ORNL/TM-2022/1497

Tools for Visualization and Analysis of Small-Angle Neutron Scattering Data: Descriptions and Examples

William T. Heller

January, 2022

OAK RIDGE
National Laboratory

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-2022/1497

Neutron Scattering Division

Tools for Visualization and Analysis of Small-Angle Neutron Scattering Data: Descriptions and Examples

William T. Heller

January, 2022

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 v
List of Tables vii
Abbreviations ix
Acknowlegements xi
Abstract 1
1 Introduction 1
2 *multiplot* 3
2.1 Description 3
2.2 Motivation 3
2.3 Usage 3
2.4 Example Output 4
3 *kratky_plot* 5
3.1 Description 5
3.2 Motivation 5
3.3 Usage 5
3.4 Example Output 6
4 *solvent_sub* 7
4.1 Description 7
4.2 Motivation 7
4.3 Usage 7
4.4 Example Output 8
5 *debye_bueche* 9
5.1 Description 9
5.2 Motivation 9
5.3 Usage 9
5.4 Example Output 10
6 *rg_explorer* 11
6.1 Description 11
6.2 Motivation 11
6.3 Usage 11
6.4 Example Output 12
7 References 13

LIST OF FIGURES

1 Example output from *multiplot* 4

2 Example output from *kratky_plot* 6

3 Example output from *solvent_sub* 8

4 Example output from *debye_bueche* 10

5 Example output from *rg_explorer* 12

LIST OF TABLES

Abbreviations

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

Acknowlegements

This research used resources at the High Flux Isotope Reactor and Spallation Neutron Source, a DOE Office of Science User Facility operated by the Oak Ridge National Laboratory. The author would like to thank John Hetrick, Jose Borreguero Calvo, Pete Peterson and Donnie Earnest for discussing how to make tools such as these available to the ORNL neutron scattering facility users.

Abstract

A great deal of progress has been made in improving the data reduction experience for the SANS instruments at the SNS and HFIR at ORNL. The existing data reduction toolset, **drtsans** [1], makes it possible to integrate data analysis and visualization tools into the data reduction scripts, thereby providing new opportunities for more automated data processing for users of the SNS and HFIR. Here, the first set of tools developed is described with usage examples.

1 Introduction

ORNL is home to three world-class SANS instruments and one world-class USANS instrument [4]. These instruments serve a diverse user community that performs research in topics that include structural biology, materials science, physics and chemistry. The neutron fluxes provided by the SNS and HFIR result in a large number of experiments per year that can each produce up to hundreds of individual data files that must be analyzed before the data can be turned into publishable results. Data analysis is one of the greatest bottlenecks facing the productivity of the SANS and USANS instruments at ORNL. An equally pressing challenge for users of the instruments is obtaining feedback of some kind during the experiment that allows them to make decisions about what to do next based on the data that has been recently collected.

The most logical place to employ these tools is during data reduction. The development of **drtsans** [1] provided an opportunity to do so because it is possible to integrate visualization and analysis tools written in Python (https://www.python.org) into the Python data reduction scripts that call **drtsans**. Here, a set of tools is presented that provides functionality for both data visualization and data analysis. Each tool is independent and self-contained to eliminate inter-dependency, and the tools can be called from Python scripts that are not also performing data reduction. Each tool provides one or more graphs in PNG format that are generated using the *matplotlib* library (https://matplotlib.org). The results can then be viewed without having to load the data and results into a plotting program. In some cases, the user is also provided one or more ASCII data files with the results in the event that they desire to prepare a publication-quality figure using other software.

The tools that are currently available and are described in this report are:

- *multiplot* - create a plot from one or more data sets
- *kratky_plot* - create a Kratky plot ($q^2 I(q)$ vs. q) from a data set
- *solvent_sub* - perform solvent (background) subtraction using $I(q)$ vs. q
- *debye_bueche* - perform a Debye-Bueche fit [2] to a data set
- *rg_explorer* - perform a set of Guinier fits [3] to a data set

Each tool and how it is called in a script is presentd herein. Example results are also presented for each tool. In this document. example code, such as how to use one of the tools, is shown in blue, as are directories on the computer systems and file names output by the tools.

