

Reflectometry STAP Report: Software

Andrew R. McCluskey

Data Management and Software Centre, European Spallation Source ERIC, Ole Maaløes vej 3, 2200 København, Denmark
andrew.mccluskey@ess.eu

October 4, 2021

Andrew McCluskey has now been in the role of Instrument Data Scientist (IDS) for ESTIA for approximately 9 months (started Jan 2021). Work conducted during this time, and reported in this document, has involved close collaboration between DMSC, instrument teams, and in-kind partners to develop reflectometry data processing, reduction and analysis at the European Spallation Source (ESS). In particular, attention is drawn to the close collaboration with the Amor team at the Paul Scherrer Institut (PSI) SINQ neutron source (Jochen Stahn and Artur Glavic) and the `scipp` team (Simon Heybrock, Neil Vaytet, Jan-Lukas Wynen, Owen Arnold, and Matthew Jones) on the data processing and reduction, and the `easyScience` team (Piotr Rozyczko, Simon Ward, and Andrew Sazonov) at DMSC and BornAgain team from the Scientific Computing Group at the Heinz Maier-Leibnitz Zentrum (MLZ) on the analysis work.

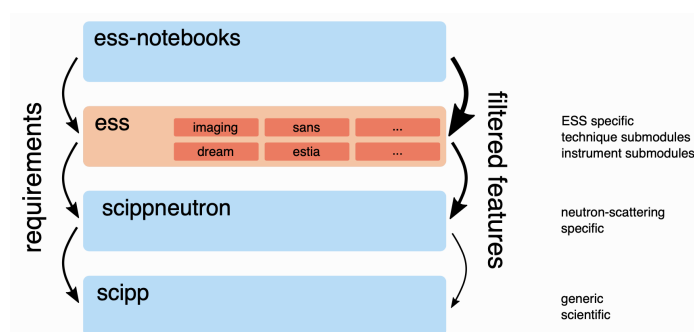


Figure 1: The software framework for reduction at ESS; from <https://scipp.github.io/ess/>.

1 Data reduction

The previous STAP report (available from indico: https://indico.ess.lu.se/event/2705/attachments/12019/21317/DMSC_STAP_Report.pdf), provided a brief introduction to the collaboration between the IDS, the `scipp` development team and the Amor instrument team. This project is building on the existing collaboration between the Amor team and ESS as a whole, where the Amor instrument is assisting with the testing for ESTIA components, including the multi-blade detector system. Since the previous STAP meeting, there have been significant developments in the area of data reduction for reflectometry in `scipp` and the organisation of the data reduction framework at ESS. Additionally, following an internal GUI workshop for data reduction and consultation with Robert Jacobs, a prototype design for a data reduction user interface is proposed.

Software framework

The framework for data reduction for ESS (Figure 1) has been defined as a result of work in 2021 on both the `scipp` package and instrument/technique-specific developments arising, in part, from the Amor collaboration mentioned above. The packages contained within this framework are all developed as open-source projects, using Github, and welcome contributions and collaboration.

The framework is defined as a series of levels, with `scipp` (<https://scipp.github.io>) sitting at the bottom of the stack, providing generic and scientific functionality, such as labelled data arrays and uncertainty propagation, and enabling high-throughput from the C++ implementation. The `scippneutron` module (<https://scipp.github.io/scippneutron/>) builds on `scipp` to provide neutron scattering-specific, but importantly, facility-independent functionality, including unit conversions (i.e. time-of-flight to wavelength) and live- and written-data parsing. Above, the `scippneutron` package is the `ess` specific code (<https://scipp.github.io/ess/>), this code contains instrument- and technique-specific submodules for the ESS, enabling the sharing of software between techniques (for example, the `ess.amor` reduction uses a large amount of general reflectometry code contained in

the `ess.reflectometry` submodule). The `ess` modules include online, API-level documentation which improves the ease-of-use of the reduction code by expert users and instrument staff (note that the `scipp` and `scipp-neutron` modules also have online documentation). Finally, to further enable ease-of-use and transparency in neutron scattering data reduction, the `ess-notebooks` repository has been created (<https://scipp.github.io/ess-notebooks/>). This resource is designed to provide examples of reduction workflows, as Jupyter Notebooks, for given instruments. We hope that this will become valuable in future for the user community to scrutinise and better understand the reduction processes being utilised. Furthermore, the availability of this clear documentation will empower users, and instrument scientists, to investigate the raw experimental data and subject this to appropriate data analysis.

