Science & Technology Facilities Council

Technical Report
RAL-TR-2017-012

Overview of developments in MANTID relating to indirect inelastic spectroscopy July 2016-July 2017

L McCann

September 2017

Enquiries concerning this report should be addressed to:

RAL Library
STFC Rutherford Appleton Laboratory
Harwell Oxford
Didcot
OX11 0QX

Tel: +44(0)1235 445384
Fax: +44(0)1235 446403
email: libraryral@stfc.ac.uk

Science and Technology Facilities Council reports are available online at: http://epubs.stfc.ac.uk

ISSN 1358-6254

Overview of Developments in MANTID relating to Indirect Inelastic Spectroscopy July 2016 – July 2017

Louise McCann

16 August 2017

Abstract

The Manipulation and Analysis Toolkit for Instrument Data (MANTID) is an open source cross-platform application that provides a framework for data reduction and analysis relating to neutron and muon techniques.

This document aims to detail the development completed in MANTID used Indirect Inelastic neutron spectroscopy (INS) and covers MANTID releases 3.8, 3.9 and 3.10. This development was focused on maintaining the Graphical User Interfaces (GUIs), the underlying algorithms, and introducing new algorithms designed to speed up the initial workflow of data reduction and analysis of INS datasets.

Table of Contents

Acknowledgements

I would like to thank Dr Spencer Howells and Dr Sanghamitra Mukhopadhyay for their support and contributions throughout the year. I would also like to thank the Molecular Spectroscopy Group (MSG) for their scientific contributions and aid in testing. Finally, I would like to thank the rest of the MANTID development team for help and support, and for testing the technical aspects of the code.

1. Introduction

The MANTID framework is an open source project with cross-platform support that provides data reduction, analysis and visualisation of neutron and muon scattering experiments [1].

The goal of the package is to present a high-performance framework that provides a common interface utilised by several facilities worldwide. This includes the Spallation Neutron Source (SNS) at Oak Ridge National Laboratory, US, the ISIS Muon and Neutron Source based at Rutherford Appleton Laboratory, UK, and the Institut Laue-Langevin (ILL), France. The MANTID project is maintained by dedicated teams from these institutions and other partner facilities. There is also an ongoing effort to add support for the European Spallation Source (ESS), Sweden when it is functional.

The project covers several areas of neutron science, including single crystal and powder diffraction, reflectometry, inelastic scattering, small angle scattering, and muon science. Previously, there were many small packages for each area of neutron science, which proved to be complicated and unmaintainable. The MANTID project aims to combine all of these small packages into one collaborative framework. The indirect inelastic areas of MANTID are originally based on the MODES package [2] and VESUVIO data analysis software [3].

The core of the MANTID framework is written in C++ which is used with Python. The main MantidPlot window is based on QtiPlot, allowing for visualisation and manipulation of data. In addition to the main MantidPlot interface, there is also support for scripting in Python.

The primary data structure used in MANTID is a workspace. This is a multi-dimensional object containing data and information about the data within. Data from workspaces can be easily visualised and manipulated by users. The main method for manipulating workspaces is through algorithms. There are also several Graphical User Interfaces (GUIs) that provide a simple interface for users. These GUIs typically call one or more algorithms and have simplified input options and formatted output. The MANTID framework also provides a fitting wizard, which provides several pre-defined graph fitting routines for data analysis. Users can also customise fitting to their needs.

This report aims to documents all the changes and improvements made to the areas of MANTID dedicated to Inelastic Neutron Scattering (INS) and Quasi-Elastic Neutron Scattering (QENS). The changes described mainly impact workflows used at ISIS Neutron and Muon Source, as well as comments on the recent development of a workflow for ILL data. Future changes and plans for the improvement of the MANTID project are also covered.

2. S(Q, ω) in Data Reduction

The Indirect Data Reduction interfaces in MANTID are used to reduce the raw time of flight (TOF) data from all indirect geometry instruments, ready for analysis. The Energy Transfer tab converts data from TOF to enerfy transfer and allows for modifications such as grouping and background removal. The calibration tab is used for IRIS and OSIRIS only to produce the calibration and resolution workspaces that cater to the Energy Transfer tab and many other workflow algorithms. In

the SofQW and Moments tabs, reduced data is converted into momentum transfer, Q, against energy transfer, ΔE, to calculate the first four moments. Further details of all Data Reduction tabs can be found in Ref [4] and the methods used in An Introduction to Inelastic Neutron Scattering and MANTID at Molecular Spectroscopy Group at ISIS [5].

The Indirect Data Reduction interfaces have not been particularly modified over the MANTID releases described in this report. There have been bug fixes and minor improvements to the underlying workflow algorithms. One change to note is that the SofQW tab, which is shown in Figure 1, no longer presents options for the different methods of rebinning workspaces, and instead only uses the Normalised Polygon method. This method uses fractional area tracking and a weighted sum of fractional polygons. Details can be found here:
http://docs.mantidproject.org/nightly/algorithms/SofQWNormalisedPolygon-v1.html

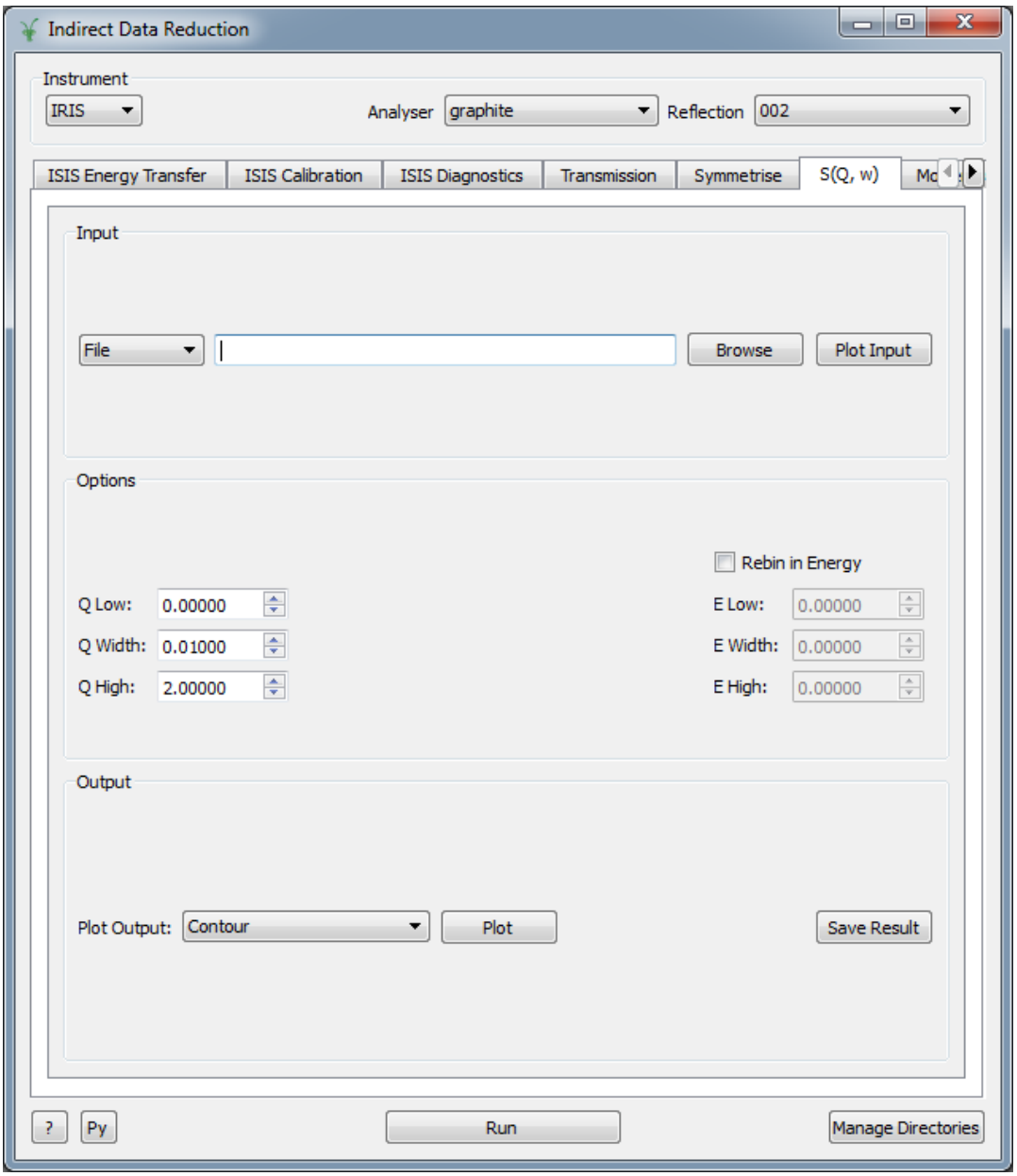

Figure 1 The S(Q,w) tab within the Indirect Data Reduction interface

3. Data Analysis for QENS Instruments

3.1 New models

The MANTID framework provides a fitting wizard to aid in fitting data to mathematical models. This fitting wizard is shown in Figure 2. There is a user interface provided for peak functions and users have the ability to create their own functions. The fitting engine can also be accessed via Python scripts. In the last year of development, two new models relevant to QENS have been developed, mainly through contributions from developers at ORNL. The new models are:

TeixeiraWaterSQE models the dynamic structure factor for a particle undergoing jump diffusion [6]. http://docs.mantidproject.org/nightly/fitfunctions/TeixeiraWaterSQE.html

IsoRotDiff models the dynamic structure factor for a particle undergoing continuous and isotropic rotational diffusion. http://docs.mantidproject.org/nightly/fitfunctions/IsoRotDiff.html

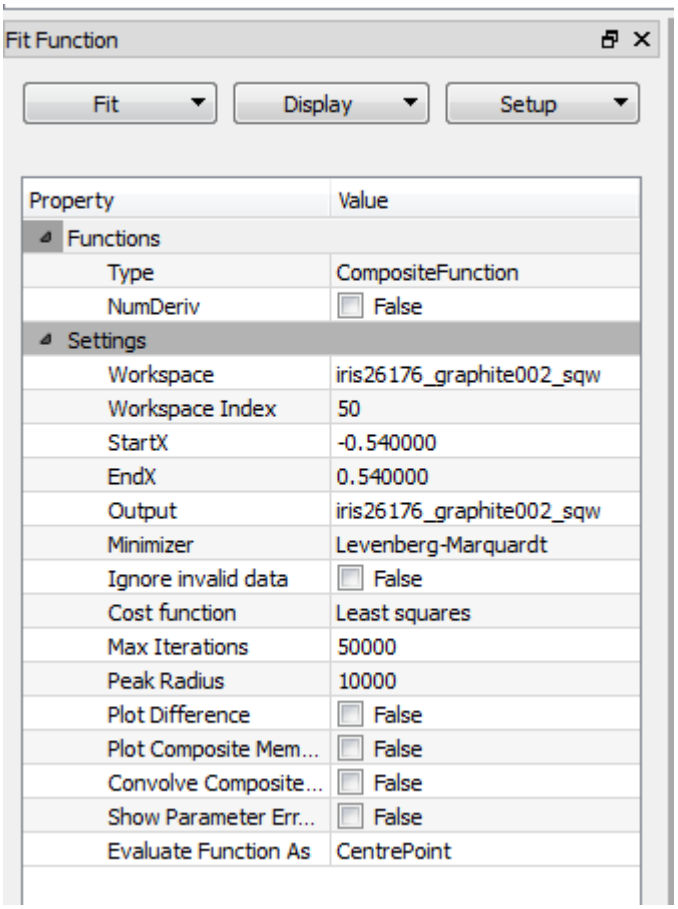

Figure 2 Details of the Fitting Wizard from the MantidPlot window

3.2 Iqt & IqtFit

The I(Q,t) and I(Q,t)Fit tabs in the Indirect Data Analysis interface convert reduced data to an intermediate scattering function and fits exponential decay functions, as described in [5]. These techniques are helpful in analysing QENS data

The underlying algorithms IqtFitMultiple and IqtFitSequential previously used user defined functions to fit exponential decay functions. However, there are now fit functions pre-defined within the MANTID framework that execute the same model. These fit functions are tested to a higher standard and using them would mean more shared functionality with the rest of the code base. The algorithms have been updated to use the ExpDecay function

$$Height \times exp^{-(\frac{x}{lifetime})}$$

and StretchedExp function

$$Height \times exp^{-(\frac{x}{lifetime})^{stretching}}$$

Both fit functions can be found in the MANTID Project documentation
http://docs.mantidproject.org/nightly/fitfunctions/index.html

3.3 ConvFit

Also in the Indirect Data Analysis interface is the ConvFit tab. ConvFit fits a scattering function S(Q, ω) using convolution theory.

There is an option to use the Fitting Algorithm for Bayesian Analysis of Data (FABADA) [7], a fitting minimizer provided with the MANTID framework. The advanced options of this minimiser have been exposed to the ConvFit tab to allow for finer control of the fitting. The layout of the options is shown in Figure 3.