2 multiplot

2.1 Description

multiplot is a tool for creating plots of one or more data sets from ASCII text data files, such as those produced during data reduction. Data can be supplied in 2-column (q, I), 3-column (q, I, dI) and 4-column (q, I, dI, dq) format, which is output by **drtsans** [1].

2.2 Motivation

The ability to plot several data sets, such as those from a concentration series or a temperature series, in a single graph is useful for assessing how an experiment is progressing. At present, **drtsans** [1] outputs a plot in PNG format during the reduction process for every data set. This is useful, but somewhat limited because several data sets collected in different instrument configurations spanning different q-ranges may be used to create a single complete data set that would be subjected to data analysis. It is possible to "merge" these data sets for a single sample collected using different instrument configurations during data processing, doing so and plotting the results is not intrinsic to the data reduction being performed by **drtsans**. The tool *multiplot* was developed to allow users to visualize merged data sets and groups of data sets as an integral part of the data reduction process.

2.3 Usage

To use multiplot, 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/datsans/
```

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

```
/home/myhome/data/
```

It is possible to call multiplot by providing a list of data filenames, the desired name of the output file and the working path. A code snippet follows.

```
import sys
sys.path.append(’/SNS/EQSANS/shared/datsans/’)
import multiplot

    other code may be here

data_path = ’/home/myhome/data/’
files = [’1ao6.miq’, ’1ovt.miq’, ’dm000_37c300.txt’]
output_file = ’set_plot’
multiplot.multiplot(name=files, ofile=output_file, path=data_path)
```

The tool will save the graph in to a file named set_plot.png. The ’.png’ is appended to the file name automatically. The plot that is output only shows q, I, and dI (if present). The image from the example above is shown in Figure 1.

2.4 Example Output

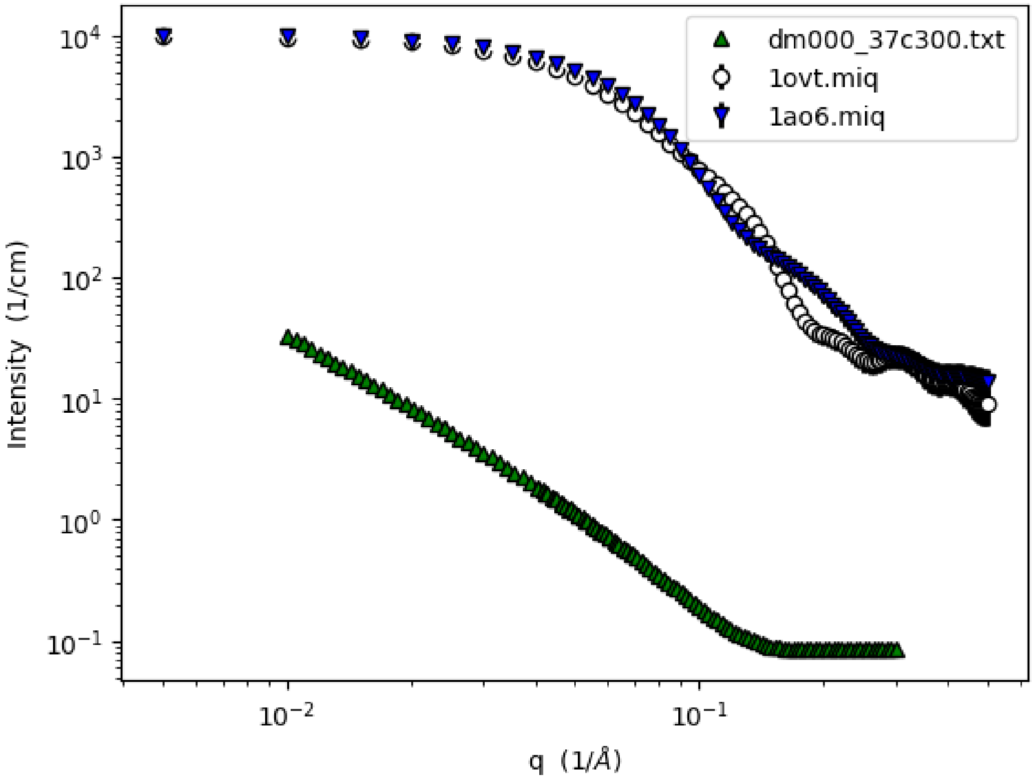

Figure 1. Example output from *multiplot*. Note that the data does not need to be from the same source. The example data sets shown here are a model intensity profile without uncertainties calculated from a 3-layer vesicle and data simulated from protein crystal structures of ovalbumin (1OVT [5]) and human serum albumin (1AO6 [6]) using ORNL_SAS [7]. Simulated uncertainties were added to the ORNL_SAS-generated data. The legend of the plot shows the names of the data files supplied to *multiplot*.

3 *kratky_plot*

3.1 Description

kratky_plot is a tool for creating a plot of data transformed into the Kratky representation of $q^2 I(q)$ vs. q from an ASCII text data file, such as those produced during data reduction. Data can be supplied in 2-column (q, I), 3-column (q, I, dI) and 4-column (q, I, dI, dq) format, which is output by **drtsans** [1].

3.2 Motivation

A commonly used approach for assessing the compactness of a scattering particle in dilute solution, such as proteins and polymers, is to create what is called a Kratky plot. In a Kratky plot, data are transformed to $q^2 I(q)$ vs. q and plotted. A compact particle, such as a folded protein or a polymer in a non-ideal solvent, has a well-defined peak that decays at high q in a Kratky plot. In contrast, a Kratky plot of an unfolded protein or a polymer in a good solvent does not have such a feature. The tool *kratky_plot* was developed to allow users to quickly assess whether or not their protein or polymer is in a compact or extended conformation.

3.3 Usage

To use kratky_plot, 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/datsans/
```