Amor reduction

A workflow for the reduction of Amor data, in the unique Selene-guided convergent beam mode, which will also be available at ESTIA, has been produced. This workflow allows written raw detector data (as a NeXus file) to be read into `scipp` and the appropriate reduction to be performed, including accurate uncertainty propagation and resolution handling. This workflow, in its current format, is described completely in the ESS notebook webpages (https://scipp.github.io/ess-notebooks/reflectometry/amor_reduction_detailed.html), and the reduced data that is produced is shown in Figure 2. The presentation of this complete workflow aims to enable complete transparency in reduction practices using `scipp`, in addition to, facilitating user understanding of the reduction workflows.

The complete reduction process has been implemented as a series of three function calls in the `ess.amor` submodule; the first to read in the sample data, the second for the reference data, and the final for the normalisation process between these [1]. In addition to this, it is possible to introduce different masking processes and visualise wavelength-angle maps using single-line commands. An example of a notebook that shows this in action can be found in the ESS notebooks webpage (https://scipp.github.io/ess-notebooks/reflectometry/amor_reduction.html). These notebooks have been shared with the team at PSI and may be used in the reduction of data collected at the Amor instrument.

This work to produce reduced reflectometry data from the Amor Selene-guided measurements is a continually developed piece of code. Following the generation of the documentation discussed above, feedback has been received from Jochen Stahn and Artur Glavic from PSI on improvements that can be made to the process to ensure the highest

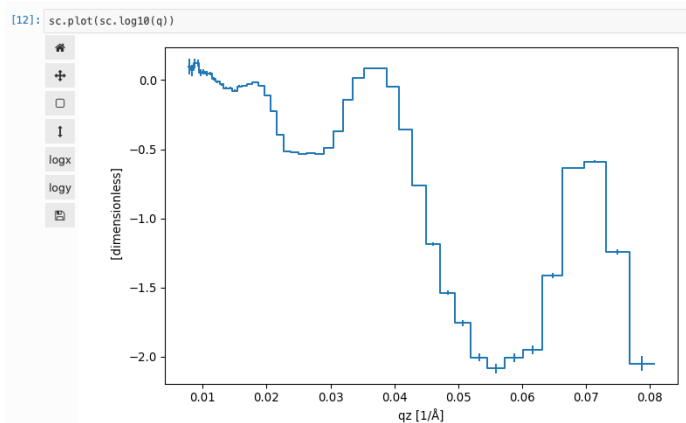


Figure 2: Reflectometry data of a NiTi-multilayer system, collected at the Amor instrument, reduced using the `ess scipp` workflow (for this data the exact measurement parameters, specifically beam width and sample size, were not known).

quality data reduction, such as histogramming on a detector pixel level to ensure that pixel discontinuities are accounted for in the normalisation step. Additionally, the capability of the `scipp` reduction code can enable novel reduction methodologies and there is interest to use the wavelength-theta map functionality to ensure that sample alignment can be accurately determined. Furthermore, this development represents just a single operating mode for the ESTIA instrument, there have been discussions with Alessandra Luchini to ensure that all operating modes for the ESTIA instrument are fully supported.

Live data reduction

In addition to the reduction of files that have already been written, it is important that reduced data can be visualised as it is collected. This functionality is completely supported in the current implementation of the reduction workflows. Furthermore, functionality has been developed in `scipp` to produce live-updating plots that can be integrated into a Jupyter Notebook interface. Currently, this aspect of the project has not been rolled out for use at the Amor instrument, as it is still to be tested on ESS infrastructure at the DMSC in København. We hope to have this live data reduction visualisation to be tested before Christmas on ESS hardware and shared with the Amor team soon after.

User interface

It was mentioned in the early 2021 STAP report that there was a plan to host a GUI workshop to gain expectations of the user-facing aspects related to data reduction. An internal workshop was held in August 2021 for ESS and in-kind instrument teams, following this workshop Robert Ja-

cobs was consulted from the reflectometry STAP for additional input on the design of a data reduction interface. The ambition is that this user interface will be made available to all users and will handle the vast majority of data reduction at ESTIA (and FREIA), with the local contact ensuring that the correct reduction steps are being used. Users with more custom reduction needs will be expected to collaborate with the Instrument Scientist and Instrument Data Scientist to develop necessary Jupyter Notebooks/Python scripts, building on the functionality made available in the `scipp`, `scippneutron`, and `ess` modules.

