ORNL/TM-2023/3126

Another Set of Python Tools for Visualizing and Manipulating Small-Angle Neutron Scattering Data: Descriptions and Examples

William T. Heller

October, 2023

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-2023/3126

Neutron Scattering Division

Another Set of Python Tools for Visualizing and Manipulating Small-Angle Neutron Scattering Data: Descriptions and Examples

William T. Heller

October, 2023

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-Battelle LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abbreviations

ORNL	Oak Ridge National Laboratory
SNS	Spallation Neutron Source
HFIR	High Flux Isotope Reactor
SANS	Small-Angle Neutron Scattering

Acknowlegements

This research used resources at the Spallation Neutron Source, DOE Office of Science User Facilites operated by the Oak Ridge National Laboratory.

Abstract

The GP-SANS, Bio-SANS and EQ-SANS instruments at ORNL [5] utilize *drtsans* for data reduction [6]. *drtsans* is built on Python (https://www.python.org), and it can be run using python scripts and Jupyter notebooks (https://www.jupyter.org). The flexibility afforded by Python makes it possible to incorporate additional actions into the scripts used for data reduction, such as analysis and visualization. Here, a new set of tools for visualizing and manipulating SANS data that can be incorporated into the data reduction scripts for the ORNL SANS instruments [5], or employed during post–processing, is presented that expands the capabilities of the two previously-released tool sets [4, 3].

1 Introduction

The ability to rapidly evaluate SANS data during an experiment, particularly on the high-flux instruments at ORNL [5] is crucial for efficiently using the limited beamtime that is given to each experiment. Equally important is the ability to visualize the data in a manner that is appropriate for the experiment at hand. Previous reports describing tools that have been developed for this purpose included those useful for visualization as well as rapid, simple analyses that can help a user evaluate data at the beam line [4, 3]. The implementation of the tools in python using libraries that are also required by *drtsans* [6] makes it possible to directly incorporate the tools into data reduction scripts.

There remains benefit in the continued development of the capabilities of the set of tools that is available in the *sanstools* repository availble at the link below.

https://code.ornl.gov/wt3/sanstools

In this report, the following new tools are presented:

- *plot2d* - create linear or log 2D plots of data;
- *roi1d* - integrate a q-range in a 1D file;
- *manual_sub* - perform a background subtraction with user-specified scale factor and baseline;
- *paramseries* - make a 2D color map plot from a series of 1D data files.

Each of these new tools is described in this report. The usage of the tool is also presented, which complements the example scripts that are also provided in the repository found at the link above. Data that can be used to test each tool is also provided in the repository, as are this report and the two previous ones [4, 3].

In this report, example code, such as code demonstrating how a tool is used, is shown in blue. Directories on the computer systems, file names output by the tools, and input variables called out in sentences are also colored blue.

2 *plot2d*

2.1 Description

plot2d is a tool for creating 2D plots from data reduced by *drtsans* [6]. The plots can be either linear or logarithmic in z, and the range of the z-axis can be specified by the user. Data is supplied as ASCII text data files, such as those produced during data reduction. Data can be supplied in 3-column format (q_x, q_y, I), 4-column format (q_x, q_y, I, dI), or 6-column format (q_x, q_y, I, dI, dq_x, dq_y). The 6-column format is output by *drtsans* [6].

2.2 Usage

To use *plot2d*, it must be imported in a Python script. For the sake of this usage example, assume that the tool has been installed in the following directory.

/SNS/EQSANS/shared/sanstools/

Further assume that the data is saved into the following directory.

/home/myhome/data/

To call *plot2d*, the user provides the data filename, the working path and The minimum and maximum z-values for the scaling, and whether to use “linear” or “log” scaling in z. A code snippet follows.

```
import sys
sys.path.append(“/SNS/EQSANS/shared/sanstools/”)
import plot2d

    other code may be here

data_path = “/home/myhome/data/”
file = “EQSANS_137614_Iqxqy.dat”
zmin = 0.01
zmax = 500
zstyle = “log”
plot2d.plot2d(path=data_path, name=file, set_min = zmin, set_max = zmax, style=zstyle)
```

The tool will save the graph to a file named EQSANS_137614_Iqxqy.png.

