Overview of use cases (DAS-33)

[DAS-36] dataset Created: 2019-Mar-26 Updated: 2019-Apr-09 Resolved: 2019-Apr-09

Status:	Done
Project:	Diffraction Analysis Software
Component/s:	None
Affects Version/s:	None
Fix Version/s:	None

Type:	Sub-task	**Priority:**	Normal
Reporter:	Thomas Holm Rod	**Assignee:**	Neil Vaytet
Resolution:	Done	**Votes:**	0
Labels:	VISPY		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original Estimate:	Not Specified		

Comments

Comment by Neil Vaytet [2019-Apr-09]

Introduction

The variety of python libraries available for data visualization is large and sometimes hard to follow these days, with solutions ranging from native GUI applications build on the openGL framework (e.g. Paraview), through Javascipt interactive plots (e.g. Plotly), to static figure generation for publications (e.g. Matplotlib).

I refer the reader to a summary I found on the Anaconda website: https://www.anaconda.com/python-data-visualization-2018-why-so-many-libraries/

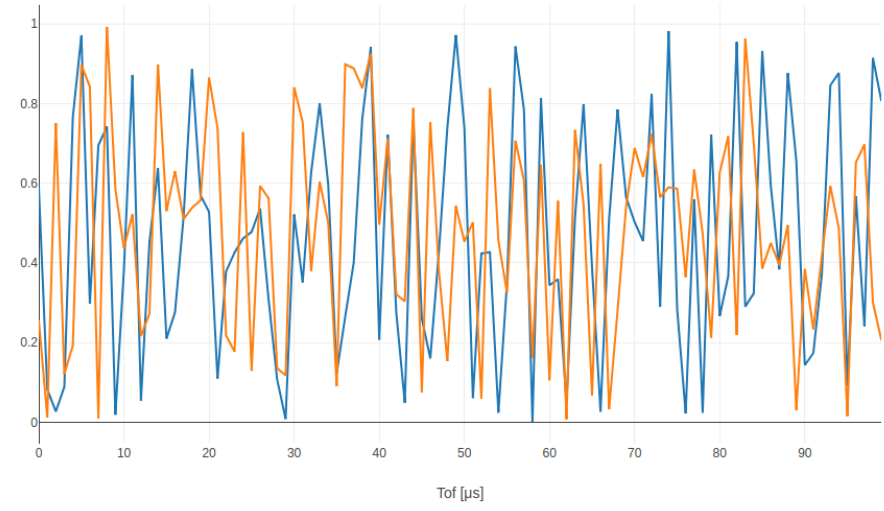
It is very likely that a single package will not be able to provide everything that we need for our visualization purposes. It could be the VIP project's goal to actually build that one library on top of the others in a unified and intuitive form.

One big question is **what is the purpose of each plot**: is it meant to be a figure included in a publication or is it supposed to be interactive visualization of data (does there need to be feedback between the visualization and the underlying data? e.g. selecting points inside a 3D volume for integration)

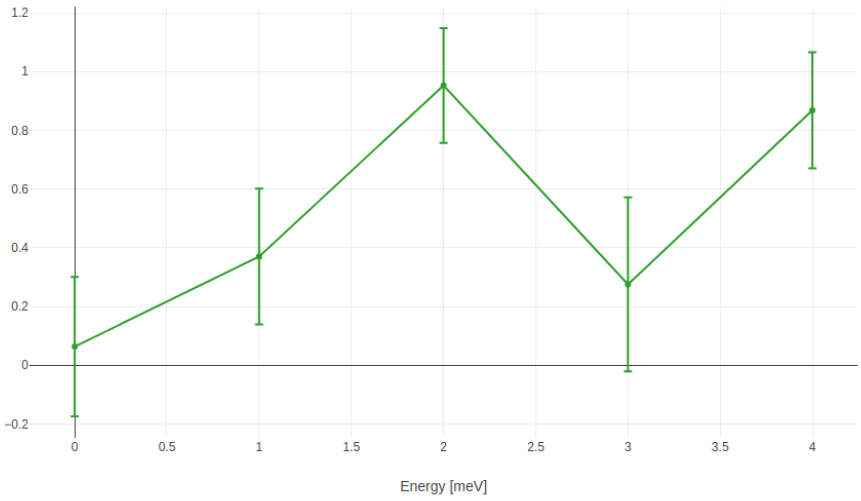
Plots in Sci++

So far, Sci++ is making use of the Plotly library to visualize the contents of Datasets. Here is a list of the different types of plots it can or needs to be able to generate.

