BioXTAS RAW Documentation

Release 2.1.2

Jesse B. Hopkins

May 23, 2022

Contents

CHAPTER 1

What is BioXTAS RAW?

BioXTAS RAW is a GUI based, free, open-source Python program for reduction and analysis of small-angle X-ray solution scattering (SAXS) data. The software is designed for biological SAXS data. It is available on windows, macOS (and OS X), and Linux. It provides an alternative to closed source programs such as Primus and Scatter for primary data analysis. Because it can calibrate, mask, and integrate images it also provides an alternative to synchrotron beamline pipelines that scientists can install on their own computers and use both at home and at the beamline.

Find out how to get it!

CHAPTER 2

Features

- Analysis of radius of gyration (R_g) and I(0) via Guinier fit.
- Analysis of molecular weight via I(0) comparison to standards, absolute calibration, correlation volume (V_c), corrected Porod volume (V_p), the ATSAS Shape&Size and the ATSAS Bayesian methods.
- Calculation of inverse Fourier transforms (IFTs) via GNOM and a Bayesian indirect Fourier transform (BIFT).
- Calculation of envelopes (dummy atom models) using DAMMIF, DAMMIN, DAMAVER, and DAMCLUST.
- Calculation of electron density using the DENSS algorithm
- Easy processing of in-line chromatography coupled SAXS data, including size-exclusions coupled SAXS (SEC-SAXS) data.
- Deconvolution of SAXS data using singular value decomposition (SVD) and evolving factor analysis (EFA), and regularized alternating least squares (REGALS).
- Standard data operations such as averaging, subtraction, merging, and rebinning.
- Creation and plotting of 1D scattering profiles from 2D detector images, including Pilatus, CBF, Eiger, and more than 20 other types of images.

CHAPTER 3

What is the RAW API?

You can also *install BioXTAS RAW as a python package* without the GUI. This lets you import RAW directly into your python scripts and use the API to call any of the functions in RAW. This is great for creating custom processing scripts, either for unusual datasets that aren't handled in the GUI, or to ensure reproducibility in your analysis. It can also be used as the basis for an open-source automated SAXS data processing pipeline at a beamline or homesource.

CHAPTER 4

History and Usage

RAW was first developed in 2008 by Soren Skou as part of the biological x-ray total analysis system (BioXTAS) project. Since then it has been extensively developed, with recent work being done by Jesse Hopkins.

RAW is actively used as the primary analysis software at the MacCHESS ID7A BioSAXS beamline at CHESS and the BioCAT (18-ID) beamline at the APS. It is also used at various other beamlines, including:

- BL19U2 (SSRF)
- LiX (NSLS II)

SAXSLAB distributes RAW with some of its homesources, and RAW is used at various other homesources around the world.

Do you use RAW? *Let us know!*

CHAPTER 5

Licensing

RAW source code is released under a GPLv3 license. Both the source code and the prebuilt versions of RAW are free for anyone to download and use.

5.1 Installation

RAW can be downloaded from sourceforge. There are prebuilt installers for Windows (.msi), MacOs (.dmg), and Debian/Ubuntu (.deb). Installation on other Linux distributions requires building from source.

RAW is also available through SBGrid.

Some features in RAW require a separate installation of the ATSAS package.

For installation instructions for the API, see the *API documentation.*

5.1.1 Detailed installation instructions

RAW Install Guide for Microsoft Windows

There are two ways to install RAW on Windows. The easiest, and recommended, way is to use a prebuilt installer (`.msi` file). If you want to install a version for which there is no prebuilt installer, you will need to install RAW from the source code.

Here you can find information on:

Using a prebuilt installer

The recommend way to install RAW on Windows is using a prebuilt installer. To install from a prebuilt installer, simply download the `RAW-x.y.z-win64.msi` (where `x.y.z` is the version number) file from sourceforge (https://sourceforge.net/projects/bioxtasraw), and double click it to run the installer.

Important Notes:

- Because the RAW team is an 'unknown' developer, you will probably see some security warnings when you install RAW. When you do, just give the installer and program permission to run on your computer.
- RAW is used by (relatively) few people, which means some virus scanners may not have seen the RAW software before. Occasionally virus scanners will mark a file (typically `RAW.exe`) as a threat (it will usually be in the 'general malware' category). If this happens, please do the following:
  - Upload the file to https://www.virustotal.com/ and see if any other virus scanners identify it as a virus (it is always possible someone hijacked the installer somehow!).
  - If most or all of the virus scanners on virustotal.com clear the file, make an exception for it in your virus scanner.
  - Contact the RAW developers, so we can report the false identification to the virus scanner company and get the file whitelisted in future definitions files.

General instructions for installing from source (advanced users)

1. Install Microsoft Visual C++ 14.2 Standalone: Build Tools for Visual Studio 2019
2. Install Python 3.X (if it isn't already installed) and add it to your system path (tested on 3.7 and 3.8).
3. Install the following Python packages (version indicated if less than most recent):
   - numpy
   - scipy
   - matplotlib
   - pillow
   - wxpython
   - h5py
   - cython
   - fabio
   - pyFAI
   - hdf5plugin
   - numba
   - reportlab
   - mmcif_pdbx
4. Download the RAW source file (`RAW-x.y.z-Source` where `x.y.z` is the version number) from sourceforge (https://sourceforge.net/projects/bioxtasraw)
5. Extract RAW to a directory of your choice.
6. In the top level RAW directory run `python setup.py build_ext --inplace` to build the extensions.
7. In the `bioxtasraw` subdirectory run `RAW.py` using python.
8. Enjoy!
   - If you have problems, please consult the *detailed installation guide* and the *solutions to common problems*. If that doesn't help, please contact the developers.

Notes for python 2 installation

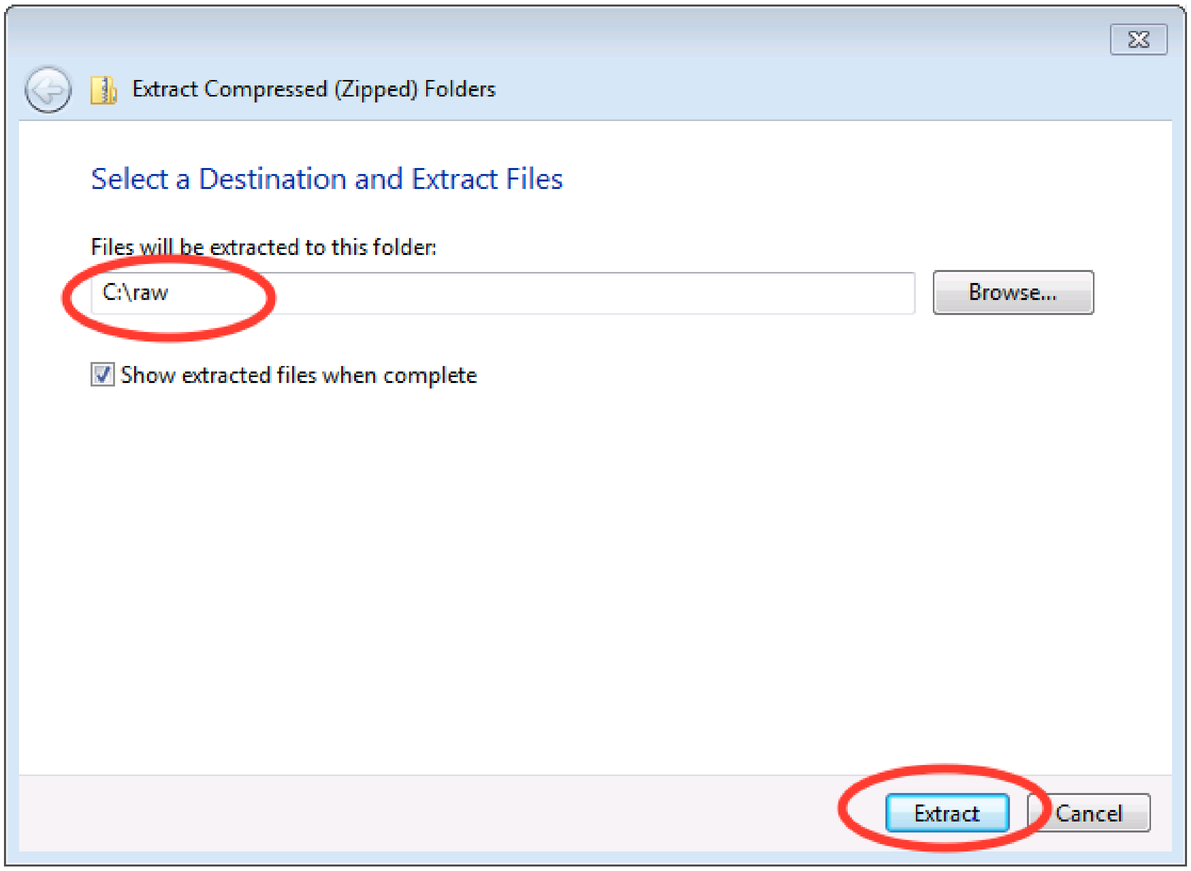
As of version 2.0.0, RAW is Python 3 compatible. The last guaranteed Python 2 compatible version of RAW is 2.0.0. However, it may still be possible to install RAW for Python 2. A few additional notes for that:

1. Install Microsoft Visual C++ 2008 Redistributable and Visual C++ Compiler for Python.
2. Install Python 2.7
3. Additional dependencies:
    - future
4. Required version of pyFAI is 0.17

Windows 7, 8.1, and 10 install from source instructions

1. RAW on windows can be installed using 64 bit (x64) or 32 bit (x86) python. Unless you know you need a 32 bit build, you should install the 64 bit version. Some libraries, such as pyFAI, maybe hard to install on 32 bit windows.
2. Download and install the Microsoft Visual C++ 14.2 Standalone: Build Tools for Visual Studio 2019.
    - https://wiki.python.org/moin/WindowsCompilers#Microsoft_Visual_C.2B-.2B-_14.2_standalone:_Build_Tools_for_Visual_Studio_2019_.28x86.2C_x64.2C_ARM.2C_ARM64.29
    - Download from here: https://visualstudio.microsoft.com/downloads/#build-tools-for-visual-studio-2019
    - Run the installer and install the C++ build tools with the default options.
3. Install Miniconda python distribution
    - Download the free miniconda python 3.x, e.g. 3.8, installer from: https://docs.conda.io/en/latest/miniconda.html
        - Make sure you get the python 3.x version!
        - Pick the appropriate 64 bit/32 bit version (64 bit recommended!).
    - Run the installer with the default options.
    - More detailed install instructions are available here: https://conda.io/docs/user-guide/install/windows.html
4. Install the necessary python packages
    - Open an anaconda prompt by clicking on the start menu -> All Programs -> Anaconda3 -> Anaconda Prompt
    - Run the following commands in the anaconda prompt:
    - `conda upgrade conda pip wheel setuptools`
    - `conda install numpy scipy matplotlib pillow numba h5py cython numexpr reportlab`
    - `conda install -c conda-forge wxpython hdf5plugin fabio pyfai mmcif_pdbx`
5. Download RAW from sourceforge (https://sourceforge.net/projects/bioxtasraw)
    - Go to the Files tab on the linked website and download the `RAW-x.y.z-Source.zip` file, where `x.y.z` is the version number (for example, 1.0.0).
6. Expand the downloaded zip file into the downloads folder

- Right click on the download and select *Extract All*
- Accept the default location for files to be extracted.



Select a Destination and Extract Files

Files will be extracted to this folder:

C:\raw

[✓] Show extracted files when complete

7. In Windows Explorer, confirm that the file named `setup.py` is in your top level expanded raw directory. If it isn't, it's likely that when you expanded the RAW download, you ended up with unnecessary layers of directories. Find the directory with `setup.py` in it, and make that the top level folder.
8. Build the extensions
   - Open an anaconda prompt as in Step 4 of these instructions.
   - Type `cd C:\raw`
   - Hit enter
   - Type `python setup.py build_ext --inplace`
   - Hit enter
9. Run `RAW.py` from the command line
   - Open an anaconda prompt as in Step 4 of these instructions.
   - Type `cd C:\raw\bioxtasraw`
   - Hit enter
   - Type `python RAW.py`
   - Hit enter
10. Enjoy!

- If you have trouble with the installation, please see the *solutions to common problems* section below.

Common problems/troubleshooting

Prebuilt installer:

- Because the RAW team is an 'unknown' developer, you will probably see some security warnings when you install RAW. When you do, just give the installer and program permission to run on your computer.
- RAW is used by (relatively) few people, which means some antivirus programs may not have seen the RAW software before. Occasionally virus scanners will mark a file (typically `RAW.exe`) as a threat (it will usually be in the 'general malware' category). If this happens, please do the following:
  - Upload the file to https://www.virustotal.com/ and see if any other antivirus programs identify it as a problem (it is always possible someone hijacked the installer somehow!).
  - If most or all of the antivirus programs on virustotal.com clear the file, make an exception for it in your virus scanner.
  - Contact the RAW developers, so we can report the false identification to the virus scanner company and get the file whitelisted in future definitions files.

From source:

- If you are updating your RAW installation, you should completely delete the old RAW source files, and then replace them with the new ones.

RAW Install Guide for macOS and OS X

There are two ways to install RAW on Mac. The easiest, and recommended, way is to use a prebuilt app package (`.dmg` file). If you want to install a version for which there is no prebuilt installer, you will need to install RAW from the source code.

Here you can find information on:

Using a prebuilt app package

The recommended way to install RAW on Mac is using a prebuilt app package. To install from a prebuilt app package simply download the `RAW-x.y.z-mac-platform.dmg` (where `x.y.z` is the version number and `platform` is x86_64 for Intel macs and arm64 for Apple Silicon macs) file from sourceforge (https://sourceforge.net/projects/bioxtasraw), double click it to open the dmg, and drag the `RAW.app` file to your Applications folder (or wherever you want to install RAW).

Direct links fo the downloads:

- Apple Silicon macs (M1 or newer chips)
- Intel macs (pre-M1 chips)

Important Notes:

- Because the RAW team is an unidentified developer, you may get a warning message the first time you run the program. If that happens, right click on RAW and select *Open* from the right click menu, and then click the *Open* button in the window that appears.
  - *Note:* This requires administrator permissions.

General instructions for installing from source (advanced users)

1. Install a standalone version of python 3.X (recommended, not required), RAW is tested on 3.7 and 3.8.
2. Install the following python packages (most recent version of each recommended):
    - numpy
    - scipy
    - matplotlib
    - pillow
    - wxpython
    - h5py
    - cython
    - fabio
    - pyFAI
    - hdf5plugin
    - numba
    - reportlab
    - mmcif_pdbx
3. Download the latest RAW sourcecode from sourceforge (https://sourceforge.net/projects/bioxtasraw)
4. Extract RAW to a directory of your choice.
5. In the top level RAW directory run `python setup.py build_ext --inplace` to build the extensions.
6. In the `bioxtasraw` subdirectory run `RAW.py` using python.
7. Enjoy!
    - If you have problems, please consult the detailed installation guide and the *solutions to common problems* below. If that doesn't help, please contact the developers.

Notes for python 2 installation

As of version 2.0.0, RAW is Python 3 compatible. The last guaranteed Python 2 compatible version of RAW is 2.0.0. However, it may still be possible to install RAW for Python 2. A few additional notes for that:

1. Additional dependencies:
    - future
2. Required version of pyFAI is 0.17

OS X and macOS install from source instructions

1. Install Miniconda python distribution
    - Download the free miniconda python 3.x, e.g. 3.8, installer from: https://docs.conda.io/en/latest/miniconda.html
    - Open a Terminal window.

- In the terminal window type `cd ~\Downloads` and hit enter.
- In the terminal window type `bash Miniconda3-latest-MacOSX-x86_64.sh` and hit enter.
- Agree to all of the prompts.
- More detailed install instructions are available here: https://conda.io/docs/user-guide/install/macos.html
- Close the terminal window.

2. Install the necessary python packages.
   - Open a new terminal window as in the previous step
   - Type `conda upgrade conda pip wheel setuptools` and hit enter. Agree to all the prompts.
   - Type `conda install numpy scipy matplotlib pillow numba h5py cython numexpr reportlab` and hit enter. Agree to all the prompts.
   - Type `conda install -c conda-forge wxpython hdf5plugin fabio pyfai mmcif_pdbx` and hit enter. Agree to all prompts.
3. Download RAW from sourceforge
   - https://sourceforge.net/projects/bioxtasraw
   - Navigate to the *Files* tab and download the latest source code, `RAW-x.y.z-Source.zip`. Or download the latest development version from the git by navigating to the *Code* tab.
4. Expand the downloaded zip file in the Downloads folder by double clicking on it.
   - This step may not be necessary, some browsers may automatically expand zip files.
5. In the terminal or in the graphical file manager, confirm that the file named `setup.py` is in your expanded raw directory. If it isn't, it's likely that when you expanded the RAW download, you ended up with unnecessary layers of directories. Find the directory with `setup.py` in it, and make that the top level folder.
6. Move the RAW files to Applications folder
   - Move the folder that contains all of the RAW files to the `Applications` folder. As above, this would be the folder with `setup.py` in it.
   - Rename the folder that you just moved to `raw`.
7. In a terminal, change directory into the top level RAW folder
   - If you used the suggested path of `Applications/raw` type: `cd /Applications/raw`
8. Build the extensions.
   - `python setup.py build_ext --inplace`
9. Navigate to the `bioxtasraw` subfolder
   - From the top level RAW folder it should be `cd ./bioxtasraw`
10. Run RAW
    - `pythonw RAW.py`
11. Enjoy!
    - In the future, you can start RAW as in the previous step.
    - If RAW doesn't work, check out the *solutions to common problems*

Common problems/troubleshooting

Installing the prebuilt app package:

- Because the RAW team is an unidentified developer, you may get a warning message the first time you run the program. If that happens, right click on RAW and select *Open* from the right click menu, and then click the *Open* button in the window that appears.
  - This requires administrator privileges
- If the above doesn't work, you can run the RAW app from the command line. Navigate to `RAW.app/Contents/MacOS` and run the RAW executable file (`./RAW`) in that directory.

Installing from source:

- If you fail to build the extensions before running RAW, RAW will crash at some point. Be sure to run the `python setup.py build_ext --inplace` before starting RAW.
- Using the RAW autorg function requires a relatively recent version of numba. If you get an error running this function update your numba version to the most recent.
- If you are installing on an older Macbook (older than ~2012) you need to install all packages through conda forge (`conda install -c conda-forge <package_name>`). The new ones in the main conda channel cuase an illegal instruction error. See this thread: https://github.com/conda/conda/issues/9678. It's possible this will be resolved at some point and the main channel will be usable for older machines again.

RAW Install Guide for Linux

There are two ways to install RAW on Linux. The easiest, and recommended, way is to use a prebuilt debian/ubuntu package (`.deb` file). This will work on Debian 8 and newer, Ubuntu 14.04 LTS and newer, and any similar Ubuntu-based OSes like Linux Mint.

If you are on a different OS, such as CentOS, or you want to install a version for which there is no prebuilt installer, you will need to install RAW from the source code.

Here you can find information on:

Using a prebuilt .deb package

The recommended way to install RAW on Linux is using a prebuilt .deb package. To install from a prebuilt app package:

1. Download the `RAW-x.y.z-linux-amd64.deb` (where `x.y.z` is the version number) file from source-forge:
   - https://sourceforge.net/projects/bioxtasraw
   - Be sure to save the file, rather than opening it with a program.
2. Open a terminal and navigate to where you saved the downloads (usually: `cd ~/Downloads`)
3. Run the following command to install, replacing the version number with the correct version:
   - `sudo dpkg --install RAW-x.y.z-linux-amd64.deb`
4. You may now run RAW either by:
   1. Using the command `bioxtas-raw` from the command line
   2. Opening the 'BioXTAS RAW' program from your Activities menu or similar.

Note: if you wish to uninstall a version of RAW installed this way, simply use the following command: `sudo apt-get remove bioxtas-raw`

General instructions for installing from source (advanced users)

1. Install python 3.X (if it isn't already installed), RAW is tested on 3.7 and 3.8.
2. Install python3 development tools and gcc (if they are not already installed).
3. Install the following python packages (version indicated if less than most recent):
   - numpy
   - scipy
   - matplotlib
   - pillow
   - wxpython
   - h5py
   - cython
   - fabio
   - pyFAI
   - hdf5plugin
   - numba
   - dbus-python
   - reportlab
   - mmcif_pdbx
4. Download RAW source code from sourceforge (https://sourceforge.net/projects/bioxtasraw/files)
5. Extract RAW to a directory of your choice.
6. In the top level RAW directory run `python setup.py build_ext --inplace` to build the extensions.
7. In the `bioxtasraw` subdirectory run `RAW.py` using python.
8. Enjoy!
   - If you have problems, please consult the detailed installation guides and check out the *solutions to common problems*.
   - If you want, see the section on *making a desktop shortcut for RAW*.

Notes for python 2 installation

As of version 2.0.0, RAW is Python 3 compatible. The last guaranteed Python 2 compatible version of RAW is 2.0.0. However, it may still be possible to install RAW for Python 2. A few additional notes for that:

1. Additional dependencies:
   - future
2. Required version of pyFAI is 0.17

Linux install from source instructions

1. Open a new terminal window (in many distros you can right click on the desktop and select *New Terminal* or *Open in terminal*).
2. Download Miniconda Python (Python 3.x, e.g. 3.8) distribution and install.
   - https://docs.conda.io/en/latest/miniconda.html
   - Make sure you chose the python 3.x installer.
   - Save to the downloads folder
   - Open a terminal and run the following commands:
     - `cd ~/Downloads`
     - `bash ./Miniconda3-version.sh` (where `-version` which will depend on the version you download)
   - Accept the default installation location.
   - At the end, say "yes" to have the conda python install put in your system path.
   - Close the terminal window.
3. Install python packages. Open a new terminal window and run the following commands (in many distros you can right click on the desktop and select *New Terminal* or *Open in terminal*).
   - `conda upgrade conda pip wheel setuptools`
   - `conda install numpy scipy matplotlib pillow numba h5py cython numexpr reportlab`
   - `conda install -c conda-forge wxpython dbus-python fabio pyfai hdf5plugin mmcif_pdbx`
4. Download RAW source code from sourceforge
   - https://sourceforge.net/projects/bioxtasraw/files
   - Go to the linked website and download the `RAW-x.y.z-Source.zip` file, where `x.y.z` is the version number (for example, 1.0.0).
5. Expand the RAW download to your location of choice.
   - We suggest `~/raw`
   - Make sure there are no spaces in the file path (you can check by navigating to the raw directory in a terminal window and using `pwd`).
   - In the terminal or in the graphical file manager, confirm that the file named `setup.py` is in your raw directory. If it isn't, it's likely that when you expanded the RAW download, you ended up with unnecessary layers of directories. Find the directory with `setup.py` in it, and make that the top level folder.
6. In a terminal, change directory into the top level RAW folder
   - If you used the suggested path of `~/raw` type: `cd ~/raw`
7. Build the extensions.
   - `python setup.py build_ext --inplace`
8. Navigate to the `bioxtasraw` subfolder
   - From the top level RAW folder it should be `cd ./bioxtasraw`
9. Run RAW

   - `python RAW.py`

10. RAW is now installed. Enjoy!
    - If you want, see the section on *making a desktop shortcut for RAW*.
    - If RAW doesn't work, check out the *solutions to common problems*.

Instructions for setting up a RAW desktop shortcut from source

All files referred to are initially located in the RAW `LinuxLib` folder.

1. Add the `start_raw` file to your path:
   - `sudo cp ~/raw/LinuxLib/start_raw /usr/local/bin`
2. Make the `start_raw` file executable:
   - `sudo chmod +x /usr/local/bin/start_raw`
3. Copy the `RAW.desktop` file to the desktop:
   - `cp ~/raw/LinuxLib/RAW.desktop ~/Desktop/`
4. Right click on the `RAW` file on the desktop, and select *Properties*
5. Click on the Permissions tab, and make sure *Allow executing file as program* is checked.
6. Note: depending your distribution/shell, you have to edit the `start_raw` file to use a different shell. By default it uses bash.