It is possible to use autoscaling for the z-axis of the plot by setting either set_min or set_max to None.

plot2d gracefully handles negative intensities. When negative intensity values are found when using log plotting, these values are assigned a value of 10^{-6}. This value can be readily ignored when creating the graph by setting the parameter set_min to a larger, positive value.

2.3 Example Output

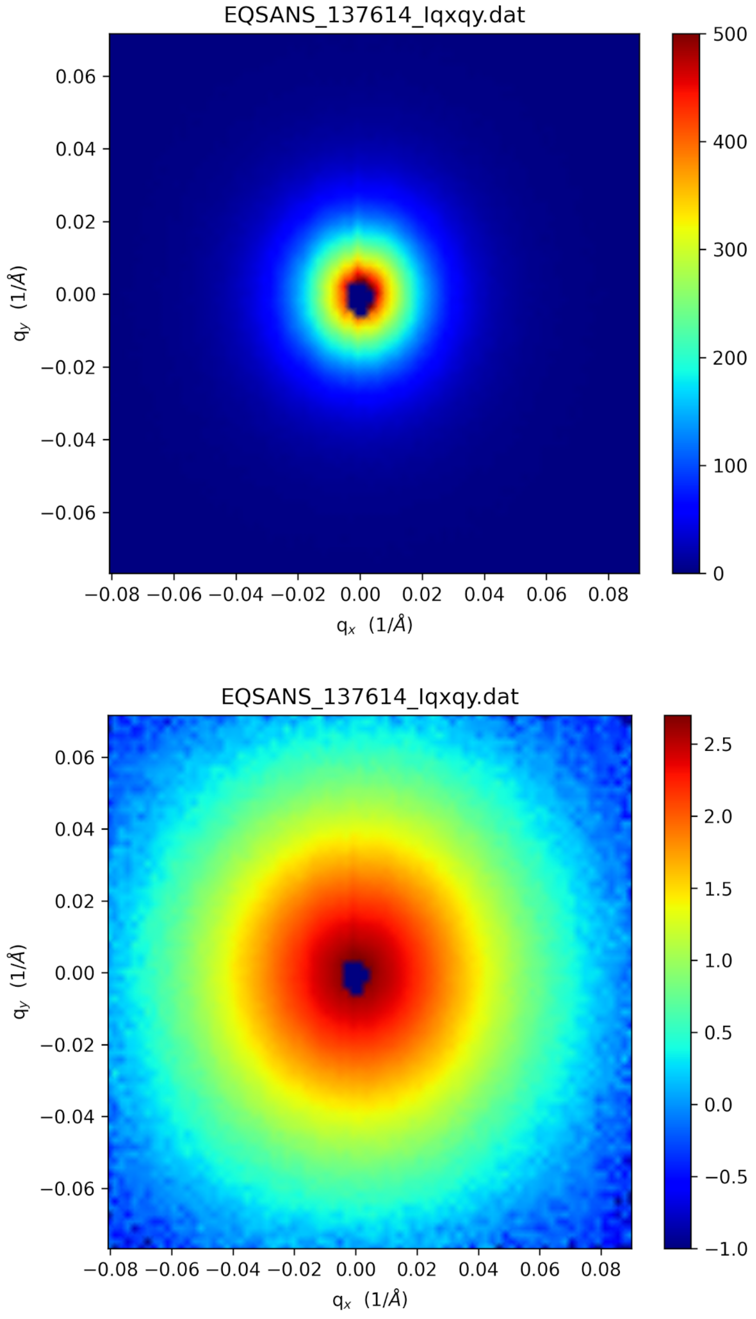

Figure 1. Example output from *plot2d*. Example output from *plot2d* from a Porasil B data set. The top image uses linear scaling on the *z*-axis and a range from 0.0 to 500.0. The bottom plot uses log scaling on the *z*-axis and a range from 0.01 to 500.0.

3 *roi1d*

3.1 Description

When working with data from parametric studies, it can be helpful to understand how a feature, such as a diffraction peak, or the overall scattering power in a given q-range changes with the parameter being varied. The tool *roi1d* is a tool for doing so from a user-specified file and q-range. In addition to a simple summation of the values in the q-range, *roi1d* can perform a Riemann summation using the width of the q bins. The user-specified q-range only indicates which points to use, it does not sub-divide either of the end bins for use in the summation to use only the specified q-range.

