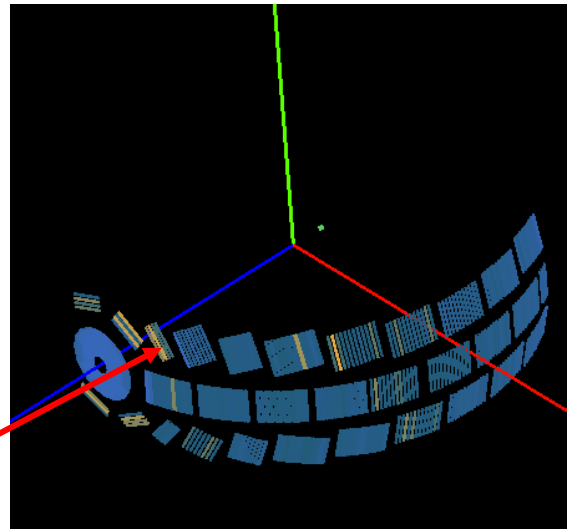
	0 266.8K	1 258.654K	2 249.923K
0 sp-0	0,144177	0,151535	0,161796
1 sp-1	0,178321	0,187072	0,198562
2 sp-2	0,190916	0,200196	0,213337
3 sp-3	0,159553	0,16689	0,179411
4 sp-4	0,17331	0,180447	0,195429
5 sp-5	0,157991	0,16699	0,179352
6 sp-6	0,145446	0,152051	0,166156
7 sp-7	0,109014	0,114242	0,125274
8 sp-8	0,145847	0,154115	0,16687
9 sp-9	0,221201	0,236249	0,256957
10 sp-10	0,200287	0,213504	0,232027

Map between spectrum number to detector pixel

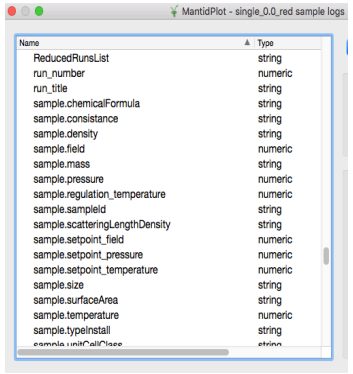
084446-Detectors-1

	Index	Spectrum No	Detector ID(s)	R	Theta	Q
1	0	1	1	2,000000000…	15,92499999…	0,580251921…
2	1	2	2	2,000000000…	14,97499999…	0,545840788…
3	2	3	3	2,000000000…	14,02499999…	0,511392140…
4	3	4	4	2,000000000…	13,07499999…	0,476908344…
5	4	5	5	2,000000000…	13,07499999…	0,476908344…
6	5	6	6	2,000000000…	14,02499999…	0,511392140…
7	6	7	7	2,000000000…	14,97499999…	0,545840788…
8	7	8	8	2,000000000…	15,92499999…	0,580251921…
9	8	9	9	2,000000000…	19,07499999…	0,694053604…
10	9	10	10	2,000000000…	20,02499999…	0,728275737…
11	10	11	11	2,000000000…	20,97499999…	0,762447816…
12	11	12	12	2,000000000…	21,92499999…	0,796567493…
13	12	13	13	2,000000000…	24,97499999…	0,905727922…
14	13	14	14	2,000000000…	25,92499999…	0,939601280…
15	14	15	15	2,000000000…	26,87499999…	0,973410060…

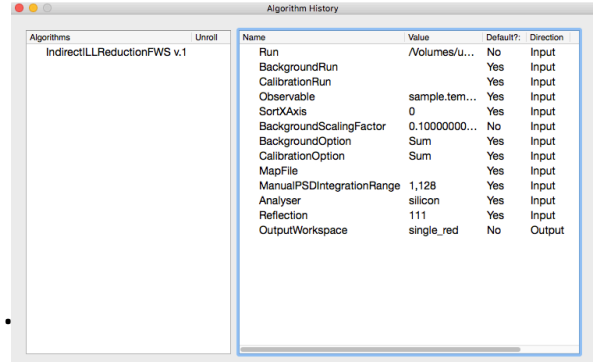
Full 3D instrument geometry



Metadata



History of algorithm execution

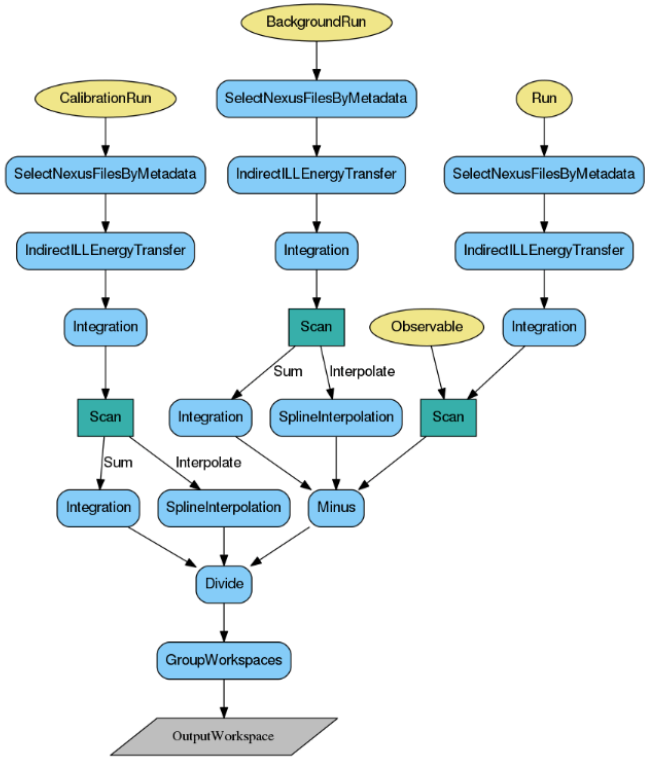


In a typical raw TOF data, each row represents a time-of-flight spectrum of a single detector pixel.