Common problems/troubleshooting

- Sometimes, compilers can have trouble if there are spaces in the filepath. Try installing RAW so that there are no spaces in the file path (navigate the folder in the terminal, type `pwd` and see what the result is).
- If you have installed a standalone python distribution (such as Enthought Canopy or miniconda/anaconda), it is possible that it isn't set to default, so when you run `python RAW.py`, you are using the wrong python.
  - You can verify which python you are using the command `which python` in the terminal.
  - You can set the correct python to default by modifying your appropriate profile file (such as the .bash_profile), or setting the `$PATH` environmental variable.
  - You can also specify the full path to the version of python you want to use in the command, such as `~/miniconda2/bin/python`
- Note that when you change environmental variables in one terminal window, you need to restart other windows for this to take effect. If you aren't using the right python (or compiler, etc), trying closing all of your terminal windows and opening a new one.
- If you fail to build the extensions before running RAW, RAW will crash at some point. Be sure to run the `python setup.py build_ext --inplace` before starting RAW.
- Using the RAW autorg function requires a relatively recent version of numba. If you get an error running this function update your numba version to the most recent.
- On Scientific Linux 6, and thus probably on Red Hat 6 (untested), RAW completely fails to work with wxpython 3.0 and certain python distributions (namely the Enthought python).
  - If RAW completely doesn't start, check and make sure you have wxpython 2.8 installed. This requires that you have matplotlib<=1.4.

  - In general, RAW should work with wxpython 3 and 4.

- As for the last time we tested it (2017) the Enthought Canopy python package DOESN'T WORK on Ubuntu or Linux Mint with wxpython.

5.2 Getting Help

5.2.1 General Questions/Comments

For questions on how to use the software, first check out the documentation on this site.

If that doesn't answer your question, or for other comments, please contact the RAW mailing list.

5.2.2 Reporting A Bug

We appreciate any and all bug reports, it helps us improve the software! If you have a bug to report, the best way is to contact the RAW mailing list.

If you have a bug report, please include: operating system, RAW version number, installation type (package or from source), what you were doing when the bug occurred, the appropriate error message(s) and sample data so we can recreate the bug. That will enable us to find and solve the problem as quickly as possible.

5.3 RAW Tutorial

5.3.1 Introduction

Overview

This tutorial covers SAXS data processing with RAW. You will learn how to:

- Process images into scattering profiles
- Average, subtract and save scattering profiles
- Find R_g and I(0) by Guinier analysis
- Find molecular weight by six different methods
- Do Kratky analysis and dimensionless Kratky analysis
- Test the similarity of scattering profiles
- Load and process SEC-SAXS data
- Carry out singular value decomposition (SVD) and evolving factor analysis (EFA) to evaluate and deconvolve SEC-SAXS data
- Carry out regularized alternating least squares (REGALS) analysis to deconvolve SAXS data.
- Do baseline correction on SEC-SAXS data
- Merge SAXS/WAXS data from two detectors
- Carry out Pair-distance distribution analysis (BIFT and GNOM)
- Evaluate ambiguity of 3D shape reconstructions (AMBIMETER)

- 3D reconstructions with bead models (DAMMIF/N and DAMAVER)
- 3D reconstructions with electron density (DENSS)
- Save your analysis information
- Calibrate RAW for integrating images
- Mask images for integration
- Set up normalization and save processing settings
- Set absolute scaling in RAW using water and glassy carbon
- Set a molecular weight standard in RAW

Section 1 covers basic processing with RAW, and *Section 2* covers advanced processing with RAW. *Section 3* covers how to set up RAW for integrating images for those who do not already have a configuration file.

This tutorial is focused on how to use RAW, it is not necessarily a tutorial on best practices for SAXS data processing. The *SAXS tutorial* covers some basic processing and analysis best practices.

Requirements

- BioXTAS RAW >= v2.1.0 (most recent is best).
  - *Install instructions*
- Tutorial data.
  - Available from: https://sourceforge.net/projects/bioxtasraw/files/?source=navbar
- ATSAS programs, >= v2.8.0 (for parts of *Section 2* of the tutorial).
  - Download and installation instructions are available from: https://www.embl-hamburg.de/biosaxs/download.html
  - Requires a free registration for academic users. Industrial users must pay to use.

Other useful materials

1. Video lectures from BioCAT's Everything BioSAXS workshops, which can help you learn more about best practices for SAXS data processing.
2. Each tutorial section has a linked video tutorial. A full playlist of the videos is available here:
3. ATSAS resources:
   - Manuals: https://www.embl-hamburg.de/biosaxs/manuals/
   - User forum: https://www.saxier.org/forum/
4. Electron density (DENSS) resources available at DENSS.org
   - Particularly useful is the section on visualizing the results and aligning with known structures.

Notes

If you are only interested in using RAW to process data, and are not interested in how to set up RAW to calibrate your data, you do not need to look at *Section 3*.

RAW depends on user feedback to get better. If you have questions, find bugs, or think a part of this tutorial is unclear, *please let the developers know.*

You can find additional developer contact information on the RAW website: https://sourceforge.net/projects/bioxtasraw/

5.3.2 Basic processing

This section will guide you through basic processing of files in RAW. It includes: integrating images into scattering profiles, averaging and subtracting scattering profiles, doing Guinier fits, MW analysis, processing SEC-SAXS data, processing data from 2 detectors (SAXS and WAXS data), saving analysis information, and a few other things. It refers to the *RAW tutorial data*.

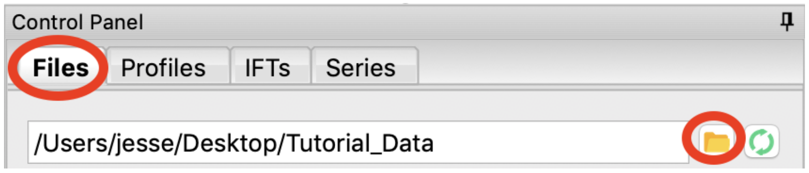
Select a section below to view the tutorial, or use the next and back buttons at the bottom of the page to navigate through it in order.

Loading configuration files and images, creating subtracted scattering profiles, saving profiles

A video version of this tutorial is available:

The written version of the tutorial follows.

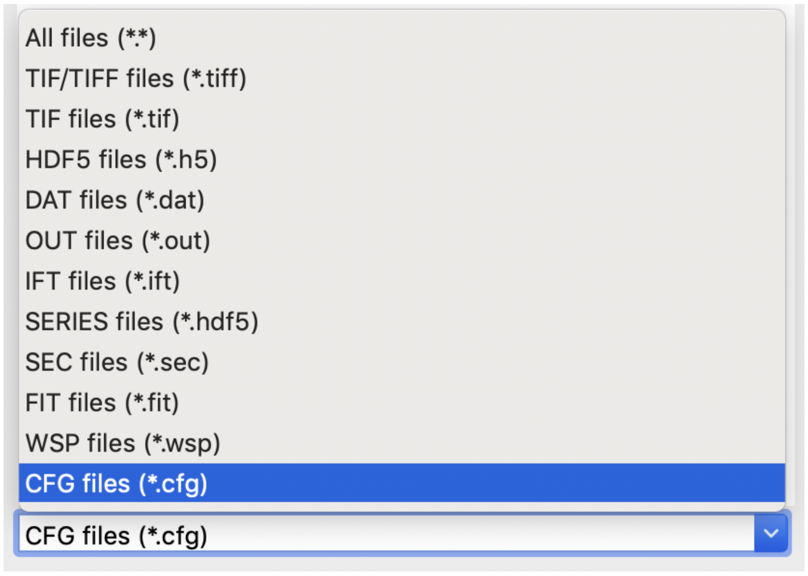
1. Open RAW. The install instructions contain information on installing and running RAW.
2. In the files tab, click on the folder button and navigate to the **Tutorial_Data/standards_data** folder. Click the open button to show that folder in the RAW file browser.



/Users/jesse/Desktop/Tutorial_Data

   - *Tip:* You can navigate within the file list as well. Use the up arrow (filename: "**..**"") to move up a directory level or double click on a directory to open it.

3. At the bottom of the File tab in the Control Panel, use the drop-down menu to set the file type filter to "CFG files (*.cfg)".



4. Double click on the **SAXS.cfg** file to load the SAXS configuration. This loads the beamline configuration into the program.

   - *Note:* Any time you are going to process images, you need to load the appropriate configuration!

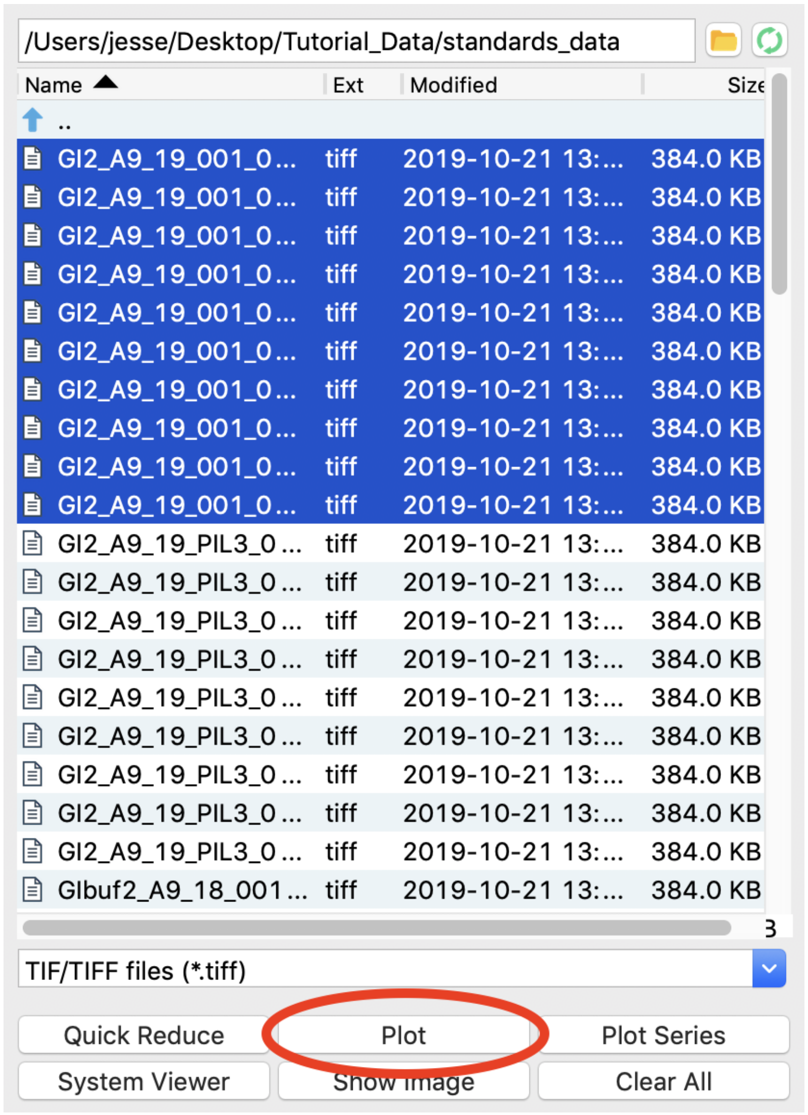
5. Change the file type filter to "TIFF files (*.tiff)".

6. At the CHESS G1 station, where this data was taken, typically ~10 images were collected from a given sample. To load in the 10 images for the glucose isomerase (GI) sample, start by selecting the files **GI2_A9_19_001_xxxx.tiff**, where **xxxx** will range from **0000** to **0009**. These files are measured scattering from 0.47 mg/ml GI.

   - *Tip:* you can hold down the ctrl key (apple key on macs) while clicking to select multiple files individually. You can also click on a file, and then shift click on another file to select those files and everything between them.
   - *Warning:* Don't load the files with **PIL3** in their name. Those are the wide-angle scattering (WAXS) data, which we will process separately later.

7. Click the plot button to integrate all of the images and plot the integrated scattering profiles on the Profiles plot.

   - *Note:* Typically, once the images are integrated we work only with the scattering profiles. However, it is useful to keep the images around in case you want to reprocess the data.

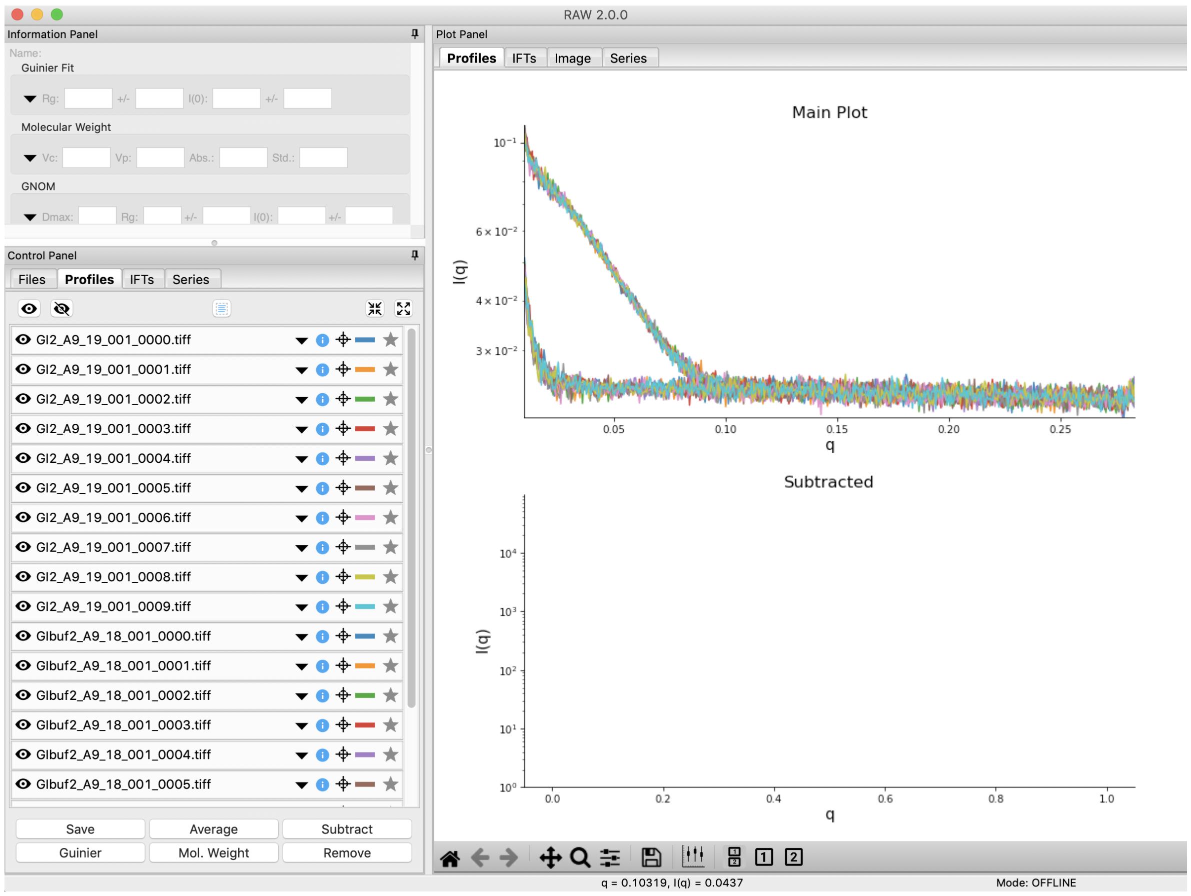


/Users/jesse/Desktop/Tutorial_Data/standards_data

Name	Ext	Modified	Size
..			
GI2_A9_19_001_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_001_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_001_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_001_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_001_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_001_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_001_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_001_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_001_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_001_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_PIL3_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_PIL3_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_PIL3_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_PIL3_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_PIL3_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_PIL3_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_PIL3_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_PIL3_0...	tiff	2019-10-21 13:...	384.0 KB
GI2_A9_19_PIL3_0...	tiff	2019-10-21 13:...	384.0 KB
GIbuf2_A9_18_001...	tiff	2019-10-21 13:...	384.0 KB

TIF/TIFF files (*.tiff)

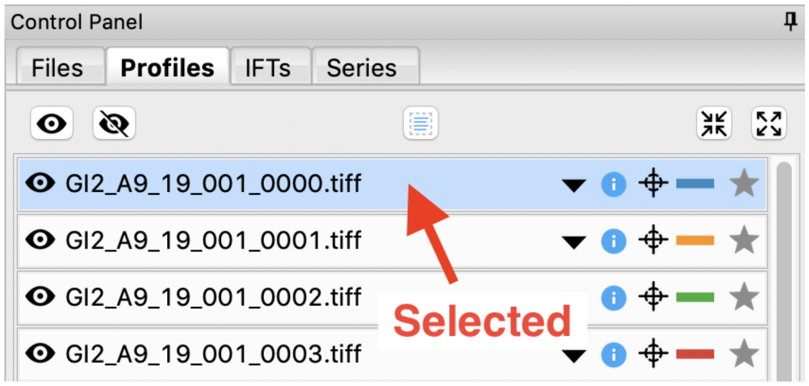
8. Plot the **GIbuf2** scattering profiles from the images. These are measured scattering from the matching buffer, without any protein, for the GI sample.
9. Click on the Profiles control tab. This is where you can see what scattering profiles are loaded into RAW, and manipulate/analyze them.
    - *Checkpoint:* If you've successfully loaded the images given, you should see twenty scattering profiles in the profiles list, with names like **GI2_A9_19_001_0000.tiff** or **GIbuf2_A9_18_001_0000.tif**.



GI2_A9_19_001_0000.tiff
GI2_A9_19_001_0001.tiff
GI2_A9_19_001_0002.tiff
GI2_A9_19_001_0003.tiff
GI2_A9_19_001_0004.tiff
GI2_A9_19_001_0005.tiff
GI2_A9_19_001_0006.tiff
GI2_A9_19_001_0007.tiff
GI2_A9_19_001_0008.tiff
GI2_A9_19_001_0009.tiff
GIbuf2_A9_18_001_0000.tiff
GIbuf2_A9_18_001_0001.tiff
GIbuf2_A9_18_001_0002.tiff
GIbuf2_A9_18_001_0003.tiff
GIbuf2_A9_18_001_0004.tiff
GIbuf2_A9_18_001_0005.tiff

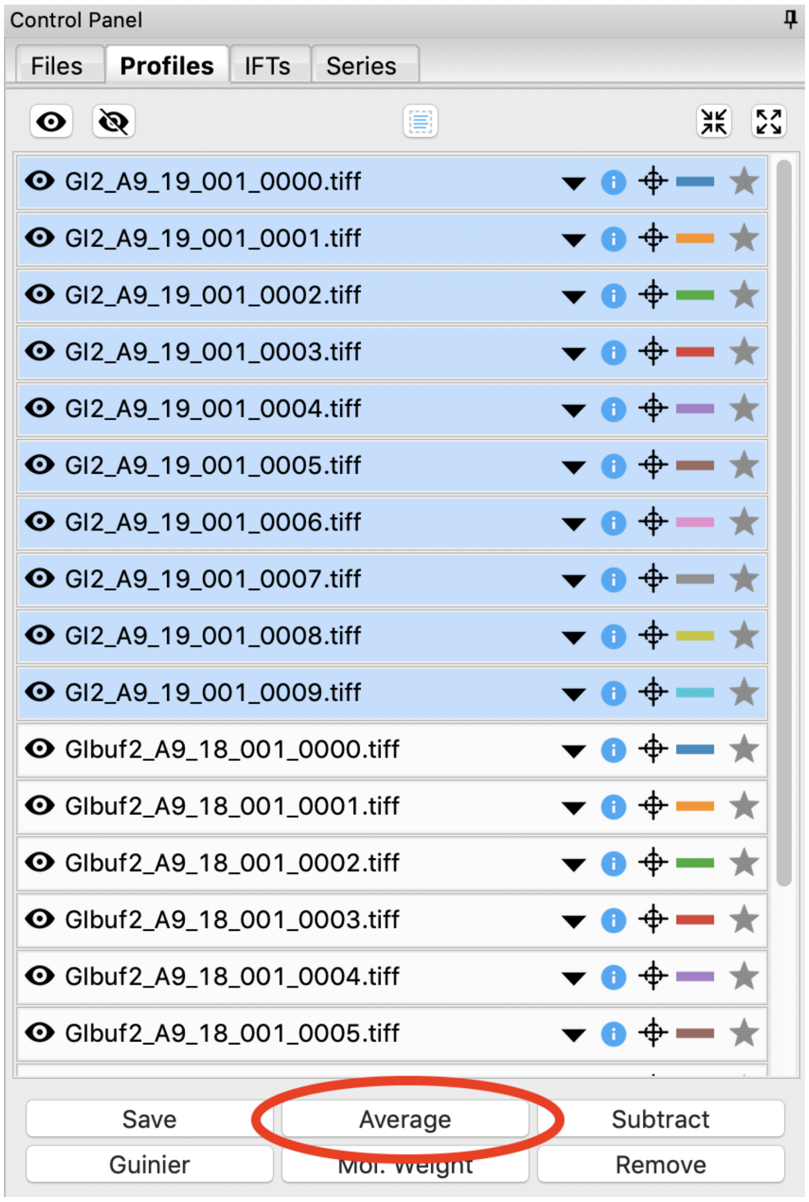
q = 0.10319, I(q) = 0.0437 Mode: OFFLINE

10. Click on a filename to select the scattering profile. The background should turn blue, indicating it is selected.



GI2_A9_19_001_0000.tiff
GI2_A9_19_001_0001.tiff
GI2_A9_19_001_0002.tiff
GI2_A9_19_001_0003.tiff

11. Select all of the GI scattering profiles
    - *Tip:* Again, the ctrl(/apple) key or the shift key can be used to select multiple scattering profiles.
    - *Warning:* Select only the GI profiles, not the GI buffer profiles.



GI2_A9_19_001_0000.tiff
GI2_A9_19_001_0001.tiff
GI2_A9_19_001_0002.tiff
GI2_A9_19_001_0003.tiff
GI2_A9_19_001_0004.tiff
GI2_A9_19_001_0005.tiff
GI2_A9_19_001_0006.tiff
GI2_A9_19_001_0007.tiff
GI2_A9_19_001_0008.tiff
GI2_A9_19_001_0009.tiff
GIbuf2_A9_18_001_0000.tiff
GIbuf2_A9_18_001_0001.tiff
GIbuf2_A9_18_001_0002.tiff
GIbuf2_A9_18_001_0003.tiff
GIbuf2_A9_18_001_0004.tiff
GIbuf2_A9_18_001_0005.tiff

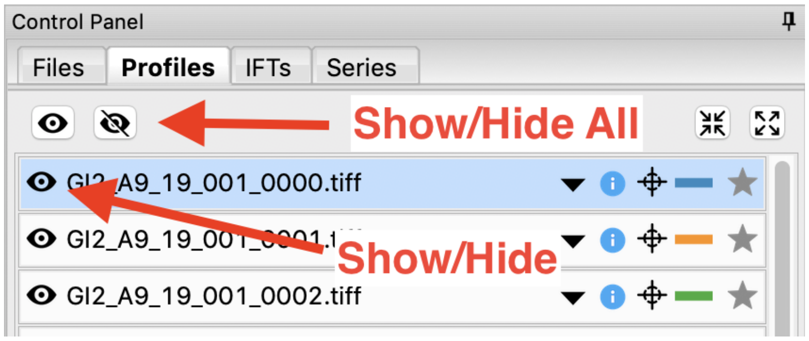
12. Use the average button to average all of the scattering profiles collected into a single curve.

    - *Checkpoint:* The averaged scattering profile should appear at the bottom of the Profiles list. You may have to scroll down to see it. The filename will be in green, and will start with **A_**, indicating it is an averaged scattering profile.