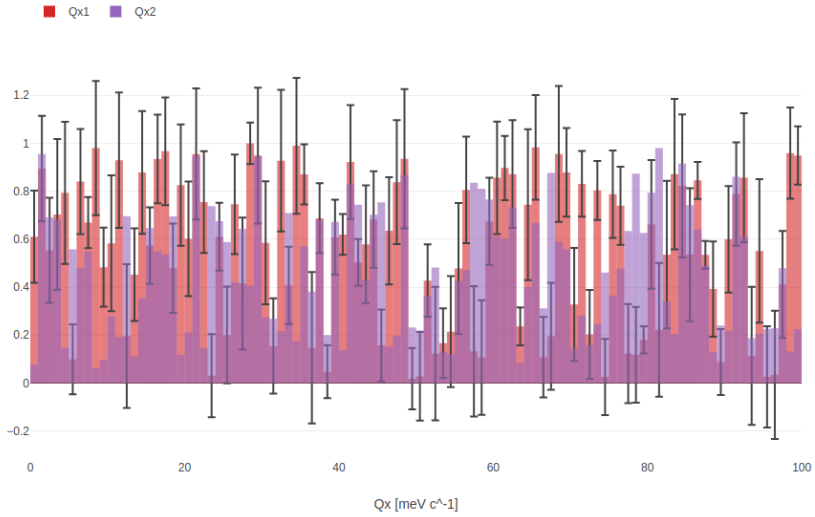
1D line plot



1D line plot with error bars



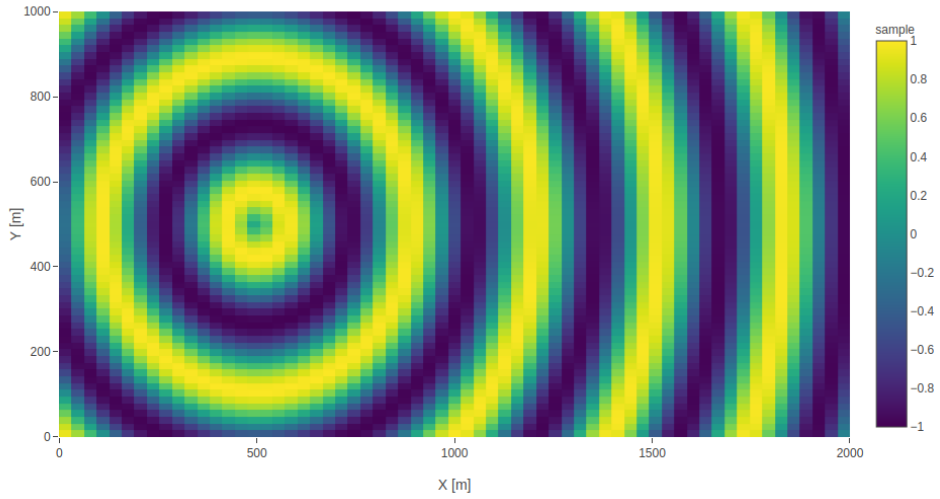
1D histogram plot (with error bars)