Additionally, the ConvFit tab supports the use of a Delta Function with some of the fitting models built into the tab. The StretchedExpFT has been modified to allow the use of a Delta Function. More information about the functions used in the ConvFit tab can be found at
http://docs.mantidproject.org/nightly/interfaces/Indirect_DataAnalysis.html

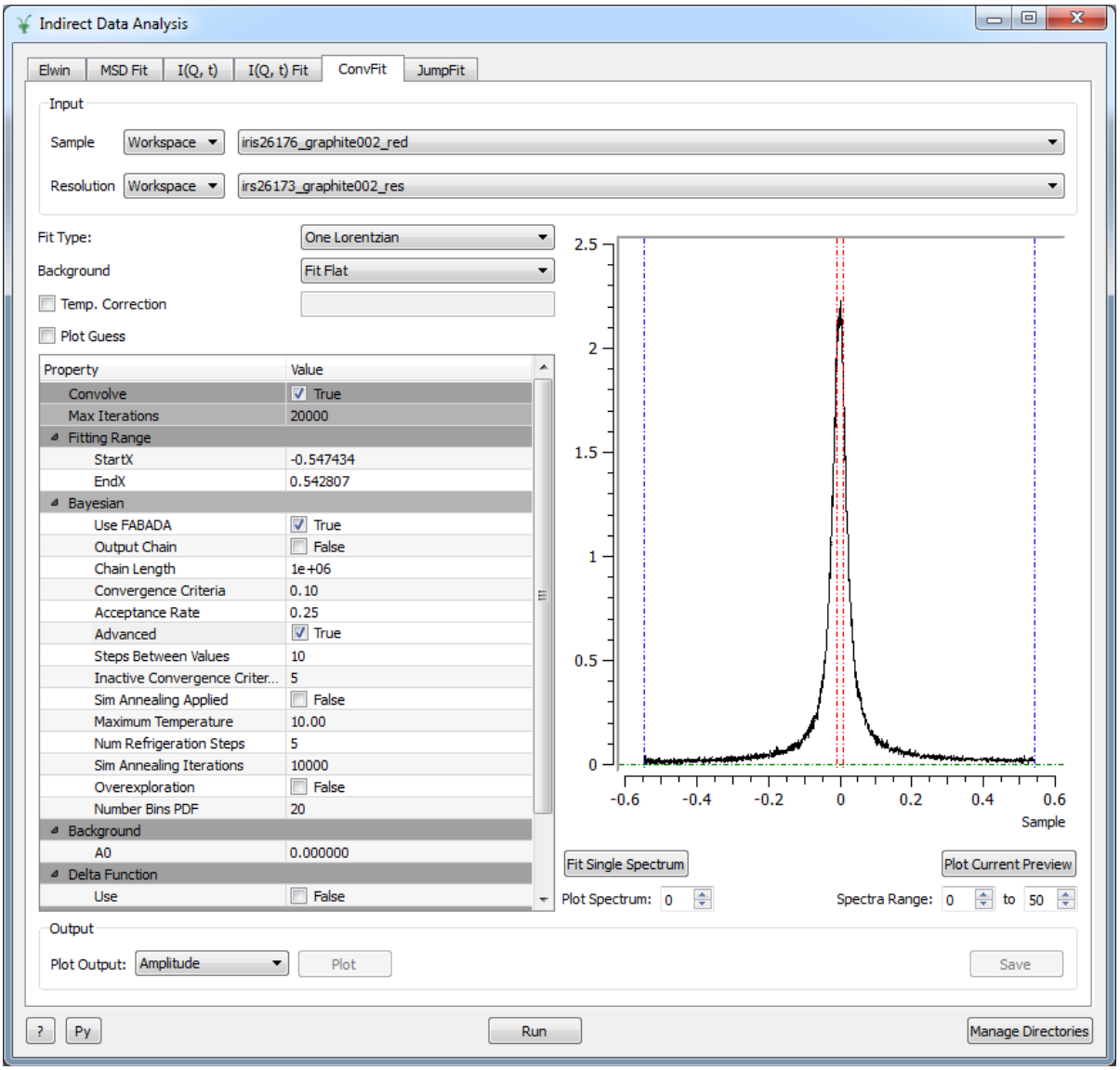

Figure 3 ConvFit tab with advanced FABADA options exposed.

4. Corrections

The Indirect Corrections interfaces are used to calculate absorption corrections for experimental data, based on the legacy MODES package [2].

The initial tab is ContainerSubtraction, which used to remove the containers contribution to a run. The tab has options for scaling or shifting the container data before performing a simple subtraction from the sample data.

The other tabs in this interface calculate factors for neutrons being absorbed within the sample and container and applies the appropriate corrections. The general purpose absorption algorithms: IndirectFlatPlate, IndirectCylinderAbsorption and IndirectAnnulusAbsorption have been refactored

and updated to use the Monte Carlo method to calculate the corrections. Details of the Monte Carlo method for absorption corrections are described here:
http://docs.mantidproject.org/nightly/algorithms/MonteCarloAbsorption-v1.html

The Monte Carlo method is a numerical way to find results to a problem that may be tricky to obtain using standard mathematical methods. The underlying technique for this method is to randomly sample data and then run the results through a deterministic algorithm, i.e. one that will always give the same results given a particular input. By following this process a numerical solution may be found.

The Absorption Correction algorithms are designed to correct the sample for factors generated by absorption and single scattering events within the sample and optionally the container.

The first stage of the algorithm sets the dimensions of the sample and the container. Currently, 3 styles of the container are supported. These are: flat plate, (a thin rectangular container), cylinder and annulus. Then the dimensions of the incident neutron beam are defined. The algorithm runs through all spectra in a run. Next, the spectrum is converted to wavelength, λ_{fixed}. Each of the bins in the spectra is then iterated over and the wavelengths before scattering, λ_1, and the wavelength after scattering, λ_2, are found. For each event, a random point in generated within the area of the beam as it strikes the sample. Another random point within the sample is created, and a scattering track between the two points is calculated. Any intersections between sample and container are recorded. Another scattering track is created from the point within the sample towards the detector position. Again intersections are recorded. The self-attenuation factor can be calculated from the two sets of intersections, given that the density of the materials involved is known.

The GUI shown in Figure 4 has been updated with additional parameters for beam size. These are taken from the instrument definition file by default. The interface has also been updated to be more user-friendly and now uses mass density. Previously, the default density option was number density, which can be hard to calculate. The MANTID framework has an algorithm, SetSampleMaterial. This can calculate the number density using tested methods, reducing the chances of incorrect values.

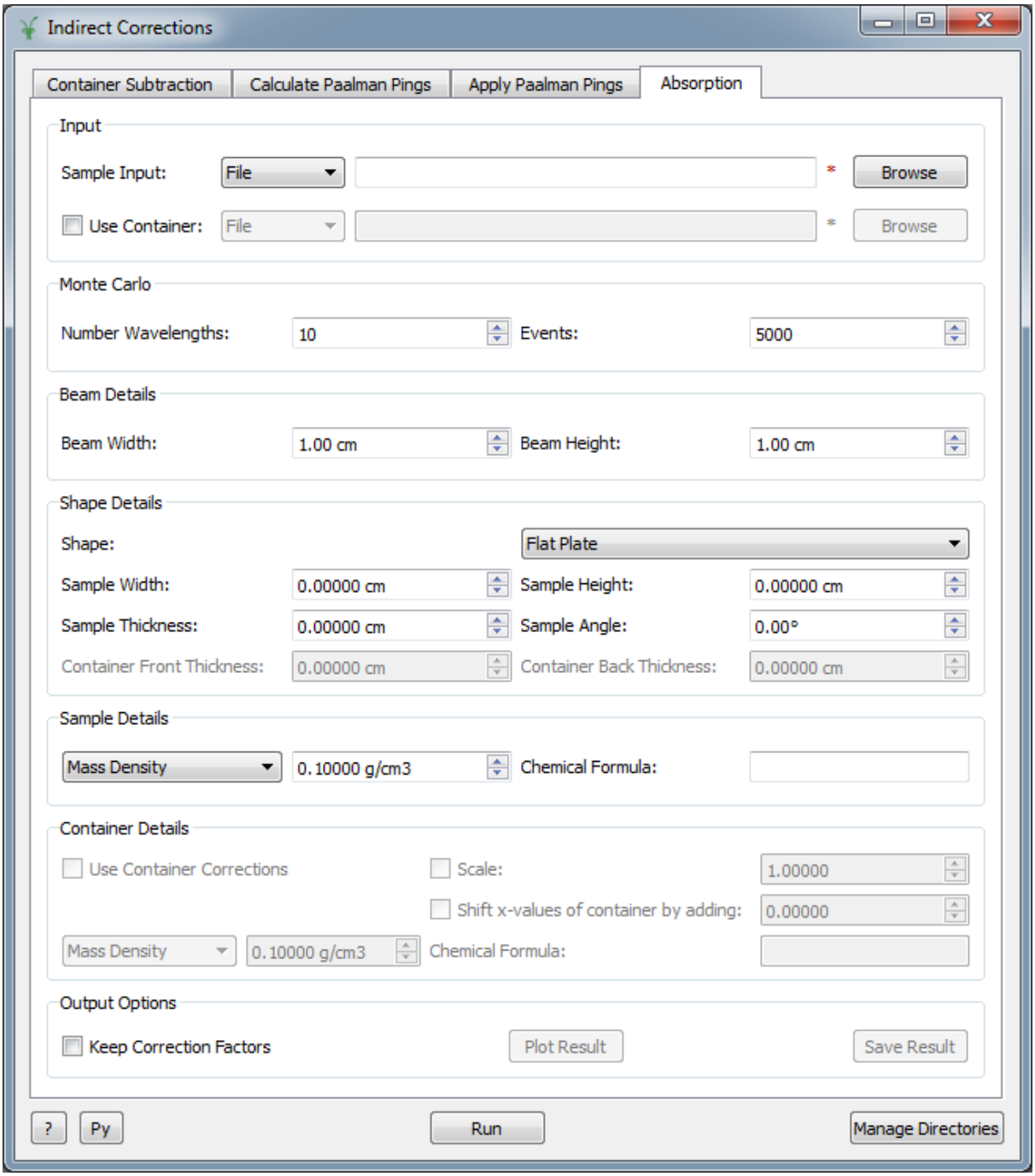

Figure 4 The Absorption tab within the Indirect Corrections interface. It has been updated so that the inputs are grouped in a user-friendly manner

The Paalman Pings GUI is split into two tabs. The first is Calculate Paalman Pings. This allows users to calculate absorption and attenuation corrections as described by Paalman & Pings in reference [8]. Various partial absorption factors are calculated, which are given in Table 1. As with absorption corrections, only flat plate, cylinder and annulus geometries are currently supported.

Symbol	Scatter From	Absorbed By	Workspace Name
$A_{s,s}$	sample	sample	ass
$A_{s,sc}$	sample	sample and container	assc
$A_{c,c}$	container	container	acc
$A_{c,sc}$	container	sample and container	acsc

Table 1 The various correction factors generated from Calculate Paalman Pings

The algorithms used to calculate the Paalman Pings factors have been extended to allow for an EFixed mode. In this mode, the analyser energy (meV) used is read from the instrument definition file by default. In the EFixed mode, the incident wavelength and final wavelengths are equal to each other and also equal to the EFixed value calculated from the analyser energy. The modes for Paalman Pings are described in more detail here: http://docs.mantidproject.org/nightly/algorithms/CylinderPaalmanPingsCorrection-v2.html. Figure 5 shows the GUI has also been updated to allow users to select between the different modes available.

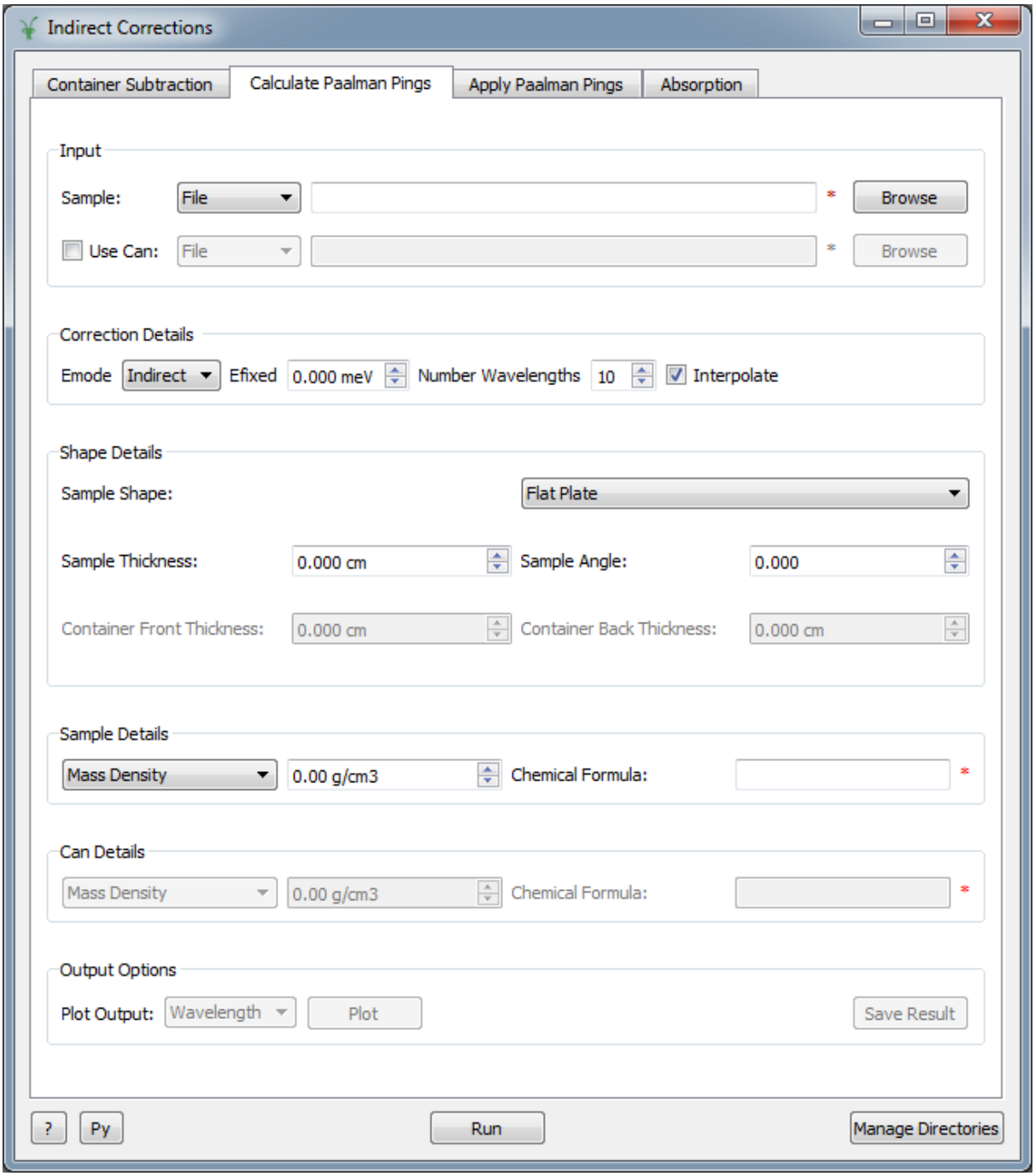

Figure 5 Calculate Paalman Pings with new Correction Details field

The second Paalman Pings tab is Apply Paalman Pings. This takes the workspaces calculated in Table 1 and applies the corrections. Any factors not supplied are assumed to have a value of 1. The workflows describing the different cases of input to Apply Paalman Pings are shown in Figures 6,7,8