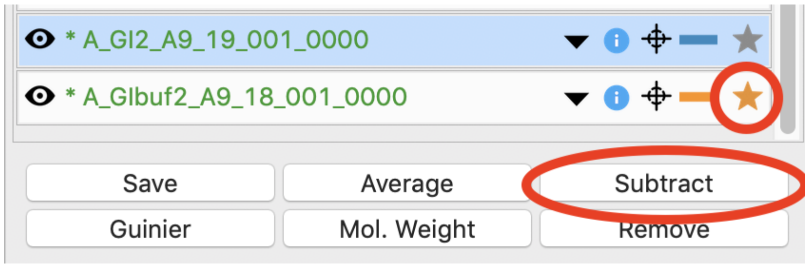
13. Average all of the GI buffer scattering profiles.

14. In order to clearly see the averaged scattering profiles, you will need to hide the individual profiles from the plot. Clicking on the eye to the left of the filename will show/hide a scattering profile. When the eye is shown, the profile is shown on the plot, when the eye has a line through it, the profile is hidden. Hide all of the profiles except the two averaged curves.

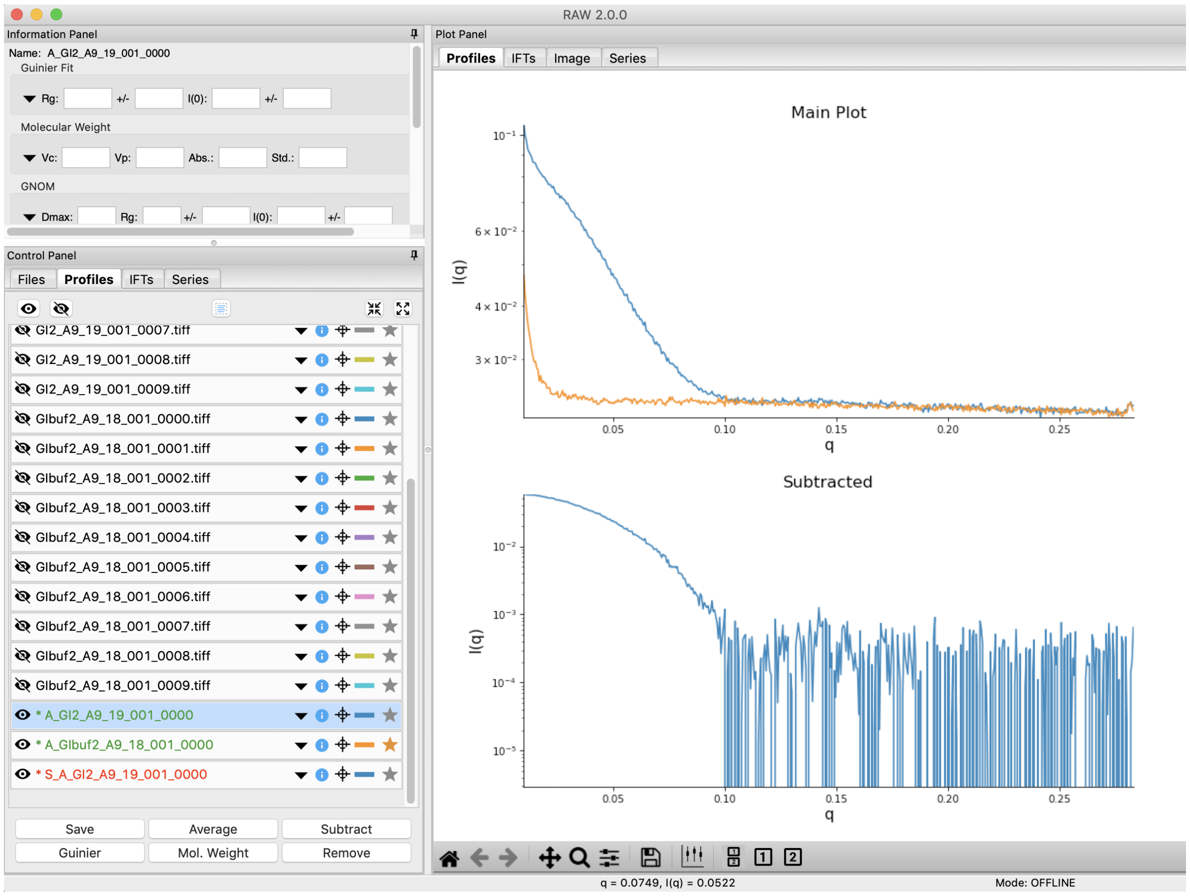
    - *Tip:* The eye and eye with the line through it at the top of the Profiles panel can be used to show/hide sets of loaded profiles at once. If no profiles are selected, these buttons show/hide all loaded profiles. If some profiles are selected, these buttons show/hide just the selected profiles. Try selecting all but the averaged files and using the show/hide all buttons.



15. Next you need to subtract the buffer scattering profile from the measured protein scattering (which is really the scattering of the protein plus the scattering of the buffer). Star the averaged buffer file, and select the averaged protein file, then click the subtract button.



- *Checkpoint:* The subtracted scattering profile should be shown in the lower plot. A new profile should be shown in the Profiles list with the name in red and a **S_** prefix indicating it is a subtracted file.



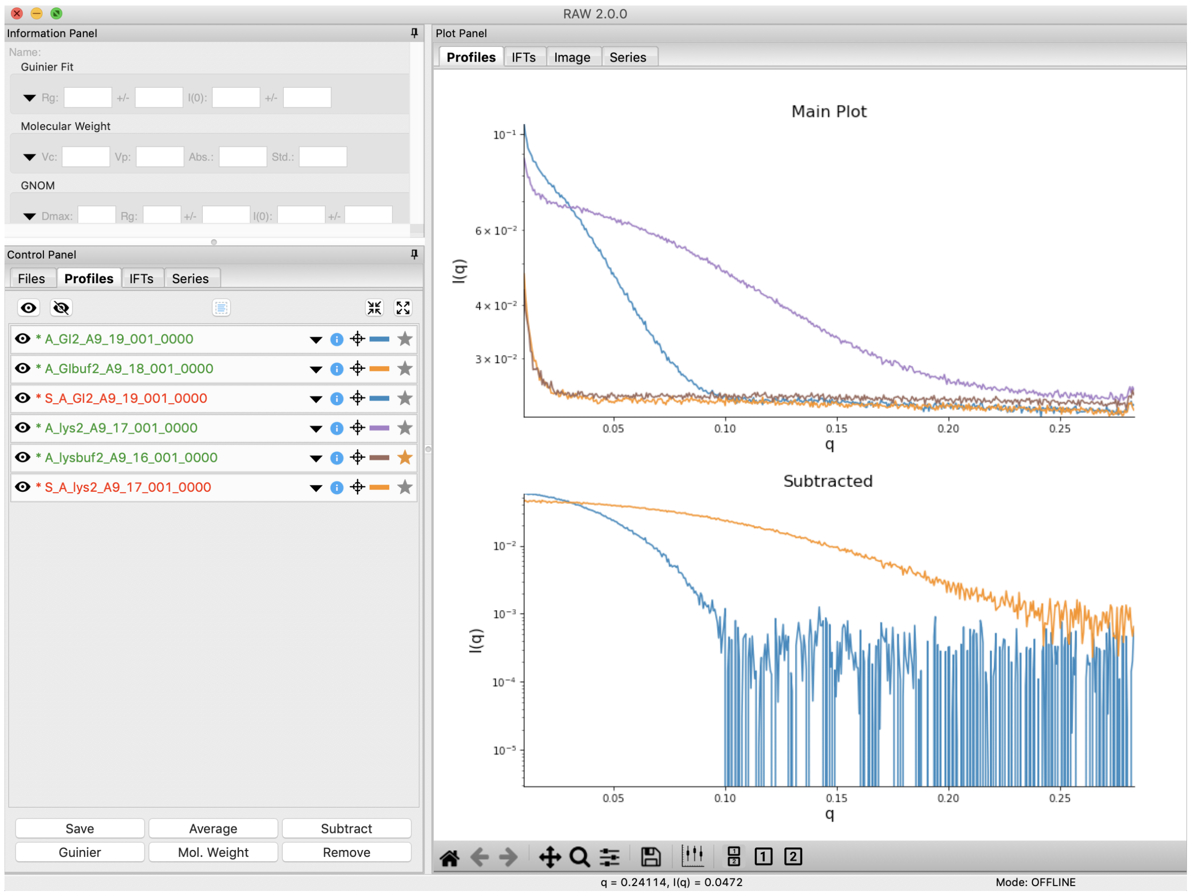
16. You don't need the individual image scattering profiles any more. Select all of those (but not your averaged or subtracted profiles!) and click remove.

    - *Note:* This only removes the scattering profiles from RAW. The images on your hard drive are unaffected.

17. You can also load files into RAW by dragging and dropping files onto the RAW window. Load in the **lys2** images by selecting them in your file browser, then dragging them onto the top plot.

18. Load in the **lysbuf2** files. Average both the lysozyme and buffer data, and subtract to create a subtracted lysozyme scattering profile. The concentration of this sample was 4.27 mg/ml. Remove all of the profiles that are not averaged or subtracted profiles.

    - *Tip:* In order to tell which curve is which in a plot, click on the target icon in the Profiles list. This should bold that curve in the plot. Click the target icon again to return the curve to normal.

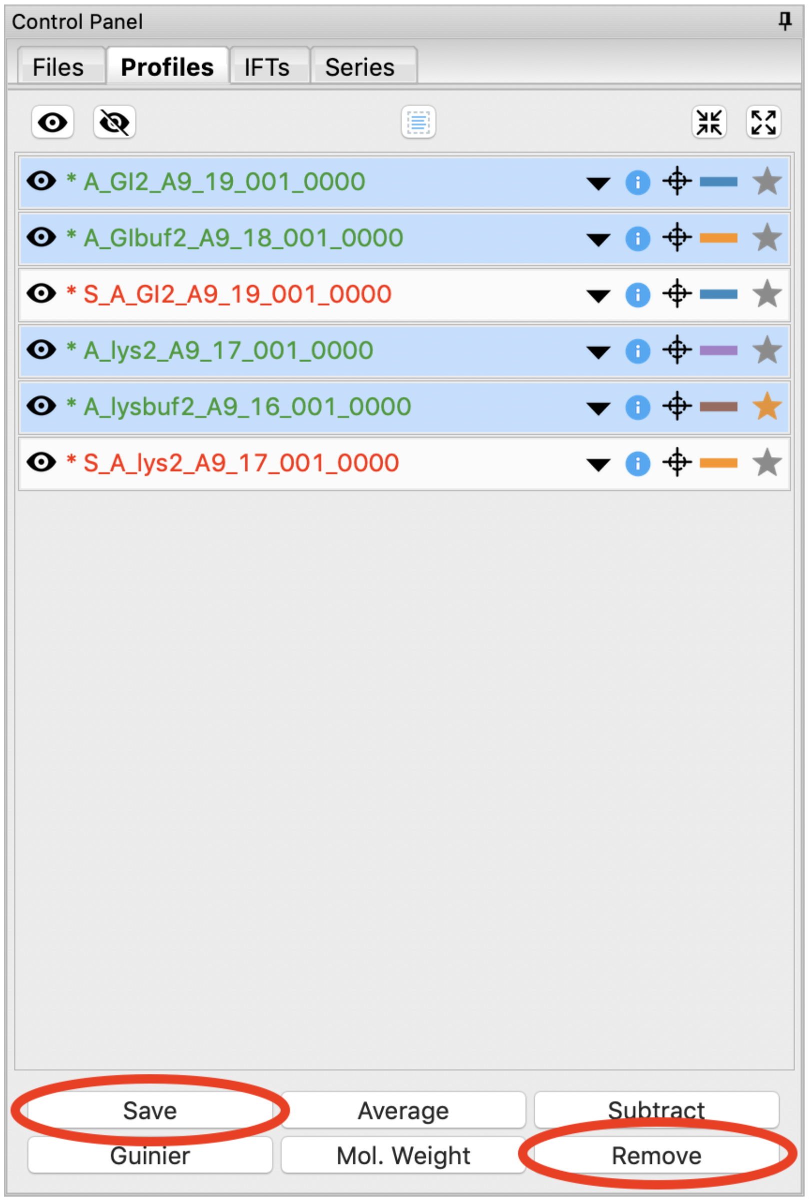


* A_GI2_A9_19_001_0000
* A_GIbuf2_A9_18_001_0000
* S_A_GI2_A9_19_001_0000
* A_lys2_A9_17_001_0000
* A_lysbuf2_A9_16_001_0000
* S_A_lys2_A9_17_001_0000

q = 0.24114, I(q) = 0.0472   Mode: OFFLINE

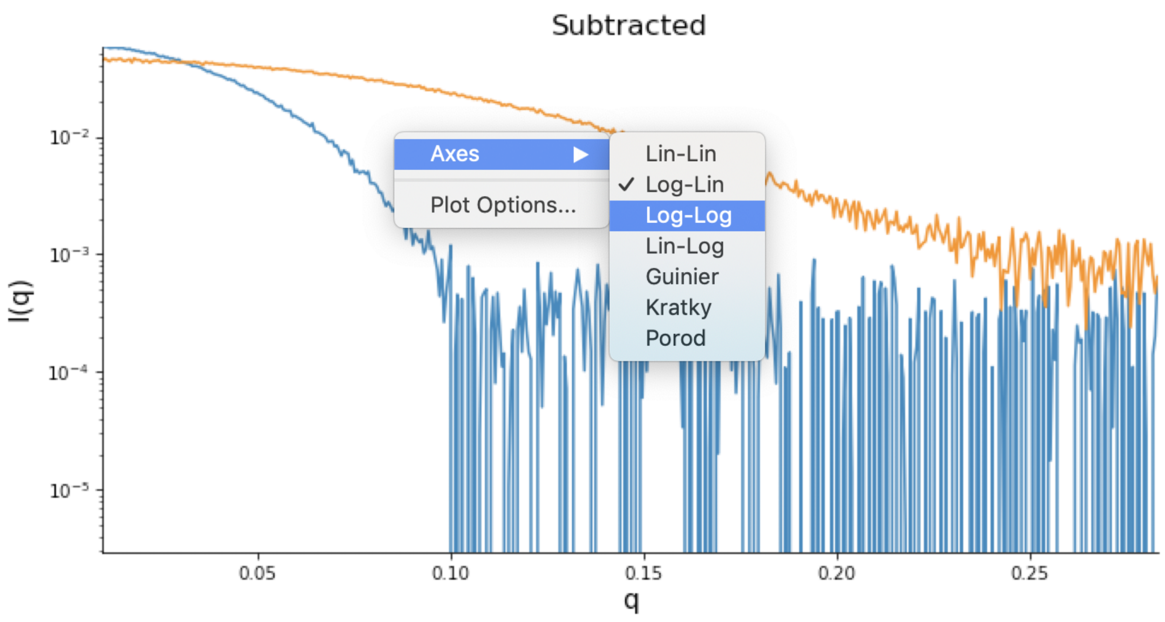
19. We're done with the averaged profiles. Select all of the averaged profiles and click the "Save" button to save them in the **standards_data** folder. Note that in the filename in the Profiles list, the * at the front goes away. This indicates there are no unsaved changes to those scattering profiles. You can now remove them.

    - *Note:* This saves them with a **.dat** extension. This is the standard format for SAXS scattering profiles, and is also human readable.



20. Right click on the subtracted plot, move the cursor over 'Axes' and select the Log-Log option.

- *Note:* It is best practice to display SAXS data, particularly in publications, on either a semi-log (Log-Lin, default option in RAW) or double-log plot (depending on the features of interest).
- *Note:* Well-behaved globular proteins will intersect the intensity axis roughly perpendicularly.



Guinier analysis

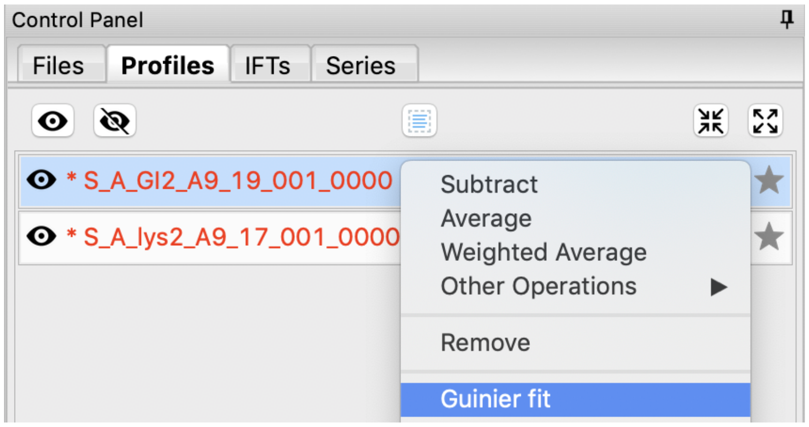
This tutorial covers how to use RAW for Guinier analysis. This is not a tutorial on basic principles and best practices for doing a Guinier analysis. For that, please see the *SAXS tutorial*.

A video version of this tutorial is available:

The written version of the tutorial follows.

1. In RAW, right click (ctrl click on macs without a right mouse button) on the subtracted GI scattering profile in the Profiles list and select “Guinier fit”. The Guinier fit window will open.

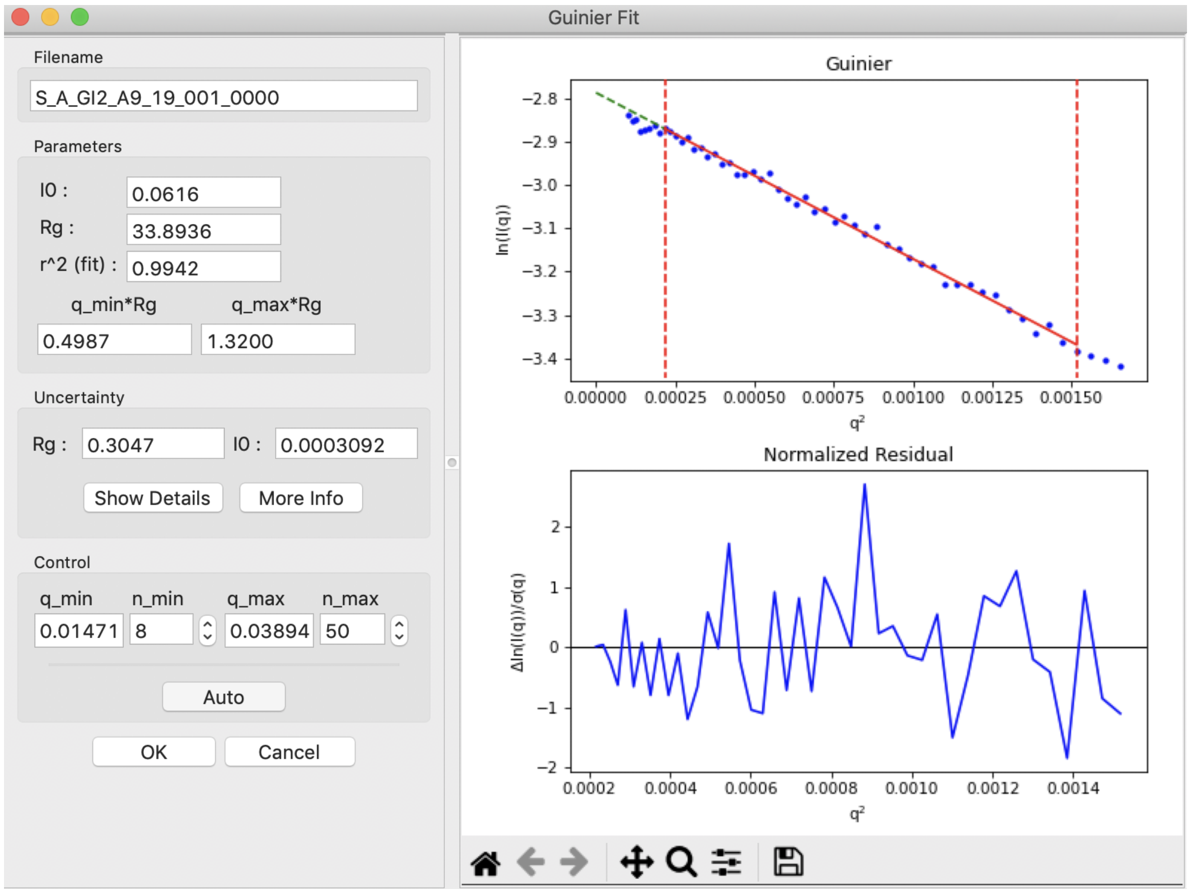
   - *Note:* You can also click the ‘Guinier’ button at the bottom of the Profiles control panel.



2. In the Guinier window, the top plot shows you the Guinier plot and the fit, while the bottom plot shows you the residual of the fit.

   - *Note:* RAW automatically tries to find the best Guinier region for you when the Guinier window is opened for the first time.

   - *Note:* The R_g value is in units of 1/q (e.g. if q is in Å^{-1} then R_g is in Å).



3. In the "Control" panel, you'll see that n_min is 8. This means RAW has skipped the first few low q points for the Guinier fit. You can see a little dip in the lowest q values, which may be why it was skipped. Use the arrow buttons next to the n_min box to adjust it down several points to include that dip and check whether the R_g changes. Once you're done return n_min to 8.

4. In the "Parameters" panel, note that $q_{max}R_g$ is ~1.32. Recall that for globular proteins like GI, it is typical to have $q_{max}R_g$ ~1.3. Adjust n_max down slightly until that is the case, watching what happens to the R_g and the residual.

   - *Question:* The literature radius of gyration for GI is 32.7 Å. How does yours compare?

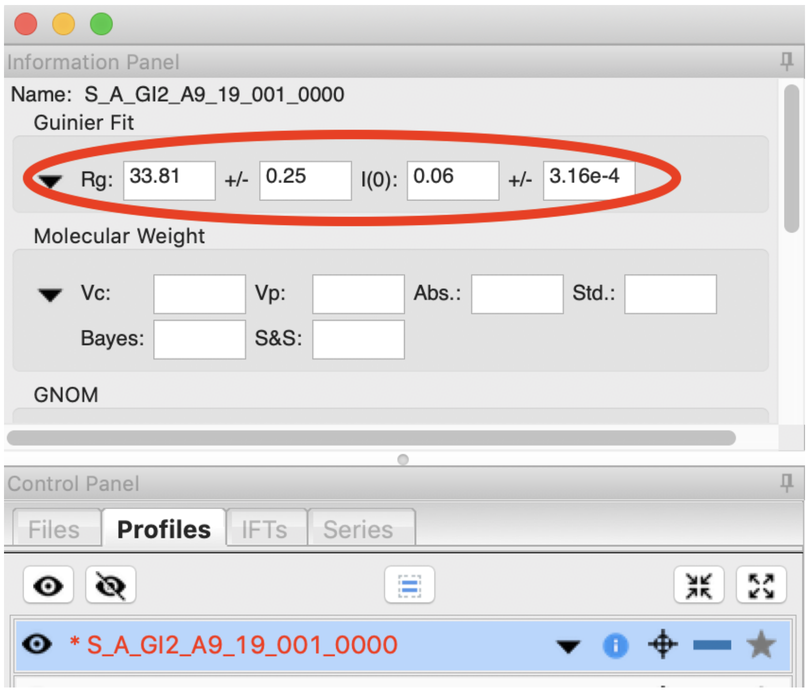
5. RAW also provides an estimate of the uncertainty in both the R_g and I(0) values for the Guinier fit, shown in the Uncertainty section.

   - *Note:* This is the largest of the uncertainties from the fit (standard deviation of fit values calculated from the covariance matrix), and either the standard deviation of R_g and I(0) across all acceptable intervals found by the automatic R_g function or an estimated uncertainty in R_g and I(0) based on variation of the selected interval start and end points.

