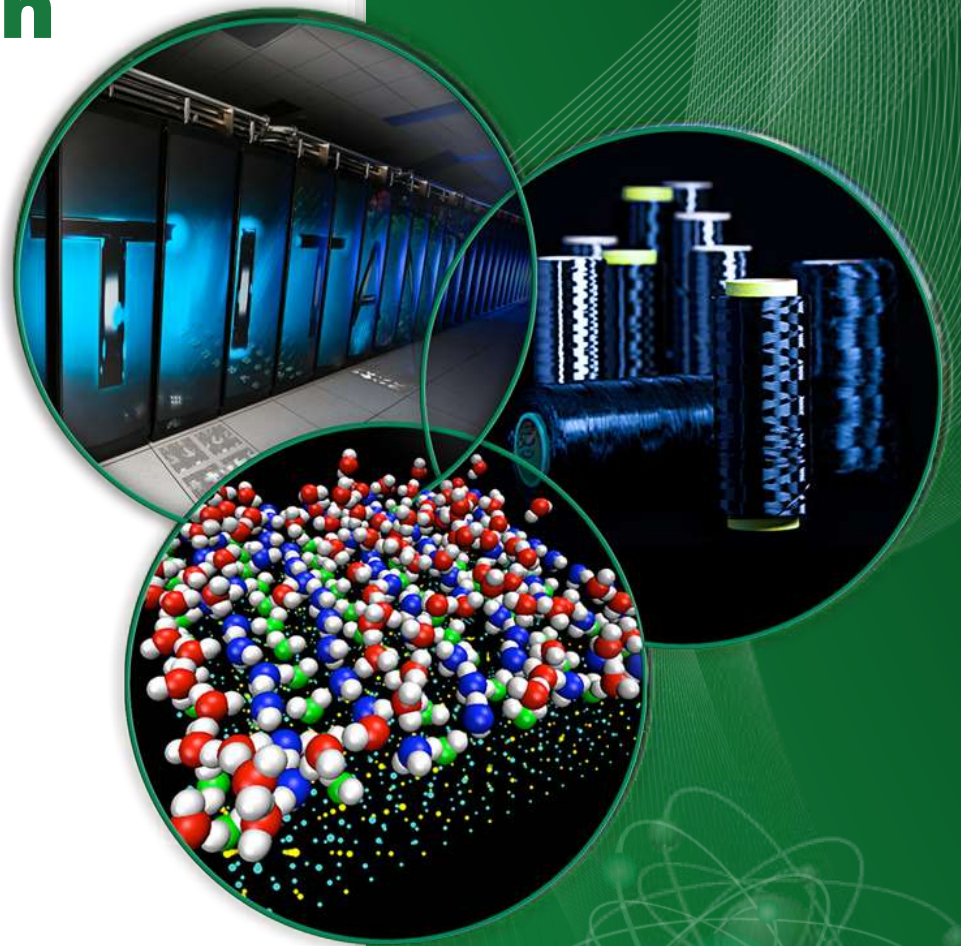


# Mantid: Now and in the future

**Garrett E. Granroth**  
Scientific Data Analysis Group  
Lead

**Sept 26, 2014**



# Mantid Team



- [www.mantidproject.org](http://www.mantidproject.org)

**NIMA 764, 156 (2014)**

# Mantid Outline

- Members, Contributors
- Deployment
- Documentation system
- Functionality Highlights and Plans
  - Python API
  - Live
  - Event processing – usage case
- Conclusions

# What is Mantid?

- A Framework for Reduction and Analysis of Neutron and Muon Data
  - Can Be accessed by
    - MantidPlot
    - Python Interface
    - C++ API
  - Has a set of data objects, methods and Algorithms well suited towards scattering science