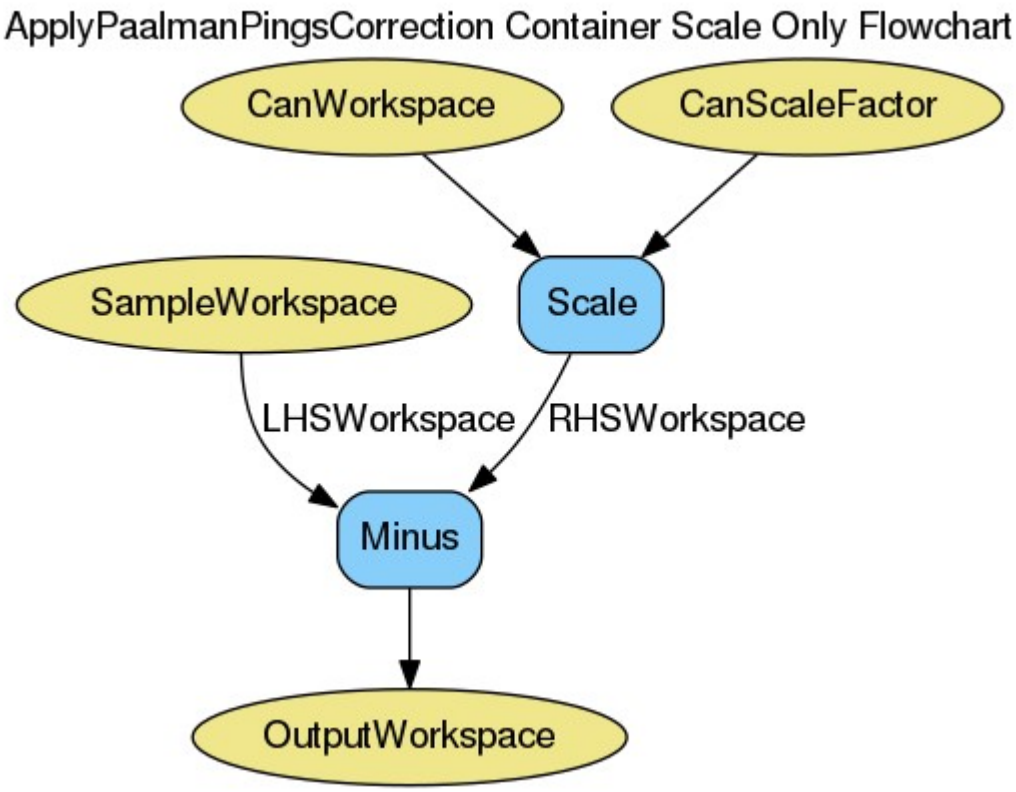

Figure 6 Workflow diagram for ApplyPaalmanPings in the case where only a container workspace and no correction factors are provided

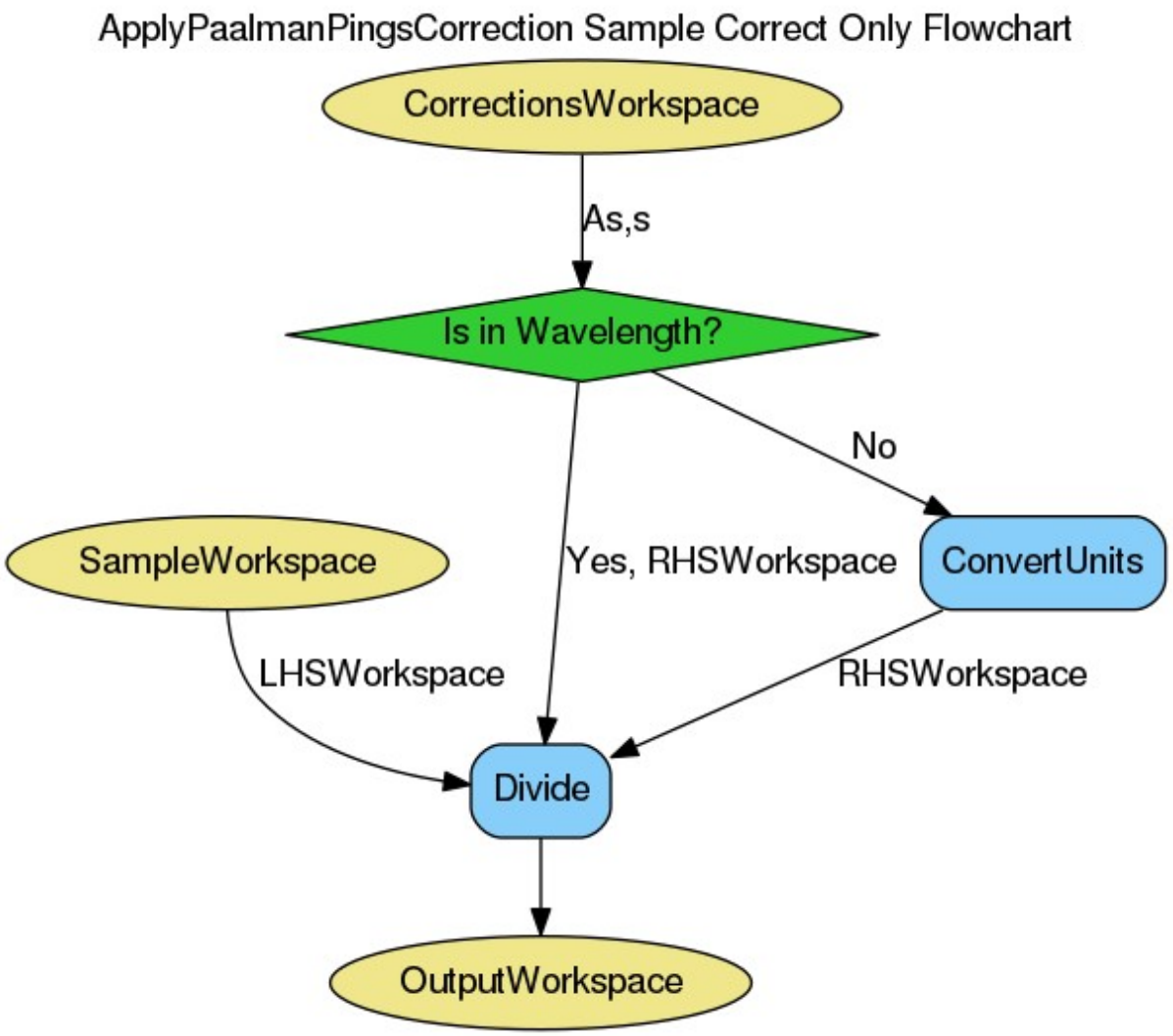

Figure 7 Workflow diagram for ApplyPaalmanPings in the case where only correction factors are provided

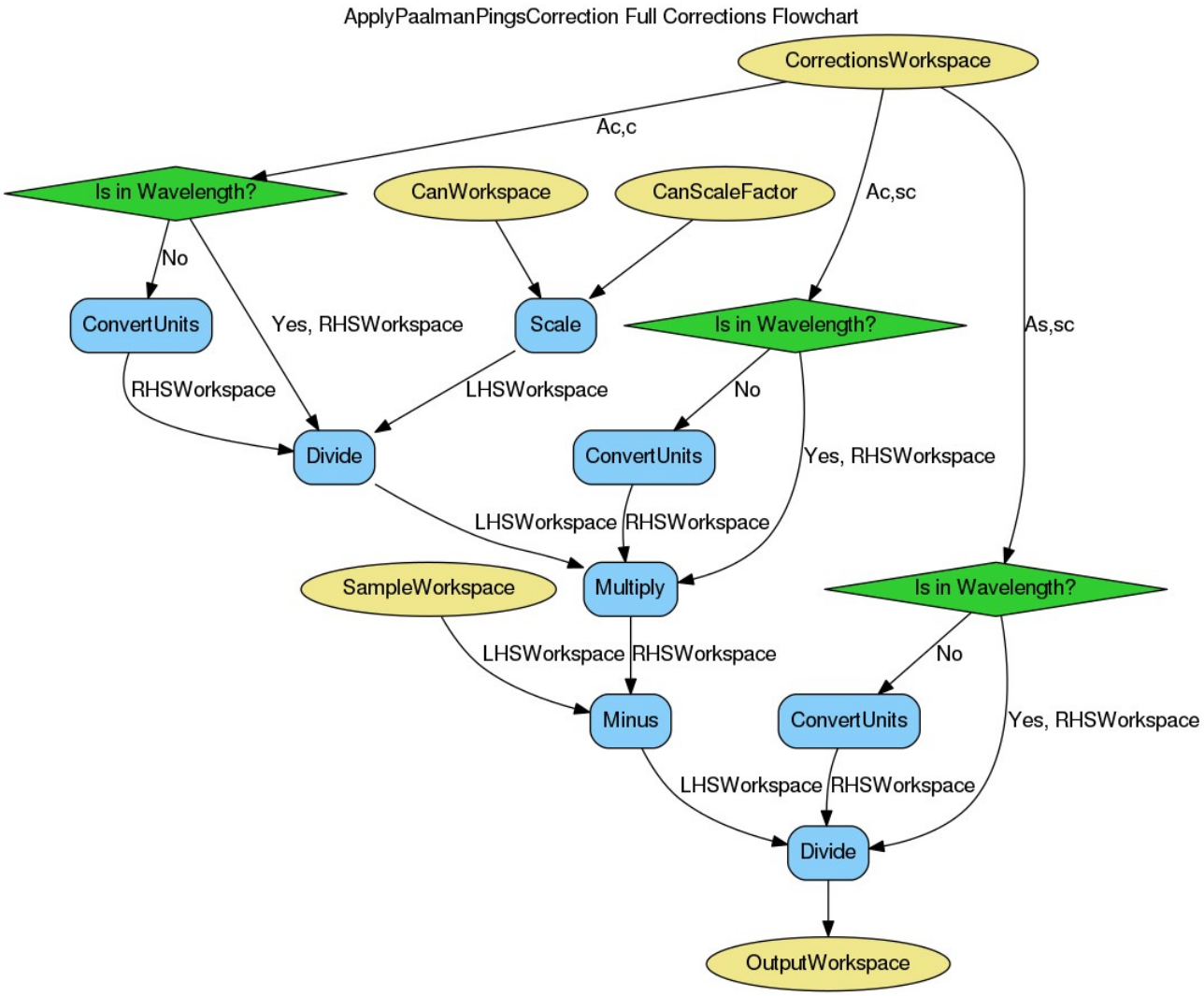

Figure 8 Workflow diagram for ApplyPaalmanPings in the case where both correction factors and a container workspace are supplied

5. Diffraction

The Indirect Diffraction Interface supports data reduction for all indirect instruments at ISIS. In Release 3.10, support for the VESUVIO instrument was added. In this interface, raw data is reduced into units of dSpacing. In the case of the OSIRIS instrument, there is an option to run the diffraction reduction in “diffonly” mode, correcting the monitor intensities using a vanadium file, generated by a vanadium target in the instrument, and stitching together d-ranges. The d-ranges are defined in the OSIRIS user manual [9]. The default diffspec mode has also been updated to allow for the sample to be normalised with a vanadium run.

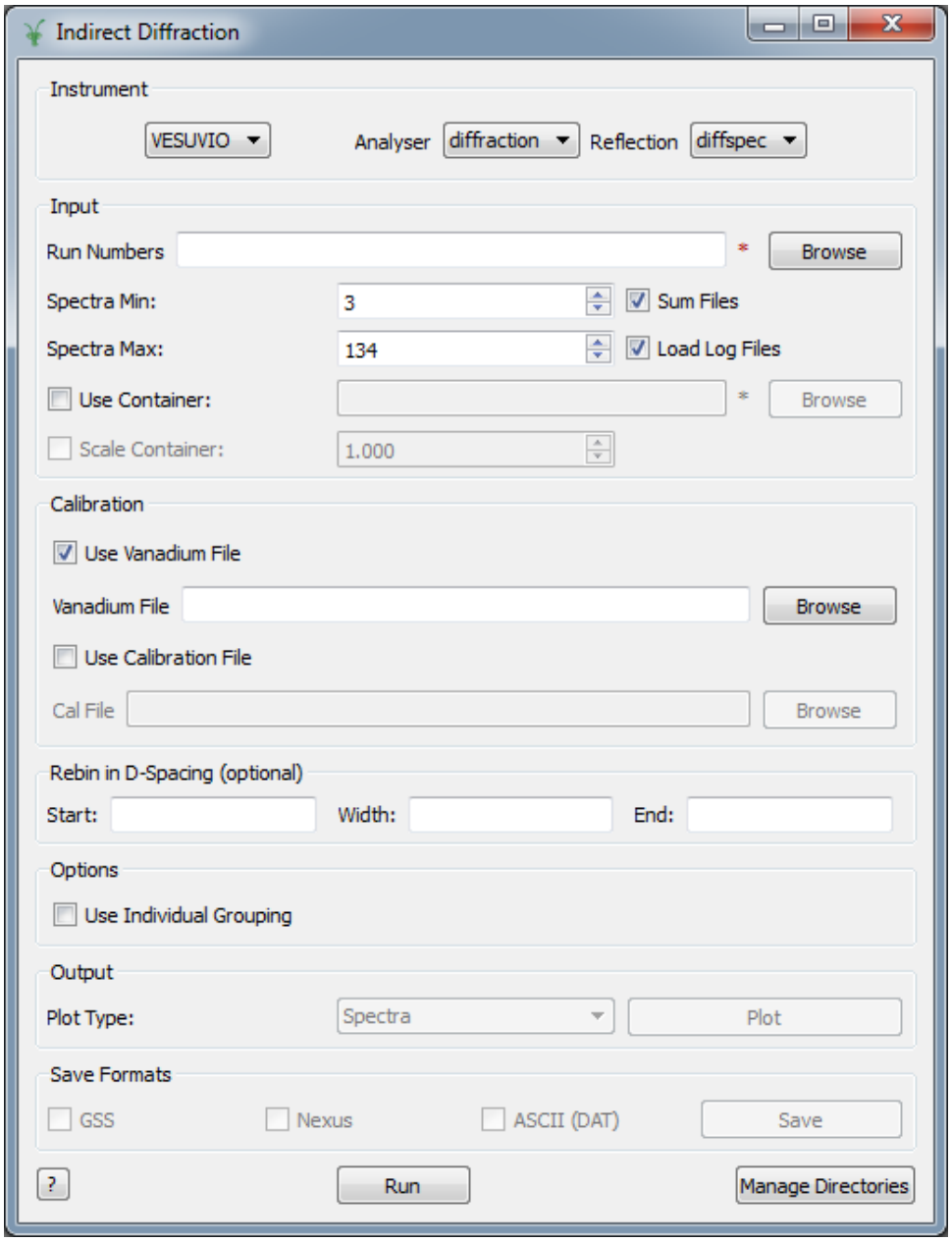

Figure 9 Diffraction interface showing VESUVIO instrument selected and vanadium file option

6. Additional Changes

6.1 Plot and save

All of the indirect interfaces have been updated to allow for plotting and saving after processing. This builds on from the work described in [10]. Previously, the plotting and saving were toggled through checkboxes, as shown in Figure 10, before the algorithm was run. Plotting and saving was previously incorporated as part of the algorithm. This functionality has been moved to the interface code where possible. As a result, for each interface, the resulting workspaces can be instantly plotted to the main MantidPlot window or saved by means of pressing a button. In many of the interfaces, there is a drop down menu which presents different options. When the interface is initiated, these buttons are disabled, ie greyed out and then re-enabled when the algorithm is complete.

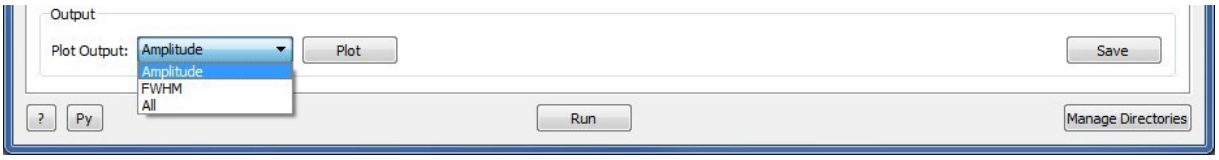

Figure 10 Showing the plot and save buttons and drop down menu.