6. Click the "OK" button to keep the results.

   - *Note:* Clicking the "Cancel" button will discard the results.

7. If you now select the GI scattering profile, in the information panel above the control panel you should see the R_g and I(0) that you just found.



Name: S_A_GI2_A9_19_001_0000

Guinier Fit

Rg: 33.81 +/- 0.25 I(0): 0.06 +/- 3.16e-4

Molecular Weight

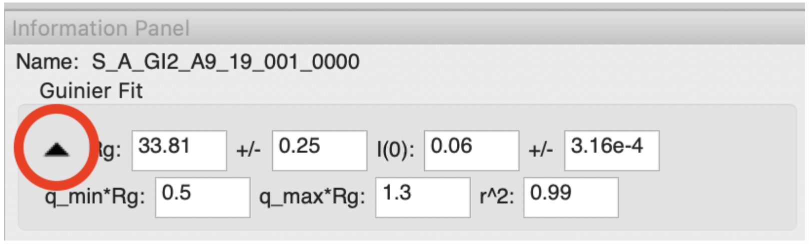
Vc: Vp: Abs.: Std.:

Bayes: S&S:

GNOM

* S_A_GI2_A9_19_001_0000

- *Tip:* Click on the triangle to expand the Guinier info section and see more details on the fit.



Name: S_A_GI2_A9_19_001_0000

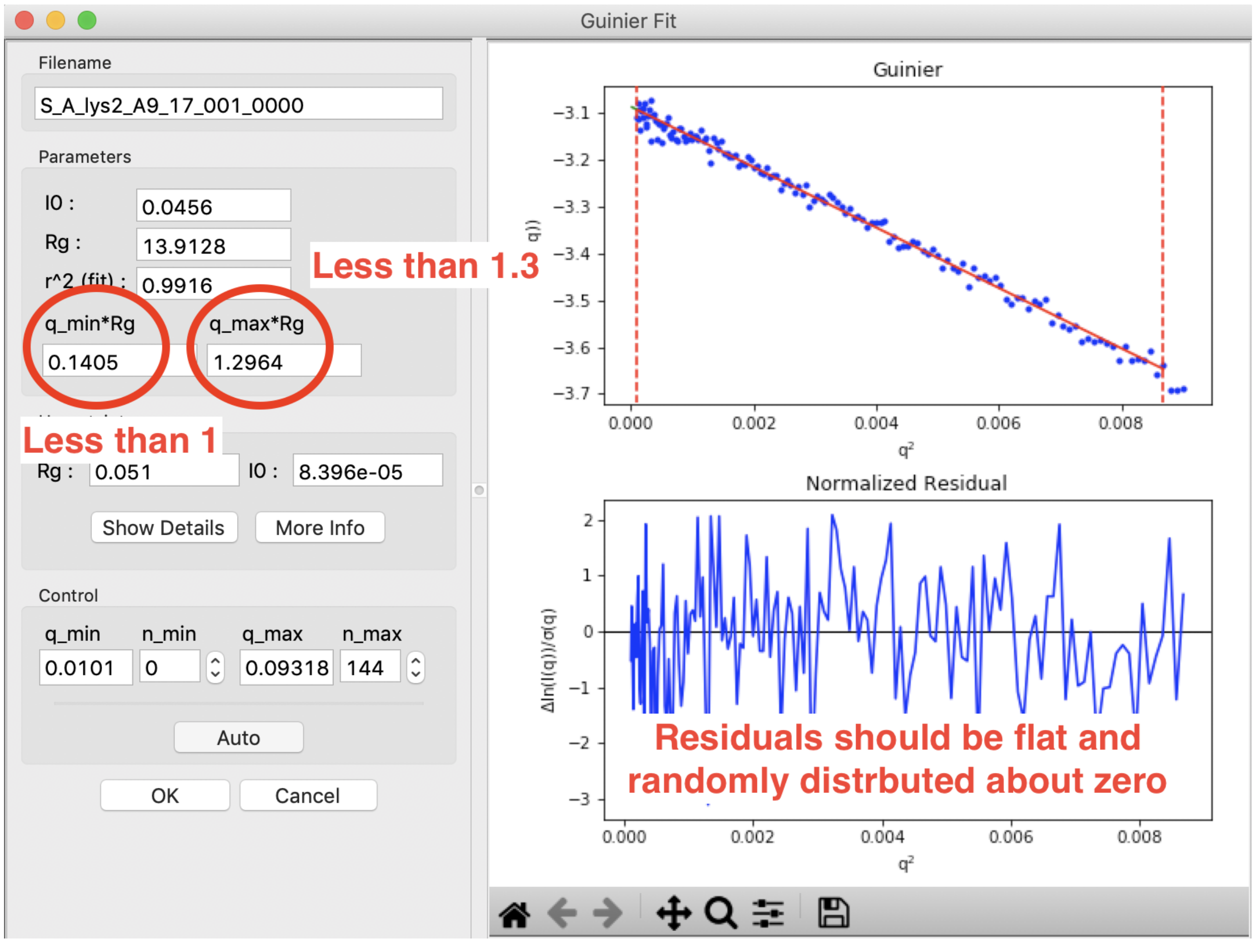
Guinier Fit

Rg: 33.81 +/- 0.25 I(0): 0.06 +/- 3.16e-4

q_min*Rg: 0.5 q_max*Rg: 1.3 r^2: 0.99

8. Repeat the Guinier analysis for lysozyme.

   - *Try:* Increase q_{min} and/or decrease q_{max} to verify that the R_g does not change significantly in the Guinier region.
   - *Tip:* If you hover your mouse cursor over the info icon (just left of the target icon) for a given scattering profile it should show you the R_g and I(0) of your Guinier analysis.



Filename: S_A_lys2_A9_17_001_0000

Parameters

I0 : 0.0456
Rg : 13.9128
r^2 (fit) : 0.9916
q_min*Rg: 0.1405
q_max*Rg: 1.2964

Rg : 0.051 I0 : 8.396e-05

Control

q_min	n_min	q_max	n_max
0.0101	0	0.09318	144

Molecular weight analysis

This tutorial covers how to use RAW for molecular weight analysis. This is not a tutorial on basic principles and best practices for molecular weight analysis. For that, please see the *SAXS tutorial*.

RAW provides four methods of molecular weight analysis:

- Referencing I(0) to that of a known standard
- From the volume of correlation using the method of Rambo and Tainer
- From the adjusted Porod volume using the method of Fisher et al.
- From the value of I(0) on an absolute scale.

If ATSAS is installed, RAW also provides an additional two methods of MW calculation using the ATSAS tools, for a total of six different methods:

- From classification by machine learning (ATSAS datclass/Shape&Size)
- From a Bayesian estimation based on concentration independent methods (ATSAS datmw bayes)

If you use these MW methods in RAW, in addition to the RAW paper please cite these papers:

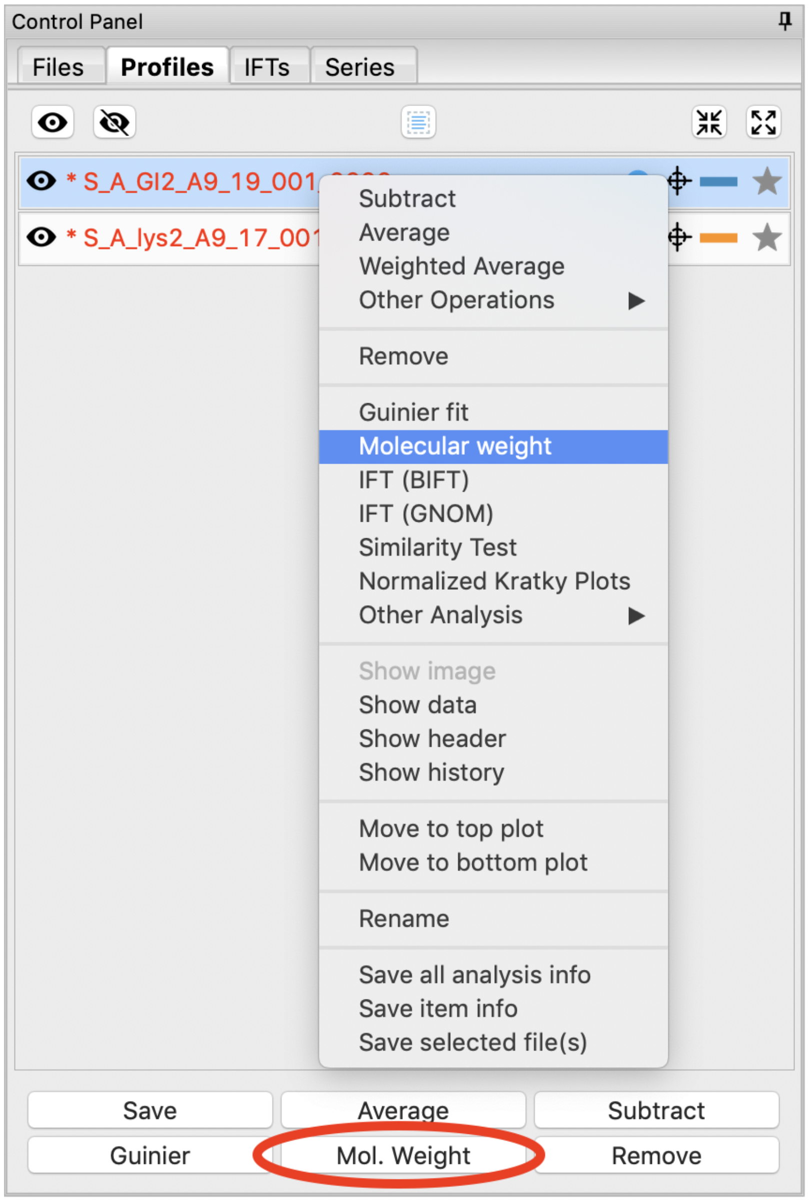
- Volume of correlation: Rambo, R.P. & Tainer, J.A. Nature (2013). 496, 477-481. DOI: 10.1038/nature12070
- Corrected Porod volume: V. Piiadov, E. Ares de Araujo, M. Oliveira Neto, A. F. Craievich, and I. Polikarpov. Protein Science (2019). 28(2), 454-473. DOI: 10.1002/pro.3528

- Bayesian inference (ATSAS): Hajizadeh, N. R., Franke, D., Jeffries, C. M. & Svergun, D. I. (2018). Sci. Rep. 8, 7204. DOI: 10.1038/s41598-018-25355-2
- Shape&Size (ATSAS): Franke, D., Jeffries, C. M. & Svergun, D. I. (2018). Biophys. J. 114, 2485–2492. DOI: 10.1016/j.bpj.2018.04.018

A video version of this tutorial is available:

The written version of the tutorial follows.

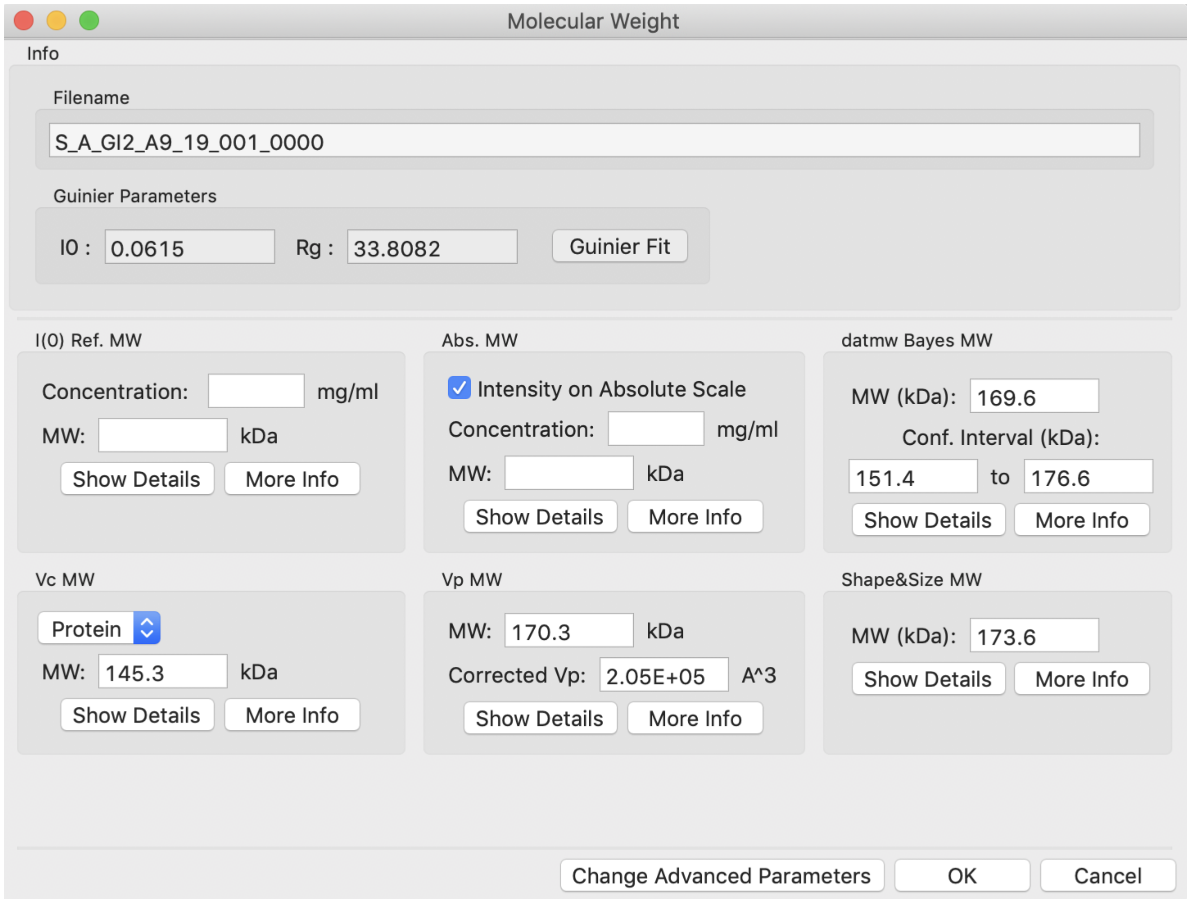
1. In RAW, right click on the subtracted GI scattering profile in the Profiles panel and select "Molecular weight." Alternatively click on the "Mol. Weight" button at the bottom of the Profiles panel.



2. At the top of the panel are the results of the Guinier fit. All methods require a good Guinier fit, so you can use that button to redo the fit if necessary. In the lower part of the panel, the results of the estimates for MW are shown.

   - *Note:* Neither the I(0) Ref. MW panel nor the Abs. MW panel should be reporting a MW.

- *Note:* If you have ATSAS installed and accessible to RAW, you should see six panels with MW, as in the image below. If you don't have ATSAS installed you will see just four panels, the rightmost panel in each row will be missing.
- *Tip:* To learn more about any of the methods, click on the "More Info" button.



Filename: S_A_GI2_A9_19_001_0000

I0 : 0.0615 Rg : 33.8082

I(0) Ref. MW — Concentration: mg/ml; MW: kDa

Abs. MW — ☑ Intensity on Absolute Scale; Concentration: mg/ml; MW: kDa

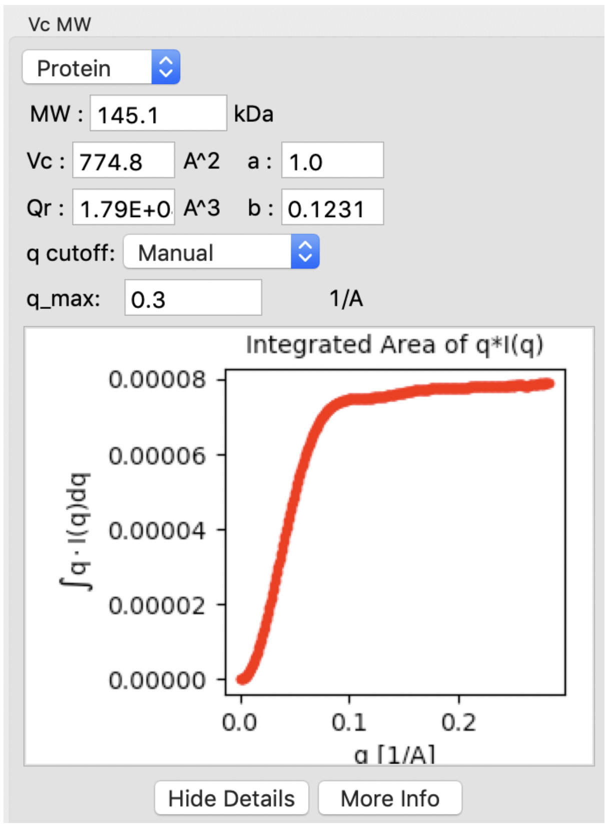
datmw Bayes MW — MW (kDa): 169.6; Conf. Interval (kDa): 151.4 to 176.6

Vc MW — Protein; MW: 145.3 kDa

Vp MW — MW: 170.3 kDa; Corrected Vp: 2.05E+05 A^3

Shape&Size MW — MW (kDa): 173.6

3. In either concentration box, enter the sample concentration of 0.47 mg/ml. Notice that you now get results from all methods of MW calculation.
   - *Question:* The expected MW value for GI is 172 kDa. How do your results compare?
4. Click on the "Show Details" button for the Vc MW panel. You should see a graph, which shows the integrated area of $qI(q)$ vs. q. For this method to be accurate, this value needs to converge at high q.


Vc MW

Protein

MW : 145.1 kDa

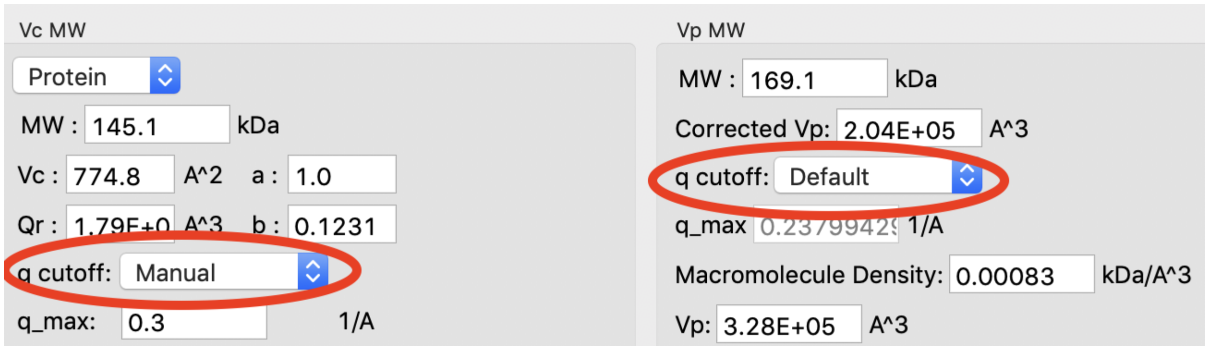
Vc : 774.8 A^2 a : 1.0

Qr : 1.79E+0 A^3 b : 0.1231

q cutoff: Manual

q_max: 0.3 1/A

5. Click on the "Show Details" for the Vp MW panel. You'll notice that both Vp MW and Vc MW have a "cutoff" selection. At higher q you can start to get scattering from flexibility or intra-molecular features that may reduce the reliability of the MW estimate. RAW automatically cuts off the scattering profile used based on the size of the object. You can change the cutoff if you need to.


Vc MW

Protein

MW : 145.1 kDa

Vc : 774.8 A^2 a : 1.0

Qr : 1.79E+0 A^3 b : 0.1231

q cutoff: Manual

q_max: 0.3 1/A

Vp MW

MW : 169.1 kDa

Corrected Vp: 2.04E+05 A^3

q cutoff: Default

q_max 0.23799429 1/A

Macromolecule Density: 0.00083 kDa/A^3

Vp: 3.28E+05 A^3

6. Click the "OK" button to save your analysis.
   - *Note:* The "Cancel" button discards the analysis.
   - *Tip:* After clicking "OK" you can now click on the GI profile in the Profiles control panel and see the MW you just found in the Info panel.
7. Repeat the MW analysis for the lysozyme sample, which had a concentration of 4.27 mg/ml. The expected MW of lysozyme is 14.3 kDa.
   - *Question:* Does the Vc method work for the lysozyme data?

Saving analysis information

There are several ways to save results in RAW, besides saving the processed files themselves. In this tutorial we go over how to create a summary PDF of your analysis, and how to save your analysis data to a spreadsheet (.csv file).

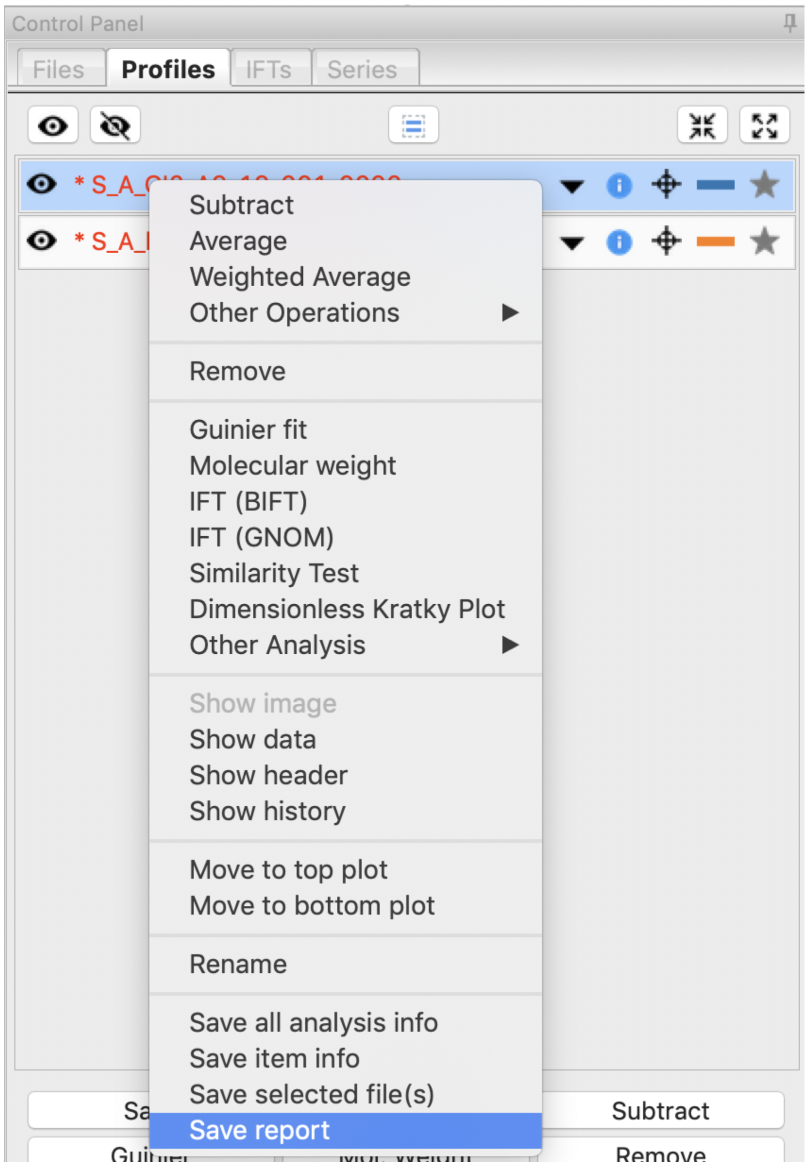
A video version of this tutorial is available:

Saving as a pdf

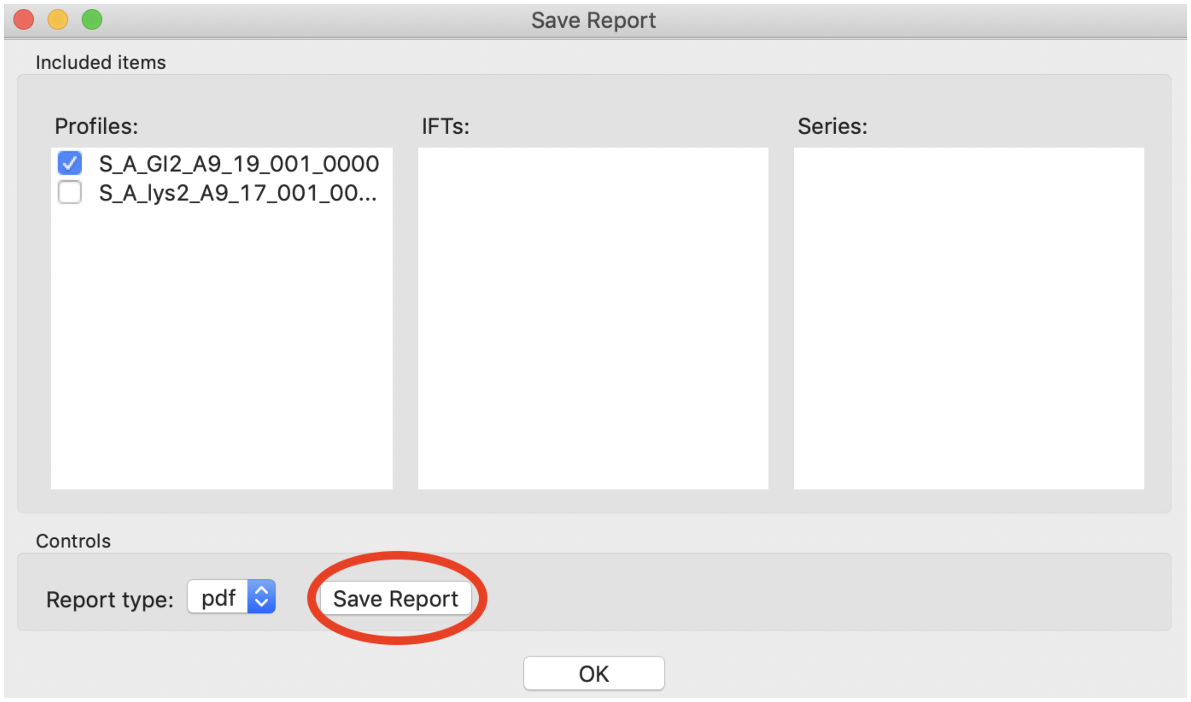
A video version of this tutorial is available:

The written version of this tutorial follows:

1. Select the subtracted GI scattering profile in the Profiles panel and right click on it and select “Save report.”



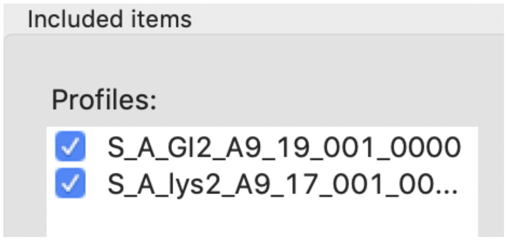
2. In the panel that opens, click the “Save Report” button and save the report as “gi_report.pdf” in the **standards_data** folder. This will save a PDF report with plots and analysis for the GI data.