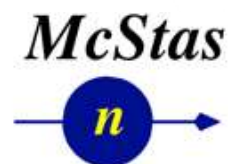
# Partners and Contributors

- Partners



joining  
soon

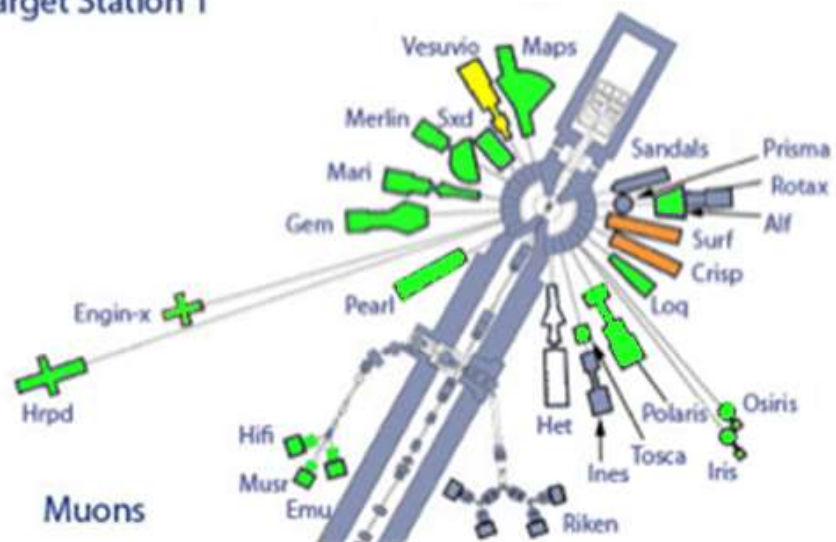
- Contributors



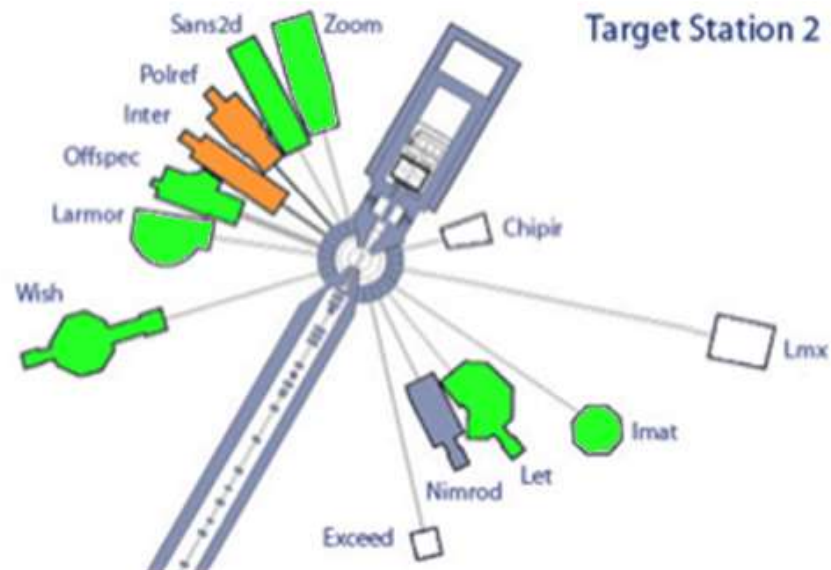
# Deployment at ISIS

## ISIS

Target Station 1



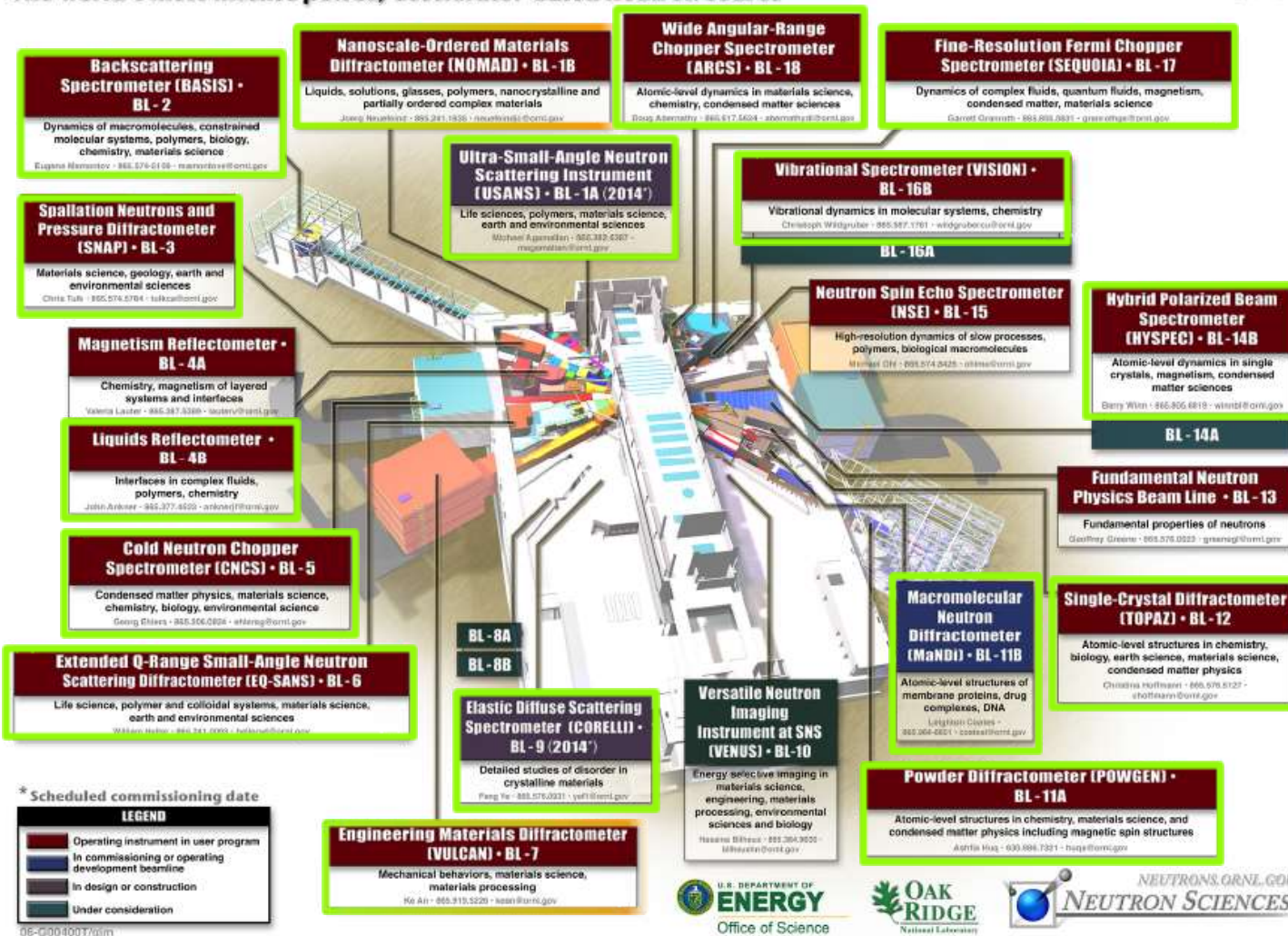
Target Station 2



# Deployment at SNS

## Spallation Neutron Source at Oak Ridge National Laboratory

The world's most intense pulsed, accelerator-based neutron source

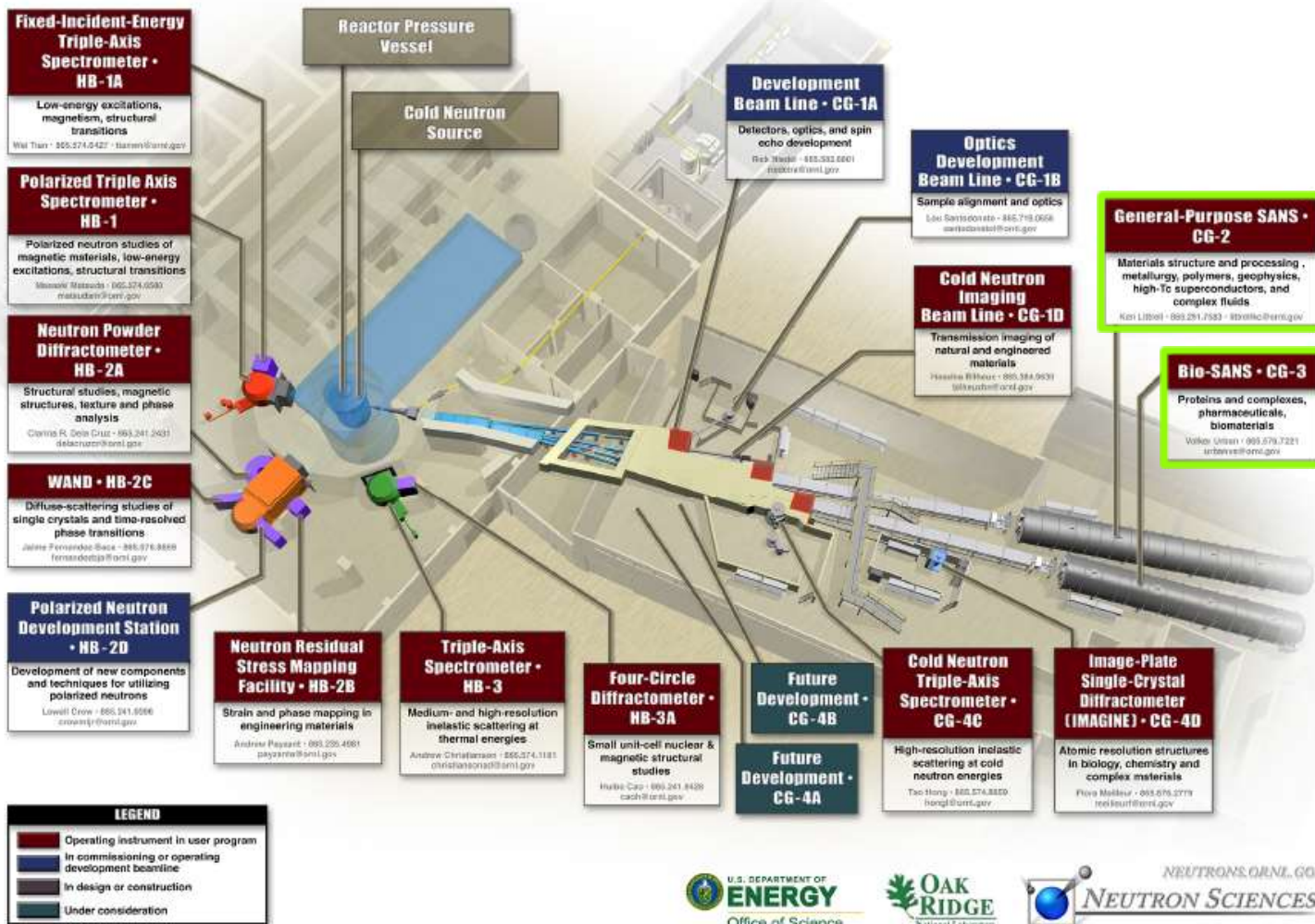




# Deployment at HFIR

## High Flux Isotope Reactor at Oak Ridge National Laboratory

The United States' highest flux reactor-based neutron source





# Other instruments



NEUTRONS  
FOR SCIENCE®

- NMI3 supported evaluation
- IN4,5 & 6
- D33



- Focus