6.2 Simulations

Although simulations are not fully a part of the MANTID framework, there is support for loading simulated data in order to compare it with experimental data. These files can be accessed through the Indirect Simulations interface. A new algorithm NMoldyn4Interpolation has been added to MANTID which will interpolate a simulated S(Q, e) workspace, onto an experimental OSIRIS S(Q, e) workspace. Simulated data can be generated from the nMoldyn software for molecular dynamics simulations [11]. This algorithm will allow users to compare between simulated and experimental data in MANTID.

6.2.1 Correlations

QENS data can be modelled using molecular dynamics simulations. The broadening of the elastic line that is a typical marker for QENS shows the movement of atoms within a molecule. Using molecular dynamics software such as CASTEP [12], the position and movements of atoms within a molecule can be calculated. Then, the increase of temperature with respect to time for each atom is simulated. Simulations of this scale often take many days for large computing clusters to complete. From the simulation, the velocity of an atom in any given direction and the rotational angle of the atom in both the horizontal and vertical directions can be obtained. From these values, velocity correlations between atoms can be obtained. Cross-correlations show the relationships between two different atoms at a particular point in time, auto-correlations show the relationship between the states of the same atom at two different times.

Two new algorithms have been implemented to read trajectory data from nMoldyn files and calculate the correlations with respect to velocity. These are VelocityCrossCorrelations and VelocityAutoCorrelations. In addition, there are two algorithms for auto-correlation functions with respect to angular rotation, AngularAutoCorrelationsSingleAxis and

AngularAutoCorrelationsTwoAxes. The difference between these two algorithms is that the first only considers a single axis, i.e. vertical or horizontal, while the second considers both.

For more information and details on the methods used see [12].

6.2.2 INS

A new plugin has been written for the MANTID framework. This plugin is called Ab-initio calculation of Inelastic Neutron Scattering, commonly known as ABINS. It is written and maintained by Krzysztof Dymkowski. It is a replacement for aClimax, 3rd party software which is no longer maintained [13]. ABINS uses the data output from density functional theory (DFT) programs such as CASTEP [14]. From this output, one-dimensional INS spectra are produced are compared to experimental data. Theoretical spectra can be created for a given type of atom and for up to fourth order quantum events. The dynamical structure factor, S, is calculated for all atoms in the system. Currently, ABINS only supports spectral data from indirect geometry instruments. A more detailed description of ABINS can be found here:
http://docs.mantidproject.org/nightly/concepts/DynamicalStructureFactorFromAbInitio.html

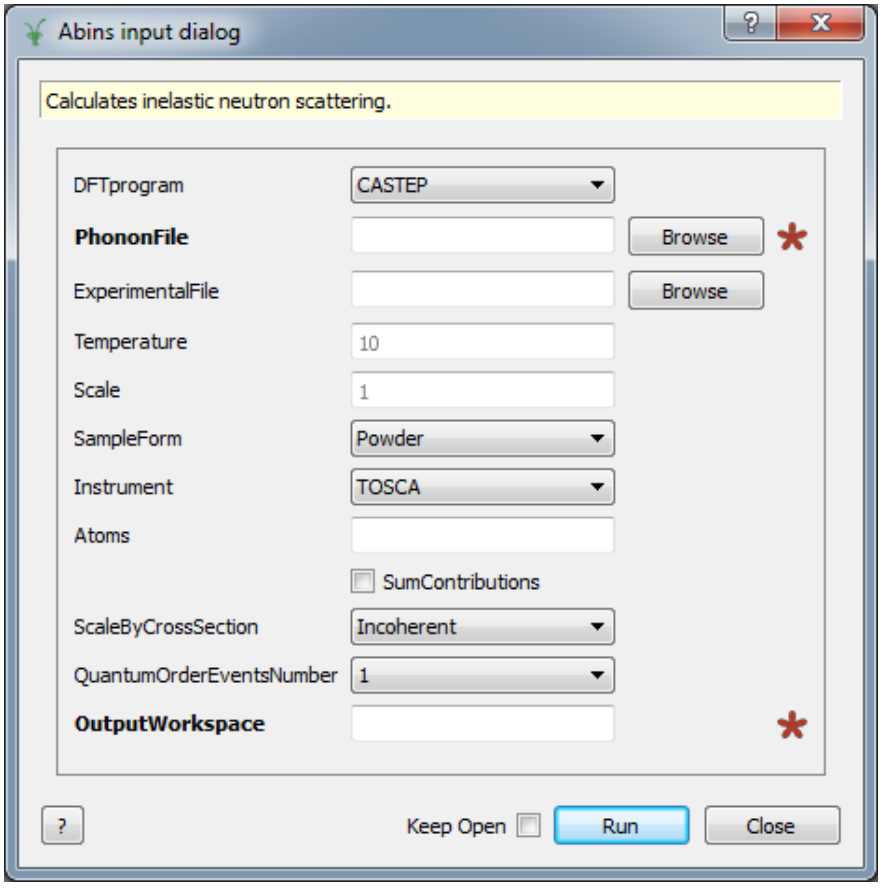

Figure 11 A GUI for the ABINS plugin automatically generated in MANTID 3.10

6.3 TOSCA update

Following an instrument review, the TOSCA instrument at ISIS underwent an upgrade to its beam guide, which provided a significant increase in the flux of neutrons that the instrument receives,

which can be seen in Figure 12. To support this change, a new instrument definition file was created, which includes updates to the positions of monitors. More work may be required to ensure that the data is normalised correctly. Information about the beam guide upgrade can be found in [15].

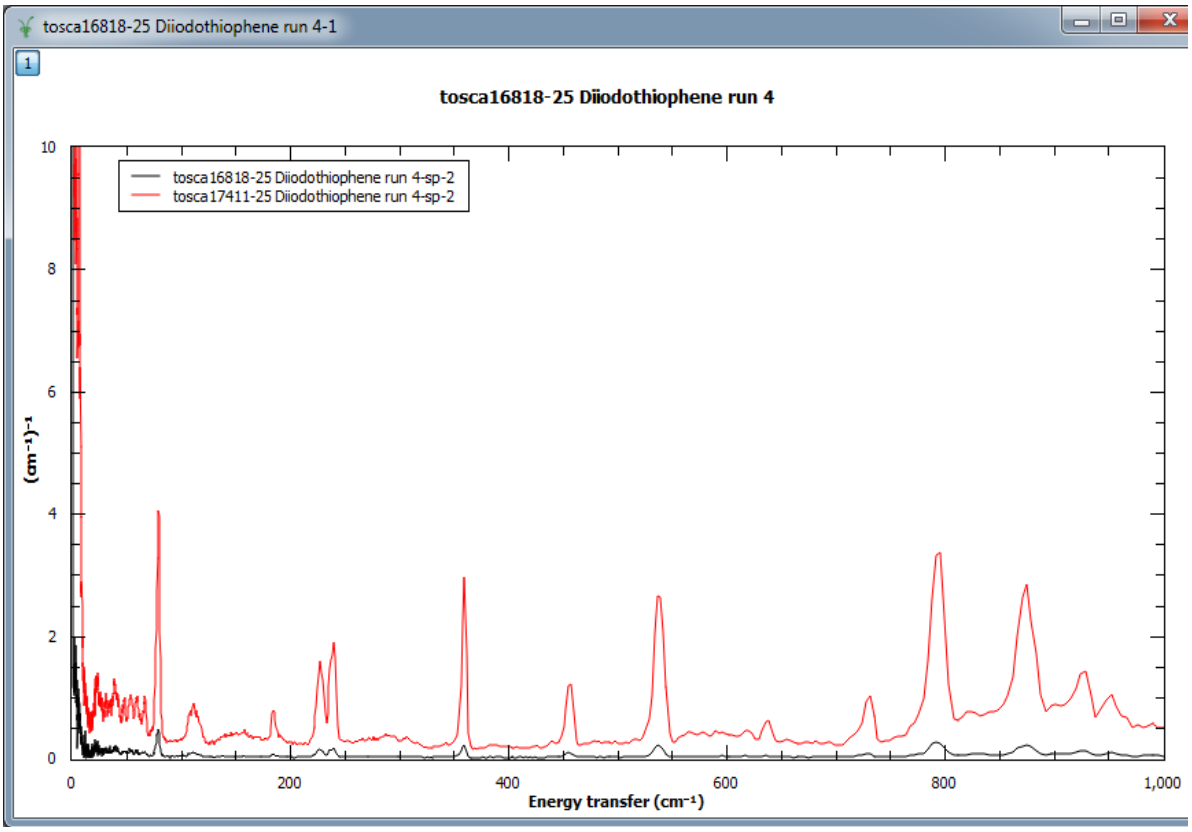

Figure 12 A standard spectrum of TOSCA data before the beam upgrade (black) and after (red)

In the MANTID framework, data and information about each instrument are stored in an instrument definition file (IDF). This file is written in XML (eXtensible Markup Language) format, which defines a set of rules for encoding information. A sample of the TOSCA instrument definition file is shown in Figure 13. When changes are made to an instrument that needs to be reflected in the MANTID framework, such as the TOSCA beam guide upgrade, it is important that the IDF before the change is stored and runs made before the upgrade can still be reduced with information from the old IDF. For this reason, multiple IDFs can be created for the same instrument, and each has a field which specifies the dates the IDF that is valid. Figure 13 shows that the most recent TOSCA definition file is valid from November 2016.

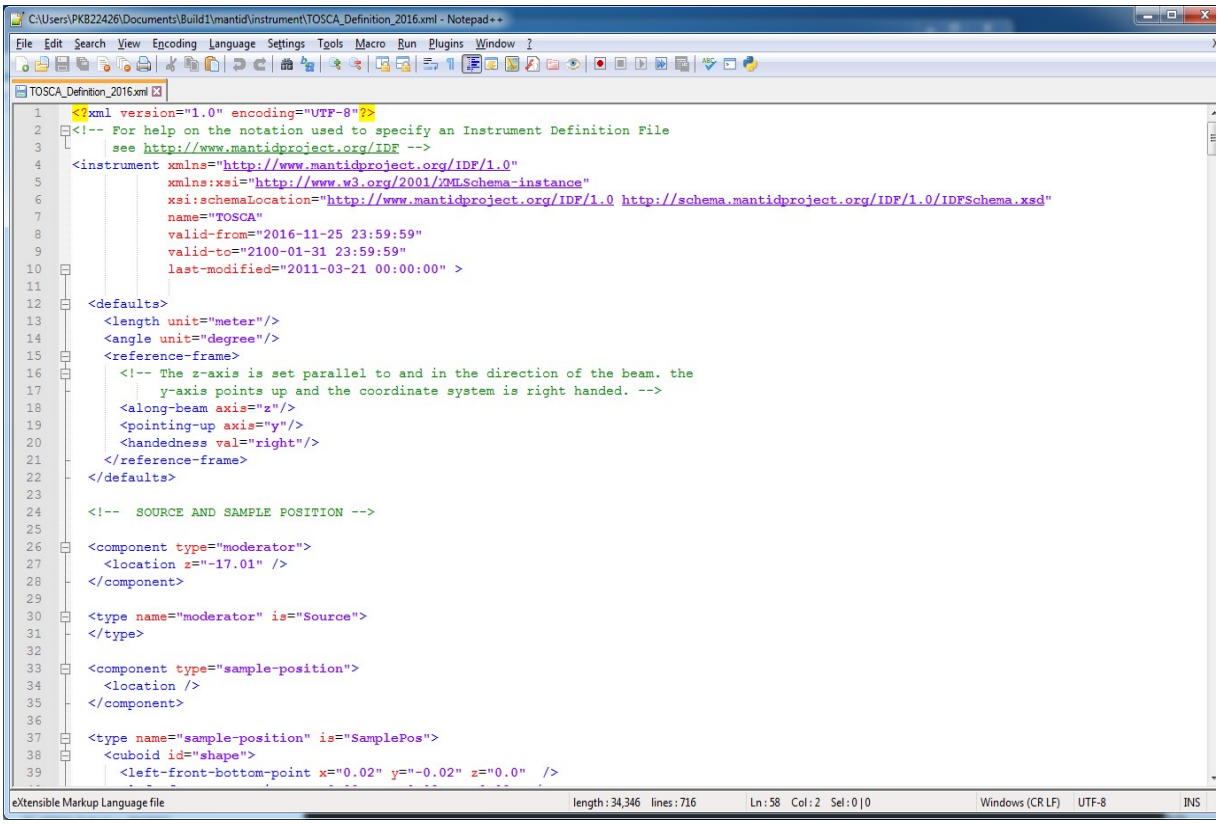

Figure 13 A section of the TOSCA instrument definition file

7. VESUVIO

During the last three years, there has been extensive development moving the existing VESUVIO data analysis scripts into the MANTID framework. This is described in [16], which also gives details of the workflows used in data reduction and analysis. During the releases covered in this document, the code relating to the VESUVIO workflow was polished and refined. However, there have been some small changes to the algorithms too.

The biggest change is that VesuvioDiffractionReduction has been deprecated. This algorithm reduced the data and converted it to d-spacing. As this functionality is covered in ISISIndirectDiffractionReduction, this algorithm has been expanded to allow for use with VESUVIO data, making the old algorithm obsolete. This includes the use of an Instrument Par File, which contains some of the instrument definition data for the VESUVIO instrument.

The code used in VESUVIO scripts is maintained in part by the instrument scientists. For example, a new algorithm was created in Python based on FORTRAN code created by the scientists. This algorithm, VesuvioPeakPrediction is used to predict the widths of peaks in reduced VESUVIO data. The data from the VESUVIO electron volt spectrometer can be used to find the neutron Compton profile as a series of Gaussian functions. The neutron Compton profile is a measurement of atomic momentum distributions. The standard deviation for the Gaussian peak is given as

$$\sigma = (\frac{3}{4} 0.1196 M \omega_D)^{1/2}$$

As given by equations 14 and 15 in reference [17]. *M* is the atomic mass of the atom and ω_D is the Debye energy. This formula for calculating standard deviation is derived from the Debye Model for the density of states.

8. High Throughput QENS

As part of the IRIS and OSIRIS review, there has been an effort to provide support for high-throughput QENS. The review promotes the development of a more user-friendly workflow for QENS data within the MANTID project [18].

Within MANTID project, four new algorithms have been created to allow for quick and simple data reduction and analysis. The existing indirect workflow algorithms have been stitched together with many of the more complex settings set to optimal defaults. The goal of these algorithms is allowing users to run these algorithms to reduce and analyse their data quickly to find regions of interest. From there, users can run their usual workflow to study their data in more detail. The algorithms have been dubbed “QuickRun” to emphasise the fact they are designed to be quick run-throughs of the typical indirect inelastic workflow.