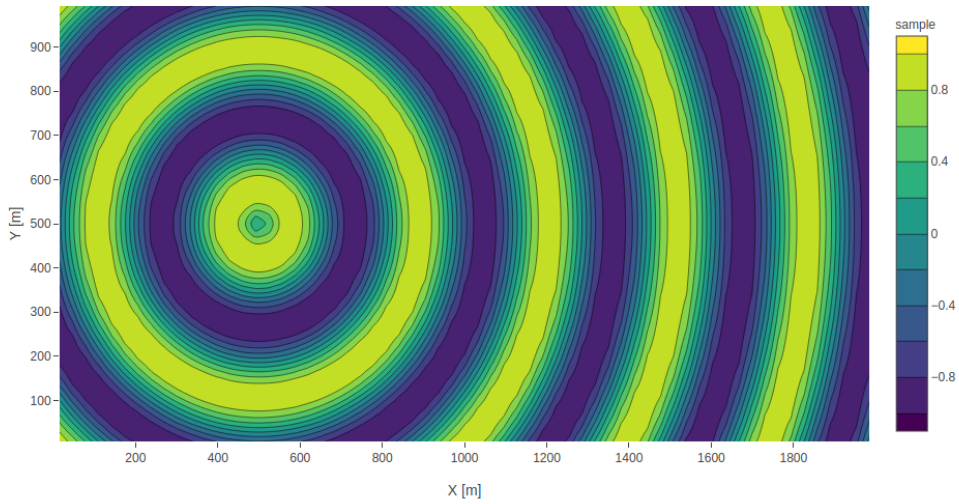
2D colour fill image

One should be able to plot images with non-regular pixel sizes, i.e. tools like matplotlib's imshow would not be sufficient as it assumes that all pixels are square.

We currently have a method to plot a low-resolution version of the image if the data is large and recover the full resolution as we zoom in (see https://community.plot.ly/t/heatmap-is-slow-for-large-data-arrays/21007/2)



or with contours



Waterfall plot

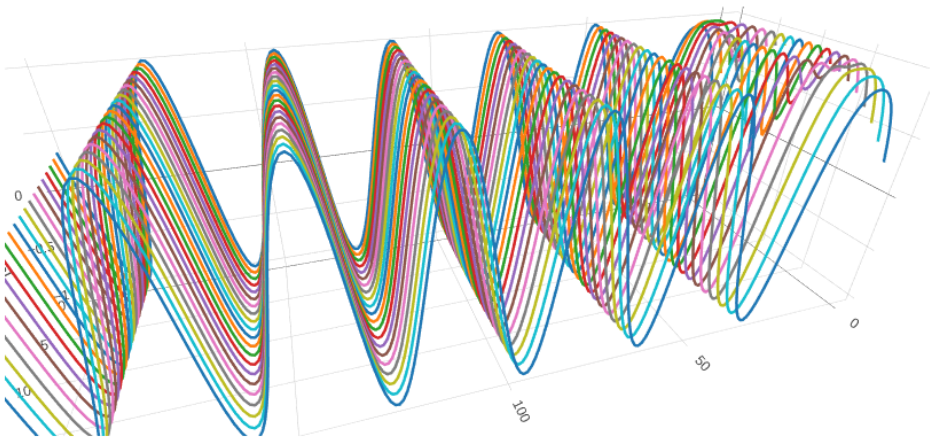
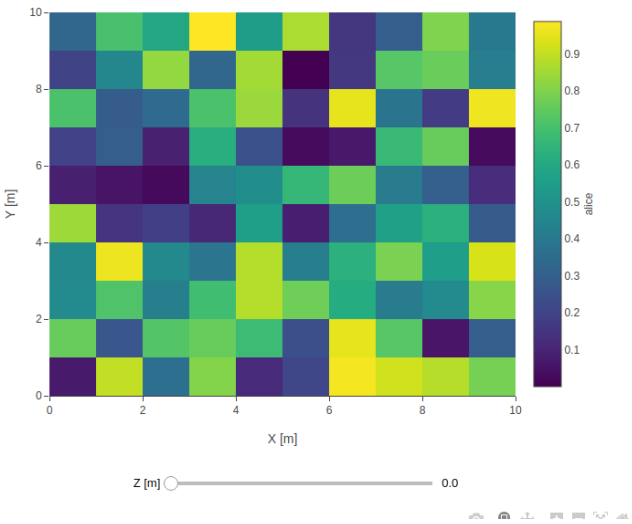
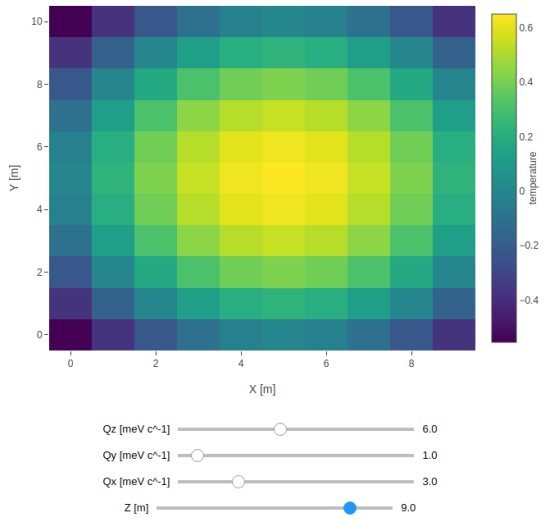