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

```
/home/myhome/data/
```

It is possible to call kratky_plot by the data file name and the working path. A code snippet follows.

```
import sys
sys.path.append(’/SNS/EQSANS/shared/datsans/’)
import kratky_plot

    other code may be here

data_path = ’/home/myhome/data/’
input_file = ’1ao6.miq’
kratky_plot.kratky_plot(filename=input_file, path=data_path)
```

The tool will save the graph to a file named 1ao6_kty.png and the data transformed into $q^2 I(q)$ vs q to a file name 1ao6_kty.txt so it can be replotted with software of the users choice. The ’.png’ and ’.txt’ are appended to the file names automatically. The plot that is output only shows q, I, and dI (if present). The transformed data that is output only contains q, I, and dI (if present). The image from the example above is shown in Figure 2.

3.4 Example Output

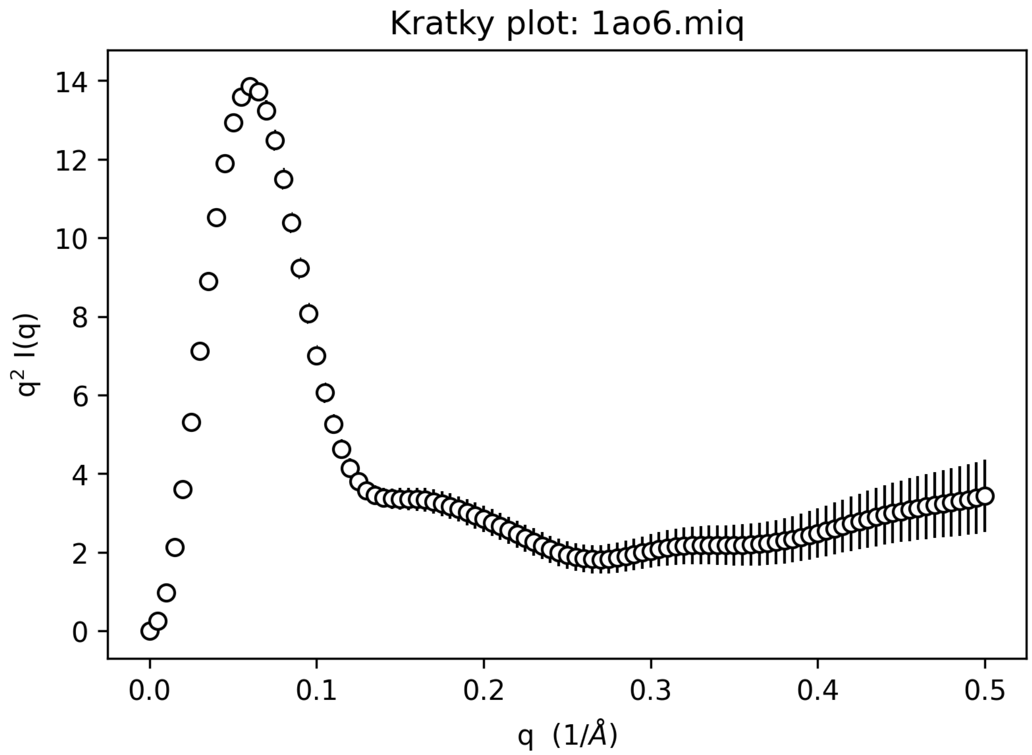

Figure 2. Example output from *kratky_plot*. The data set shown here is data simulated from the protein crystal structure of human serum albumin (1AO6 [6]) using ORNL_SAS [7]. Simulated uncertainties were added to the data.

4 *solvent_sub*

4.1 Description

solvent_sub is a tool for performing automated background subtraction, but it is intended more for cases of weak scattering above background than when strong signal exists. Two methods for performing the subtraction exist. In the first, the tool derives a scale factor that is applied to the background prior to subtraction. In the second, the tool derives a constant value that is used to shift the baseline of the background during the subtraction. In both cases, a target value of average minimum signal above background in the region used to derive the scale or shift is used to determine the right value. At present, the average minimum signal above background at high q is assumed to be 0.001 cm^{-1}. The most common usage would be for proteins or polymers in solution. Data can be supplied in 2-column (q, I), 3-column (q, I, dI) and 4-column (q, I, dI, dq) format, which is output by **drtsans** [1].

4.2 Motivation