Figure 3 shows a design prototype for the reflectometry reduction interface, it is important to be clear that this is not finalised and we welcome the STAPs feedback on this design. This interface will be visualised using a web browser and utilise compute based at the DMSC, allowing users to remotely access the data and reduce as necessary from their home university. The interface will consist of three panes; a SciCat search pane, a tabular result pane and a plotting pane.

Before the start of data collection, the local contact will prepare a data reduction workflow with input from the user as necessary, where multiple reduction workflows may be required these will all be available. This workflow will have default parameters that are being used in the live reduction and will be editable in the table discussed below. Following the completion of the data collection, the reduction workflow will be run with the default parameters and the reduced data written to disk.

The top of the reduction window will have a pane with SciCat search functionality. SciCat (<https://scicatproject.github.io>) is the ESS data catalogue and will contain both raw and reduced reflectometry data. The search pane will enable the user to search across the whole ESS catalogue (that they have access to), allowing the comparison of data from both the current beamtime and previous beamtimes. In addition to searching by proposal ID, it will also be possible to search by any metadata that is associated with a given experiment, for more discussion of catalogued metadata see Section 3.

Once the SciCat search is performed, the table pane will be populated with the results. This table is designed to be flexible in the information that can be presented, i.e. right-clicking will give information of available columns that can be selected or unselected and these values can be modified as appropriate. To re-reduce data with different parameters from those defined at the start, the Run button in the Reduce column is selected, the job will then be submitted to a job queue and once complete the plotting will be re-enabled, by checking the relevant checkbox. Re-reduced data (following changes to metadata/parameters in the table) will not overwrite the original reduction but create a new reduced file. All reduced data will be linked with the original data by a hierarchy link in the SciCat database.

When the plot column checkbox is checked for a given

dataset, it will be shown in the plot pane. This interface will be tabbed to allow different visualisations to be investigated, and similar to the table these tabs will be flexible, and the plotting of different curves will be facilitated by the ability to offset different plots. In addition to presenting collected data, there will be a “show live” check box, which will subject the data currently being collected to a given reduction workflow and present this in the plot view.

This is not the only interface that can be created, for example, there is a clear use-case to present the ratio between two datasets or a waterfall plot of a kinetic run of measurements. The tools that will be used and developed to produce the interface shown in Figure 3 are flexible and easy to use, meaning that the production of custom interfaces should be straightforward given some clear specifications.

2 Data analysis

The Scientific Computing Group at Heinz Maier-Leibnitz Zentrum (MLZ) has extended the BornAgain framework to also handle specular reflectometry as an in-kind contribution to ESS. The development of this functionality has been slowed due to a high turnover of staff. While this is unfortunate, though it shows that the BornAgain framework can manage the setback of losing key developers, because it is institutionally backed. Indeed, Forschungszentrum Jülich (FZJ) has committed to continue the development in the years to come. We look forward to the in-kind relationship between ESS and the BornAgain team transitioning to a close collaboration, with FZJ carrying the burden of maintaining the BornAgain framework, details of this collaboration can be found below.

In parallel to the BornAgain extension, ESS has initiated the easyScience project (<https://easyscience.software>). This was initially an attempt to address the lack of user-friendly and sustainable analysis software for diffraction, in collaboration with Institut Laue-Langevin (ILL). This resulted in the easyDiffraction software (<https://easydiffraction.org>) which is capable of using any of CrysFML, CrysPy, or GSAS as a diffraction calculation engine. For the user, this means that the same graphical interface and Python scripting interface can be used to run any of these engines.

The easyScience project was extended to reflectometry as it is designed such that it enables rapid development of new applications. Early testing of this involved the development of a reflectometry mockup that was used in discussion regarding a graphical user interface for specular reflectometry with the BornAgain team. Further development, by the Instrument Data Scientist and the easyScience team, in this area has led to easyReflectometry, the second member of the easyScience family. Like easyDiffraction, easyReflectometry can link to a range of different reflectometry calculation engines; `bornagain` [2] (with help from the BornAgain