8.1 EnergyWindowScan

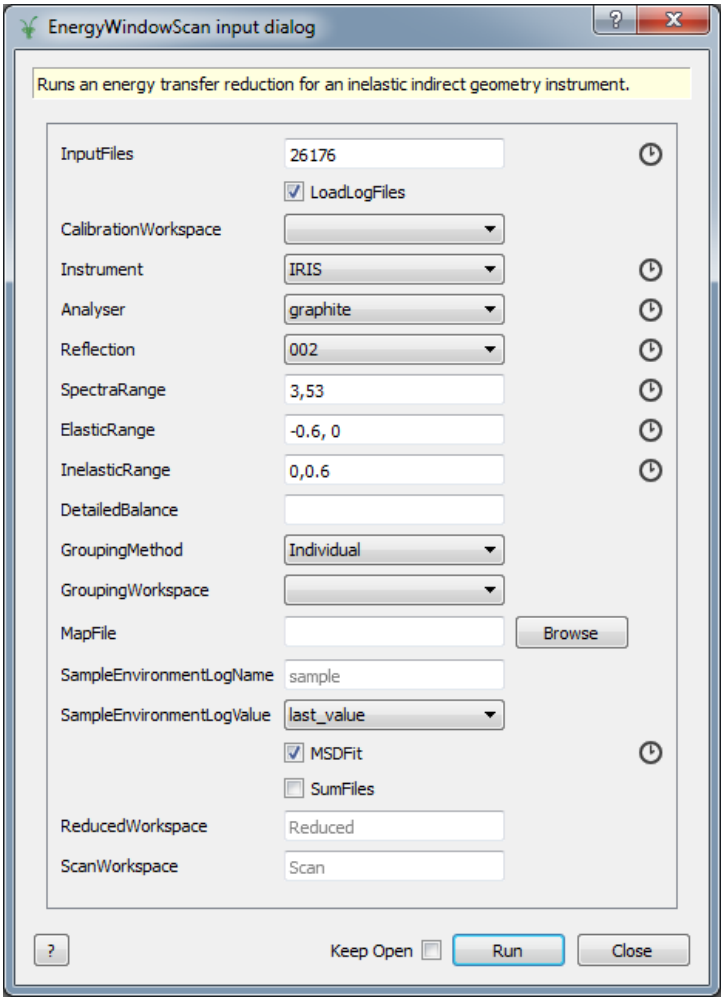

Figure 14 A GUI for EnergyWindowScan automatically generated by the MANTID framework

The primary QuickRun algorithm is EnergyWindowScan. This is formed of three of the most used workflow algorithms. First, it runs through IndirectEnergyTransfer. This loads raw data files, processes the monitors, applies calibration and then converts the original time-of-flight data to energy transfer in units of meV.

Once the raw data has been reduced, ElasticWindowMultiple, Figure 15, is called. This algorithm performs a series of elastic scans. This process involves integrating over each spectrum twice to get a workspace in momentum transfer Q, and Q^2. The Q workspaces are then converted so the vertical axis is in units of temperature if temperature can be read from the sample logs. The EnergyWindowScan algorithm runs this twice, once over the elastic region and once over the inelastic region. It then divides the elastic scattering factor by the total scattering factor to give the elastic incoherent scattering factor, EISF, which gives information about the motion of the molecule.

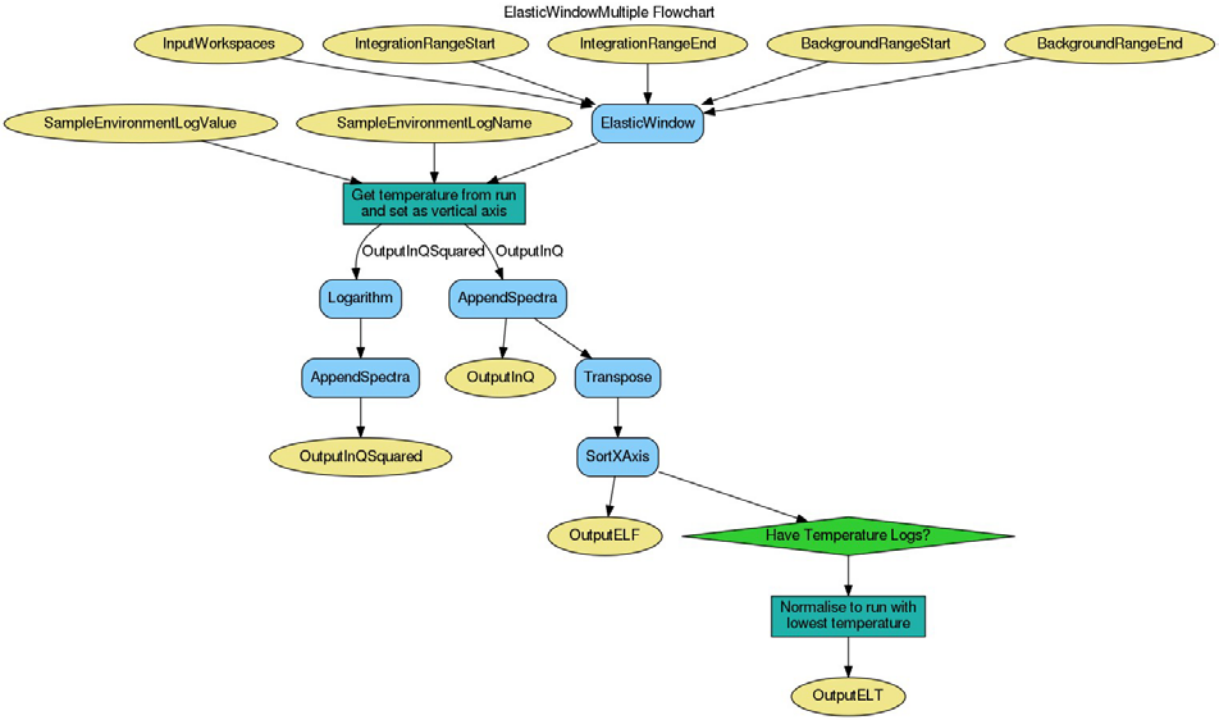

Figure 15 Workflow diagram for ElasticWindowMultiple

Finally, the user has the option to run MSDFit. The MSD stands for Mean Square Displacement. This algorithm takes the Q^2 output of the elastic region and fits it against the log of intensity. This measurement of elastic intensity as a function of temperature is a good technique for locating phase transitions and other dynamical changes.

8.2 SofQWMomentsScan

The SofQWMomentsScan algorithm focusses mainly on reducing raw data to calculate and investigate the scattering function, S(**Q**, ω). This scattering function contains details of the physical properties of the system and gives the probabilities that the scattering event changes the momentum, **Q**, and the energy transfer, ω.

As with EnergyWindowScan, the first step in SofQWMomentsScan is to reduce the raw time-of-flight data to data with units of energy transfer. Then, the algorithm SofQW converts this workspace to one with units of **Q** and ω. This is hard-coded to use the Normalised Polygon method, as mentioned in the Data Reduction section of this report.

Finally, the algorithm SofQWMoments, Figure 17 is run. This algorithm takes the output of SofQW and calculates the moment M_n for the scattering function S(**Q**, ω) for all n =0 to 4.

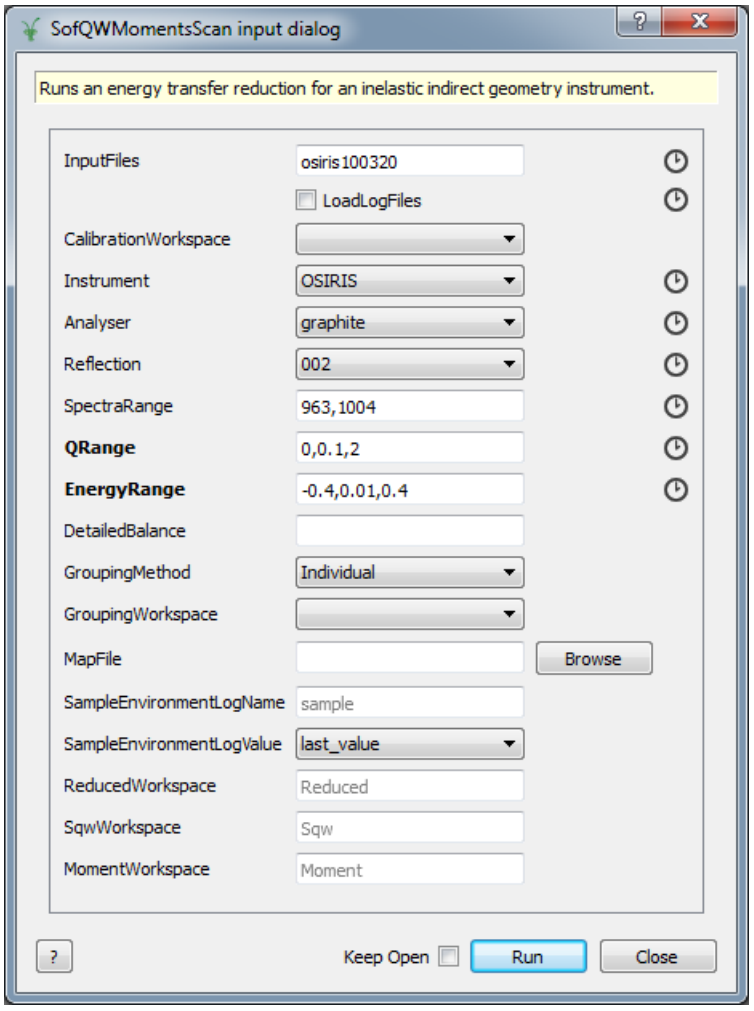

Figure 16 A GUI for SofQWMomentsScan automatically generated by the MANTID framework

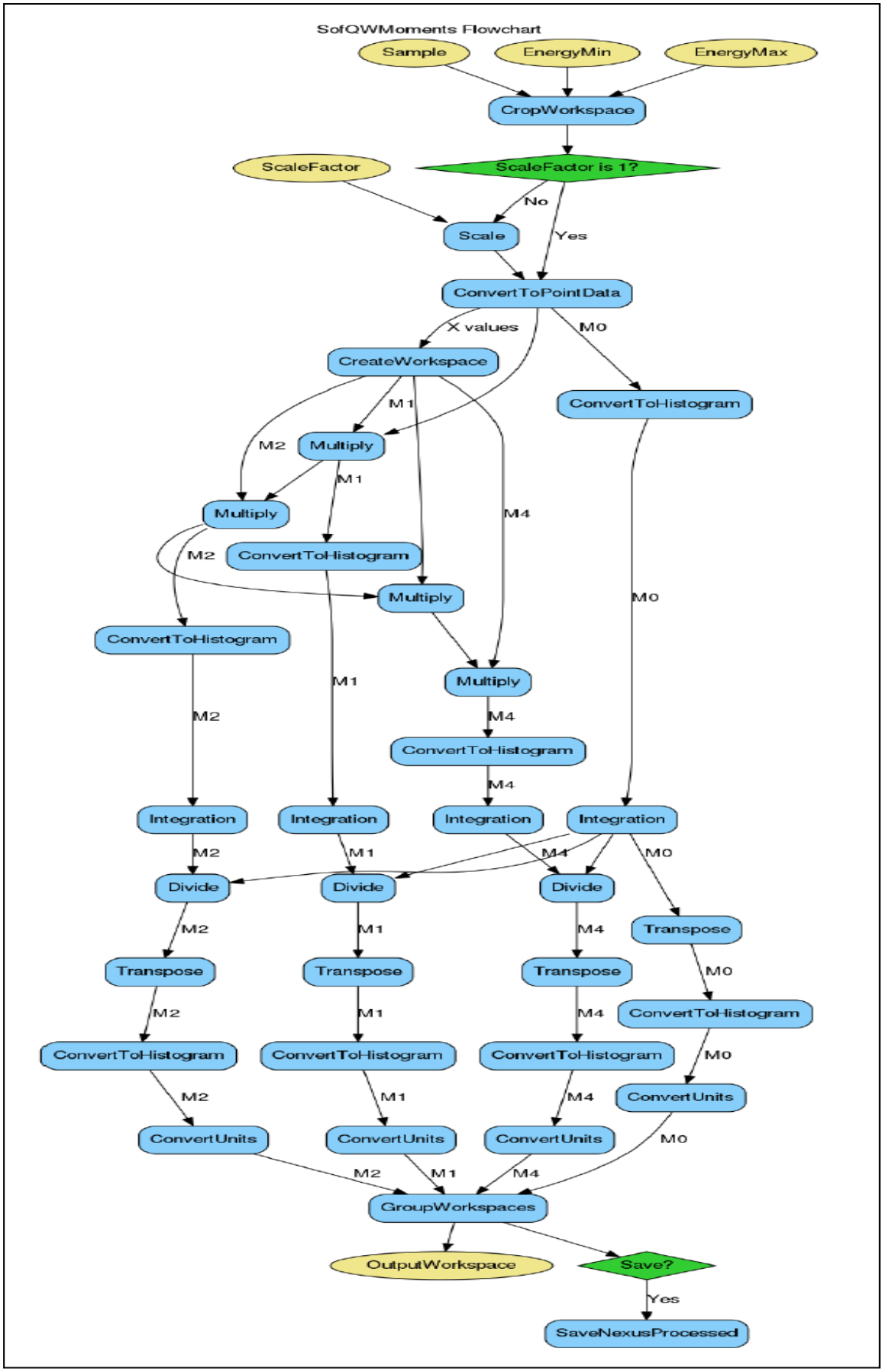

Figure 17.Workflow diagram for SofQWMoments

8.3 IndirectDiffScan

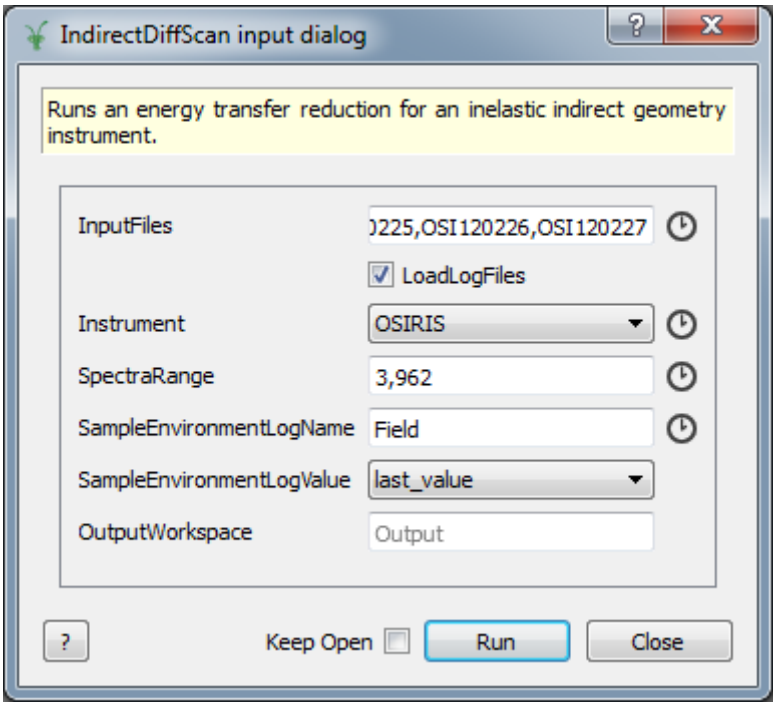

Figure 18 A GUI for IndirectDiffScan automatically generated by the MANTID framework

This algorithm is essentially a wrapper for the pre-existing ISISIndirectDiffractionReduction algorithm, Figure 19.