Image with slider for 3D data



or arbitrary number of sliders



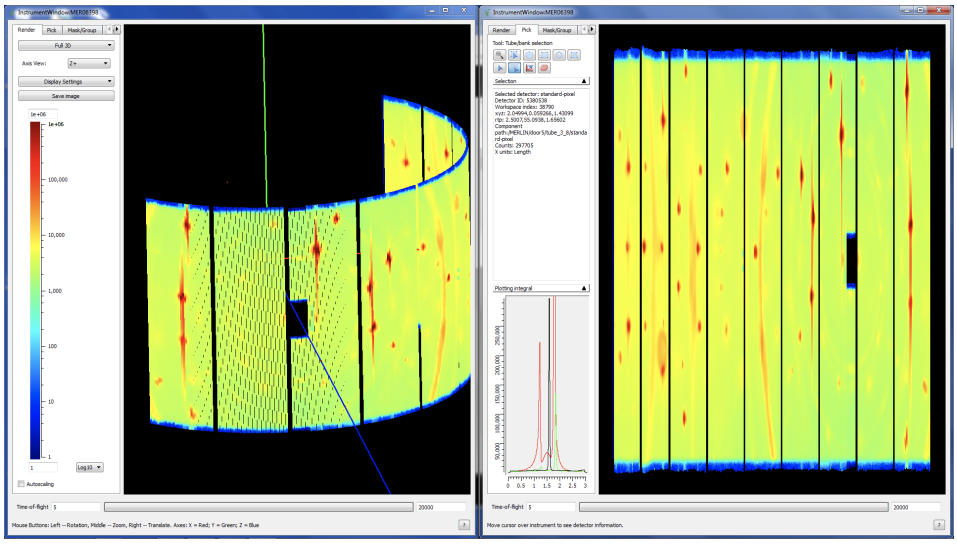
Plots in Mantid (will focus on the new Workbench interface)

Here is a good first look at the possible ways Matplotlib is now used in Mantid Workbench: https://docs.mantidproject.org/nightly/plotting/index.html

Note that some interactivity has been implemented on workbench figures: double click on axes or labels/titles to edit. Zooming is also possible. I'm not sure about showing/hiding a single curve, like plotly can do.

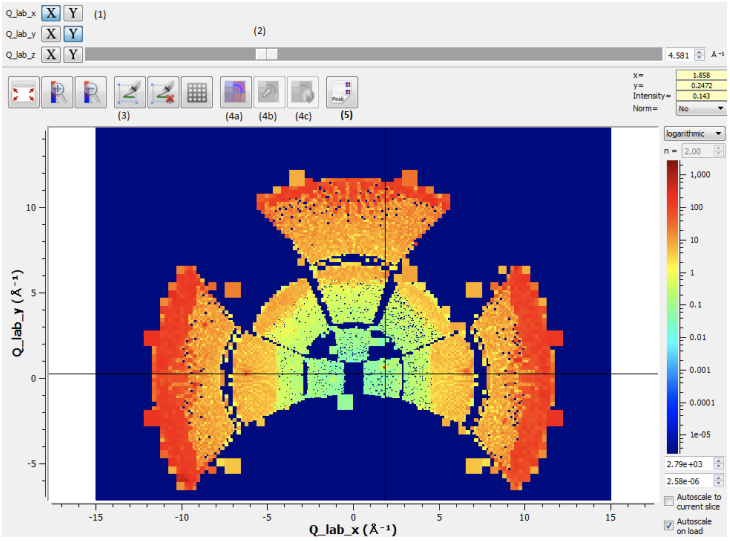
Other plots in Mantid:

Instrument view

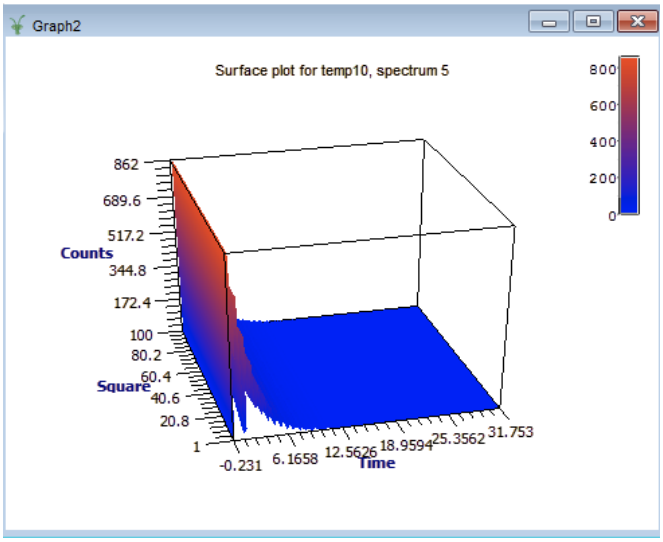


Probably the most advanced Mantid visualization tool: allows for full 3D view of data on instrument panels. Uses OpenGL. Allows for selecting detectors with a pick tool and plotting individual spectra.

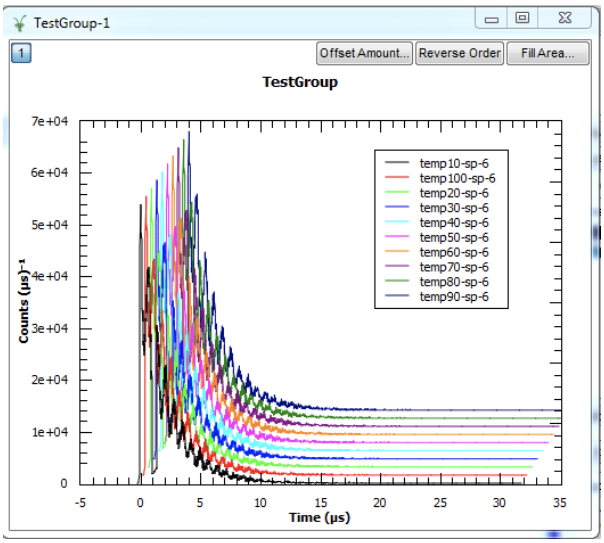
Slice viewer (3D data)