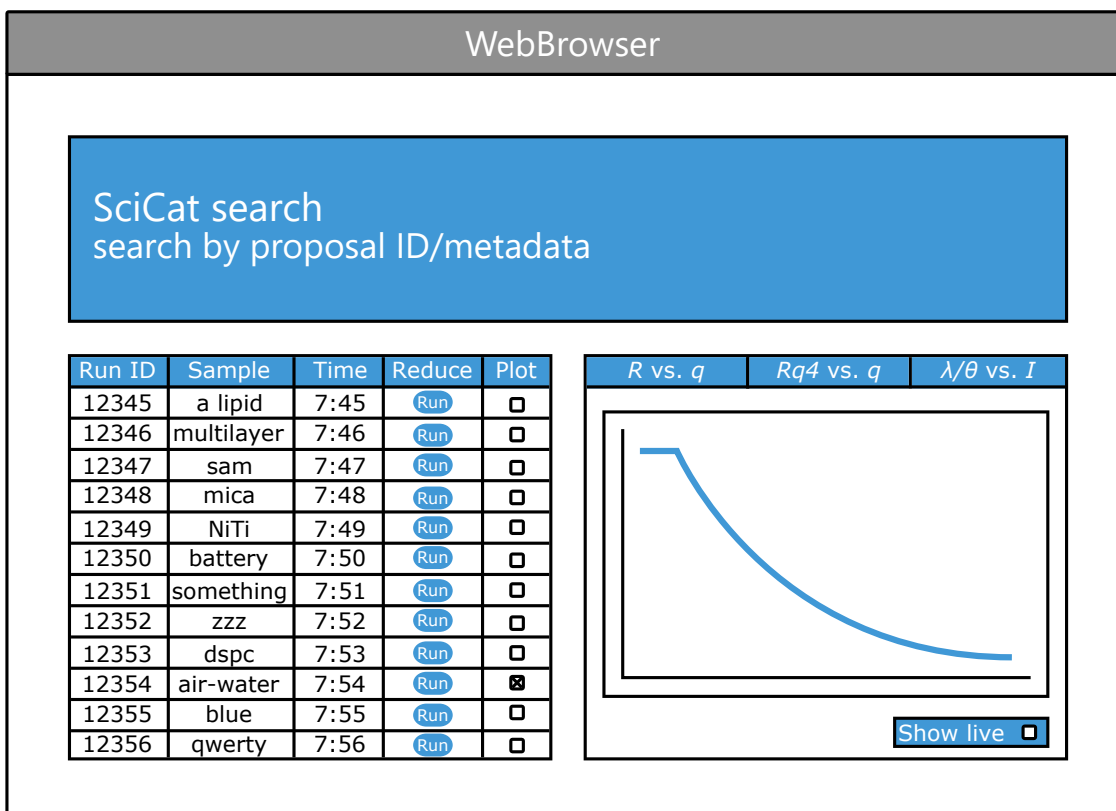


Figure 3: The proposed design for the reflectometry reduction interface.

team), `refnx` [3], and `refl1d` [4] have all been interfaced so far. The architecture of the easyScience framework allows any reflectometry calculator to be included with minimal effort (i.e. adding `refl1d` took approximately one day). There are already plans in place to add other reflectometry calculators, such as GenX.

The feedback from the reflectometry community has been positive so far (including a talk at the ORSO AGM which is available on YouTube <https://www.youtube.com/watch?v=t3GdAi8hq4k>), including interest in collaboration from developers of other reflectometry packages. In collaboration with Malmö University and ISIS, we have applied for funding from the Swedish Research Council to use easyReflectometry as the graphical interface to the new RAT package (the successor to Rascal [5]), extending the current capability to handle life science problems. Furthermore, we are in close communication with the development team of `refl1d`, who are considering the possibility of using easyReflectometry as a model builder interface, and the model description file would be used as an input in `refl1d`.

With easyReflectometry and the easyScience framework, ESS has chosen to focus on users experience and interoperability and to a large extent relying on existing calculators where possible. The easyScience framework has demonstrated that it enables rapid development of new

analysis applications (in just a single six week, two-person development sprint, it was possible to get easyReflectometry from idea to first release). However, it remains to be seen what the maintenance cost of developing the specular reflectometry user interface in-house will be. Given that the user interface shares its basic framework, i.e. the easyScience framework, with that of easyDiffraction and other future analysis applications it is foreseen that the additional cost of maintenance will be rather modest.