Unlike many of the tools in *sanstools*, *roi1d* does not create its own plots or write results to a file by itself. It use must be part of a script that saves a data file of the results, creates a plot from the results, or both. The example_roi1d.py script included in the repository shows files being saved and a plot being made.

Data is supplied as ASCII text data files, such as those produced during data reduction. Data can be supplied in 2-column format (q, I), 3-column format (q_x, I, dI), or 4-column format (q, I, dI, dq). The 4-column format is output by *drtsans* [6].

3.2 Usage

To use *roi1d*, it must be imported in a Python script. For the sake of this usage example, assume that the tool has been installed in the following directory.

/SNS/EQSANS/shared/sanstools/

Further assume that the data is saved into the following directory.

/home/myhome/data/

To call *roi1d*, the working path, data file names, the minimum and maximum q-values for the integration and the flag specifying whether or not to use the bin width in the summation must be provided by the user.

A code snippet follows.

```
import sys
sys.path.append("/SNS/EQSANS/shared/sanstools/")
import roi1d
import numpy as np

    other code may be here

data_path = "/home/myhome/data/"
files = ["EQSANS_1000s_139479_1_Iq.dat", "EQSANS_1000s_139479_2_Iq.dat", ...]
times = [500, 1500, ...]
qmin = 0.015
qmax = 0.045
dx = False
vals = np.zeros(len(files))
for i in range(0, len(files)):
    vals[i] = roi1d.roi1d(path=data_path, name=files[i], qmin=qmin, qmax=qmax, use_dx=dx)
```

The vals array can then be written to a file or create a plot, as can be seen in the Example Output, below.

3.3 Example Output

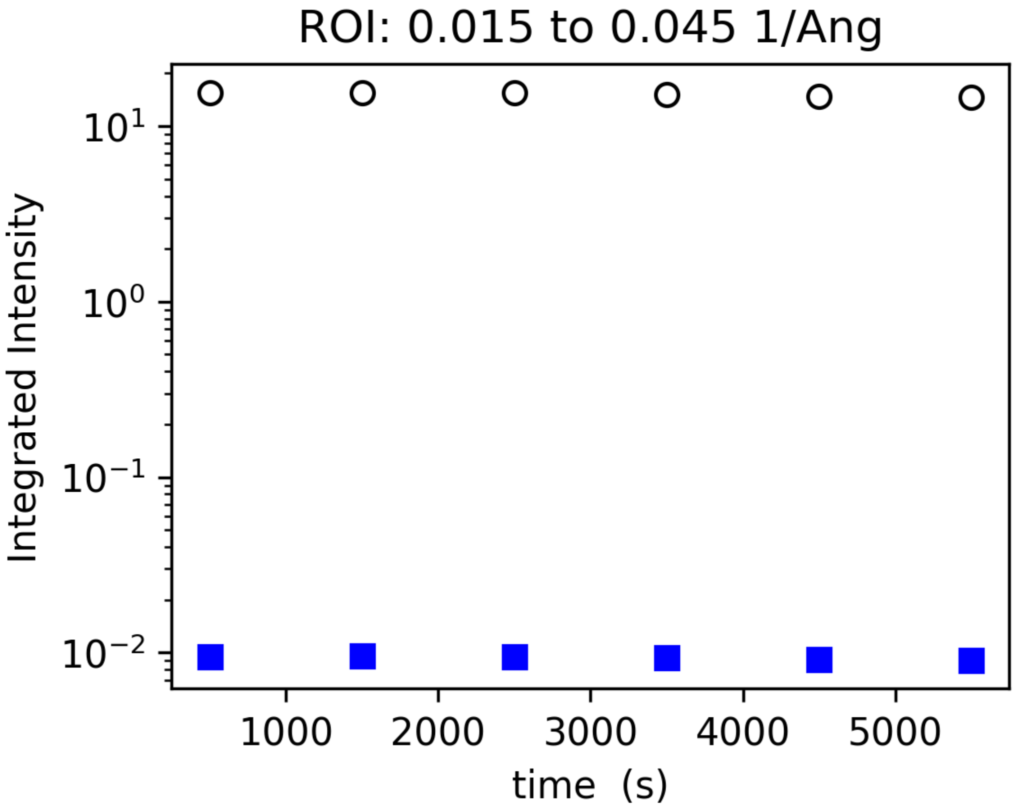

Figure 2. Example output from *roi1d*. The results obtained from integrating a region of interest in the same series of files as a straight sum of the values in the region of interest (○) and by accounting for the width of each bin in the region of interest (■).