Weakly scattering samples generally require correcting for the scattering from the solvent, which includes the incoherent scattering present in the sample/particle of interest and the surrounding solvent or other media. For example, when working with a predominantly hydrogenated sample volume that has particles dispersed in it, it is possible that the incoherent scattering from the sample is lower than that of the background media. Another possible cause of a mismatch between the inherent background of the sample and its background happens in contrast matching experiments where the amount of hydrogen in the sample and background is not exactly the same. Performing the background subtraction is often a manual process, which can be time-consuming in experiments that produce many data sets. *solvent_sub* is intended to correct for these differences by automatically performing the operations that a user or instrument scientist would otherwise perform by hand.

4.3 Usage

To use solvent_sub, 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/datsans/

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

/home/myhome/data/

It is possible to call solvent_sub by the sample data file name, the background data file name, the working path and the desired method for adjusting the background data to ensure that neither over- nor under-subtraction takes place. The method can either be 'shift' or 'scale', which work as described above. A code snippet follows.

```
import sys
sys.path.append('/SNS/EQSANS/shared/datsans/')
import solvent_sub
```

```
other code may be here
data_path = '/home/myhome/data/'
sample_file = 'hsa_tris_10.txt'
backgnd_file = 'tris.txt'
sub_method = 'scale'
solvent_sub.solvent_sub(samplename=sample_file, solventname=backgnd_file, path=data_path,
method=sub_method)
```

The tool will save the graph to a file named hsa_tris_10_sub.png and the data to a file name hsa_tris_10_sub.txt so it can be replotted with software of the users choice and analyzed. The '.png' and '.txt' are appended to the file names automatically. The plot that is output only shows q, I, and dI (if present). The data file that is output contains q, I, dI (if present) and dq (if present). The image from the example above is shown in Figure 3.