The sample model builder in easyReflectometry (Figure 4) is designed around the concept of flexible “items” that the user can select from. The use of these items will allow easyReflectometry to harness domain-specific descriptions of experimental systems, including data obtained from complementary techniques. Currently implemented item types in the latest version of easyReflectometry are `Multi-layer` (to which a series of layers defined in terms of scattering length densities, layer thicknesses and roughnesses may be added) and `Repeating Multi-layer` (which allowed a given number of repetitions to be defined for the multi-layer). Planned future item types include those for surfactants or proteins embedded in lipid bilayers and those with the ability to decouple magnetic and nuclear structure or model some dopant gradient. Additionally, it is planned that in-future support for custom items types will be introduced.

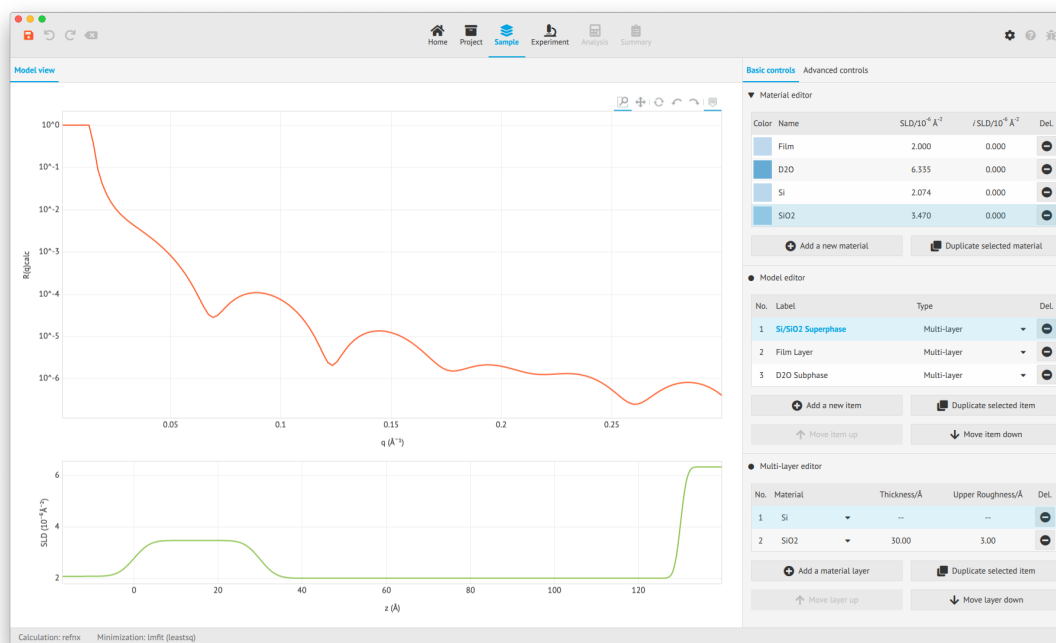


Figure 4: A screenshot of the easyReflectometry software, showing the current Sample tab, where multi-layers can be defined based on scattering length density, thickness and roughness.

As mentioned previously, the use of the easyScience framework allows the reflectometry calculation to be performed by a range of reflectometry engines. In the analysis pane of the user interface, it is easy to switch between these different calculation engines. Furthermore, the minimisation method that is used in the analysis process is also flexible with the bumps [6], `lmfit` [7] and `DFO_LS` [8] packages all supported currently. In addition to offering the user broad flexibility, this functionality also reduces the development time for easyReflectometry as there is no need to develop new reflectometry calculation engines or optimisation routines.

The current development plan for easyReflectometry includes, but is not limited to, the following items:

- multiple isotopic contrast support in graphical user interface;
- functionality for ellipsometry analysis (with support from Tilväkstverket SREss3 funding);
- support for polarised and magnetic neutron reflectometry analysis;
- output of a standardised model description;
- integration of the RAT analysis package (with support from ISIS);
- Bayesian sampling methods; and
- general custom item types support.

The collaboration between the BornAgain team and ESS will focus on two areas; the further development of the BornAgain specular reflectometry engine (for use by both easyReflectometry and BornAgain) and ensuring that BornAgain can be used for co-refinement of specular reflectometry/off-specular reflectometry/grazing-incidence small-angle scattering at ESS instruments. Some specific areas of collaboration will include:

- improvements to the BornAgain Python interface (including making BornAgain installable via `pip`) to improve flexibility and efficiency in interfacing with other packages;
- development of new physical models and analysis methodologies, such as those to describe structural roughness in the presence of magnetisation;
- enabling non-Gaussian resolution functions in the BornAgain Python interface; and
- producing BornAgain instrument descriptions for ESTIA and FREIA.