- Poldi



- Mibemol



FRM II  
Forschungs-Neutronenquelle  
Heinz Maier-Leibnitz

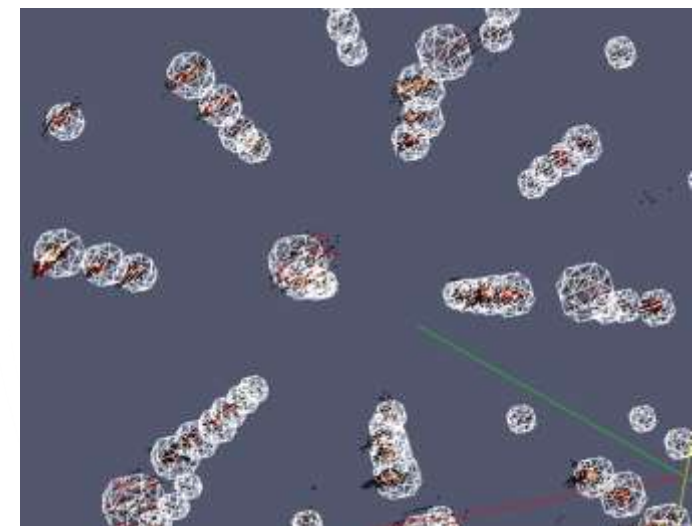
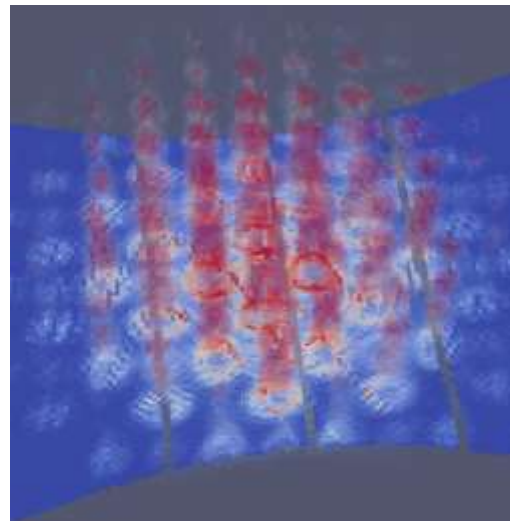
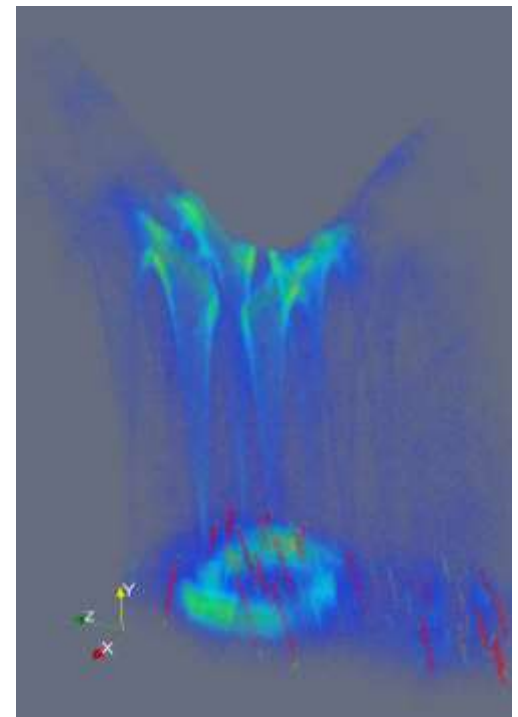
- TofTof



- Pelican
- Bilby

# Advanced Visualization

- Vates /integrated Paraview
- Future
  - Moving to Paraview 4.2
  - Production interface in Mantid workflow
  - Improved GUI



# Improved Sphinx based documentation



```
.. algorithms::
.. summary::
.. alias::
.. properties::
-----
Description
```

The algorithms will attach an `OrientedLattice` object to a sample in the workspace. For MD workspaces, you can select to which sample to attach it. If nothing entered, it will attach to all. If bad number is entered, it will attach to first sample.