4.4 Example Output

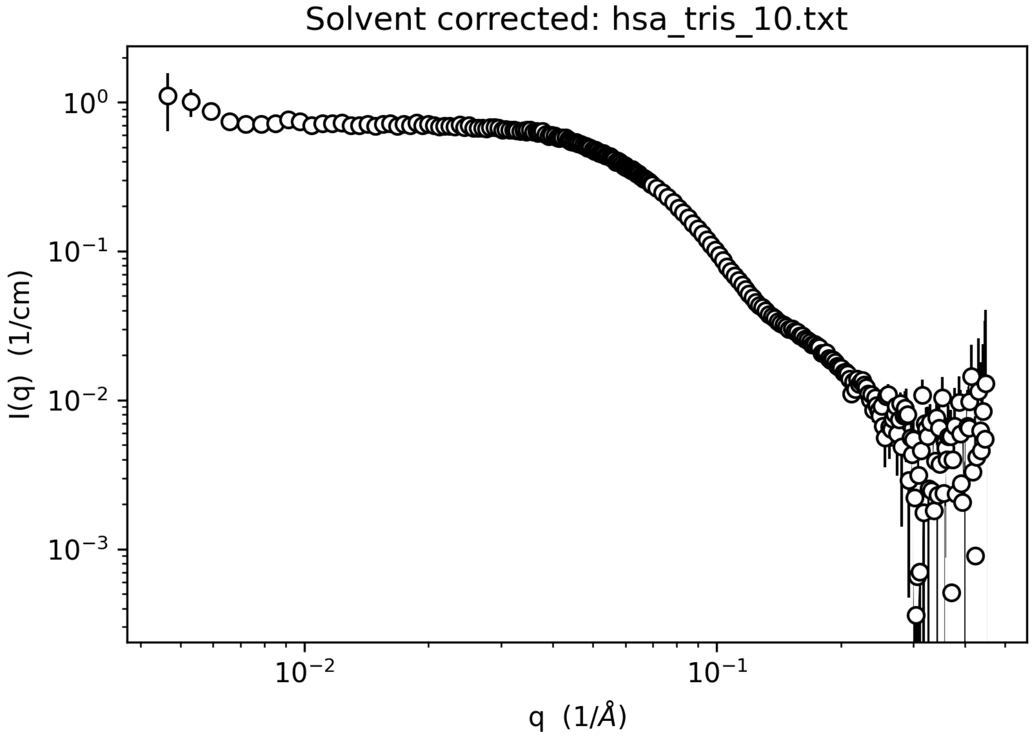

Figure 3. Example output from *solvent_sub*. The data shown is from human serum albumin in tris buffer (Heller, unpublished result) that was collected on EQ-SANS [4]. The sample and buffer signal were reduced separately against the empty cell prior to subtraction of the buffer signal from the sample signal.

5 *debye_bueche*

5.1 Description

debye_bueche is a tool for performing a fit of data transformed into $1/\sqrt{I(q)}$ vs. q^2, which is the Debye-Bueche model [2]. Data can be supplied in 3-column (q, I, dI) and 4-column (q, I, dI, dq) format, which is output by **drtsans** [1].

5.2 Motivation

The Debye-Bueche fit [2] is almost exclusively used for determining the scale factor that is needed for **drtsans** to place SANS data in absolute units of cm^{-1} [8] from a measurement of the Porasil B porous silica standard samples that are maintained at the SANS beamlines. Automating this fit will save some time when configuring the data reduction for an experiment.

5.3 Usage

To use *debye_bueche*, 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/datsans/
```

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

```
/home/myhome/data/
```

It is possible to call *debye_bueche* by the data file name and the working path. The desired minimum and maximum q-values for the fit are also required. A code snippet follows.

```
import sys
sys.path.append(’/SNS/EQSANS/shared/datsans/’)
import debye_bueche