Although the IRIS, OSIRIS and VESUVIO instruments are mainly used for INS and QENS experiments, they also have backscattering detectors that allow for rudimentary diffraction data to be recorded at the same time. This allows users to find Bragg peaks, which can be used to simplify finding the structure of the sample.

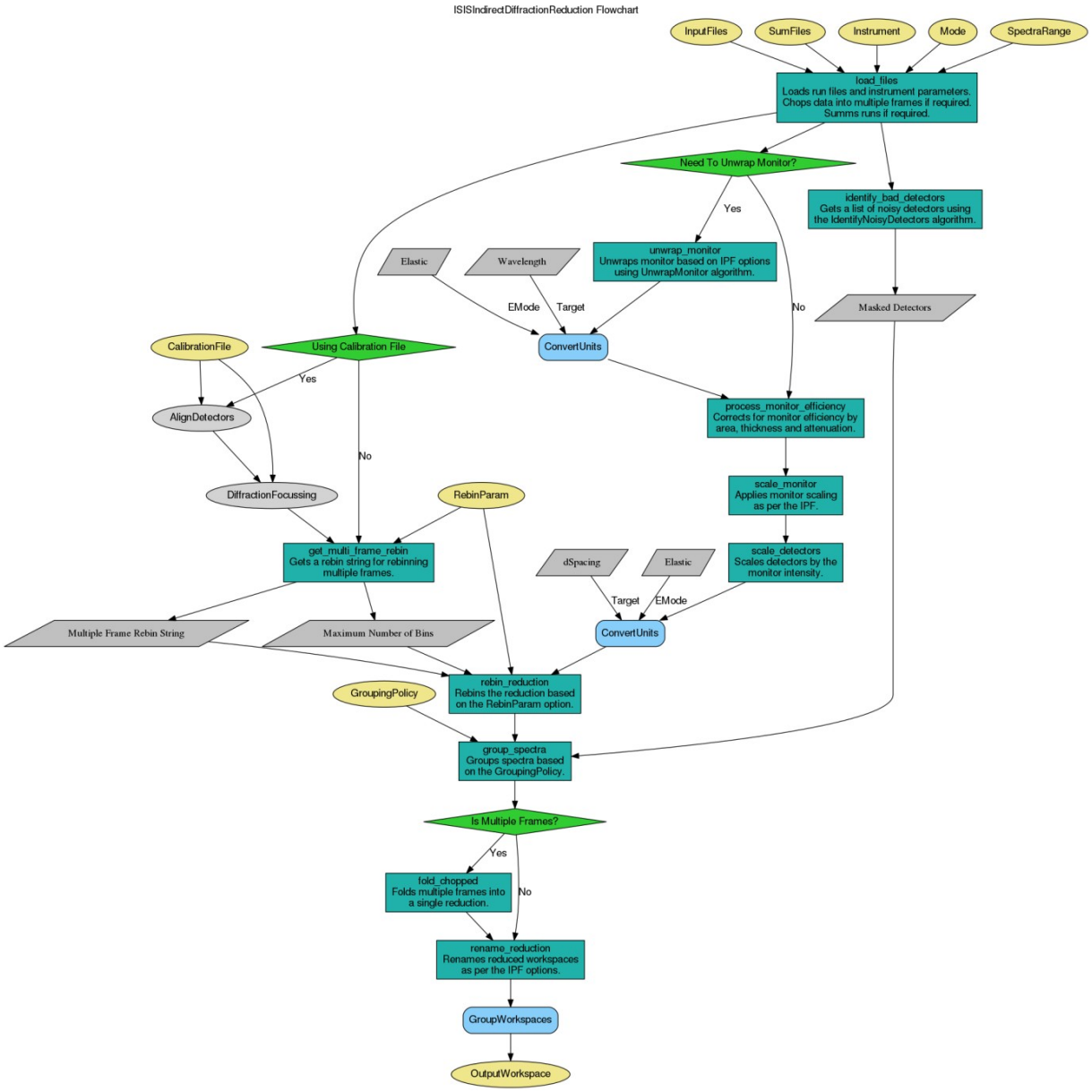

Figure 19 Workflow diagram for ISISIndirectDiffractionReduction

IndirectDiffScan is intended for use with the IRIS and OSIRIS instruments and many of the ISISIndirectDiffractionReduction parameters are set to optimal defaults. Table 2 shows all the default parameters.

Parameter	Value	Description
Container files	''	No container files are used
Vanadium files	''	No vanadium files are used
Sum files	False	Files are not summed and are run individually
Mode	diffspec	Only runs diffspec mode – simple reduction and no need for vanadium files.
Grouping method	All	All detectors are grouped together for each run.

Table 2 The default parameters and their values within IndirectDiffScan

IndirectDiffScan reduces the raw time-of-flight data and then converts it into d-spacing in Å. The data is then stored in a single workspace as a function of temperature, as shown in Figure 21Figure 20. If the temperature is not in the logs for that run, it is stored as a function of the run number.

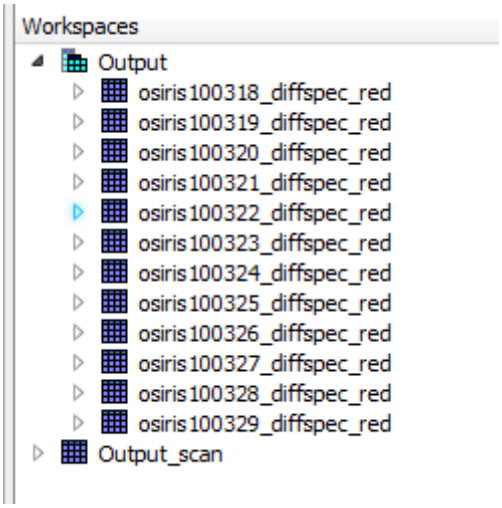

Figure 20 The output workspaces resulting from IndirectDiffScan

Output_scan

Y values | X values | Errors

	77 2.36426Å	78 2.36489Å	79 2.36552Å	80 2.36615Å	81 2.36678Å	2.3
0 200.021K	13.4305	13.3599	13.5749	13.1976	13.1477	12.3052
1 219.959K	14.3593	13.8477	13.2706	12.9648	13.1521	13.2691
2 230.013K	15.6387	15.4681	14.4934	14.8739	13.9964	13.1478
3 240.007K	16.0806	15.5334	15.0538	14.2172	13.6359	13.8799
4 250.002K	16.5861	15.9082	15.1536	14.5779	14.5952	14.1577
5 260.028K	16.4101	15.5475	15.9039	15.8656	14.976	14.6416
6 269.955K	17.0237	16.8729	16.2052	16.195	16.0001	15.5232
7 279.983K	17.91	17.6045	16.3228	15.6427	15.9525	15.4915
8 290.01K	18.054	17.7507	17.1528	17.2813	16.5268	16.3796
9 300.038K	18.3364	17.9369	17.3695	16.9189	17.1051	16.9685
10 300.038K	19.2815	18.2004	17.8131	17.2454	16.8338	16.9887
11 309.966K	18.2507	17.9995	17.5625	17.6223	16.6942	16.6639

Figure 21 Data in output_scan as shown above. Each ws in the output group workspace "output" has temperature extracted and is stored as a function of that temperature.

8.4 IndirectSampleChanger

This algorithm is similar to 8.1 EnergyWindowScan.

IRIS and OSIRIS have recently been upgraded to use a sample changer, which allows for multiple samples to be placed in the instrument at once, and for each temperature, the samples are rotated. This allows for data to be obtained for up to 3 samples without having to change the temperature for each sample individually. The sample changer is shown in Figure 22.

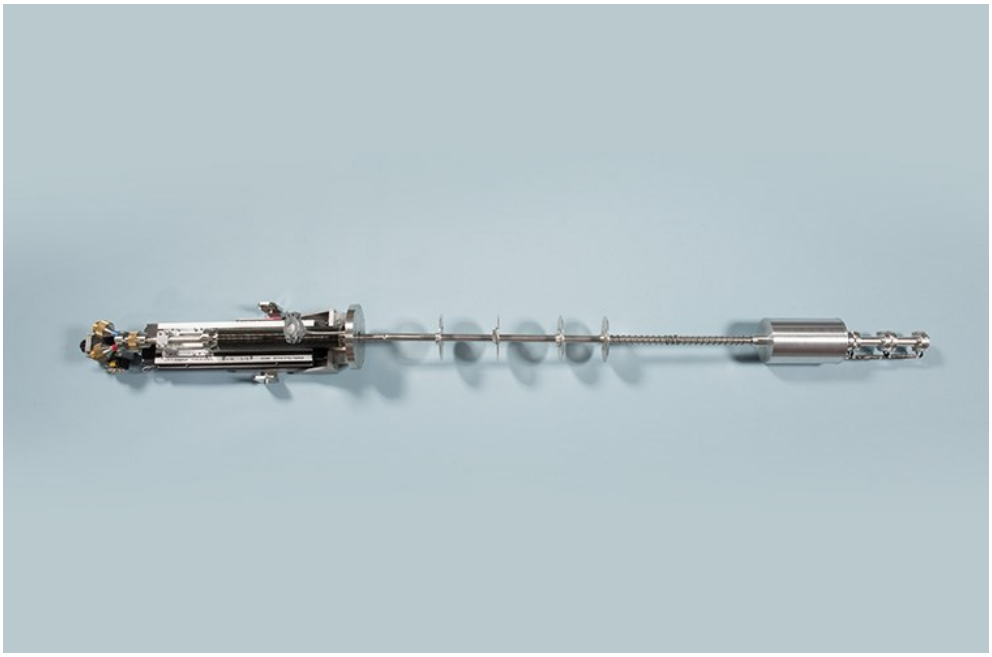

Figure 22 The three stage sample changer for use of the IRIS instrument. Three samples can be inserted vertically and the stick can be moved so each sample is incident to the beam in turn. Picture taken from http://www.isis.stfc.ac.uk/instruments/iris/sample-environment/iris-three-position-sample-changer15987.html

Input run numbers are associated with certain general temperature points. The algorithm used the position of the sample, which is stored in the sample logs, to determine which specific temperature from the logs should be used for EnergyWindowScan. This is repeated for the number of different materials in the sample changer.

For example, if there are 2 different materials in the sample changer, at 3 different temperatures 10K, 20K and 300K. The algorithm would execute 2 scans, one for each sample: The first scan would be the 1st, 3rd and 5th runs combined. The second scan would be the 2nd, 4th, and 6th runs combined.

The names of the sample position have been hard coded into the algorithm. In the sample logs, it is expected that an entry named “Position” is supplied with the current position of the sample on the sample changer, ie 0, 1 or 2. This is then used to read the temperature from the sample logs, taken from either the log named “Bot_Can_Top”, “Middle_Can_Top”, or “Top_Can_Top” where position 0 is the bottom, and so forth.

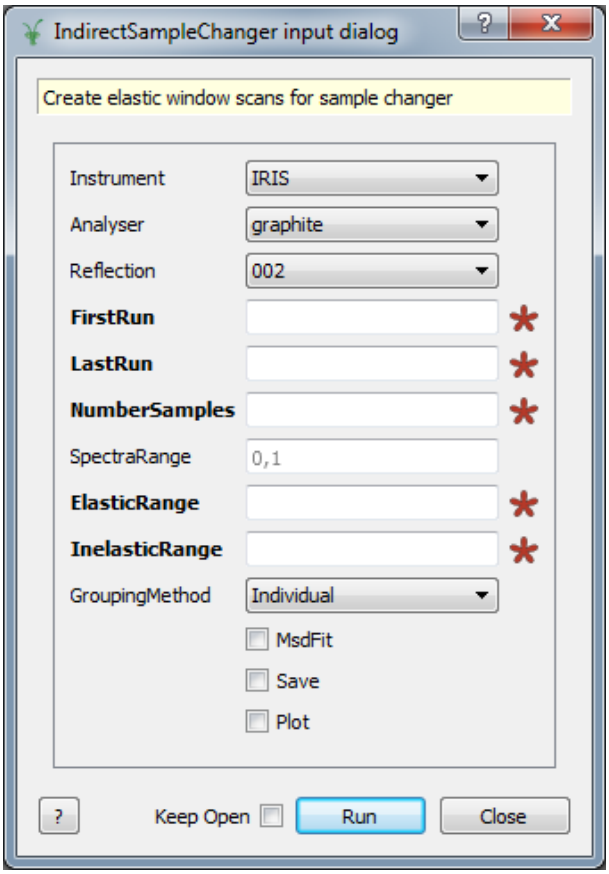

Figure 23 A GUI for IndirectSampleChanger automatically generated by the MANTID framework

9. The Future

Over the last year of development, in addition to improvements to the existing code base, there have been a couple of major changes and introduction of large sections of new code. These changes and additions have laid the groundwork for further enhancements, and are described below.

9.1 QuickRun interface

As mentioned previously, there have been several new algorithms designed for a quicker and easier workflow. At the moment, these algorithms are accessed through a python script or a small, automatically generated interface, as shown in Figure 14Figure 16Figure 18Figure 23.

To make the algorithms easier for users to access, additional GUIs should be written in the future. Any default options could be provided and updated automatically when the instrument is changed, for example. The GUIs should present a layout similar to the existing indirect interfaces, such that all options are laid out in a logical, streamlined manner.