Profiles:
- [✓] S_A_GI2_A9_19_001_0000
- [ ] S_A_lys2_A9_17_001_00...

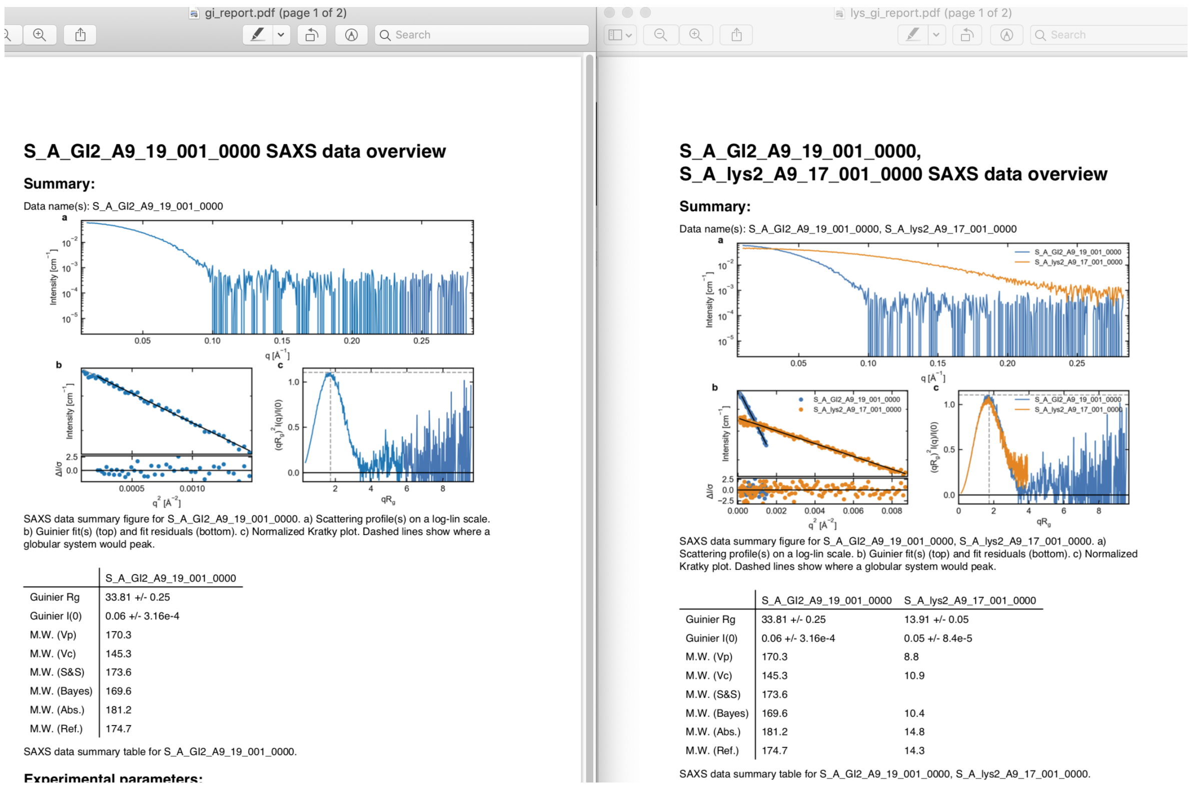
Report type: pdf

3. You can also save a report for multiple datasets at once. Select both the lysozyme and GI datasets by checking both of the checkboxes. Then click "Save Report" to save a report with both datasets as "lys_gi_report.pdf" in the **standards_data** folder.



Profiles:
- [✓] S_A_GI2_A9_19_001_0000
- [✓] S_A_lys2_A9_17_001_00...

4. Open the two reports in a pdf viewer. You should see just the GI data in the first, whereas the second pdf should have plots and columns in the table for both the lysozyme and GI data.



S_A_GI2_A9_19_001_0000 SAXS data overview

Summary:

Data name(s): S_A_GI2_A9_19_001_0000

SAXS data summary figure for S_A_GI2_A9_19_001_0000. a) Scattering profile(s) on a log-lin scale. b) Guinier fit(s) (top) and fit residuals (bottom). c) Normalized Kratky plot. Dashed lines show where a globular system would peak.

	S_A_GI2_A9_19_001_0000
Guinier Rg	33.81 +/- 0.25
Guinier I(0)	0.06 +/- 3.16e-4
M.W. (Vp)	170.3
M.W. (Vc)	145.3
M.W. (S&S)	173.6
M.W. (Bayes)	169.6
M.W. (Abs.)	181.2
M.W. (Ref.)	174.7

SAXS data summary table for S_A_GI2_A9_19_001_0000.

Experimental parameters:

S_A_GI2_A9_19_001_0000, S_A_lys2_A9_17_001_0000 SAXS data overview

Summary:

Data name(s): S_A_GI2_A9_19_001_0000, S_A_lys2_A9_17_001_0000

SAXS data summary figure for S_A_GI2_A9_19_001_0000, S_A_lys2_A9_17_001_0000. a) Scattering profile(s) on a log-lin scale. b) Guinier fit(s) (top) and fit residuals (bottom). c) Normalized Kratky plot. Dashed lines show where a globular system would peak.

	S_A_GI2_A9_19_001_0000	S_A_lys2_A9_17_001_0000
Guinier Rg	33.81 +/- 0.25	13.91 +/- 0.05
Guinier I(0)	0.06 +/- 3.16e-4	0.05 +/- 8.4e-5
M.W. (Vp)	170.3	8.8
M.W. (Vc)	145.3	10.9
M.W. (S&S)	173.6	
M.W. (Bayes)	169.6	10.4
M.W. (Abs.)	181.2	14.8
M.W. (Ref.)	174.7	14.3

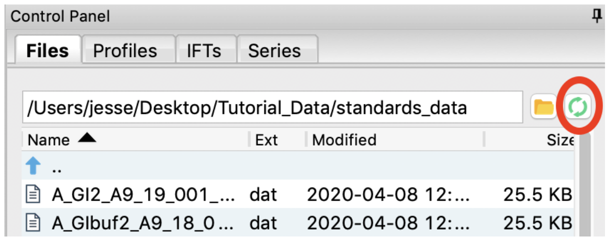
SAXS data summary table for S_A_GI2_A9_19_001_0000, S_A_lys2_A9_17_001_0000.

5. Once you're done saving reports, click the "OK" button to close the window.

Saving as a spreadsheet (.csv)

The written version of the tutorial follows.

1. Save your subtracted scattering profiles in the **standards_data** folder.
2. Select both subtracted profiles, right click on one of them, and select 'Save all analysis info.' Give it an appropriate name and save it in the **standards_data** folder.
   - *Note:* This saves a **.csv** file with all of the analysis information for the selected scattering profiles.
   - *Try:* Open the **.csv** file in Microsoft Excel or Libre/Open Office Calc. You should see all of the analysis that you just did.
3. Remove the subtracted scattering profiles from RAW by selecting both of them and clicking the "Remove" button.
4. Load the saved subtracted scattering profiles back into RAW. Note that if you select one in the Profiles list, the information panel in the upper left corner of RAW populates with analysis information. The analysis information is saved with the scattering profile, so if you forget to save it in a **.csv**, you can load in the profiles later and do it then.
   - *Note:* To get new files to show up in the file tab, you may have to click the refresh button. Also, make sure to that your file type filter is either All files or DAT files.



/Users/jesse/Desktop/Tutorial_Data/standards_data

Name	Ext	Modified	Size
..			
A_GI2_A9_19_001_...	dat	2020-04-08 12:...	25.5 KB
A_GIbuf2_A9_18_0...	dat	2020-04-08 12:...	25.5 KB

- *Try:* Open the saved subtracted scattering profile **S_A_GI2_A9_19_001_0000.dat** in a text editor such as Notepad (windows) or TextEdit (mac). You should see all of the data in three columns, followed by header information. If you scroll down far enough, the header information contains all of the analysis information, as well as the files that were averaged and subtracted to make the scattering profile.

Kratky analysis

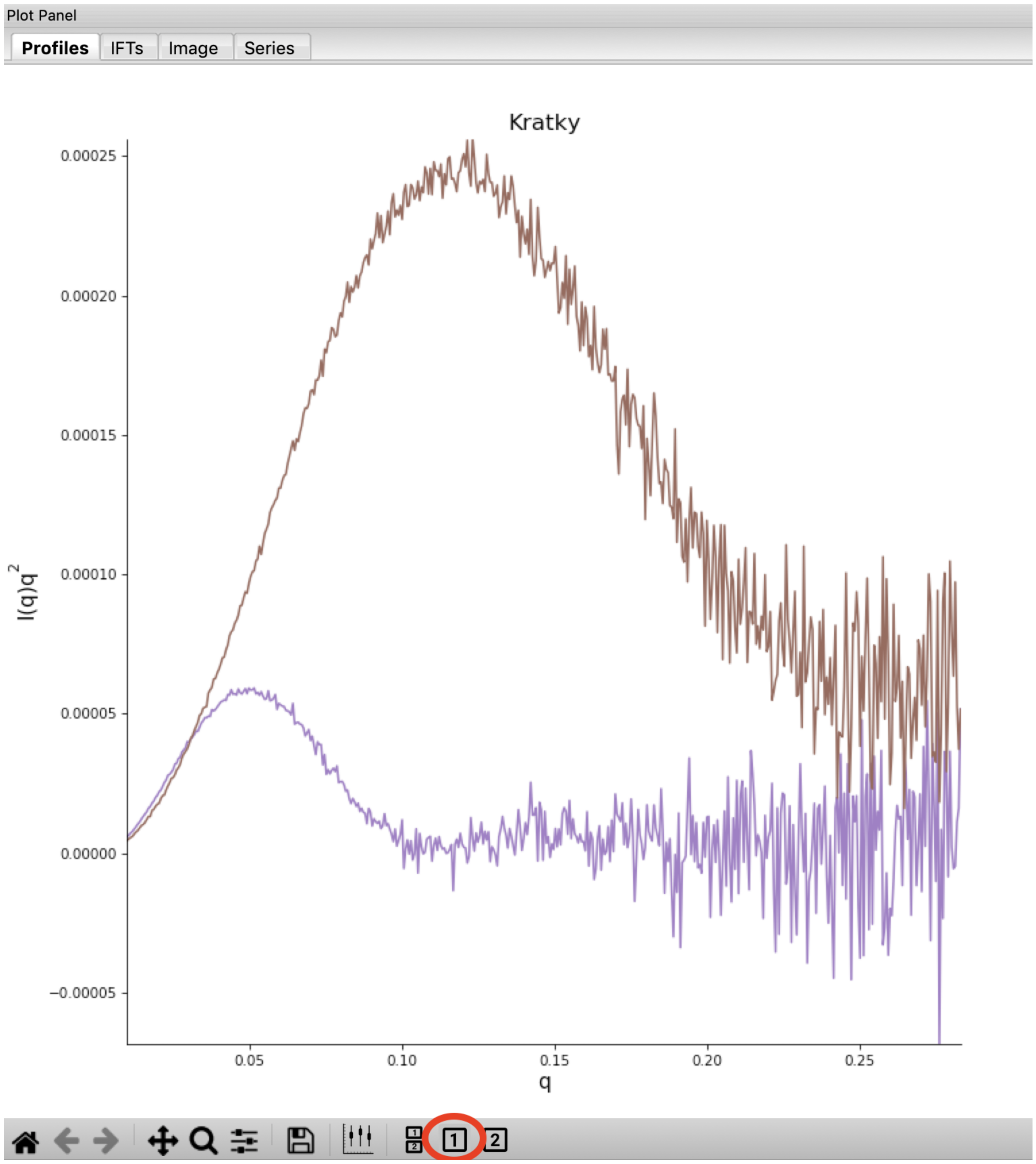
A Kratky plot is a plot of $q^2 I(q)$ vs. q. Kratky plots can qualitatively assess the flexibility and/or degree of unfolding in samples. Unfolded (highly flexible) proteins should have a plateau in the Kratky plot at high q, while compact, globular proteins will have a bell-shaped (Gaussian) peak. A partially unfolded (flexible) protein may have a combination of the bell-shape and plateau, or a plateau that slowly decays to zero.

Normalized Kratky plots are plots of $q^2 I(q)/I(0)$ vs. q. This normalizes scattering profiles by mass and concentration. Dimensionless Kratky plots are presented as either $(qR_g)^2 I(q)/I(0)$ vs. qR_gor $(q^2 V_c)I(q)/I(0)$ vs. $q(V_c)^{1/2}$. These dimensionless plots can provide semi-quantitative analysis of flexibility and disorder. More information about can be found here and references therein: https://www.bioisis.net/tutorials/21.

A video version of this tutorial is available:

The written version of the tutorial follows.

1. If you haven't already, save the loaded scattering profiles, remove all loaded profiles from RAW, and reload your saved GI and Lysozyme profiles, as described in *the previous section*.
2. Put the top plot on Kratky axes.
   - *Tip:* Right click on the plot to change the plot type.
3. Show only the top plot by clicking on the 1 in the plot control bar below the plots.



4. Both GI and lysozyme show the classic bell shape, indicating they are completely folded.

   • *Warning:* Bad buffer subtraction can also result in a Kratky plot that appears to show some degree of flexibility. Excellent buffer subtraction is required for accurately analysis with this technique.

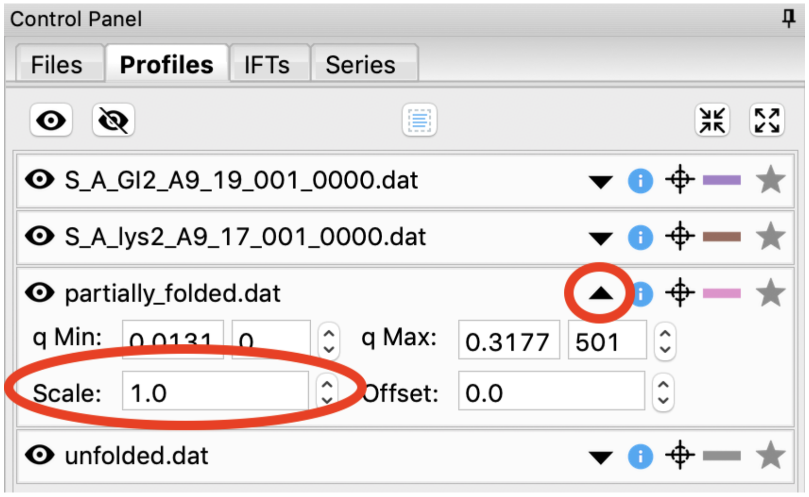
5. Load the two scattering profiles in the **Tutorial_Data/flexibility_data** folder.

   • *Note:* The **unfolded.dat** file is the scattering profile of an unfolded lysine riboswitch. The **partially_folded.dat** file is same lysine riboswitch, but in the biologically functional configuration. The data were downloaded from the BIOISIS database, and has the BIOISIS ids of 2LYSRR and 3LYSRR.

6. SAXS data can be presented on an arbitrary scale, which is why these two profiles have intensity that is much larger than the lysozyme and GI data (which is on an absolute scale). Use the triangle button for each item in the Profiles list to show more options. Hide one of the newly loaded data sets, and adjust the scale factor on the

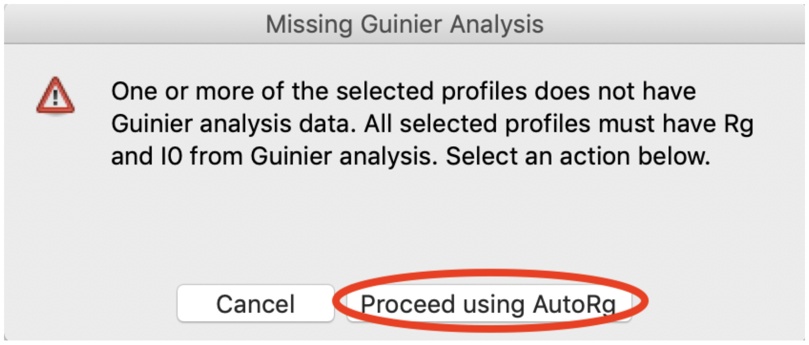
other until you can comfortably see it and your lysozyme and GI data. Repeat the scale adjustment for the other data set.

- *Tip:* The up and down arrows will only adjust the last digit of the scale factor.



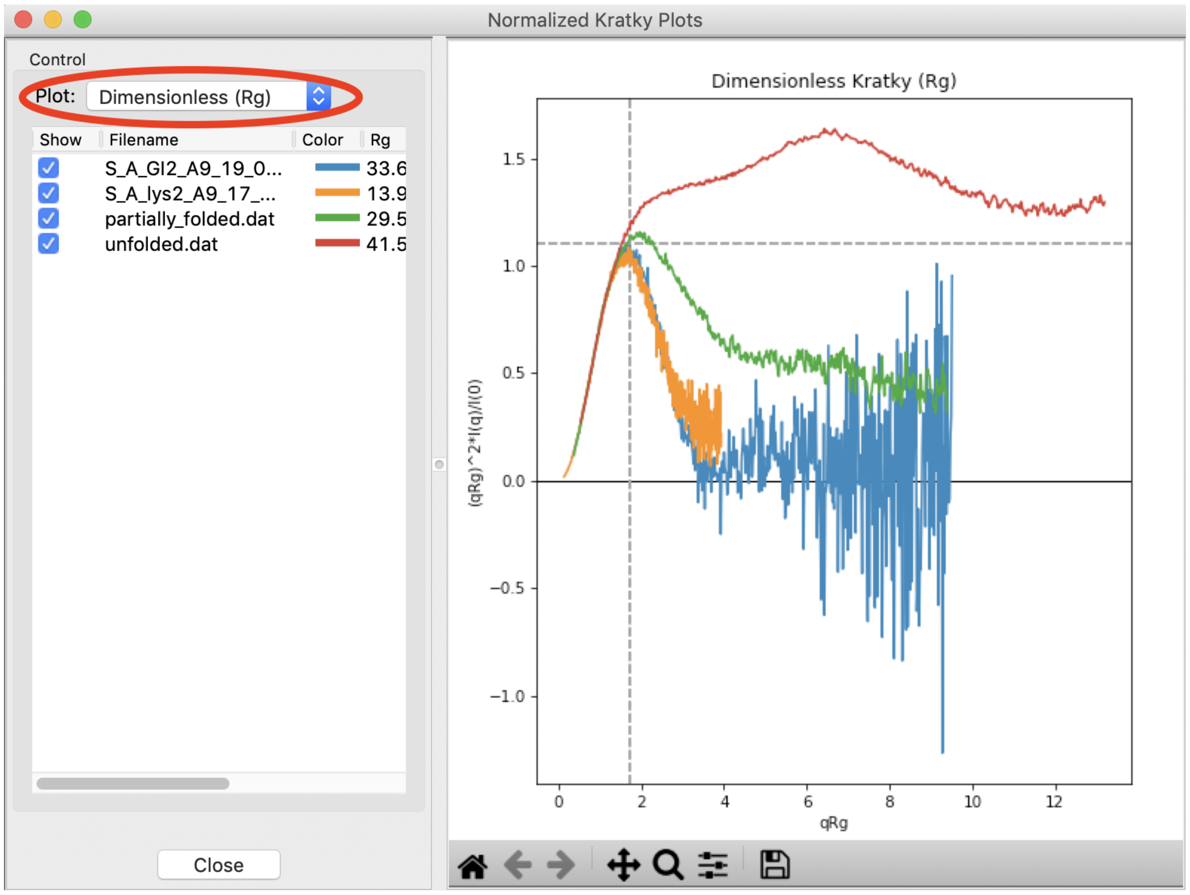
S_A_GI2_A9_19_001_0000.dat
S_A_lys2_A9_17_001_0000.dat
partially_folded.dat
q Min: 0.0131 0 q Max: 0.3177 501
Scale: 1.0 Offset: 0.0
unfolded.dat

7. Kratky analysis can also be done on normalized or dimensionless data. RAW supports normalization by I(0), and non-dimensionalization by R_g and Vc (the volume of correlation).
8. Select all four loaded scattering profiles, right click, and select the Dimensionless Kratky Plot option.
9. Normalized and dimensionless Kratky plots require Guinier analysis to be done. If one or more profiles are missing this information, RAW will show the following window. You can either cancel, and do the fits manually, or you can proceed with RAW's automatic determination.


Missing Guinier Analysis

One or more of the selected profiles does not have Guinier analysis data. All selected profiles must have Rg and I0 from Guinier analysis. Select an action below.

10. Click the Proceed using AutoRg button to proceed to the Dimensionless Kratky Plot window using RAW's automatic fitting for R_g.
11. By default, the plot is the Dimensionless R_g plot. Use the drop-down "Plot" menu at the top to select the Normalized (by I(0)) and Dimensionless Vc plots.
    - *Tip:* The dashed gray lines on the Dimensionless (Rg) plot are guidelines for a globular protein. For a globular protein the peak position should be at $qR_g = \sqrt{3} \approx 1.73$, while peak height should be $3/e \approx 1.1$.