If UB matrix elements are entered, lattice parameters and orientation vectors are ignored. The algorithms will throw an exception if the determinant is 0. If the UB matrix is all zeros (default), it will calculate it from lattice parameters and orientation vectors (Mullice and Horace style). The algorithms will throw an exception if  $u$  and  $v$  are collinear, or one of them is very small in magnitude.

#### Usage

```
.. testcode:: SetUB
# create a workspace (or you can load one)
ws=CreateSingleValuedWorkspace(5)

#set a UB matrix using the vector along k_i as 1,1,0, and the 0,0,1 vector in the horizontal plane
SetUB(ws,a=5,b=6,c=7,alpha=90, beta=90, gamma=90, u="1,1,0", v="0,0,1")

#check that it works
from numpy import *
mat=array(ws.sample().getOrientedLattice().getUB())
print "UB matrix"
print array_str(mat,precision=3, suppress_small=True)

.. testcleanup:: SetUB
DeleteWorkspace(ws)
```

#### Output:

```
.. testoutput:: SetUB
UB matrix
[[-0.  0.  0.143]
 [ 0.129 -0.128  0. ]
 [ 0.154  0.107 -0.  ]]
```

.. categories::



## SetUB v1

### Table of Contents

- Summary
- Properties
- Description
- Usage

### Summary

Set the UB matrix, given either lattice parameters and orientation vectors or the UB matrix elements



### Properties

Name	Direction	Type	Default	Description
Workspace	In/Out	Workspace	Mandatory	An input workspace.
a	Input	number	1	Lattice parameter a
b	Input	number	1	Lattice parameter b
c	Input	number	1	Lattice parameter c
alpha	Input	number	90	Lattice parameter alpha (degrees)
beta	Input	number	90	Lattice parameter beta (degrees)
gamma	Input	number	90	Lattice parameter gamma(degrees)
u	Input	dbl list	1,0,0	Vector along $k_i$ when goniometer is at 0
v	Input	dbl list	0,1,0	In plane vector perpendicular to $k_i$ when goniometer is at 0
UB	Input	dbl list	0,0,0,0,0,0,0,0	UB Matrix
MDSampleNumber	Input	number	Optional	For an MD workspace, the sample number to which to attach an oriented lattice (starting from 0). No number, or negative number, means that it will copy to all samples

### Description

The algorithms will attach an `OrientedLattice` object to a sample in the workspace. For MD workspaces, you can select to which sample to attach it. If nothing entered, it will attach to all. If bad number is entered, it will attach to first sample.



# Python API - auto reduction

- Simple shell driven script after a run completes
- Introduced as a help
- Users now **expect** autoreduction to work.
- Prototyping web driven reduction



## Results for Reduction for 29846

Job 11125

Status: COMPLETED

Start time: April 30, 2014, 1:32 p.m.

End time: April 30, 2014, 1:33 p.m.

Reduction parameters

Sample data: 29846

Beam center data: 29838

Sample transmission: 29841

Empty transmission: 29838

Background: 29844

Background transmission: 29838

Background empty transmission: 29838

Job directory:

[/lustre/snsfs/scratch/apache/m2d\\_221](#)

[11125.fermi-mgmt3.ornl.gov.ER](#)

[11125.fermi-mgmt3.ornl.gov.OU](#)

[29846\\_lq.txt](#)

[29846\\_lq.xml](#)

[29846\\_lqxy.dat](#)

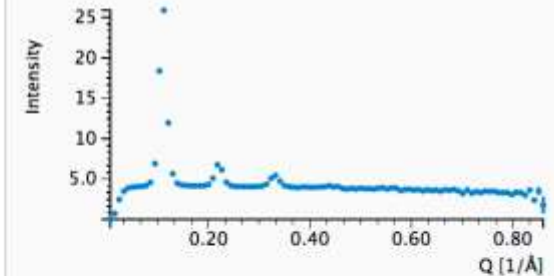
[29846\\_lqxy.nxs](#)

[29846\\_reduction.log](#)

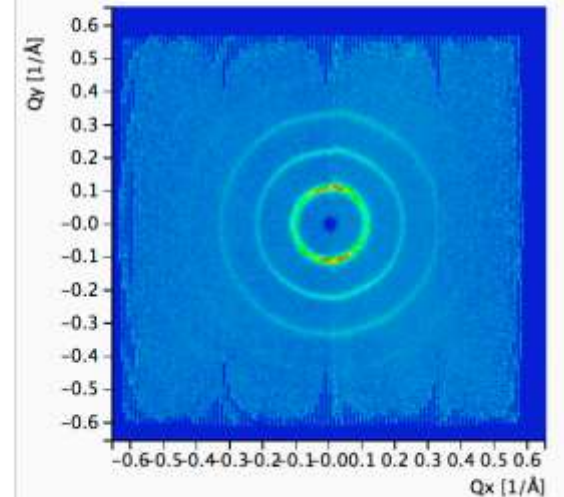
[submit.sh](#)

[web\\_submission.py](#)

Need to print? Click the 'adjust plot' link and use the print functionality of your browser.