    other code may be here

data_path = ’/home/myhome/data/’
input_file = ’porasil_merged.txt’
debye_bueche.debye_bueche(filename=input_file, path=data_path, qmin=0.01, qmax=0.03)
```

The tool will save the graph of $1/\sqrt{I(q)}$ vs. q^2 to a file named porasil_merged_db.png. The plot that is output only shows q, I, and dI (if present). The ’.png’ is appended to the file name automatically. The image from the example above is shown in Figure 4.

5.4 Example Output

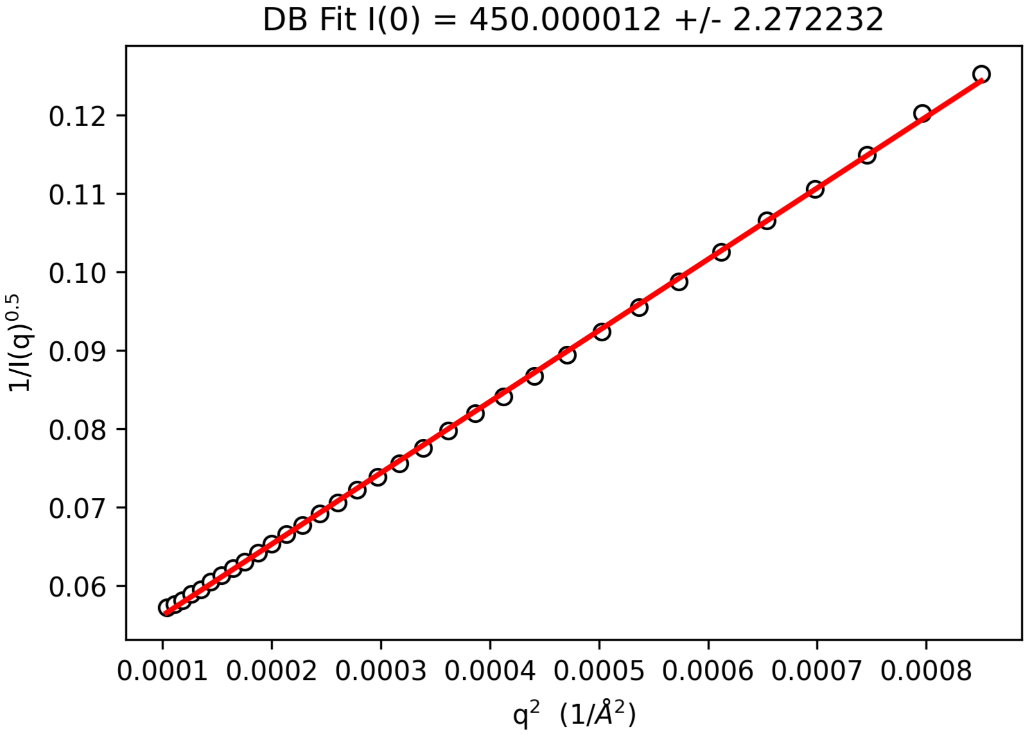

Figure 4. Example output from *debye_bueche*. The data is from the calibrated Porasil B standard used at the EQ-SANS for scaling reduced data to units of cm^{-1} [8]. Note that the title of the plot provides the $I(0)$ value from the fit. In this example, the absolute scale factor has already been included in the data reduction process.

6 *rg_explorer*

6.1 Description

rg_explorer is a tool for performing a set of Guinier fits [3] to a data set within a q-range that is provided by the user. Rather than performing a single fit, the tool performs a set of fits of $Ln(I(q))$ vs. q^2 within the specified Q-range using at least 5 data points, checks the fits for consistency with the assumptions that must be satisfied for a Guinier fit to be valid [3] and outputs plots of the results. Data can be supplied in 3-column (q, I, dI) and 4-column (q, I, dI, dq) format, which is output by **drtsans** [1].

6.2 Motivation

Studies of the structures of protein and protein complexes in dilute solution by small-angle scattering require very high-quality, monodisperse and non-interacting samples. Determining the quality of samples during an experiment helps to guide the remainder of the experiment. One way to assess sample quality is to perform a Guinier fit [3] of $Ln(I(q))$ vs. q^2 for the radius of gyration, R_g, and the forward scattering, $I(0)$. The plot of the data as $Ln(I(q))$ vs. q^2 is also helpful for assessing sample quality.

6.3 Usage

To use *rg_explorer*, 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/datsans/

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

/home/myhome/data/

It is possible to call *rg_explorer* by the data file name and the working path. The desired minimum and maximum q-values for the fit are also required. A code snippet follows.