The BornAgain team is continuing its development and is in the process of redesigning its user interface to improve the user experience. In the process, the team has taken strong inspiration from easyReflectometry and plan to have a single graphical user interface for specular reflectometry, off-specular reflectometry, and GISANS, enabling e.g. co-analysis of specular and off-specular data. The collaboration

between ESS and FZJ will continue with the ESS Instrument Data Scientist participating in developer meetings with the BornAgain developers. At a higher level, European cross-facility discussions about analysis software takes place in League of advanced European Neutron Sources (LENS) regularly.

Analysis support at ESS

The STAP report delivered in early 2021 stated that the long-term plan for analysis support of reflectometry would be outlined in this report. To describe this plan, we define four broad-brush user types that are expected at ESS on the ESTIA and FREIA instruments:

- new users to reflectometry that have no prior experience in reflectometry analysis;
- users with complex analysis requirements (e.g. constrained analysis of kinetic measurements);
- experienced reflectometry users with experience in reflectometry analysis; and
- users interested in the co-refinement of specular/off-specular/GISANS.

These four categories will handle the vast majority, if not all, of users of reflectometry instrumentation at ESS.

The first category is inexperienced users, who are the main market for the easyReflectometry graphical user interface. These users will be introduced to and supported (by the full instrument staff, led by the Instrument Data Scientist) to use the easyReflectometry software package. This support will take the form of online tutorials and documentation, in addition to short introductions before or during the reflectometry beamtime. Users of this type, with specific analysis requirements, may be supported by the Instrument Data Scientist to produce custom item types as necessary.

Users with particularly complex analysis requirements, that are too complex/cumbersome to be handled by the easyReflectometry graphical interface will be offered direct support from the Instrument Data Scientist to use the associated Python interface. This support is expected to take the form of a collaboration between the Instrument Data Scientist and the user(s), with the expectation of co-authorship on any publication that would use the data analysis methodology. Over time, these complex analyses methodologies will be considered for inclusion in the easyReflectometry modular graphical user interface with appropriate acknowledgement/citation expected from subsequent users.

Experienced users of reflectometry (that do not fall into the previous category) will typically need little support in the analysis, they will be expected to use software packages that are familiar to them. The Instrument Data Scientist will offer general expertise concerning data analysis for these users (considerations regarding overfitting and expected results),

however, this will be in an ad hoc fashion. If these users are interested in transitioning to using the easyReflectometry package, this will be supported as in the previous two categories.

The final category of users are those aiming to analyse data collected from specular, off-specular, and potentially GISANS, in a coherent fashion (i.e. where a single model is applied to all techniques). This use case is currently uniquely supported by the BornAgain software package and these users will be supported in using this package (either in the Python environment or through the BornAgain GUI). It has been shown, in a collaboration between Artur Glavic, Michael Fitzsimmons, and others (that Andrew McCluskey has been a passive observer on) that, using the Python interface, BornAgain is capable of performing such analysis. Through continued communication and collaboration between ESS and the Scientific Computing Group at MLZ, this analysis methodology will continue to be developed.

For all of the user types described above, the Instrument Data Scientist will work with the users before the experiment to determine if there is an expected analysis approach. Those with an analytical model available will be supported in adding this analytical model to the automated data pipeline, such that initial results from such a model will be made available during data collection. Initially, this support will focus on those with complex analysis requirements using the easyReflectometry Python interface that will involve substantial planning before the beamtime. However, in time this will be made available to all users capable of scripting their analyses (this includes those using the easyReflectometry graphical user interface). In addition to traditional automated analysis solutions, the Instrument Data Scientist will also say up to date on recent developments in the applications of data science and artificial intelligence to reflectometry analysis (indeed, this is an active area of research interest for the Instrument Data Scientist).