Mockups of the menus and tabs have been created using the Balsamiq software suite [https://balsamiq.com]. They are shown below:

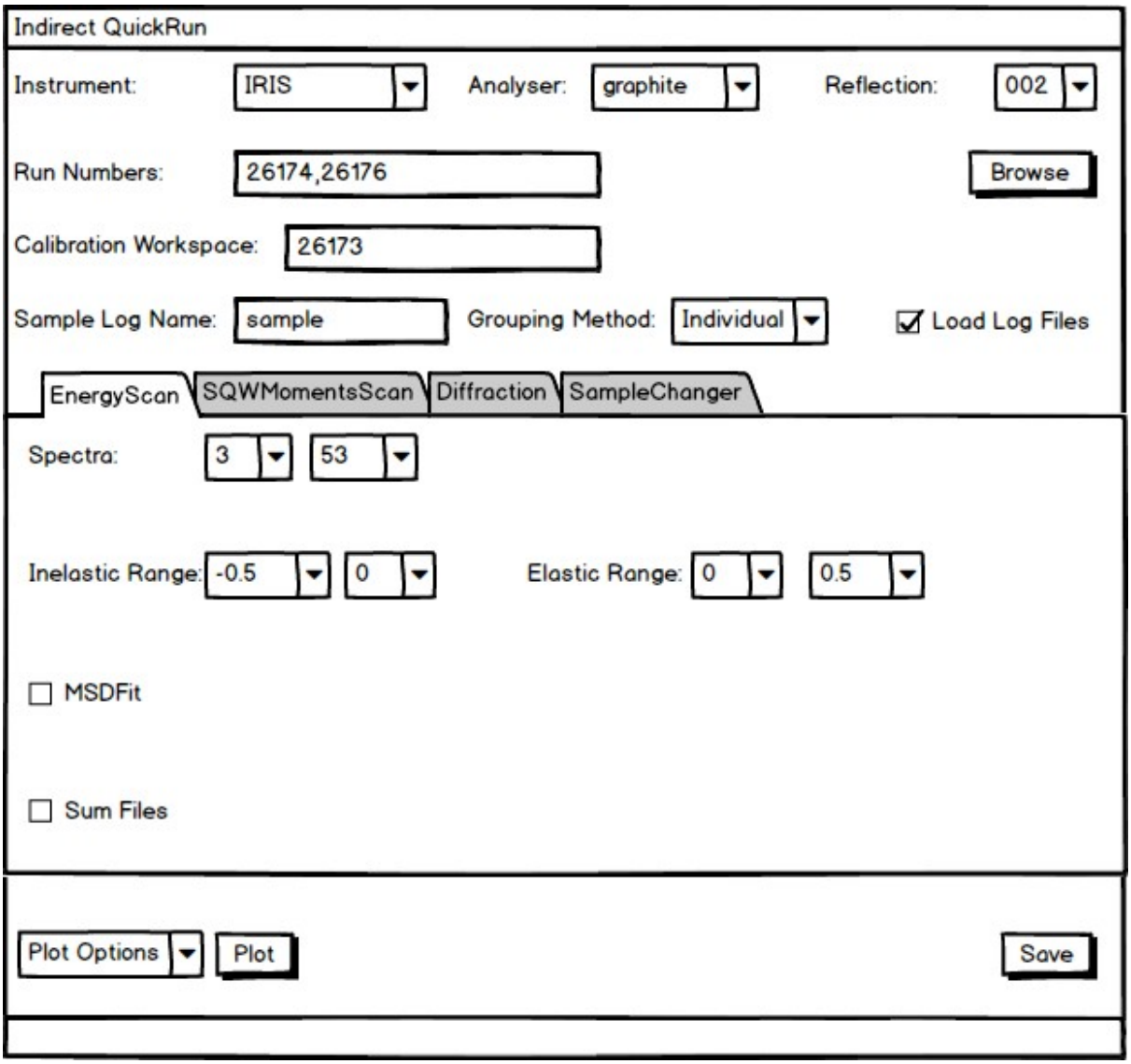

Figure 24 Mockup GUI for the EnergyScan tab in the QuickRun interface

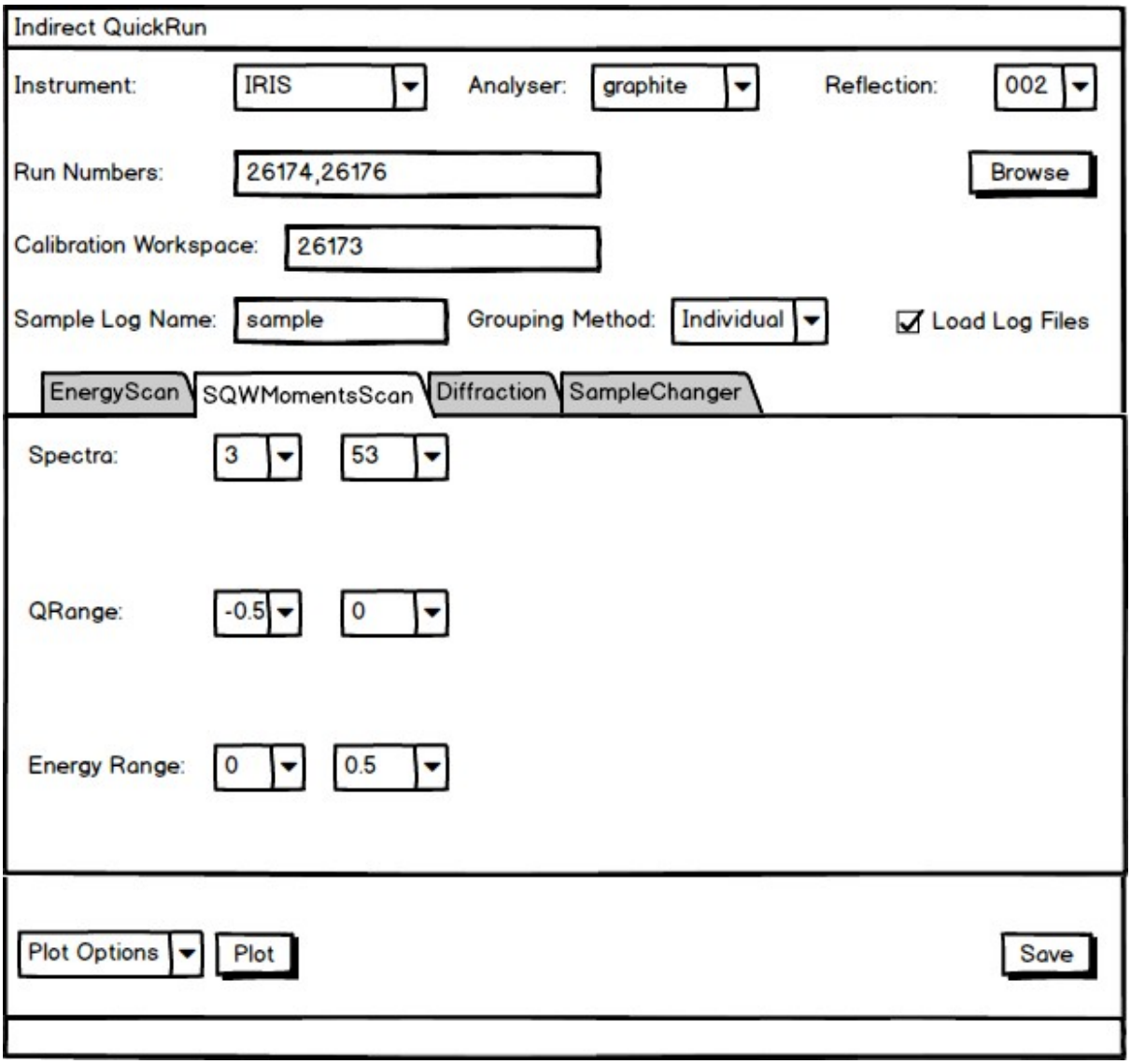

Figure 25 Mockup GUI for the SofQWMomentsScan tab in the QuickRun interface

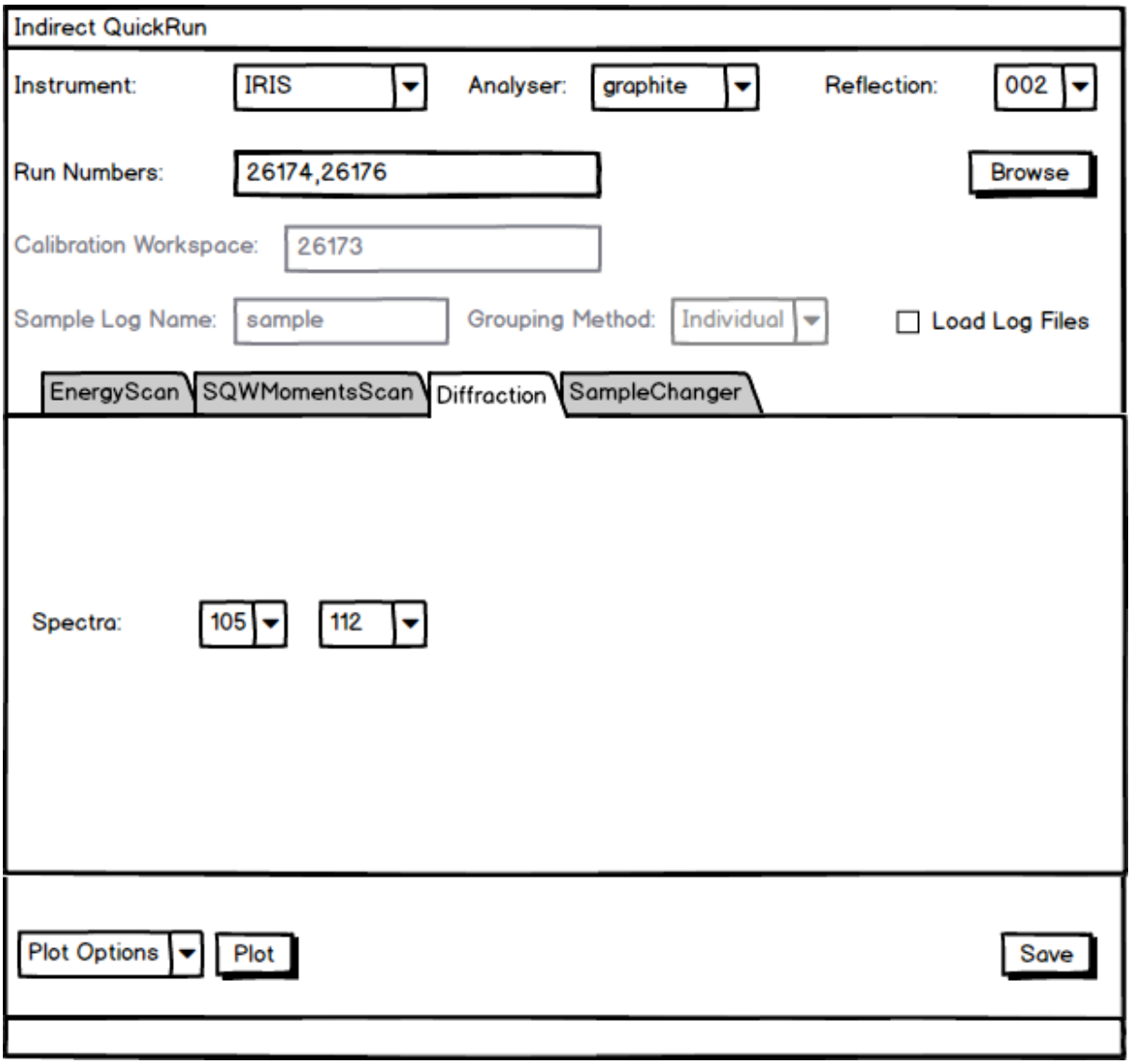

Figure 26 Mockup GUI for the Diffraction tab in the QuickRun interface

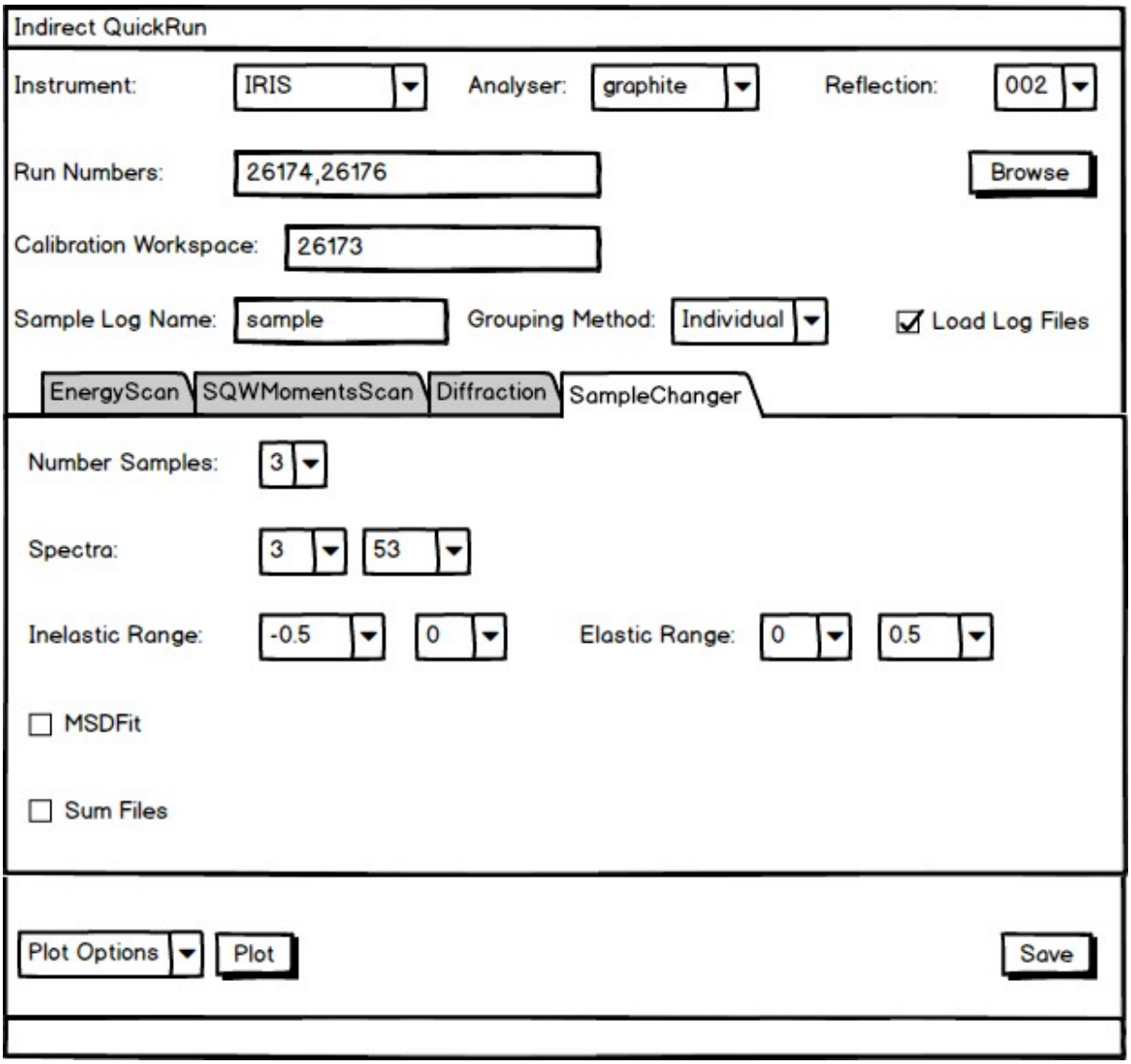

Figure 27 Mockup GUI for the SampleChanger tab in the QuickRun interface

As shown in the mockups, many of the parameters are common to all of the algorithms. These parameters are separated to the top of the interface. This enables users to be able to move the parameters between algorithms without changing anything.

In the MANTID framework, most of the user interfaces are written in C++. This creates optimised and efficient code; however, due to technical limitations, it is hard to write tests for the interfaces themselves. This means that the only way to find bugs or errors is by testing the interface manually, which can be error prone.

The first stage of manual testing is performed by the developers whenever an interface is changed. However, this often means that only a small area of the interface is tested at a time, and does not check how the change may impact the whole interface. Also, developers typically use the same default data set to run tests, so the interface may not be able to catch any flaws or quirks in data. Another factor inhibiting how the developers can efficiently test interfaces is that they might not understand or know how the users and scientists typically use the interface.