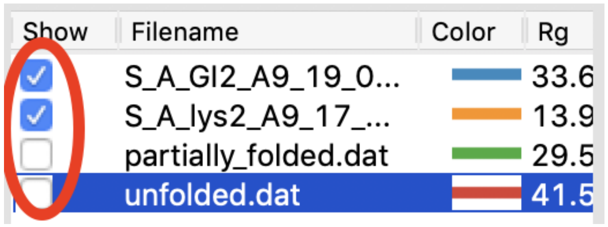
Control

Plot: Dimensionless (Rg)

Show	Filename	Color	Rg
✓	S_A_GI2_A9_19_0...	🟦	33.6
✓	S_A_lys2_A9_17_...	🟧	13.9
✓	partially_folded.dat	🟩	29.5
✓	unfolded.dat	🟥	41.5

12. Return to the Dimensionless R_g plot. Use the check boxes to hide the partially_folded and unfolded data sets on the plot. Note that both the lysozyme and GI data look very similar on this plot, showing they have similar shapes and (lack of) flexibility.

    - *Tip:* You can click on the colored line in the Color column to change the color of an item on the plot.



Show	Filename	Color	Rg
✓	S_A_GI2_A9_19_0...	🟦	33.6
✓	S_A_lys2_A9_17_...	🟧	13.9
	partially_folded.dat	🟩	29.5
	unfolded.dat	🟥	41.5

13. Right click on the plot and select "Export Data As CSV" to save the dimensionless data for further processing or plotting with another program.

14. Click the Close button to close the Dimensionless Kratky Plot window.

Similarity Testing

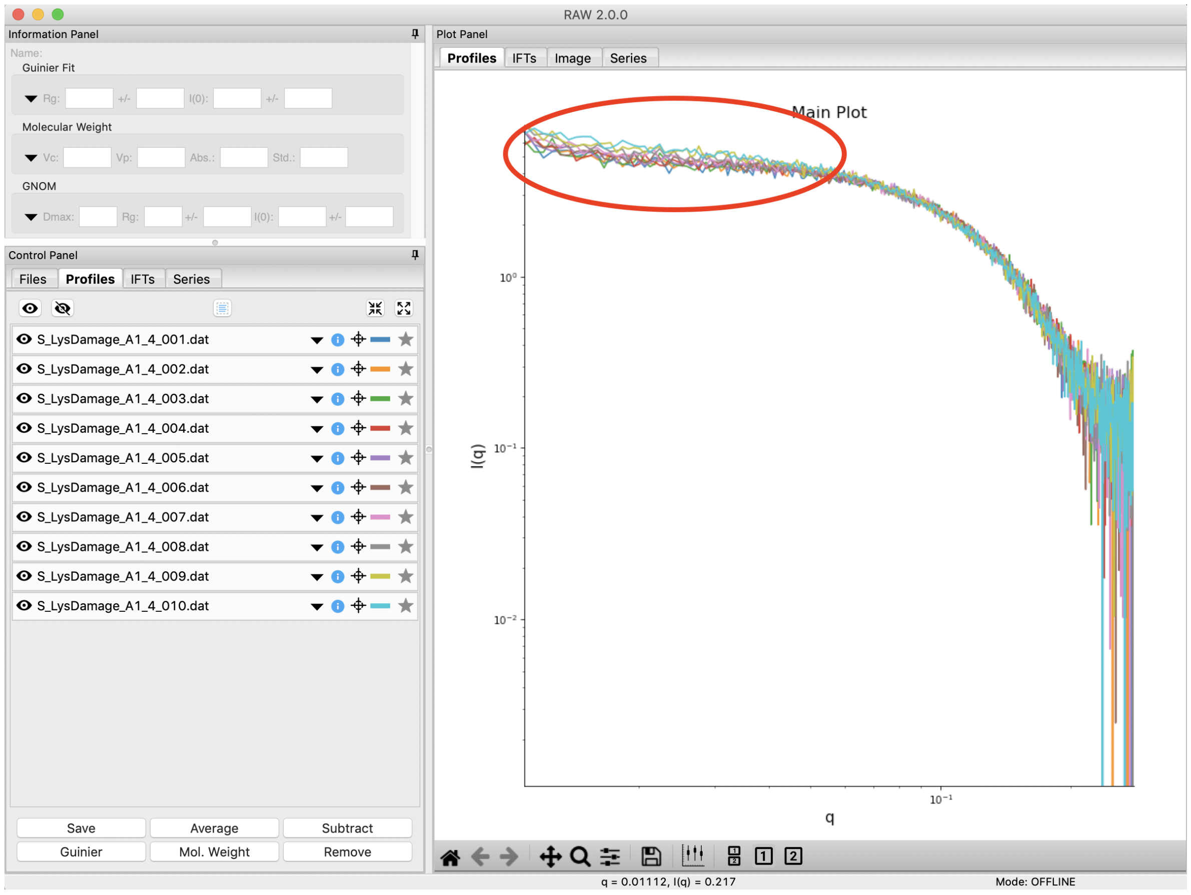
A video version of this tutorial is available:

The written version of the tutorial follows.

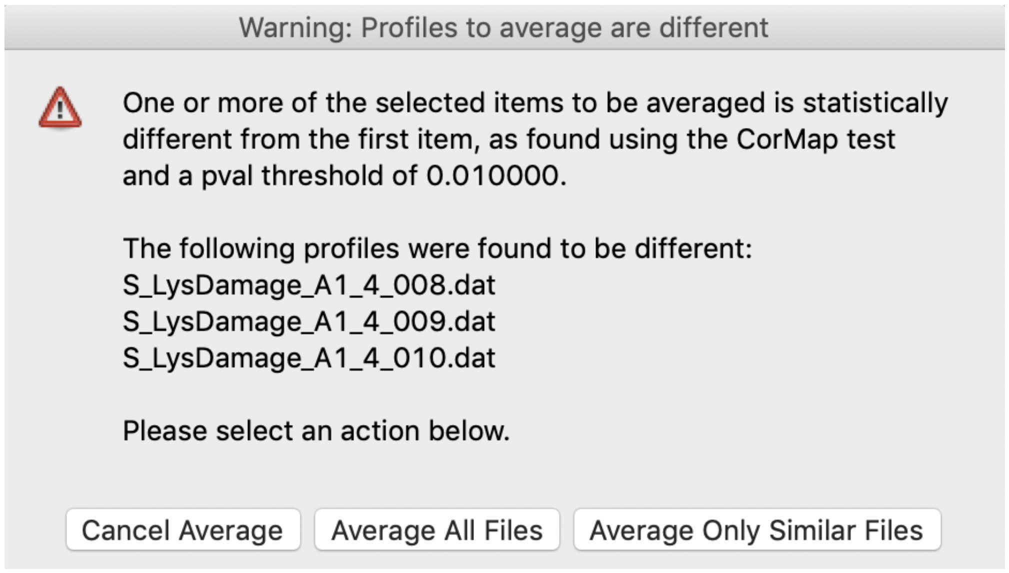
RAW has the ability to test scattering profiles for statistical similarity. Currently, only one test is available: the Correlation Map test. This can be done manually, and is also done automatically when scattering profiles are averaged.

This can be useful when you're dealing with data that may show changes in scattering from radiation damage or other possible sources.

1. Clear any data loaded into RAW. Load all of the profiles in the **Tutorial_Data/damage_data** folder into the Profiles plot. Show only the top plot.
   - *Tip:* In the Files tab, click the "Clear All" button.
2. Put the plot on a log-log scale. You should see that the profiles are different at low *q*.
   - *Note:* These data are showing what radiation damage looks like in a data set. They are consecutive profiles from the same sample, and as total exposure of the sample increase (frame number increases), the sample damages. In this case, the damage is manifesting as aggregation, which shows up as an uptick in the profiles at low *q*.



3. Select all of the profiles and average them. You will get a warning message informing you that not all the files are statistically the same.
   - *Note:* This is only as good as the statistical test being used, and the cutoff threshold selected. In the advanced options panel you can select the test, whether or not it is corrected for multiple testing, and the threshold used.



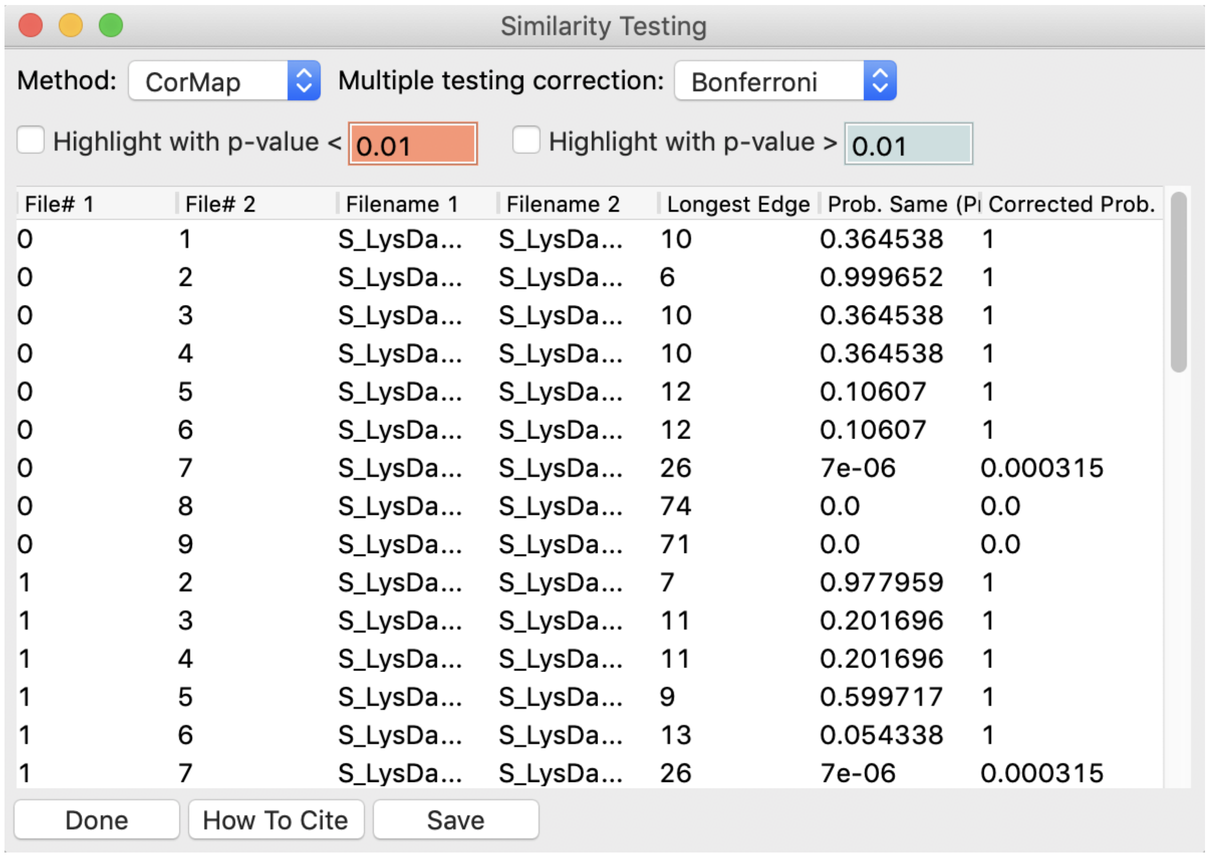
Warning: Profiles to average are different

One or more of the selected items to be averaged is statistically different from the first item, as found using the CorMap test and a pval threshold of 0.010000.

The following profiles were found to be different:
S_LysDamage_A1_4_008.dat
S_LysDamage_A1_4_009.dat
S_LysDamage_A1_4_010.dat

Please select an action below.

4. Click the "Average Only Similar Files" button.
   - Note: This averages only those profiles found to the same as the first file, for the given statistical test.
5. Select all of the profiles except the new averaged one, and right click and select "Similarity Test".



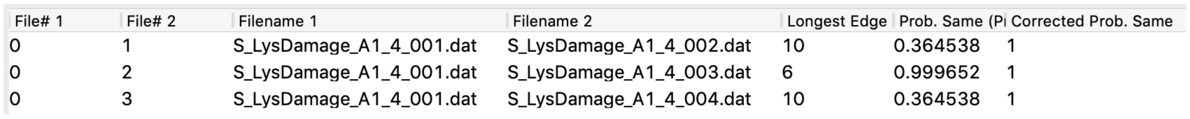
Similarity Testing

Method: CorMap Multiple testing correction: Bonferroni

Highlight with p-value < 0.01 Highlight with p-value > 0.01

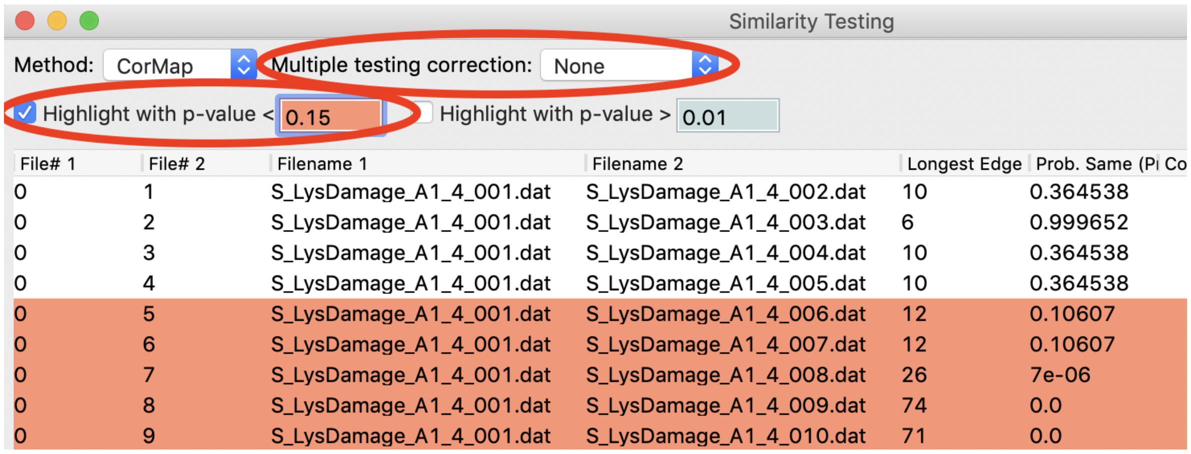
File# 1	File# 2	Filename 1	Filename 2	Longest Edge	Prob. Same (Pi	Corrected Prob.
0	1	S_LysDa...	S_LysDa...	10	0.364538	1
0	2	S_LysDa...	S_LysDa...	6	0.999652	1
0	3	S_LysDa...	S_LysDa...	10	0.364538	1
0	4	S_LysDa...	S_LysDa...	10	0.364538	1
0	5	S_LysDa...	S_LysDa...	12	0.10607	1
0	6	S_LysDa...	S_LysDa...	12	0.10607	1
0	7	S_LysDa...	S_LysDa...	26	7e-06	0.000315
0	8	S_LysDa...	S_LysDa...	74	0.0	0.0
0	9	S_LysDa...	S_LysDa...	71	0.0	0.0
1	2	S_LysDa...	S_LysDa...	7	0.977959	1
1	3	S_LysDa...	S_LysDa...	11	0.201696	1
1	4	S_LysDa...	S_LysDa...	11	0.201696	1
1	5	S_LysDa...	S_LysDa...	9	0.599717	1
1	6	S_LysDa...	S_LysDa...	13	0.054338	1
1	7	S_LysDa...	S_LysDa...	26	7e-06	0.000315

6. The similarity testing window (above) shows the results of the pairwise tests done using the CorMap method. Expand the window and the Filename columns to allow you to see the full filenames along with the probabilities.



File# 1	File# 2	Filename 1	Filename 2	Longest Edge	Prob. Same (Pi	Corrected Prob. Same
0	1	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_002.dat	10	0.364538	1
0	2	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_003.dat	6	0.999652	1
0	3	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_004.dat	10	0.364538	1

7. Using the menu at the top, turn off multiple testing correction. Change the highlight less than value to 0.15, and highlight those pairs.



Method: CorMap   Multiple testing correction: None

[✓] Highlight with p-value < 0.15   [ ] Highlight with p-value > 0.01

File# 1	File# 2	Filename 1	Filename 2	Longest Edge	Prob. Same (Pi	Co
0	1	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_002.dat	10	0.364538	
0	2	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_003.dat	6	0.999652	
0	3	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_004.dat	10	0.364538	
0	4	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_005.dat	10	0.364538	
0	5	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_006.dat	12	0.10607	
0	6	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_007.dat	12	0.10607	
0	7	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_008.dat	26	7e-06	
0	8	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_009.dat	74	0.0	
0	9	S_LysDamage_A1_4_001.dat	S_LysDamage_A1_4_010.dat	71	0.0	

8. Without multiple testing correction, and using a less stringent threshold for similarity, we see that more profiles are selected here (profiles 6-10) than were excluded from the average using the automatic test. Because we know radiation damage increases with dose, it is reasonable to suspect that we should discard profiles 6-10, not just 8-10 as in the automated version.
9. Save the similarity test data as a **.csv** by clicking the "Save" button.
10. Close the similarity testing window by clicking the "Done" button.
11. Average profiles 1-5.
12. Hide all of the profiles except the two averaged profiles on the plot.
    - *Question:* Is there a difference between the two? What about if you do a Guinier fit?
    - *Note:* In this case, the differences are subtle, a ~1-2% increase in R_g. So the automated determination did a reasonable job. However, it is generally good to double check your set of profiles both visually and using the Similarity Test panel when the automated test warns you of outlier profiles.

Basic SEC-SAXS processing

In a typical SEC-SAXS run, images are continuously collected while the eluate (outflow) of a size exclusion column flows through the SAXS sample cell. As proteins scatter more strongly than buffer, a plot of total scattered intensity vs. time, the so-called SAXS chromatograph (or scattergram), will show a set of peaks similar to what is seen by UV absorption measurement of the SEC system. RAW includes the capability to do routine processing of SEC-SAXS data. This includes creating the SAXS chromatograph from the data, plotting R_g, MW, and I(0) across the peaks, and extracting specific frames for further analysis.

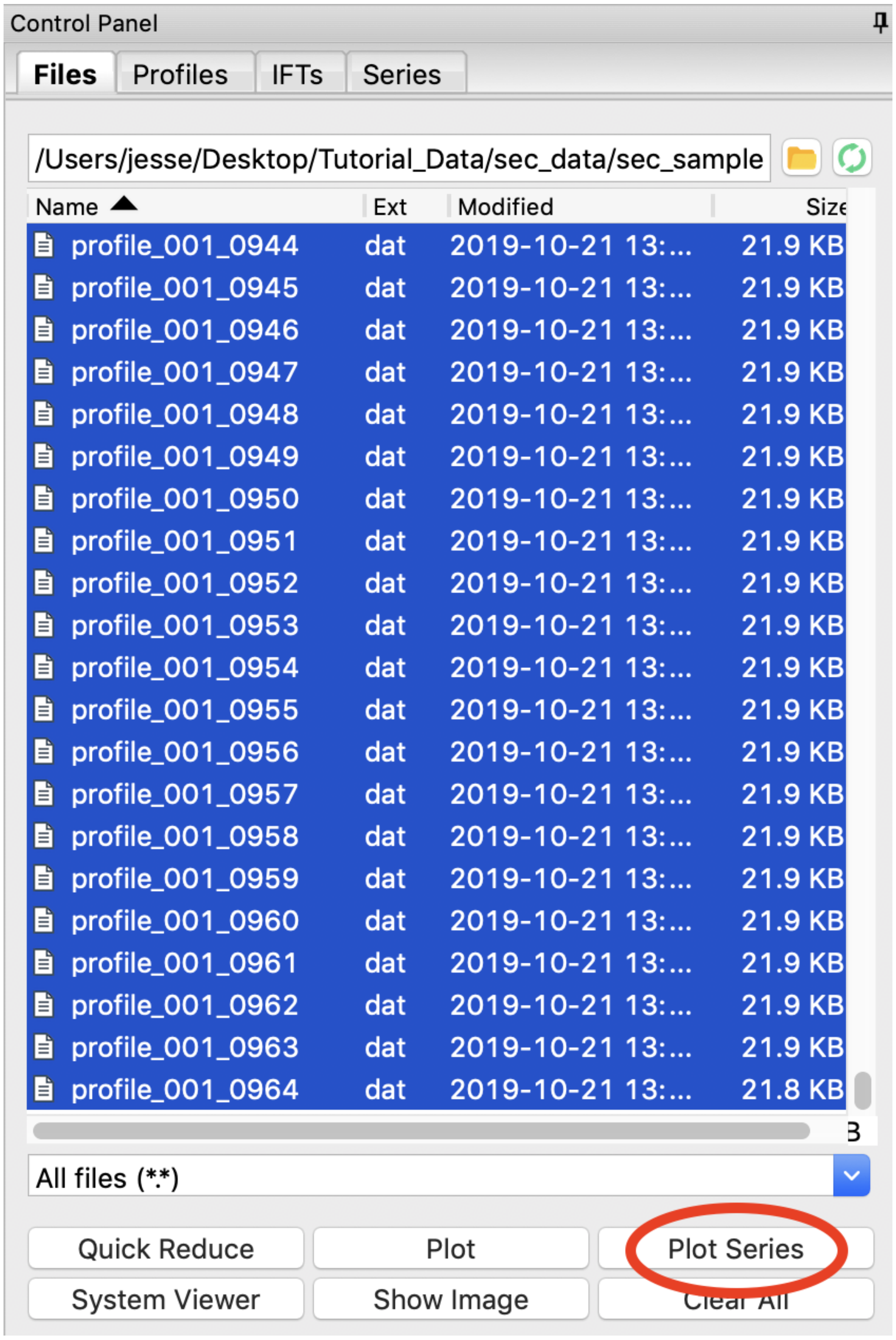
Note: In RAW, this is called Series analysis, as the same tools can be used for other sequentially sampled data sets.

A video version of this tutorial is available:

The written version of the tutorial follows.

1. Clear any data loaded into RAW.
    - *Tip:* In the Files tab, click the "Clear All" button to clear all data in RAW.

2. Go to the Files control tab and navigate to the **sec_sample_1** data directory.
3. Click on the first data file, **profile_001_0000.dat**. Scroll down to the bottom of the file list, and shift click on the last file, **profile_001_0964.dat**. This should highlight all of the files in between, as well as the two you clicked on. Click on the "Plot Series" button to load the series into RAW.