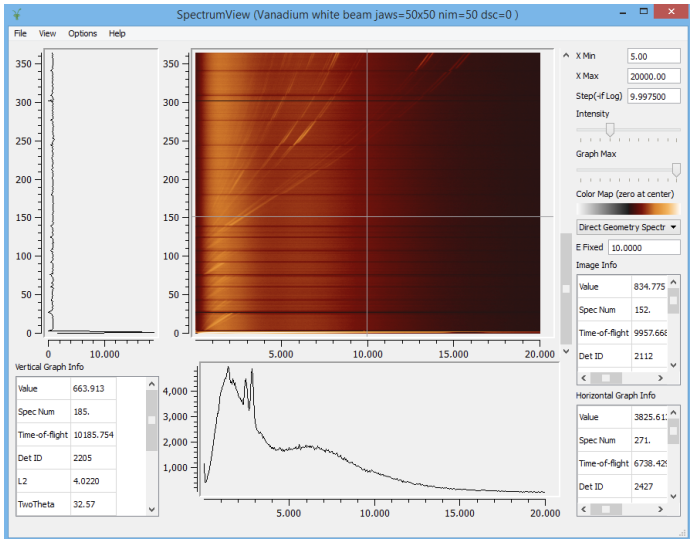
Surface plot



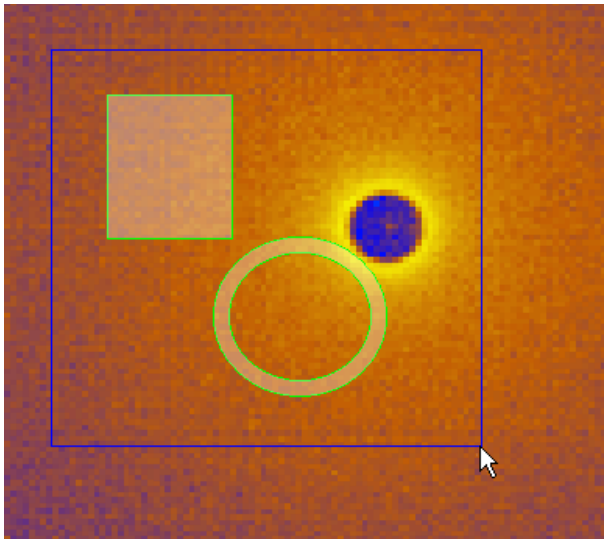
Waterfall plot



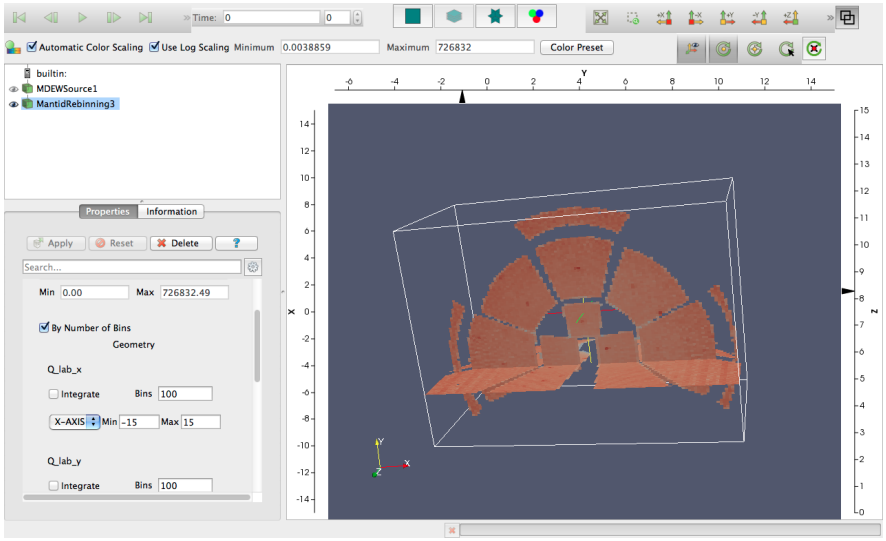
Spectrum viewer (2D data)



Masking



A note on 3D data visualization (VSI module)



Currently no one seems to be using it in Mantid. Some say they don't need it, others say they would use it if it had a few more functionalities.

For example, TOPAZ users working on SXD would like to have a tool where they can view events in 3D (like a 3D scatter plot), and roughly select points to integrate to perform a first quick computation of the UB matrix. VSI only allows to view the data, not select points and feedback to the data. What would be the best approach for this? Can we build something like this on top of Paraview (=VSI) or should we use something more like point cloud libraries which can handle much larger datasets?

Some examples here:

http://www.pointclouds.org/about/
https://developer.here.com/blog/interactive-visualization-of-10m-3d-points-with-new-open-source-python-package-pptk
http://www.open3d.org/docs/tutorial/Advanced/interactive_visualization.html

However, maybe other groups need to plot more than just points, meaning we can't simply use point clouds but we need something like paraview.

Here is a link to various examples that can be done in Paraview (examples for elastic diffuse scattering): https://corelli.pages.ornl.gov/paraview

Other ideas:

The "superplot" functionality from LAMP