Algorithms

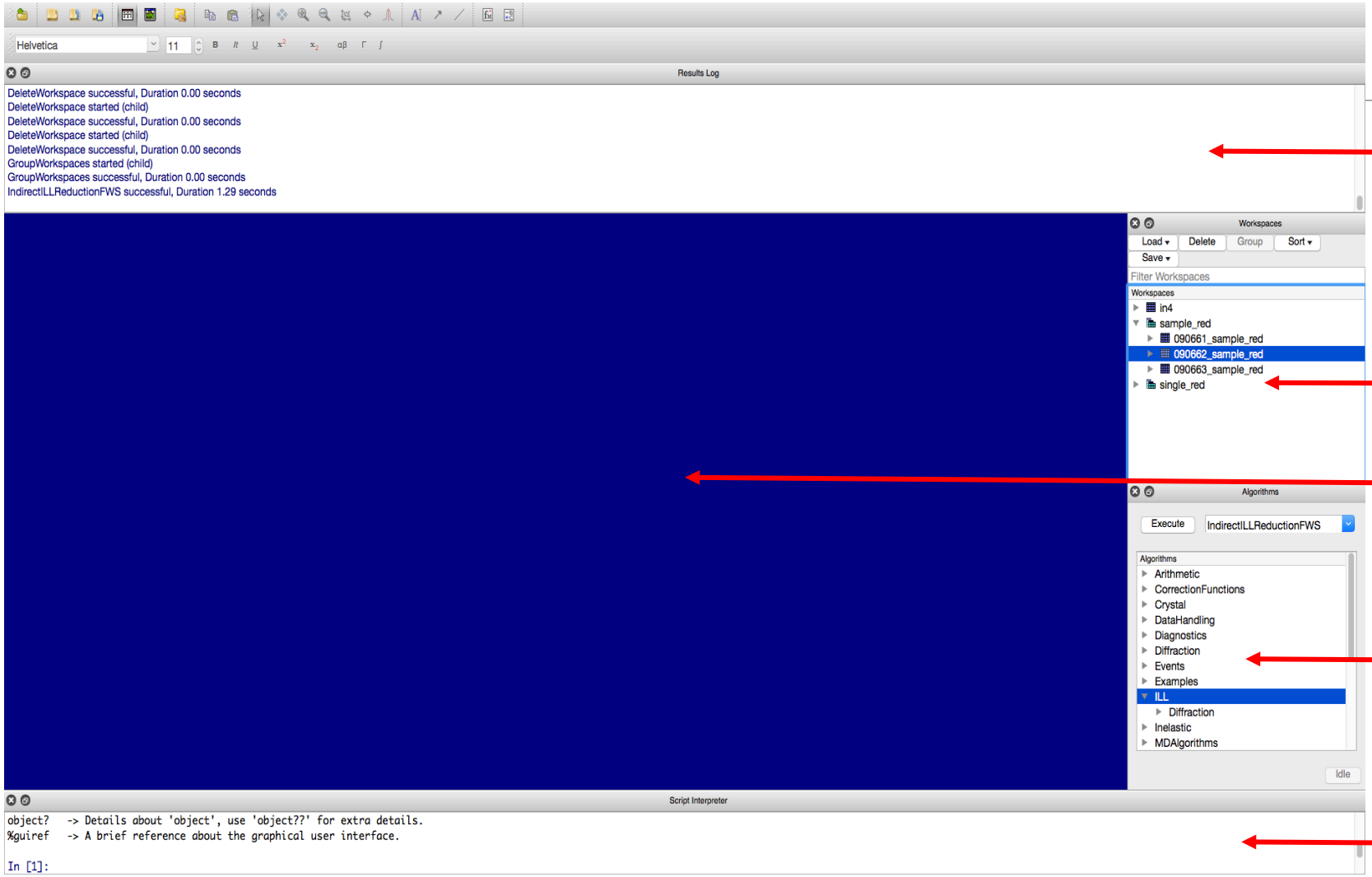
- Manipulate the data in workspaces or files
- Examples can be arithmetic operations, e.g. **Integrate**, **Sum**, **Transpose**, etc.
- New algorithms can be created and added both in C++ and python
- Reduction algorithms are written in python and represent workflow-type algorithms, which call many other algorithms in a sequence
- For each algorithm, a simple GUI dialog is automatically generated on-the-fly
- Python simple API provides function-like interface for invokation of the algorithms, no matter they are written in C++ or python

```
> result = SomeAlgorithm(InputWorkspace = inWS, Param="value"
```



A workflow algorithm flowchart.

MantidPlot GUI



Log shell

Workspace browser

Main working area

Algorithm browser

IPython shell

MantidPlot GUI

Custom interface window

Auto-generated interface dialog

Workspace matrix view

Python script editor

Plotting window

Instrument view