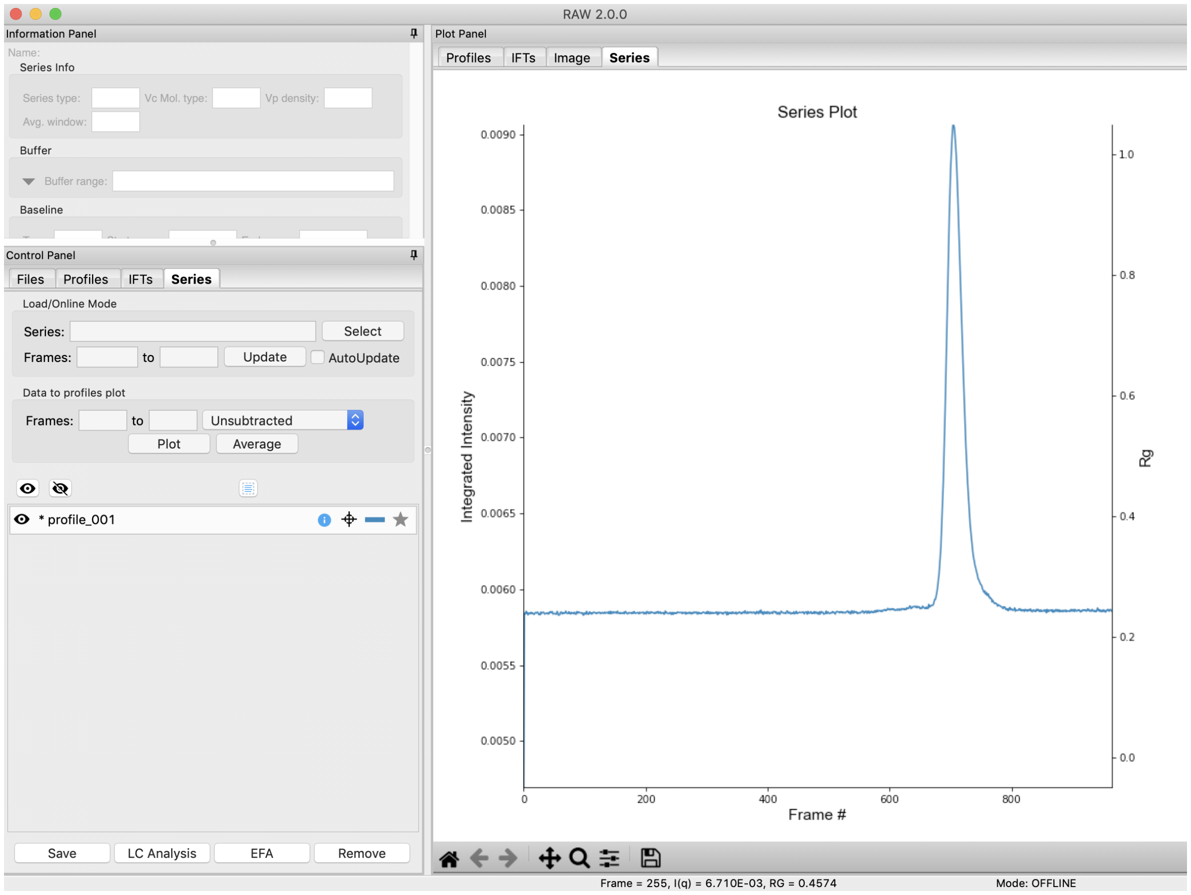
/Users/jesse/Desktop/Tutorial_Data/sec_data/sec_sample

Name	Ext	Modified	Size
profile_001_0944	dat	2019-10-21 13:...	21.9 KB
profile_001_0945	dat	2019-10-21 13:...	21.9 KB
profile_001_0946	dat	2019-10-21 13:...	21.9 KB
profile_001_0947	dat	2019-10-21 13:...	21.9 KB
profile_001_0948	dat	2019-10-21 13:...	21.9 KB
profile_001_0949	dat	2019-10-21 13:...	21.9 KB
profile_001_0950	dat	2019-10-21 13:...	21.9 KB
profile_001_0951	dat	2019-10-21 13:...	21.9 KB
profile_001_0952	dat	2019-10-21 13:...	21.9 KB
profile_001_0953	dat	2019-10-21 13:...	21.9 KB
profile_001_0954	dat	2019-10-21 13:...	21.9 KB
profile_001_0955	dat	2019-10-21 13:...	21.9 KB
profile_001_0956	dat	2019-10-21 13:...	21.9 KB
profile_001_0957	dat	2019-10-21 13:...	21.9 KB
profile_001_0958	dat	2019-10-21 13:...	21.9 KB
profile_001_0959	dat	2019-10-21 13:...	21.9 KB
profile_001_0960	dat	2019-10-21 13:...	21.9 KB
profile_001_0961	dat	2019-10-21 13:...	21.9 KB
profile_001_0962	dat	2019-10-21 13:...	21.9 KB
profile_001_0963	dat	2019-10-21 13:...	21.9 KB
profile_001_0964	dat	2019-10-21 13:...	21.8 KB

All files (**)

4. RAW should automatically show you the Series plot panel. If not, click on the Series tab in the plot. Click on

the Series tab in the control panel.



Frame = 255, I(q) = 6.710E-03, RG = 0.4574

Mode: OFFLINE

- *Try:* Each point on this curve is the integrated intensity of a scattering profile. You can figure out which one by right clicking on the filename in the Series list and selecting 'Show data'. This will show you the frame number and integrated intensity displayed on the plot, and the filename corresponding to the displayed frame number.

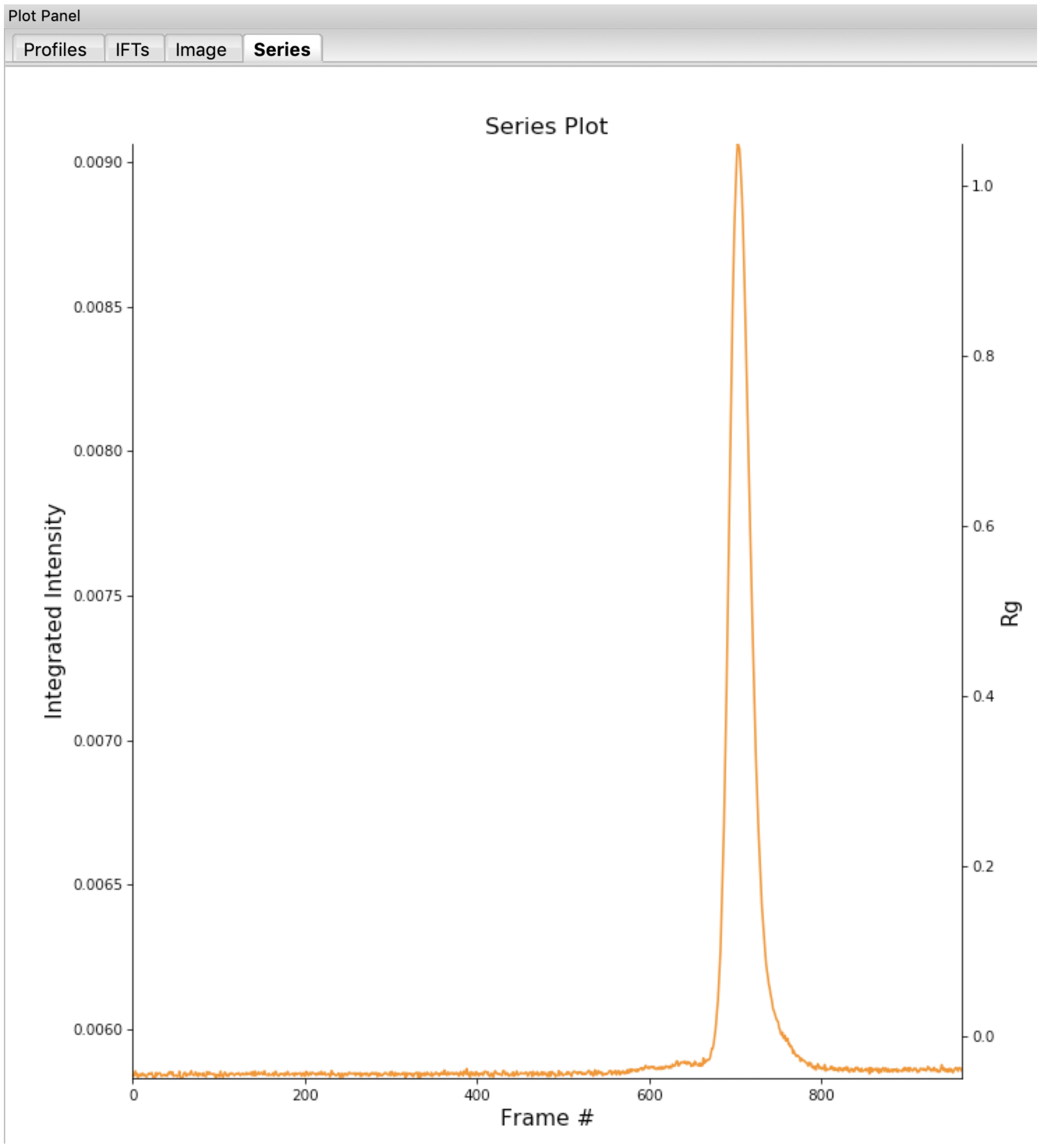
5. Drag the plot so that you can clearly the see the first frame. You'll notice it has a significantly lower intensity than the rest of the frames. This happens occasionally at the MacCHESS G1 beamline (where the data was taken). It can make it harder to tell what the data is doing.

   - *Tip:* Select the crossed arrows in the plot control bar, and then click and drag on the plot to move the curve around on the screen.

6. Reload the data as in steps 2-3, but select the second data file, **profile_001_0001.dat** as the initial file in the series.

   - *Tip:* As before, select **profile_001_0001.dat**, scroll down to the last file, and shift click to select the files to load in. Then click the "Plot Series" button.

7. You will see the same curve plotted as before, but without the very first scattering profile. Remove the other loaded data set. Now you should have a curve where the baseline is very close to the bottom of the plot.



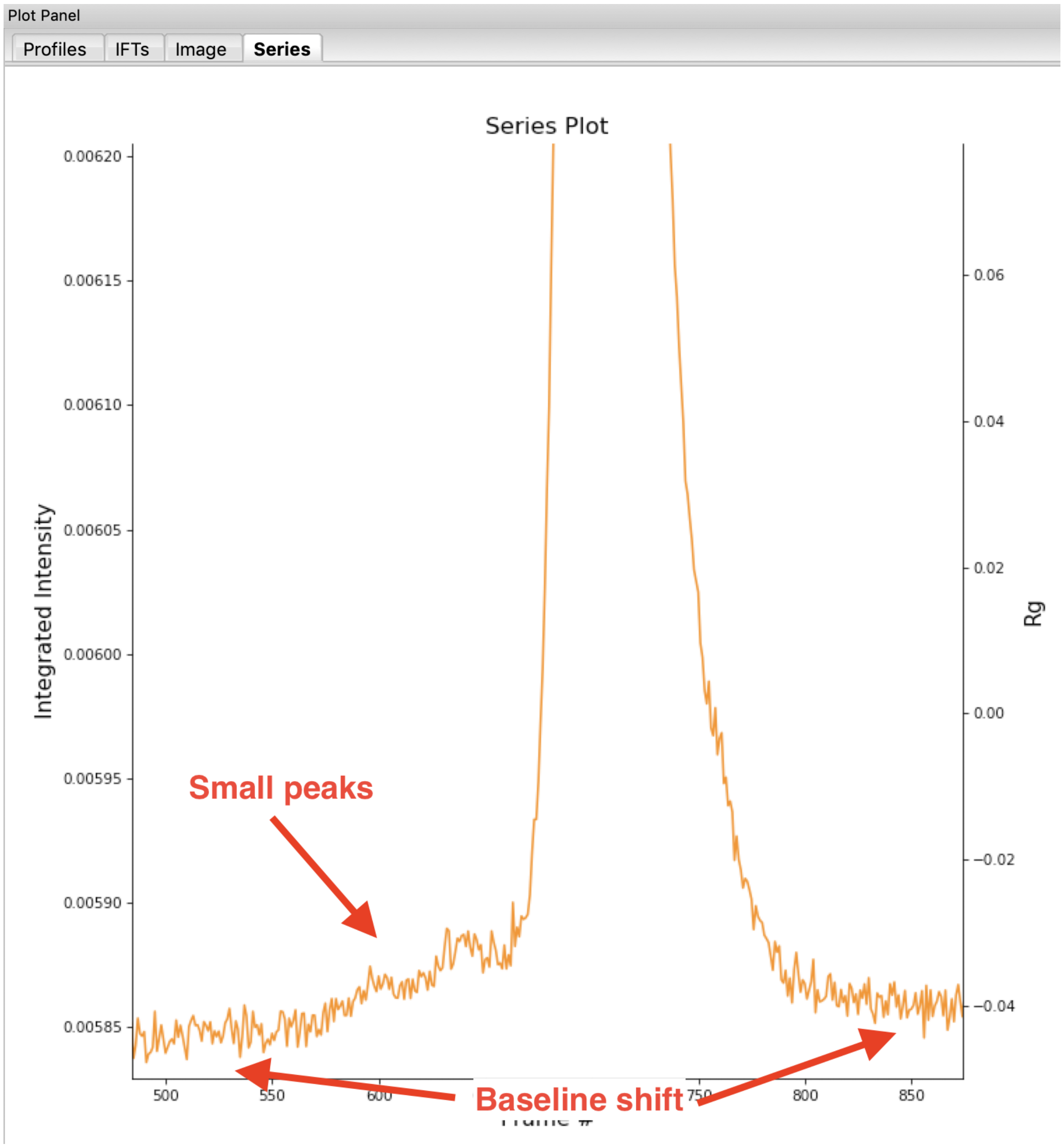
8. In some cases it is more useful to look at the mean intensity, the intensity at a specific q value, or the intensity in a range of q values than the total intensity. Right click on the plot and select mean intensity for the left axis y data. Then try the intensity at *q=0.02* and in the q range 0.01-0.03.

   - *Note:* You need to have the drag button in the plot control bar unselected to get a right click menu.
   - *Tip:* CHROMIXS, in the ATSAS software displays the average intensity over the q range 0.01-0.08 $^{-1}$. To achieve a similar display in RAW, set your q range for the intensity to that. Note that RAW will display the sum of the intensity over the range while CHROMIXS displays mean intensity for the range, so the results won't be exactly the same.

9. Return to plotting the integrated intensity. Zoom in near the base of the peak. Notice that there are two smaller peaks on the left, likely corresponding to higher order oligomers that we don't have the signal to properly resolve. Also notice that the baseline after the peak is not the same as the baseline before the peak. This can

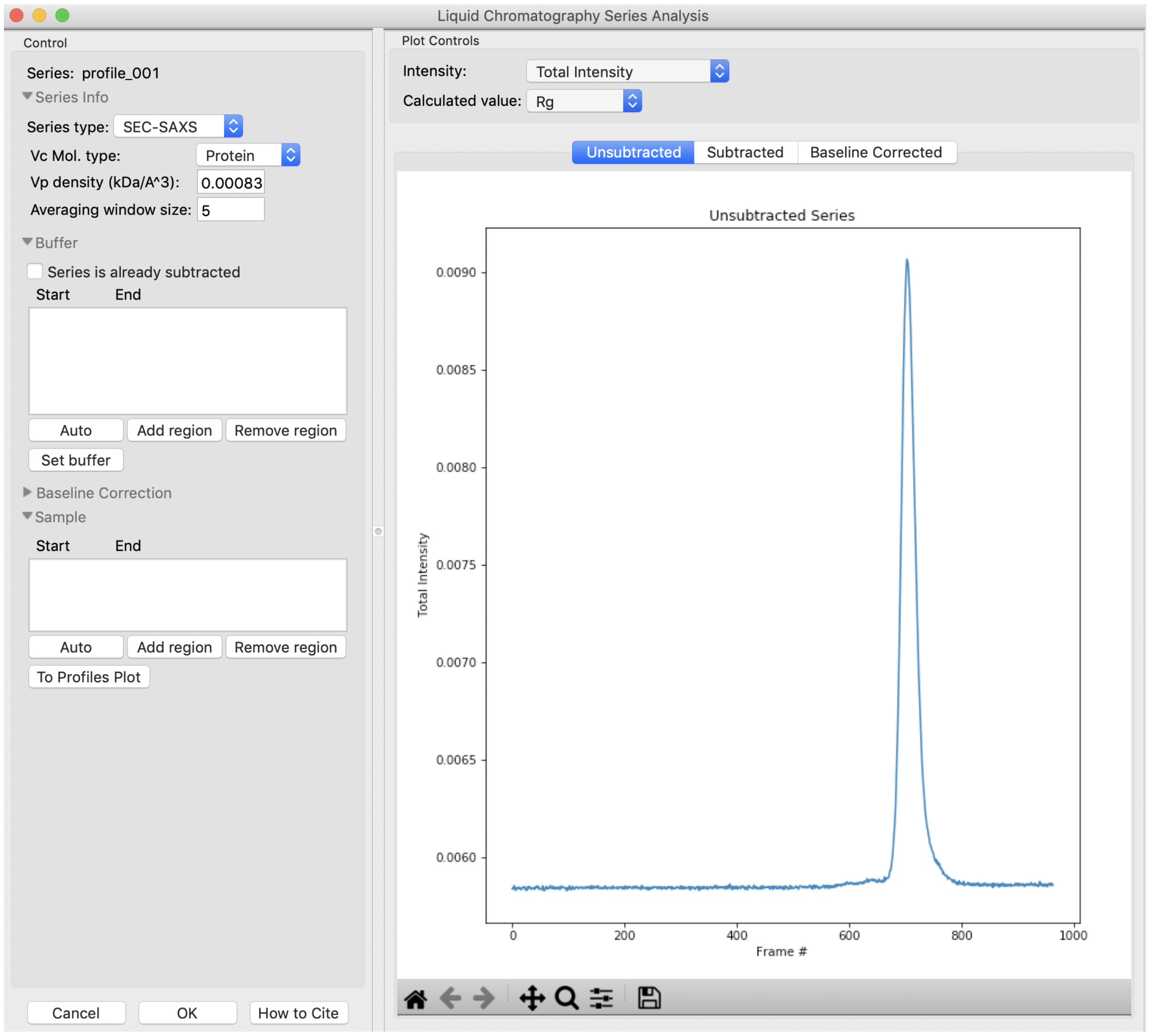
happen for several reasons, such as damaged protein sticking to the sample cell windows.

- *Tip:* Click on the magnifying glass at the bottom of the plot, then click and drag on the plot to select a region to zoom in on.



10. Zoom back out to the full plot.
    - *Tip:* Click the Home (house) button at the bottom of the plot.
11. In order to determine if we really have a single species across the peak, we will calculate the R_g and MW as a function of frame number. Click on the “LC Analysis” button at the bottom of the Series control panel or right click on the filename in the Series control panel and select “LC Series analysis” to open the LC Series analysis panel.
    - *Note:* At the top of the control panel in this window, in the ‘Series info’ section you’ll see several settings. If you had RNA instead of protein, you would use the Vc Mol. type menu to select that option. This affects

the calculation of the molecular weight. You could also change the Vp density away from the default value, or change the averaging window (discussed below).



Series: profile_001

Series type: SEC-SAXS

Vc Mol. type: Protein

Vp density (kDa/A^3): 0.00083

Averaging window size: 5

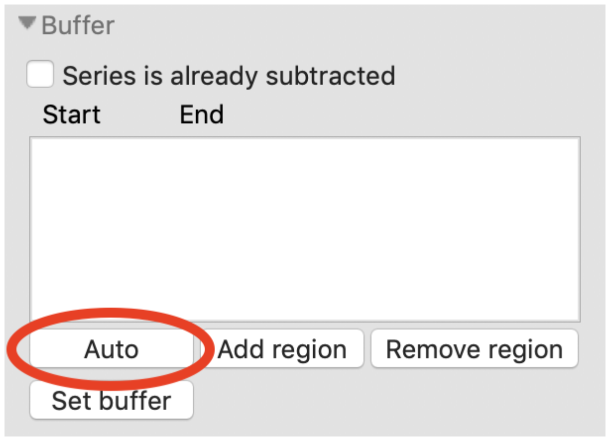
Series is already subtracted

Intensity: Total Intensity

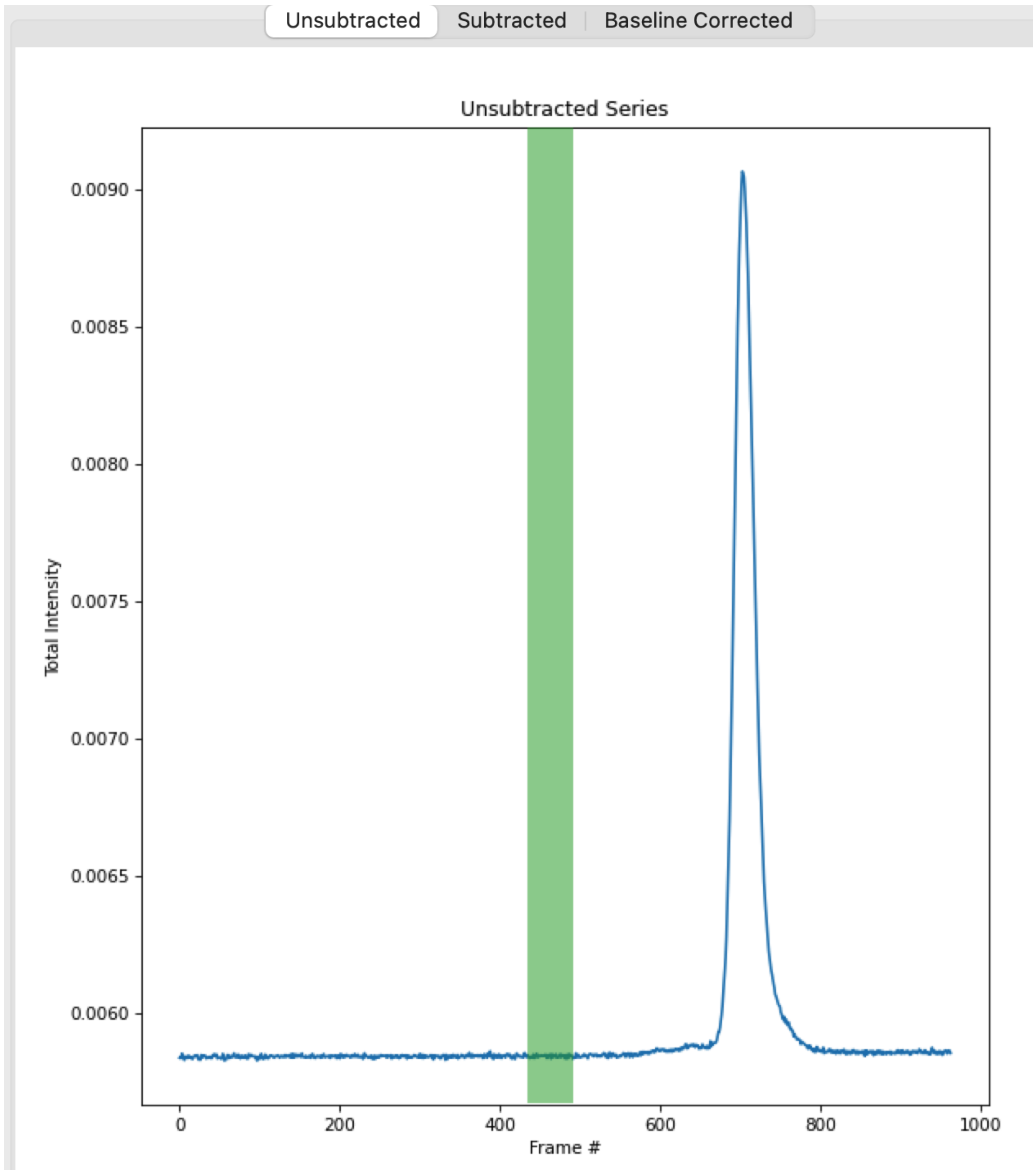
Calculated value: Rg

12. The LC Series analysis panel provides basic and advanced analysis tools for liquid chromatography experiments. Here we will show how to select buffer and sample regions, and send final processed data to the Profiles plot. The advanced baseline correction features are discussed later.

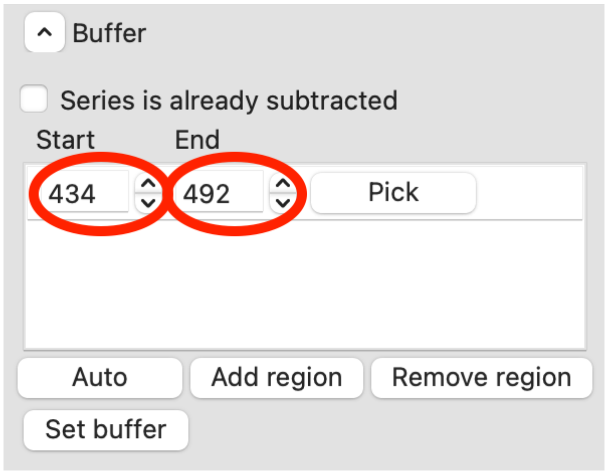
    In order to calculate R_g and other parameters as a function of elution time (Frame #), we need to define a buffer region. RAW can do this automatically. In the 'Buffer' section click the 'Auto' button.