The second stage of manual testing is during the beta testing period before the release. In this period, both developers and scientists test the software extensively. This is efficient in catching bugs, as scientists can run through their workflows and use any data available. However, this can mean that not everything is caught if they are only testing what they use each day – a particular button may never be clicked. Developers also test interfaces during beta testing, however, this is often to "break" the interface, for example entering a word where there should be a number. This is good for catching bugs but doesn't necessarily test the functionality of the interface. Finally, beta testing is very laborious and to ensure testing is thorough it may have to be repeated. This can be tedious and sections may be accidentally missed. Scientists may be too busy to participate in beta testing and will typically only test the areas relating to their workflow and not so much the general framework. Also, they might not know about new features or the best way to test them.

The new Indirect QuickRun will ideally be written in Python using the PyQt library [https://wiki.python.org/moin/PyQt]. This library provides bindings for the Qt framework, which is the framework the MANTID project uses to create GUIs. This means that the design process remains the same. A GUI can be created using QtDesigner, which is demonstrated in Figure 28 and saved as an XML text file.

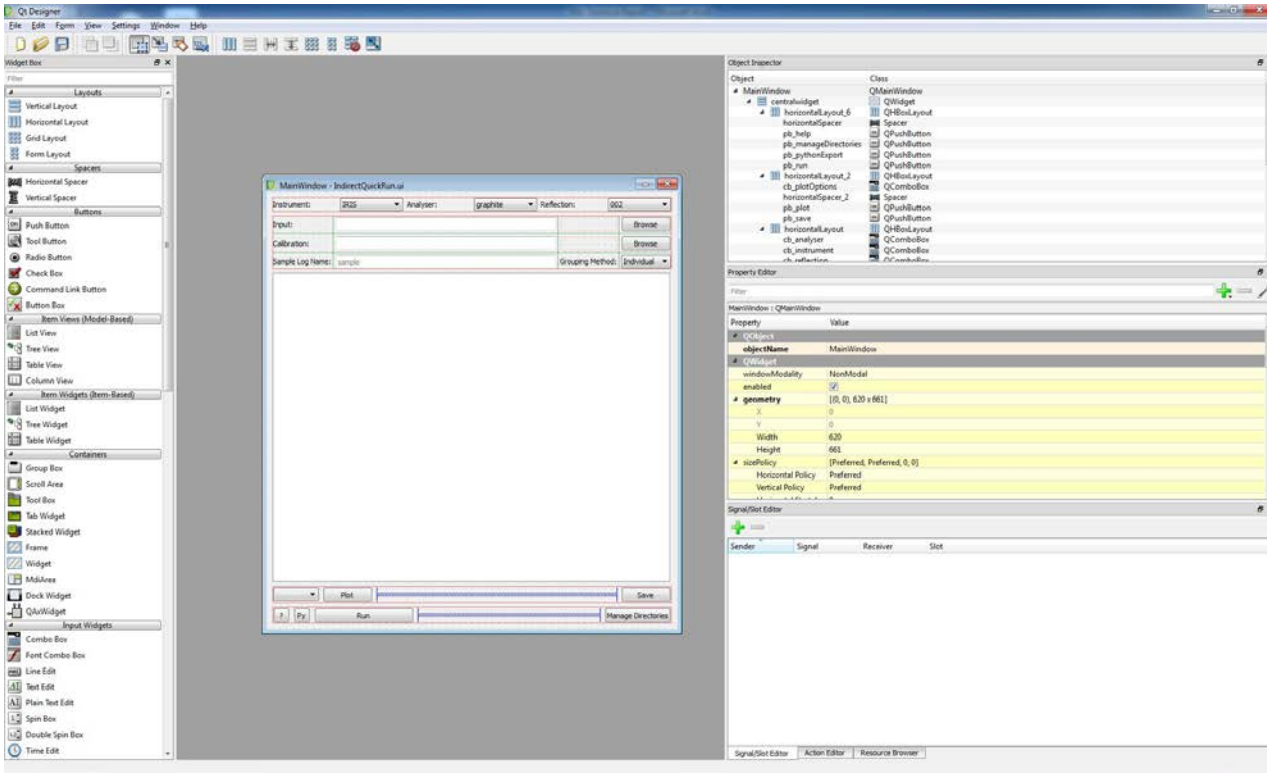

Figure 28 Initial quickrun interface design in Qt Designer

This file is read in by the Python code and all widgets are generated automatically. This allows for new additions and changes to the visual appearance of the interface to be made easily and without significant changes to the Python code.

The main difference between these planned interfaces and the current GUIs in the MANTID framework is that the Indirect QuickRun interfaces should be written in a Model-View-Presenter (MVP) pattern, as shown in Figure 29. What this means is that there are 3 sections to each interface.

The view is the initial Python code which contains the details of the interface itself. It reads an XML file, sets up the interface and any signals and slots. Signals and slots are what the Qt framework calls interactions between widgets and pieces of code. For example, a signal could be when a button is pushed and the slot could be to multiply an input number by 10 and display that number. The view also collects data from the interface, such as parameters needed for the algorithm.

In the middle is the presenter. It reads data from the view and manipulates it via the methods described in the model. This would include any validation steps. In the example, the presenter could ask the view for the input number and check it is positive. It calls the method for multiplying an input number by 10 from the model and returns the result to the view.

Finally, the model which describes the underlying mechanisms and methods of the interface. This could be the algorithm the interface calls or a simple wrapper to that algorithm with any necessary additions. For example, it could contain a simple function for multiplying a given number by 10.

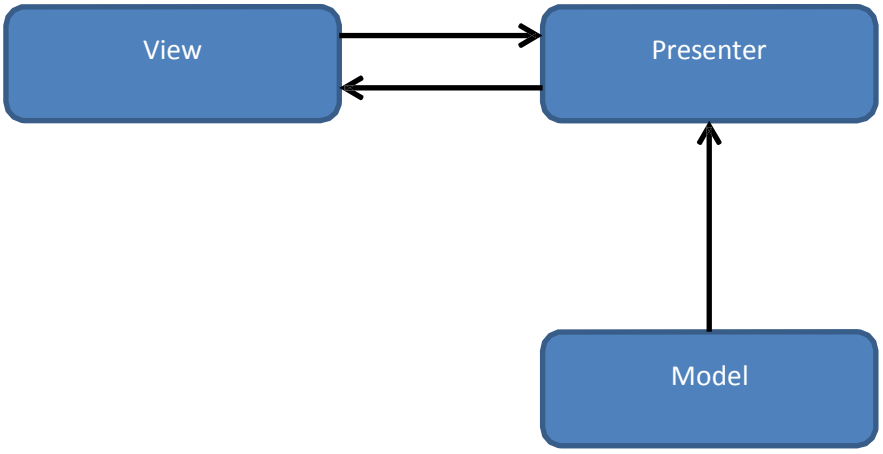

Figure 29 Basic MVP design pattern

This MVP design pattern is ideal for writing automated tests. The model contains data and functions that are separate to the rest of the design, so this can be tested using standard unit tests. To test the actual design pattern, a mock of the view and model can be created. A mock is something that programmers use to test complex pieces of code. It can be thought of as a simple substitute for the real thing. The underlying methods from the model don't need to be called.

In addition to the benefits of the design pattern, the PyQt library includes a class called QTest. This class is used to simulate physical mouse clicks and keyboard presses. This class may be used for testing the view, however, for most automated tests, it would be better to pass values directly to the presenter, although QTest could be used to see how interfaces handle bad input if run with random values. More information and documentation for this class can be found here: http://pyqt.sourceforge.net/Docs/PyQt4/qttest.html

9.2 Python 3 support

A major change which is slowly being implemented through MANTID is the upgrade to Python 3. Much of the functionality of MANTID is written in Python, particularly the indirect inelastic algorithms. As the major release of Python during the development of the MANTID framework was Python 2, this was the supported version. Also, some of the computers used by instrument scientists and users run operating systems such as Red Hat 6, which only supports Python 2.

In 2008, Python 3 was officially released, and as of December 2016, the latest release is 3.6. Updates which may affect the code base include using the print statement as a function, ranges returning an iterator object rather than a list, dictionary iterations. More changes can be found at https://docs.python.org/3/whatsnew/index.html. Since the release of Python 3, it has been announced that Python 2 will no longer be supported from 2020. Combined with the fact that the majority of Red Hat computers used at the neutron facilities have been updated to Red Hat 7, which has been updated to support Python 3, the MANTID project is being updated so all code is compatible with Python 3.

The main period over which the upgrade to Python 3 has been worked on is during the maintenance period of development. Roughly every 4 months there is a cycle of development, resulting in a new release of the MANTID software. The first three months consist of development. This is when new functionality is put in or major improvements and changes to the code base are implemented. A

month before the release deadline, the developers start testing the software and there is a beta testing period for scientists (see the QuickRun section for more information about this). After beta testing, the developers have the opportunity to fix any bugs and to begin maintenance. Maintenance includes changes to the code base that will not affect how the code behaves. This includes refactoring and restructuring of code. Also, maintenance code might not be crucial to the release, and so big changes can be done over several maintenance periods. Such is the case with the upgrade to Python 3. Every release cycle, a small amount of work is done, for example, changing all the algorithms one release, and then the associated unit tests for the next. This means that these kinds of changes get thoroughly tested before being permanently introduced and allows for many developers to work slowly rather than one person rushing through the task through before a release deadline.

9.3 MANTID 4.0

As mentioned in the introduction, the MANTID project is based on the Qt framework and the QtiPlot library. Currently the supported version is Qt4, however, there is a newer version, Qt5.
Unfortunately, QtiPlot is not updated to Qt5 and is in fact no longer supported. Also, the graphs and figures produced by QtiPlot are not officially publication quality, which has caused an issue for users. This, combined with some of the comments mentioned in previous sections of this report, such as the inability to automatically test interfaces and the update to Python 3, has prompted the senior development team to begin plans for a MANTID 4.0.

In most cases, this will not change the functionality of the MANTID framework. The plotting library and main GUI will be based on matplotlib, a popular python plotting library. This not only ensures that graphs are publication quality but also means that users can easily modify the presentation of a graph to show a style that suits them, using the matplotlib API.

10 ILL

The ILL have increased their efforts contributing to and maintaining the MANTID project. The initial routines written by Spencer Howells and the LAMPS software [19] have been built on to create a reliable workflow for data reduction of data from the IN16B instrument at ILL. This includes IndirectILLReductionQENS for multiple-file reduction for Quasi-Elastic Neutron Scattering data and IndirectILLReductionFWS for reduction of fixed-window scan data.

The developers at the ILL have also contributed to the Absorption Corrections and Paalman Pings changes mentioned in the Corrections section of this report.

10.1 GetQsInQENSDATA

A new algorithm, GetQsInQENSDATA has been added. This algorithm extracts values for momentum transfer from the vertical axis of a workspace representing the dynamic factor S(Q, E). This algorithm may be useful for future refactoring of indirect algorithms.

Bibliography

[1] *Mantid: Manipulation and Analysis Toolkit for Instrument Data.,* Mantid Project, 2013.

[2] S. Howells, V. Garcia Sakai, F. Demmel, M. T. F. Telling and F. Fernandez-Alonso, "The MODES User Guide," Rutherford Appleton Laboratory, 2010.

[3] J. Mayers, "User guide to VESUVIO data analysis," , RAL-TR-2011-003 , 2010, RAL Technical Report

[4] S Jackson, "An overview of the recent development of indirect inelastic data analysis in Mantid", RAL-TR-2014-010. 2014, RAL Technical Report; D. Nixon, "Developments in MANTID relating to indirect inelastic spectroscopy between July 2014 - July 2015," RAL-TR-2015-007, 2015, RAL Technical Report

[5] S Mukhopadhyay, "How to Use Mantid for Low Energy Inelastic Neutron Scattering Data Analysis On Indirect Geometry Instruments", RAL-TR-2014-005. 2014, RAL Technical Report ; D. Nixon, S. Howells and S. Mukhopadhyay, "An Introduction to Inelastic Neutron Scattering and MANTID at ISIS," 2015.

[6] J. Teixeira, M.-C. Bellissent-Funel, S. H. Chen and A. J. Dianoux, "Experimental determination of the nature of diffusive motions of water molecules at low temperatures," *Phys. Rev. A,* vol. 31, no. 3, pp. 1913-1917, Mar 1985.

[7] L. C. Pardo, M. Rovira-Esteva, S. Busch, M. D. Ruiz-Martin and J. L. Tamarit, "FABADA: a Fitting Algorithm for Bayesian Analysis of DAta," *Journal of Physics: Conference Series,* vol. 325, p. 012006, 2011.

[8] H. H. Paalman and C. J. Pings, "Numerical Evaluation of X-Ray Absorption Factors for Cylindrical Samples and Annular Sample Cells," *Journal of Applied Physics,* vol. 33, pp. 2635-2639, 1962.

[9] M. T. F. Telling and K. H. Andersen, "The OSIRIS User Guide," RAL-TR-2003-016 , 2008. RAL Technical Report

[10] E. Oram and L. McCann, "Simulation support and further Indirect Inelastic specific improvements within MANTID," RAL-TR-2016-019 , 2016, RAL Technical Report

[11] B. Aoun, G. Goret, E. Pellegrini and G. R. Kneller, "nMoldyn Documentation," 2014, https://www.ill.eu/nmoldyn

[12] F. Demmel and S. Mukhopadhyay, "Quasielastic neutron scattering measurements and ab initio MD-simulations on single ion motions in molten NaF," *The Journal of Chemical Physics,* vol. 144, p. 014503, 2016.

[13] A. J. Ramirez-Cuesta, "aCLIMAX 4.0.1, The new version of the software for analyzing and interpreting INS spectra," Computer Physics Communications, vol. 157, pp. 226-238, 2004.

[14] S. J. Clark, M. D. Segall, C. J. Pickard, P. J. Hasnip, M. J. Probert, K. Refson and M. C. Payne, "First principles methods using CASTEP," *Zeitschrift für Kristallographie - Crystalline Materials,* vol. 220, p. 567, 2009.

[15] J. F. Castanon, "Monte-Carlo simulations for the development of TOSCAś guide at ISIS," RAL-TR-2013-002, 2013, RAL Technical Report

[16] E. Oram, "An Overview of the development of Indirect Inelastic Data Reduction and Analysis in MANTID between July 2015 - July 2016," RAL-TR-2016-011 ,2016, RAL Technical Report

[17] J. Mayers and A. C. Evans, "Measurement of Atomic Momentum Distribution Functions by Neutron Compton Scattering; Progress of the EVS Spectrometer during 1990," Rutherford Appleton Laboratory, 1991.

[18] F Demmel et al, "IRIS & OSIRIS International Beamline Review - Report Presented to the Panel," RAL-TR-2014-014, 2014, RAL Technical Report

[19] *LAMP, the Large Array Manipulation Program,* https://www.ill.eu/instruments-support/computing-for-science/cs-software/all-software/lamp/