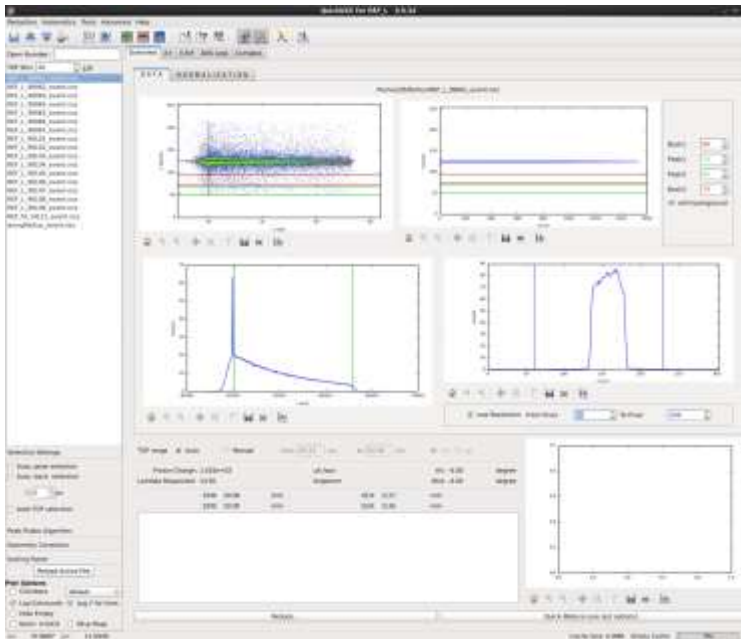


[adjust plot](#)



[adjust plot](#)

# Python API – use your favorite interface



IPython Notebook Mantid\_InelasticPowder\_Example Last Checkpoint: Jul 31 16:59 (autosaved)

File Edit View Insert Cell Kernel Help

Markdown Cell Toolbar: None

## Basic IPython Notebook Example:

### Inelastic Powder Projection Calculation and Visualization

The following IPython Notebook example illustrates the following:

1. Enables using Mantid Algorithms by setting up mantid.simpleapi
2. Reads in a Mantid workspace
3. Shows raw data
4. Calculates powder projections
5. Visualizes projected powder data
6. Saves ASCII data to file

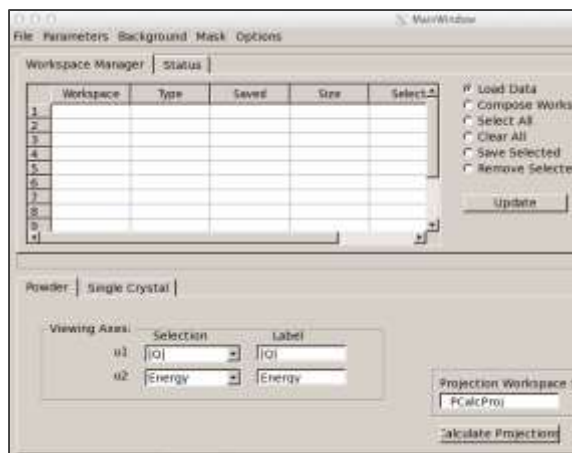
#### 1. Enable Mantid Simple api

```
In [1]: import sys,os
print "Getting up Mantid environment"
try:
    #check if MANTIDPATH environment variable exists
    mantidpath=os.environ['MANTIDPATH'] #check if MANTIDPATH environment variable is set
except:
    #case MANTIDPATH did not exist, so creates the necessary path additions
    mantidpath=r'/opt/Mantid/bin' #if not, then use the Linux path for Analysis computers
    #else set the MANTIDPATH environment variable since it seems not to be set
    os.environ['MANTIDPATH']=mantidpath
sys.path.append(mantidpath)
from mantid.simpleapi import *
print "Mantid environment initialized"
```

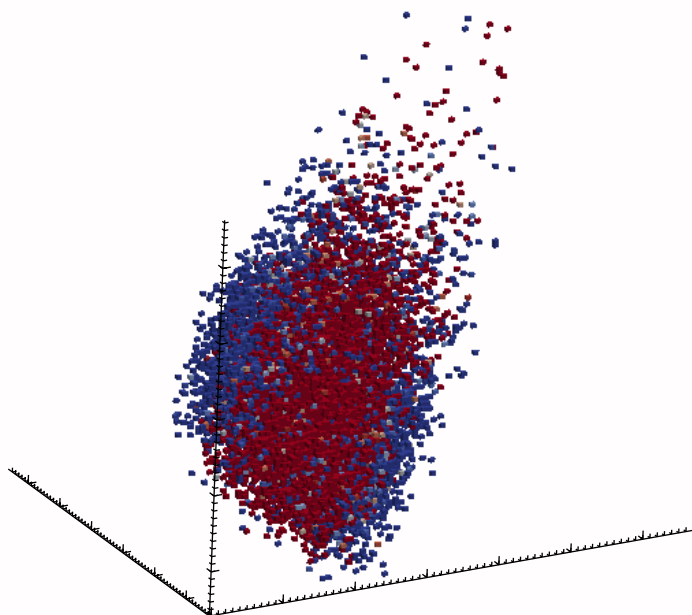
#### 2. Select File and read in raw data Mantid workspace

```
In [2]: #Load Python QT GUI environment to enable using a dialog to interact with user for file selection
from PyQt4 import Qt, QtCore, QtGui
useDialog=False #the user can change this flag - True enables a dialog pick file, False uses hardcoded
if useDialog:
    print "Using File Dialog to select a workspace file"
    from PyQt4 import Qt, QtCore, QtGui
    curdir=os.getcwd()
    filter='*.nxs'
    file = QtGui.QFileDialog.getOpenFileNames(None,'Open Workspace', curdir,filter)
else:
    print "Using hard coded path to an example data file"
    file=r'/SRG/users/public/Notebooks/data/scr_1000.nxs'

print "Loading workspace file: ",file
ws=Load(filename=file) #load Mantid workspace from file
ws.name #show some workspace info to see if the proper workspace was loaded
```



# Python API - get data in and out

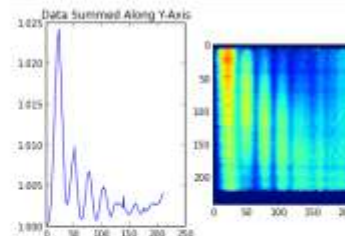


## 3. Show Raw Data

```
In [3]: #use this magic command to keep the plots within the notebook
print "Setting up matplotlib environment for basic data display"
%matplotlib inline
```

```
In [4]: import matplotlib.pyplot as plt
import numpy as np

#create a figure to hold the plots
plt.figure(1)
#extract 2D spectra
data=ws.extractY()
#sum to produce a 1D data set to visualize
data1D=sum(data,1)
#place 1D data in first figure
plt.subplot(121)
plt.plot(data1D)
plt.title("Data Summed Along Y-Axis")
#place 2D spectra data in 2D image plot
plt.subplot(122)
#show data in log scale to see detail better for this example
eps=0.00001 #place floor for log data to avoid log
plt.imshow(np.log10(data+eps))
plt.show()
```



```
In [5]: #disable inline plotting and let new windows pop-up separately
%matplotlib qt
```