4 *manual_sub*

4.1 Description

Correcting SANS data from weakly scattering, dilute solutions can be a time-consuming task, but it is vital to do it as well as possible to ensure that subsequent data analyses can be relied upon. The previously-developed *solvent_sub* [4] attempted to automatically determine either a suitable baseline to account for a difference in hydrogen content or a scale factor to account for the fraction of solvent displaced by the scattering particle. Unfortunately, an automatic determination of the necessary baseline or scale factor does not always produce a suitable result. Relying on either the baseline or a constant for proper solvent background subtraction may not be sufficient in all cases. The tool *manual_sub* provides the same functionality as *solvent_sub* [4] but with user-specified baseline and scale factor values that allow for fine-tuning of the background correction. Importantly, *solvent_sub* allows for simultaneous application of both a baseline and a scale factor, thereby affording maximum flexibility to the user.

Data is supplied as ASCII text data files, such as those produced during data reduction. Data can be supplied in 2-column format (q, I), 3-column format (q_x, I, dI), or 4-column format (q, I, dI, dq). The 4-column format is output by *drtsans* [6].

manual_sub is not well-suited to being incorporated into data reduction scripts.

4.2 Usage

To use *manual_sub*, it must be imported into a Python script. For the sake of the sample, assume taht the tool has been installed in the following directory

/SNS/EQSANS/shared/sanstools/

Further assume that the data is saved into the following directory.

/home/myhome/data/

```
import sys
sys.path.append("/SNS/EQSANS/shared/sanstools/")
import manual_sub

    other code may be here

data_path = "/home/myhome/data/"
samplefile = "hsa_bmcl_10.txt"
solventfile = "buffer_bmcl.txt"
baseline = 0.011
scalefactor = 1.0
plot2d.plot2d(samplename=samplefile, solventname=solventfile, path=data_path, base_line=baseline,
scale_factor=scalefactor)
```

The tool saves two files, one is the background-subtracted data file (hsa_bmcl_10_sub.txt) and the second is a plot of the data (hsa_bmcl_10_sub.png). The output data file has the file names, the base line value and the scale factor value in the header information.

4.3 Example Output

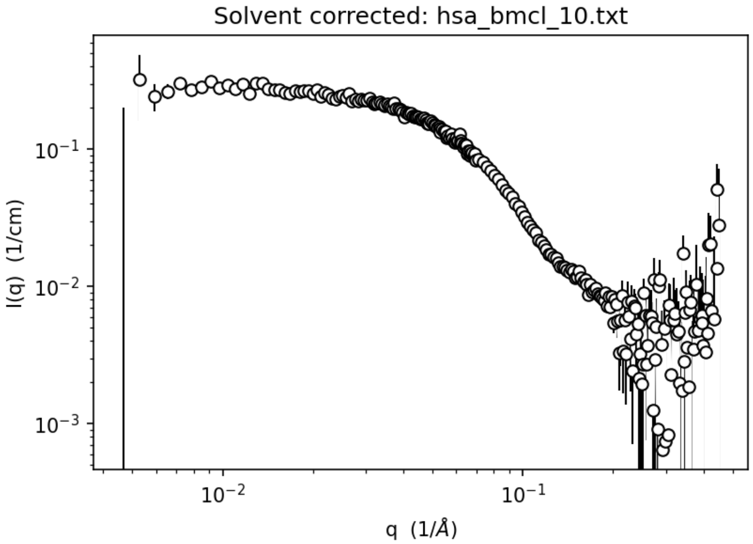

Figure 3. Example output from *manual_sub*. Example output from *manual_sub* using data from a chemical denaturation study of a protein.