```
import sys
sys.path.append('/SNS/EQSANS/shared/datsans/')
import rg_explorer

    other code may be here

data_path = '/home/myhome/data/'
input_file = '1ovt.miq'
rg_explorer.rg_explorer(filename=input_file, path=data_path, qmin=0.005, qmax=0.07)
```

The tool will save the graph of $Ln(I(q))$ vs q^2 to a file named 1ovt_rg.png. A contour plot of R_g in the q-range specified by the user is also saved as a file name 1ovt_rg_map.png. A table of the results of the fitting that only contains valid values of R_g (i.e. $q_{max}R_g < 1.0$) is also provided in a file name 1ovt_rg_table.txt. The columns of the file are q_{min}, q_{max}, χ^2, R_g, dR_g, $I(0)$ and $dI(0)$. The '.png' and '.txt' are appended to the file names automatically. The images from the example above is shown in Figure 5.

6.4 Example Output

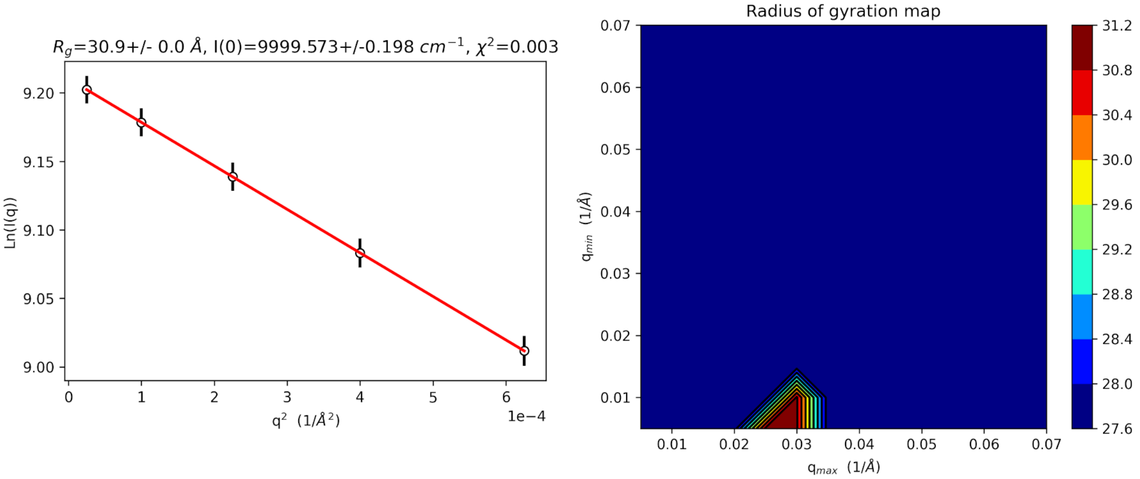

Figure 5. Example output from *rg_explorer*. *rg_explorer* outputs two plots. The left image is $Ln(q)$ vs. q^2 and the best fit line from the set. The right image shows a 2D map of valid R_g values (i.e. $q_{max}R_g < 1.0$) as a function of q_{min} and q_{max}. The example data was simulated from protein crystal structures of ovalbumin (1OVT [5]) using ORNL_SAS [7]. Simulated uncertainties were added to the data.

References

[1] drtsans; https://code.ornl.gov/sns-hfir-scse/sans/sans-backend, May 2020.

[2] P. Debye and A. M. Bueche. Scattering by an inhomogeneous solid. *Journal of Applied Physics*, 20(6):518–525, 1949.

[3] A. Guinier. La diffraction des rayons x aux très petits angles : application à l'étude de phénomènes ultramicroscopiques. *Ann. Phys.*, 11(12):161–237, 1939.

[4] 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, Apr 2018.

[5] H. Kurokawa, B. Mikami, and M. Hirose. Crystal structure of diferric hen ovotransferrin at 2.4 å resolution. *Journal of Molecular Biology*, 254(2):196–207, 1995.

[6] S. Sugio, A. Kashima, S. Mochizuki, M. Noda, and K. Kobayashi. Crystal structure of human serum albumin at 2.5 å resolution. *Protein Engineering, Design and Selection*, 12(6):439–446, 06 1999.

[7] E. Tjioe and W. T. Heller. *ORNL_SAS*: software for calculation of small-angle scattering intensities of proteins and protein complexes. *Journal of Applied Crystallography*, 40(4):782–785, Aug 2007.

[8] G. D. Wignall and F. S. Bates. Absolute calibration of small-angle neutron scattering data. *Journal of Applied Crystallography*, 20:28–40, 1987.