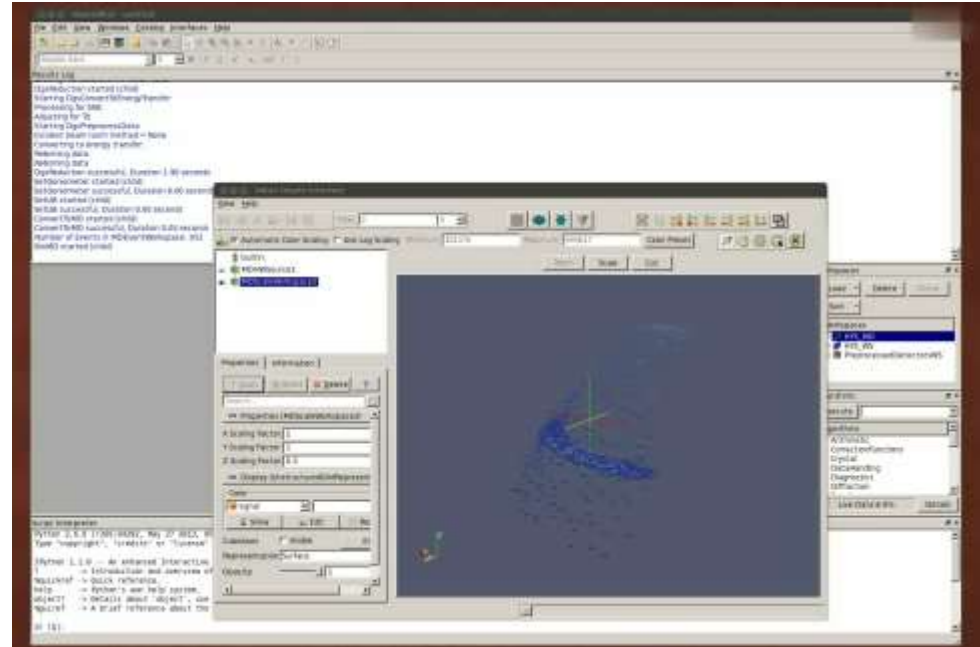
## 4. Calculate Powder Projection Data

```
In [6]: #reduce the data and converts the workspace to an multi-dimension (ND) array
ws_pwr]=ConvertFromND(ws,['Q'],'Direct') #output is the powder projection ND
```



# Live Data

- Converting to instrument units (Q and  $\omega$ ) as things come in.
- On HySpec, SEQUOIA, Vision, Correlli, and USANS at SNS
- ENGIN-X, MERLIN, LET, OFFSPEC, SANS2D at ISIS

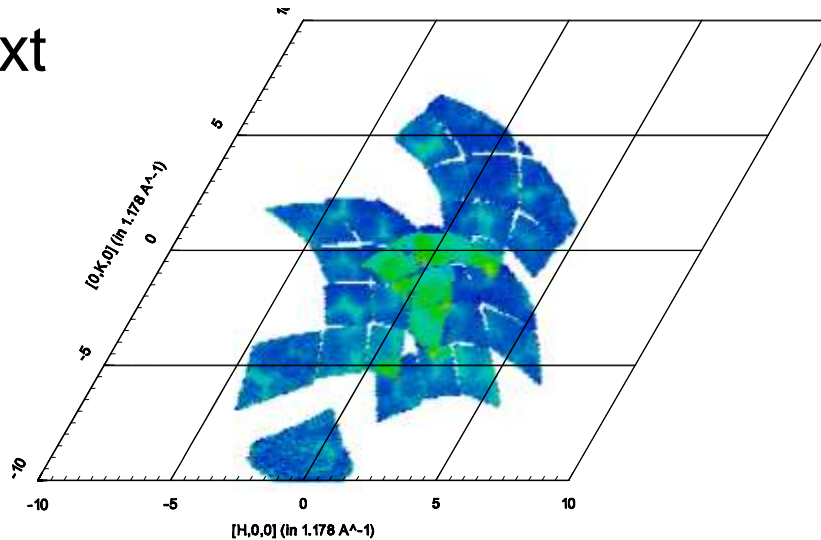
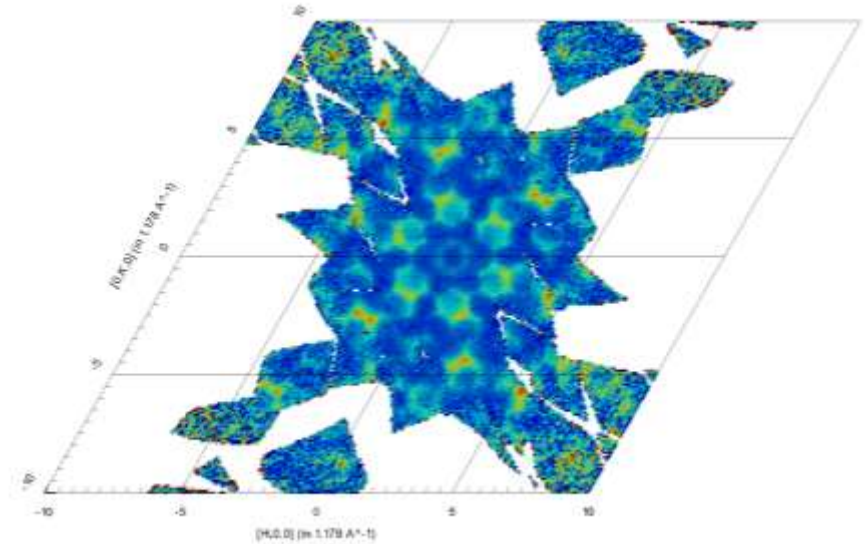


# Event Based Data reduction

- Allows for faster processing in most cases
- Currently working through normalization of different statistic runs
  - Some algorithms can be challenging
- Allows for pump probe filter experiments