5 *paramseries*

5.1 Description

There are a wide variety of ways to visualize changes in SANS data that take place when measuring a parametric series of data. Parameters often used include concentration, temperature and time. The most common method for visualizing changes in such series are with plots containing all of the data sets in a single graph. The curves can be displayed as-is, or they can be offset for clarity. Such plots can become very busy. An alternative kind of plot is to create a color map from the series of data. Such plots are perhaps more common in powder diffraction, for example, where phase transitions in materials are readily visualized as the diffraction peaks shift as a sample environment parameter is changed during an *in situ* study of a material (see, for example, Figures 2 and 3 of the *in situ* neutron diffraction study published by Haddock and coworkers [1]).

Data is supplied as ASCII text data files, such as those produced during data reduction. Data can be supplied in 2-column format (q, I), 3-column format (q_x, I, dI), or 4-column format (q, I, dI, dq). The 4-column format is output by *drtsans* [6].

5.2 Usage

To use *paramseries*, it must be imported in a Python script. For the sake of this usage example, assume that the tool has been installed in the following directory.

/SNS/EQSANS/shared/sanstools/

Further assume that the data is saved into the following directory.

/home/myhome/data/

To call *paramseries*, the working path, data file names, the minimum and maximum q-values for the integration and the flag specifying whether or not to use the bin width in the summation must be supplied.

A code snippet follows.

```
import sys
sys.path.append("/SNS/EQSANS/shared/sanstools/")
import paramseries

    other code may be here

data_path = "/home/myhome/data/"
files = ["tseries_1_Iq.dat", "tseries_2_Iq.dat", "tseries_3_Iq.dat", ...]
temps = [25.0, 27.5, 30.0, ...]
series = "temperature"
ofile = "tseries.png"
pstyle = "logxz"

paramseries.paramseries(datapath=data_path, filenames=files, series_name=series, pseries=temps,
ofile=ofile, plottype=pstyle)
```

The allowed options for the plottype are “linear”, “logx”, “logz” and “logxz”. A plot with the desired *x* and *z* scaling, which can be seen in the Example Output will be saved to a file /home/myhome/data/tseries.png. The particular plot option shown above results in Panel D of the figure below.