3 Data/metadata cataloguing

Raw data collected at ESS instrumentation will be stored in the SciCat database, from which the user will be able to access this at their home institution (or copy to a hard drive during the beamtime). Reduced reflectometry data will also be stored in SciCat, in objects that are linked to the parent raw data, with unique reductions producing unique reduced data (i.e. changing a reduction parameter will not overwrite a previous reduction). A diagrammatic description of the data and metadata flow is shown in Figure 5. To ensure that this database is easily searchable by the user, discussions are already taking place (with Massimiliano Novelli) into the metadata that will be stored alongside the relevant datasets. This metadata will be split into three types:

- Instrumental metadata
- Scientific metadata

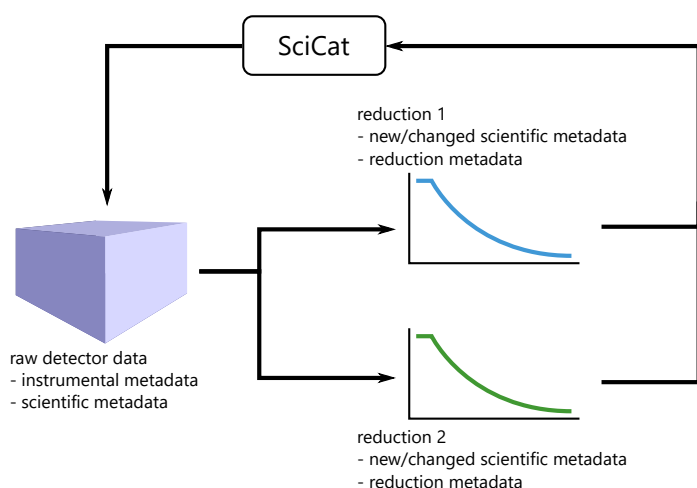


Figure 5: A diagrammatic description of the planned data flow in reduction of reflectometry data, this could be expanded to include the results of automatic analysis.

- Post-collection metadata

Instrumental metadata is that associated with instrumental parameters during the data collection (i.e. sample angle/temperature), this is associated with the raw data (and by inheritance the reduced data). The majority of instrumental metadata will be obtained automatically from the instrumentation and sample environment without the need for user interaction. Scientific metadata is that which the user will provide, such as sample names, chemical descriptors, or qualitative sample evaluations. Much of this metadata can be automatically produced, i.e. a drop box menu of sample chemical descriptors may be populated from the beamtime proposal, however, some of it will require user input to generate. Therefore, to motivate the users to input this data, it will be impressed on the user the value of this downstream in the searching of data (using the SciCat search functionality mentioned previously) or to facilitate automatic data analysis. Additionally, the inclusion of this metadata will be straightforward to achieve, leveraging SciChat as an online e-logbook that may be accessed both onsite and remotely. We must note that there is no desire to require the user to input manually large amounts

of metadata, however, we hope that by showing them the added value this will be of interest. Post-collection metadata is metadata associated with the data reduction process, such as the data reduction process and parameters used, this could also include analysis metadata in automated analysis situations. This metadata will not be directly linked with the raw data, rather associated with reduced datasets.

References

- [1] J. Stahn and A. Glavic. Focusing neutron reflectometry: Implementation and experience on the TOF-reflectometer Amor. *Nuclear Instruments and Methods in Physics Research A*, 821:44–54, 2016.
- [2] G. Pospelov, W. Van Herck, J. Burle, J. M. Carmona Loaiza, C. Durniak, J. M. Fisher, M. Ganeva, D. Yurov, and J. Wutke. BornAgain: software for simulating and fitting grazing-incidence small-angle scattering. *Journal of Applied Crystallography*, 53(1):262–276, 2020.
- [3] A. R. J. Nelson and S. W. Prescott. rfnx: neutron and x-ray reflectometry analysis in python. *Journal of Applied Crystallography*, 52(1):193–200, 2019.
- [4] P.A. Kienzle, J. Krycka, N. Patel, and I. Sahin. Refl1d (Version 0.8.14) [Computer Software]. College Park, MD: University of Maryland, 2011.
- [5] A. V. Hughes. Rascal. Sourceforge <https://sourceforge.net/projects/rscl/>, 2011.
- [6] P.A. Kienzle, J. Krycka, N. Patel, and I. Sahin. Bumps (Version 0.8.0) [Computer Software]. College Park, MD: University of Maryland, 2011.
- [7] M. Newville, R. Otten, T. Stensitzki, A. R. J. Nelson, A. In-gargiola, D. B. Allen, Austen Fox, and M. Rawlik. lmfit <https://lmfit.github.io/lmfit-py/>, 2021.
- [8] C. Cartis, J. Fiala, B. Marteau, and L. Roberts. Improving the Flexibility and Robustness of Model-Based Derivative-Free Optimization Solvers. *ACM Transactions on Mathematical Software*, 45(3):32:1–32:44, 2019.