# Event Diffraction Data Normalization

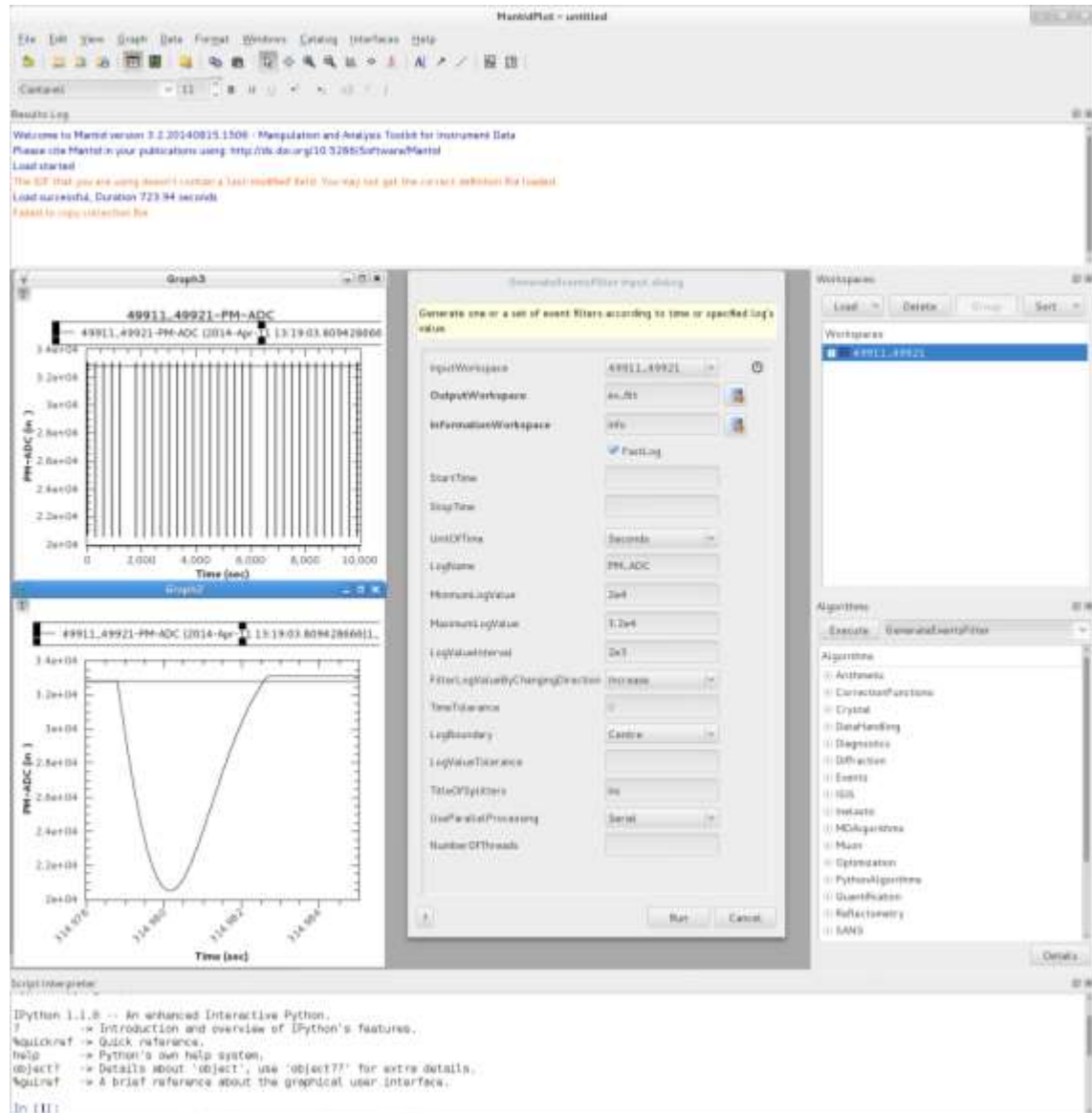
- Most Challenging for diffraction
  - Different Incident flux for every incident wavelength /time bin
- Moving to inelastic next
  - Much faster than histograms





# Filter fast logs – a use case

- Generate Filter on Pulsed Magnet log
- Prepare to put events in 5 workspaces corresponding to rising edge on magnet signal



# Filter fast logs – a use case

- Interface to perform the filtering

The screenshot displays the Mantid software interface. At the top, there is a menu bar and a toolbar. Below that, a 'Results Log' window shows the status of the software load and event filter generation. The main area contains two plots: the top one is a histogram of PM-ADC (h-) vs Time (sec) showing a dense series of vertical lines, and the bottom one is a line plot of PM-ADC (h-) vs Time (sec) showing a sharp dip. A 'Workspace' panel on the right lists the current workspace '49911\_49921'. A 'Filter Events' dialog box is open in the center, allowing the user to configure the filtering process. The dialog box includes fields for the input workspace, filter workspace, output workspace definition, information workspace, output TOF correction workspace, correction function, spectrum filter, and spectrum. The 'Filter By Phase Type' section is expanded, showing 'Group Workspaces' selected. The 'Run' and 'Cancel' buttons are at the bottom of the dialog box.

# Filter fast logs – a use case

The screenshot displays the MantidPlot interface with several key components:

- Log Console (Top):** Shows the execution of 'SumSpectra' and 'GroupDetectors' for two detector groups, '4.1.1' and '4.2.1'. It reports successful completion with durations of approximately 1.47 and 0.22 seconds respectively.
- Table (Left):** A table titled 'workspaceregru' showing PM-ADC values for various detector channels.
- Graphs (Bottom Left):** A plot of PM-ADC (B) vs Time (sec) showing a characteristic dip in the signal.
- Detector Geometry (Center):** Two windows show the detector layout for '4.2.1' and '4.1.1', with blue vertical bars representing detector positions.
- Properties (Right):** A 'MaskGroup' window for '4.1.1' showing a 'MaskingRect' with the following values:
 

Property	Value
left	0.728104
top	0.839908
right	1.928960
bottom	0.288418
- Workspace (Far Right):** A list of workspace objects including '4.1.1', '4.1.1.sum', '4.2.1', and '4.2.1.sum'.
- Algorithm List (Bottom Right):** A sidebar listing various algorithms such as 'Execute', 'SumSpectra', 'Analyze', 'CorrectionFunction', etc.
- Plot (Bottom Center):** A plot titled '4.1.1.sum' showing 'Counts' vs 'd-Spacing (Å)'. It compares two data series: '4.1.1.sum' (black bars) and '4.2.1.sum' (red bars). The red series shows significantly higher counts at higher d-spacings.
- Python Console (Bottom):** Shows the Python 3.1.3 prompt and some introductory text about the software's features.