5.3 Example Output

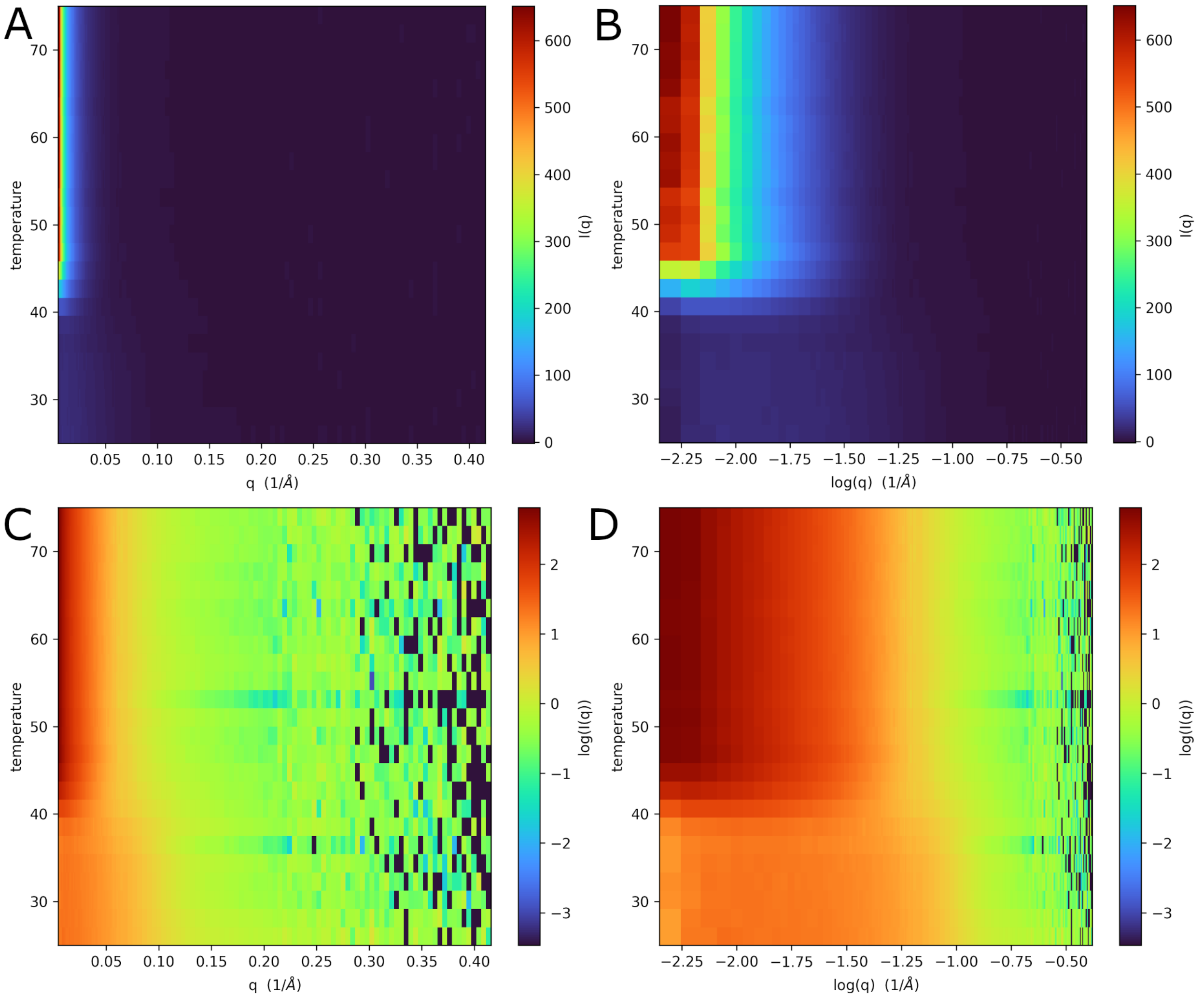

Figure 4. Example output from *paramseries*. Example output from *paramseries* using data from a thermal denaturation study of a protein [2] using (A) linear x/linear z scaling; (B) log x/linear z scaling; (C) linear x/log z scaling; and (D) log x/log z scaling. This figure has not been presented before.

6 References

References

[1] J. W. Haddock, Z. J. Barton, K. Feng, R. E. Baumbach, Q. Zhang, and S. E. Latturner. Flux growth of cerium nickel gallides studied by in situ neutron diffraction. *Inorganic Chemistry*, 61(39):15645–15653, 2022.

[2] W. T. Heller. Comparison of the thermal denaturing of human serum albumin in the presence of guanidine hydrochloride and 1-butyl-3-methylimidazolium ionic liquids. *The Journal of Physical Chemistry B*, 117(8):2378–2383, 2013.

[3] W. T. Heller. More tools for visualization and analysis of small-angle neutron scattering data: Descriptions and examples. *DOE OSTI*, DOI: 10.2172/1895215, 2022.

[4] W. T. Heller. Tools for visualization and analysis of small-angle neutron scattering data: Descriptions and examples. *DOE OSTI*, DOI: 10.2172/1842594, 2022.

[5] W. T. Heller, M. Cuneo, L. DeBeer-Schmitt, C. Do, L. He, L. Heroux, K. Littrell, S. V. Pingali, S. Qian, C. Stanley, V. S. Urban, B. Wu, and W. Bras. The suite of small-angle neutron scattering instruments at oak ridge national laboratory. *Journal of Applied Crystallography*, 51(2):242–248, 2018.

[6] W. T. Heller, J. Hetrick, J. Bilheux, J. M. Borreguero Calvo, W.-R. Chen, L. DeBeer-Schmitt, C. Do, M. Doucet, M. R. Fitzsimmons, W. F. Godoy, G. E. Granroth, S. Hahn, L. He, F. Islam, J. Lin, K. C. Littrell, M. McDonnell, J. McGaha, P. F. Peterson, S. V. Pingali, S. Qian, A. T. Savici, Y. Shang, C. B. Stanley, V. S. Urban, R. E. Whitfield, C. Zhang, W. Zhou, J. J. Billings, M. J. Cuneo, R. M. Ferraz Leal, T. Wang, and B. Wu. drtsans: The data reduction toolkit for small-angle neutron scattering at oak ridge national laboratory. *SoftwareX*, 19:101101, 2022.