# Filter fast logs – a use case

The screenshot illustrates the MantidPlot software interface used for neutron data analysis. The main window shows a data table with columns for 'left' and 'right' values. A plot titled 'd-1.h.sum' displays 'Counts' versus 'd-spacing (Å)'. An 'Algorithm Manager' window is open, showing a list of algorithms and their parameters. The 'Execution Summary' window shows the duration of the process. The 'Python Console' at the bottom displays the Python 3.1.2.8 prompt.

left	right
0.812014	1.329560
1.208078	0.288016

**Algorithm Manager**

Algorithm	Used	Name	Value	Default?	Direction
MaskFilter v.1		InputW...	d-1.h	No	Input
Load v.1		OutputW...	d-1.h.sum	No	Output
GenerateVetoData...		MaskFile	maskfile.m	No	Input
FilterEvents v.1		Spectrum...		Yes	Input
ConnectUnits v.1		Detector...		Yes	Input
Rebin v.1		Workspace...		Yes	Input
GroupDetector v.2		Keeping...	0	Yes	Input
		Behavior	Sum	Yes	Input
		Preserve...	1	Yes	Input
		Copyfile...		Yes	Input

**Execution Summary**

Duration	OS Name	Linux
0:11:52.2 seconds		
Date	OS Version	3.10.0-123.8.1.el7.x86_64
3/26/2014 18:08:17	Package Version	3.2.20140815.1208

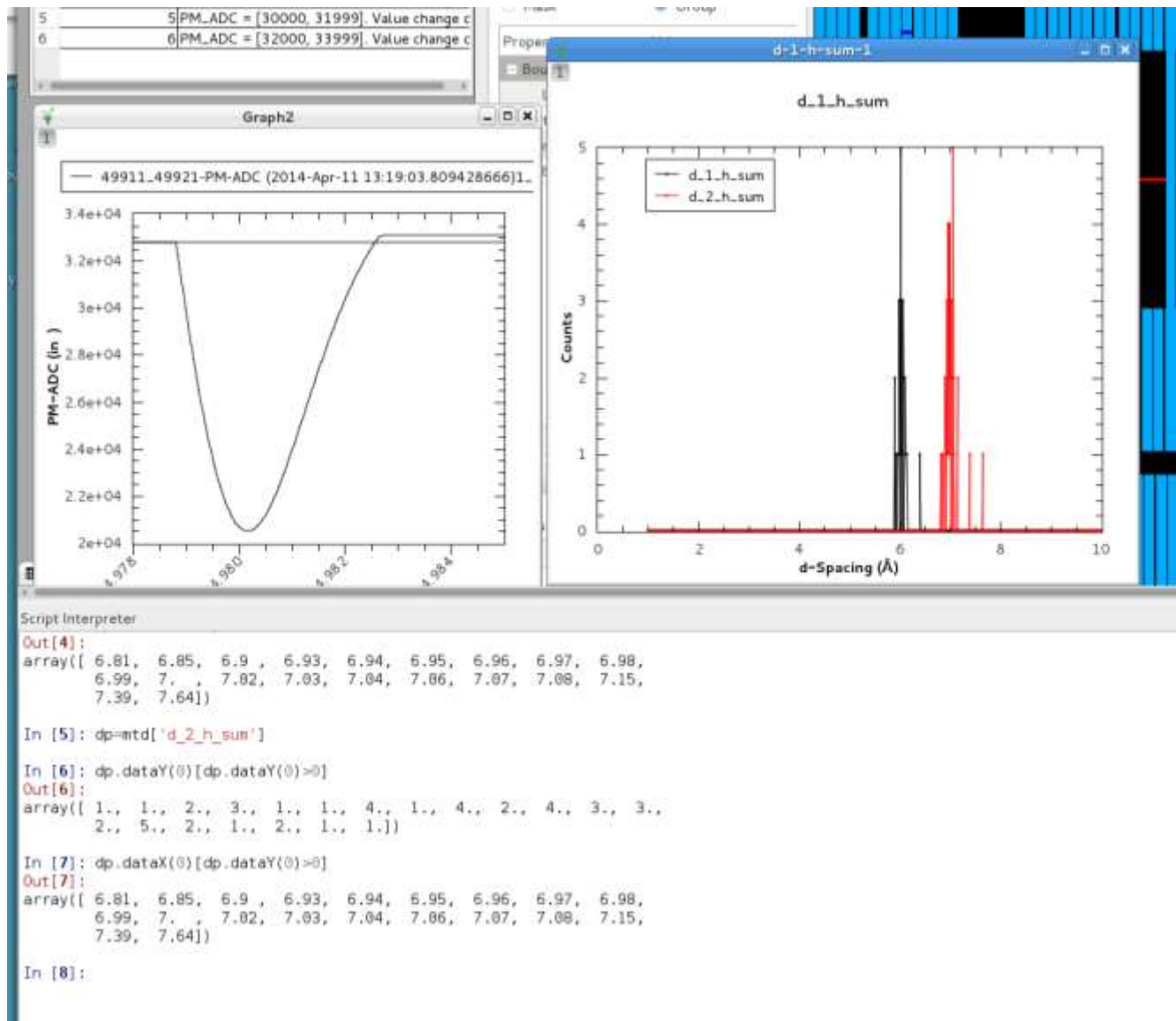
**Python Console**

```
Python 3.1.2.8 -- An enhanced Interactive Python.
?          -> Introduction and overview of Python's features.
helpref    -> Quick reference.
help       -> Python's own help system.
object?    -> Details about "object", use "object??" for extra details.
helpref    -> A brief reference about the graphical user interface.

In [1]:
```



# Filter fast logs – a use case



# Conclusions

- Mantid is seeing broad use at neutron scattering facilities
- The Python API provides an straightforward and powerful interface to the Mantid Algorithms
- Event based reduction provides added scientific functionality.
- Mantidplot is an interface useful for developing scientific workflows in Mantid.