- *Checkpoint:* You should see a buffer range show up in the buffer list, with defined start and end values. The region will be shown in green on the Unsubtracted plot.



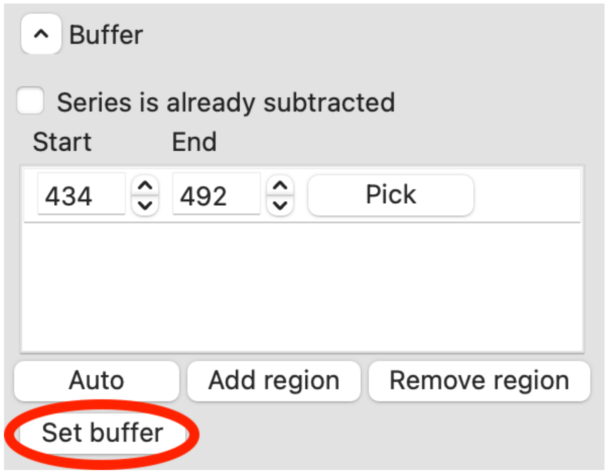
13. You can make fine manual adjustments to the buffer range if necessary. Zoom in on the baseline around the buffer region. Use the up/down arrows for the Start and End points to adjust the buffer region a little bit. You will see the region on the plot update as you make the changes.



Start 434
End 492

- *Warning:* The automatic buffer determination can be wrong! Always be sure to manually inspect the region it picked. In particular, large flat leading edge shoulders next to the main peak can look like a baseline region to the algorithm, and will often mistakenly be picked.
- *Tip:* If the SAXS data isn't clear (noisy, low signal, etc.), it can be useful to inspect the UV trace associated with the SEC elution to see where there are minor elution components that you should exclude from your buffer selection.

14. Zoom back out on the plot. Reset the buffer range to 434 to 492 by typing those values in the Start/End range boxes and hitting enter.
15. To set the buffer region, create a set of subtracted profiles, and calculate structural parameters as a function of elution time, click the 'Set buffer' button. This may take a while to calculate.



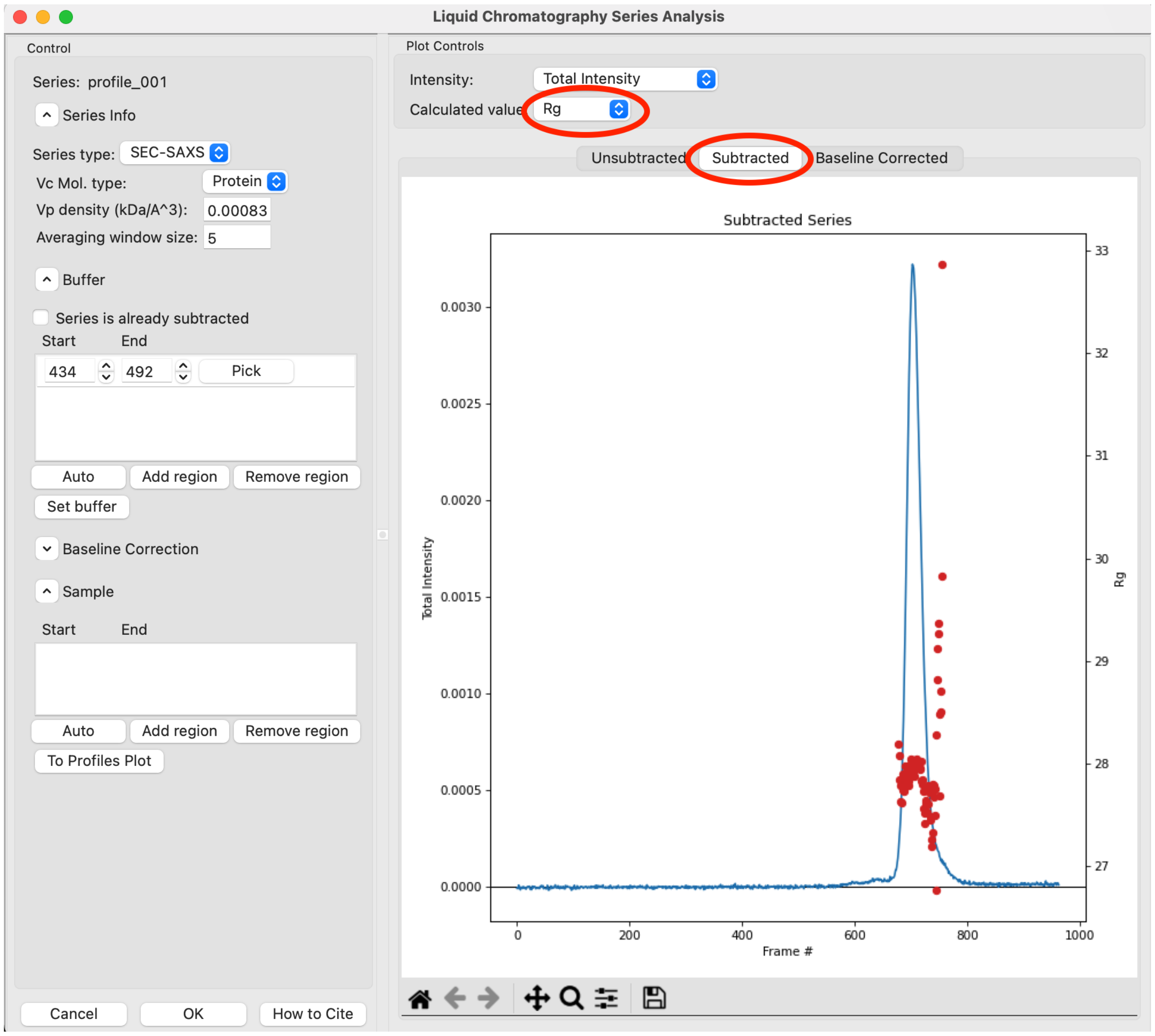
Start 434
End 492

- *Note:* All of the files in the given buffer range will be averaged and used as a buffer. A sliding average window (size defined by the 'Averaging window size' in the 'Series Info' section) is then moved across the SEC curve. So for a window of size five, the profiles corresponding to frames 0-4, 1-5, 2-6, etc will be averaged. From each of these averaged set of curves, the average buffer will be subtracted, and RAW will attempt to calculate the R_g, MW, and I(0). These values are then plotted as a function of frame number.
- *Warning:* It is important that the buffer range actually be buffer! In this case, we need to make sure to not include the small peaks before the main peak.

16. Once the calculation is finished, the window should automatically display the Subtracted plot. If it doesn't, click on the 'Subtracted' tab in the plot. On this plot there is a new Intensity vs. Frame # curve, representing the subtracted data. There is also a set of markers, showing one of the calculated parameters. By default the R_g

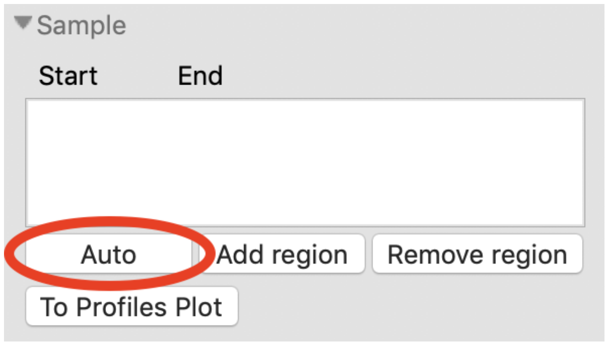
displayed. The calculated parameters are plotted on the right Y axis. You can show R_g, I(0), and MW calculated by the volume of correlation (Vc) and adjusted Porod volume (Vp) methods. Click on the 'Calculated value' menu to switch between the different displays.

- *Try:* Show the R_g, MW (Vc), and MW (Vp). Notice that the MW estimate varies between the two different methods.
- *Note:* You'll notice a region of roughly constant R_g across the peak. To either side there are regions with higher or lower R_g values. Some of these variations, particularly on the right side, are from scattering profiles near the edge of the peak with lower concentrations of sample, leading to more noise in determining the R_g values. There may also be some effects from the small peaks on the leading (left) side of the peak, and from the baseline mismatch between left and right sides of the peak.

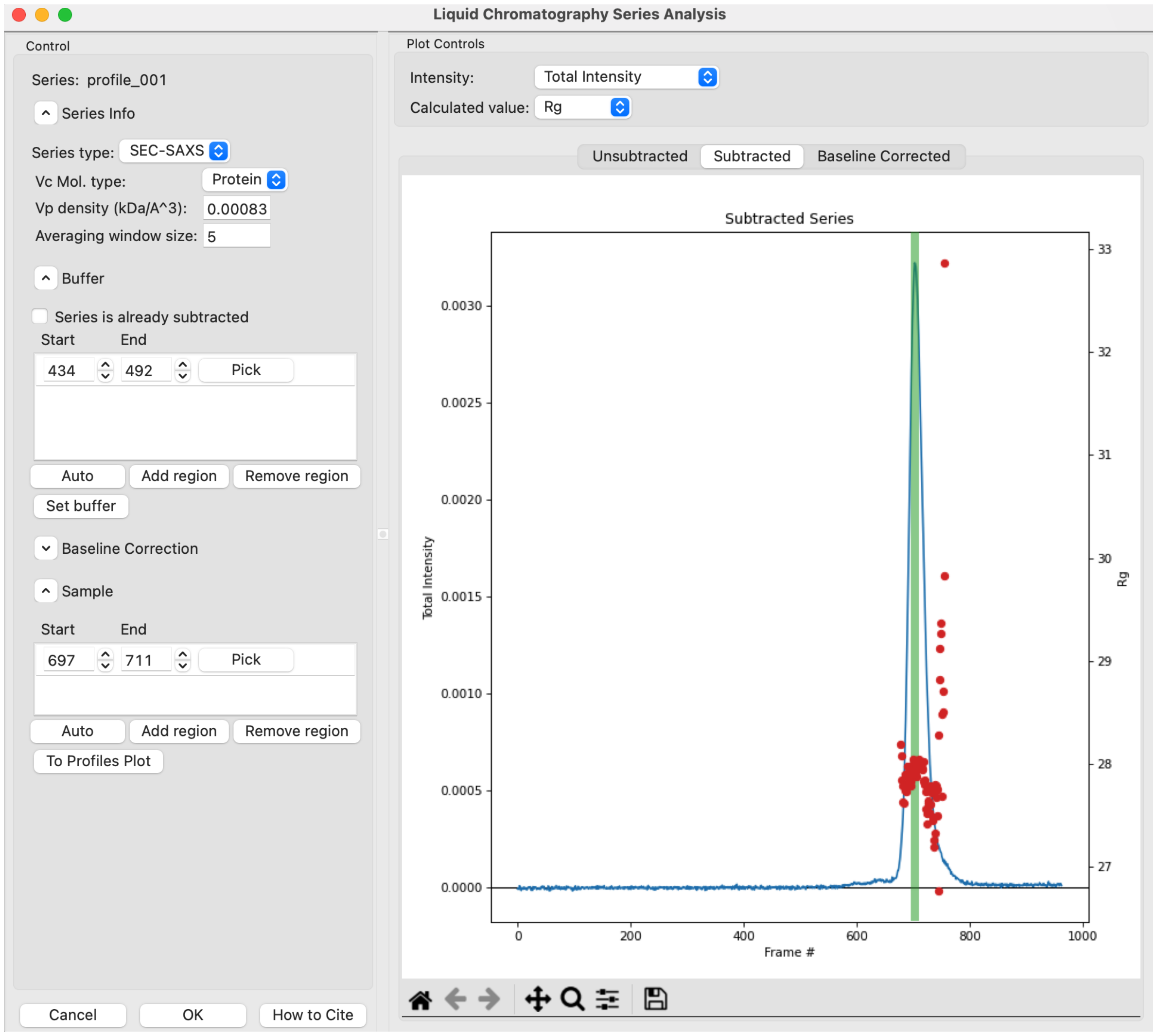


17. A monodisperse peak should display a region of flat R_g and MW near the center. Note that some spread on either edge can come from small shoulders of other components, bad buffer selection, or just the low signal to noise in the tails of the peak. Zoom in on the R_g and MW values across the peak to verify that these show a significant flat region.

RAW can automatically determine a good sample region (good being defined as monodisperse and excluding low signal to noise data). To do this, click the 'Auto' button in the Sample region.



- *Checkpoint:* You should see a sample range show up in the sample list, with defined start and end values. The region will be shown in green on the Subtracted plot.



Series: profile_001

Series type: SEC-SAXS
Vc Mol. type: Protein
Vp density (kDa/A^3): 0.00083
Averaging window size: 5

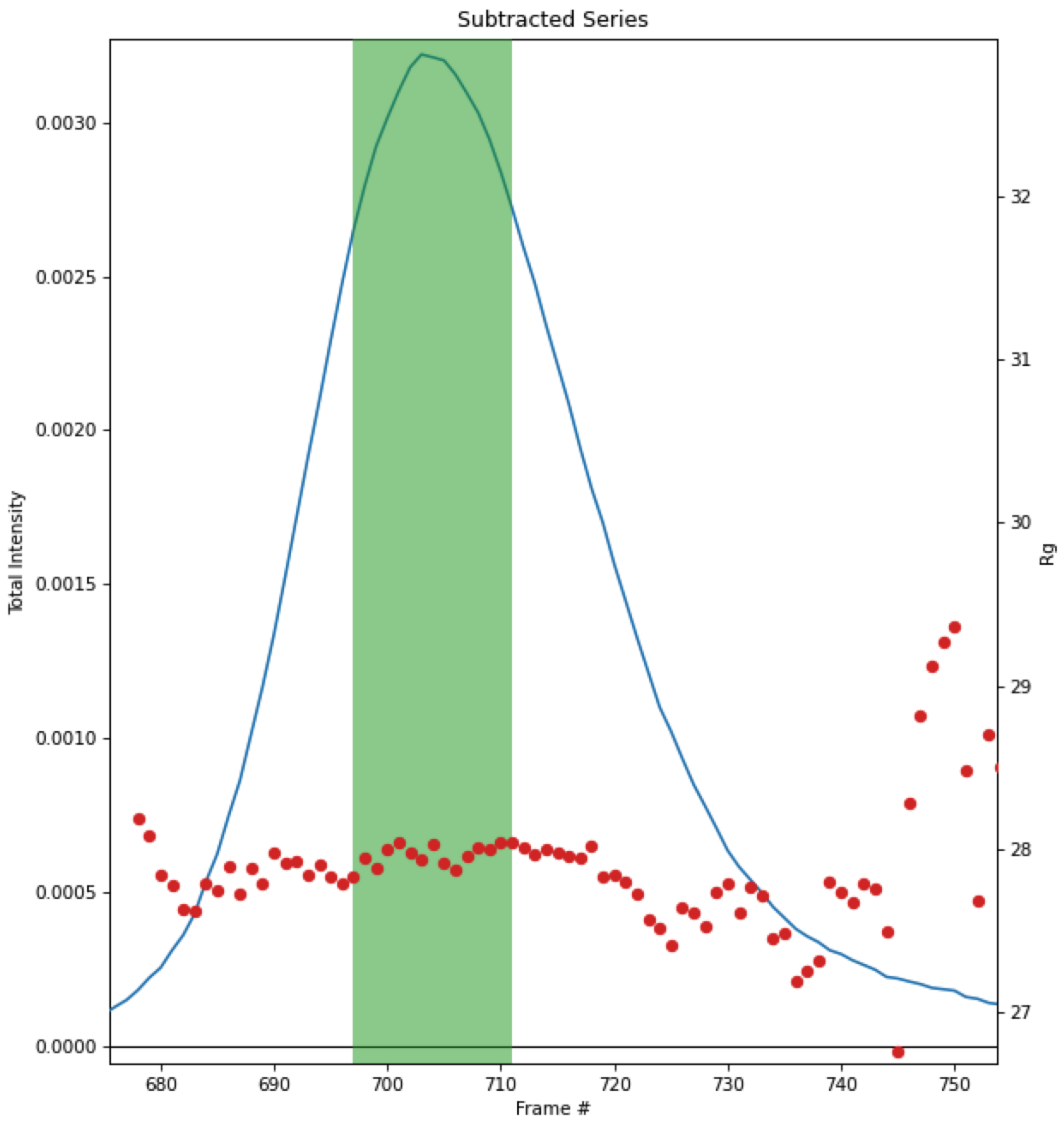
Series is already subtracted

Buffer Start 434, End 492

Sample Start 697, End 711

Intensity: Total Intensity
Calculated value: Rg

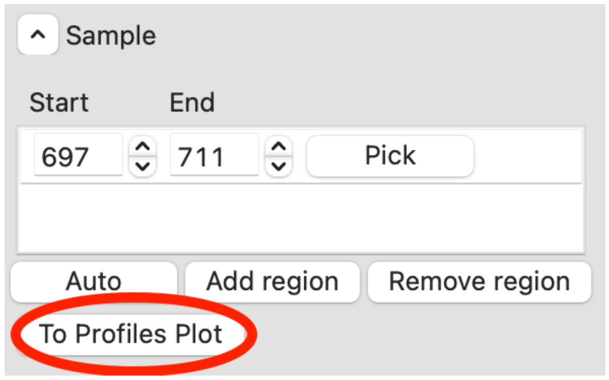
18. In the plot, zoom in on the peak region and verify that the R_g and MW seem flat in the selected sample range.



- *Tip:* You can manually adjust the sample region range in the same way as the buffer range, using the controls in the Start/End boxes.

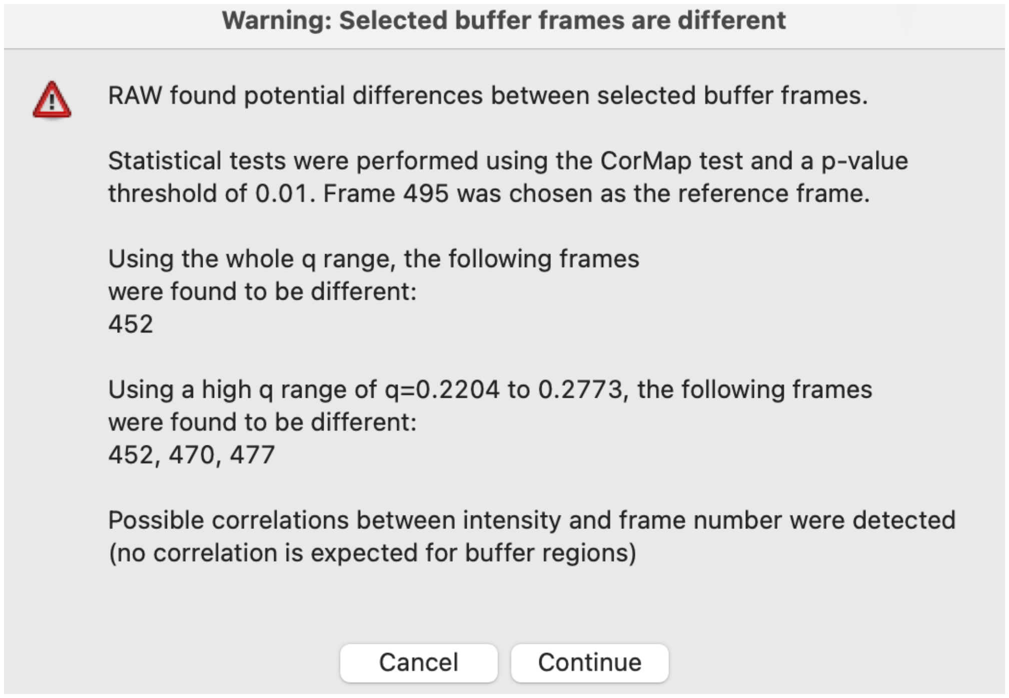
19. Once you are satisfied with the region picked (should be 697-711), click the 'To Profiles Plot' button. This averages the selected region and sends the resulting average to RAW's Profiles Plot.

    - *Note:* RAW first averages the selected sample and buffer regions in the unsubtracted data, then subtracts. This avoids the possibility of correlated noise that would arise from averaging the subtracted files.



20. If you adjust the sample or buffer region in a way that could be problematic, RAW will warn you. Try this.

    - Adjust the Buffer end to include more of the elution range, such as ending at 520. You will want to click on the 'Unsubtracted' plot to see the buffer range. Then click 'Set Buffer'. You will see a warning window telling you what might be wrong with the selected region. Click 'Cancel'.



Warning: Selected buffer frames are different

RAW found potential differences between selected buffer frames.

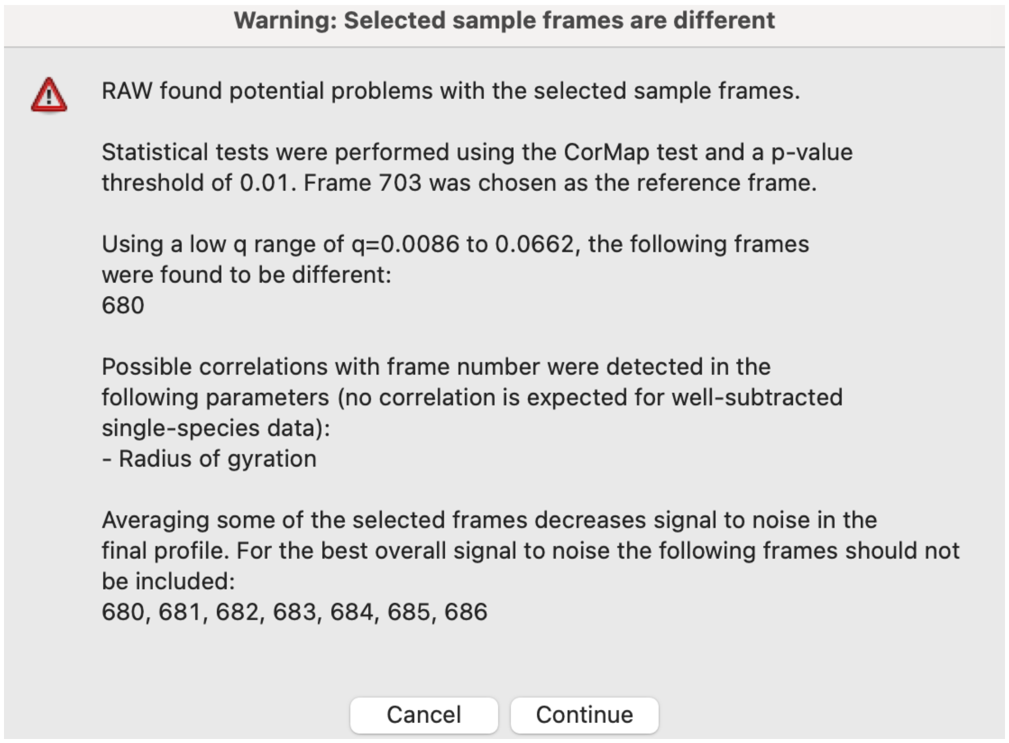
Statistical tests were performed using the CorMap test and a p-value threshold of 0.01. Frame 495 was chosen as the reference frame.

Using the whole q range, the following frames
were found to be different:
452

Using a high q range of q=0.2204 to 0.2773, the following frames
were found to be different:
452, 470, 477

Possible correlations between intensity and frame number were detected
(no correlation is expected for buffer regions)

    - Adjust the Sample start to include some of the non-flat Rg region, such as starting at 680. Then click 'To Profiles Plot'. You will see a warning window telling you what might be wrong with the selected region. Click 'Cancel'.



Warning: Selected sample frames are different

RAW found potential problems with the selected sample frames.

Statistical tests were performed using the CorMap test and a p-value threshold of 0.01. Frame 703 was chosen as the reference frame.

Using a low q range of q=0.0086 to 0.0662, the following frames were found to be different:
680

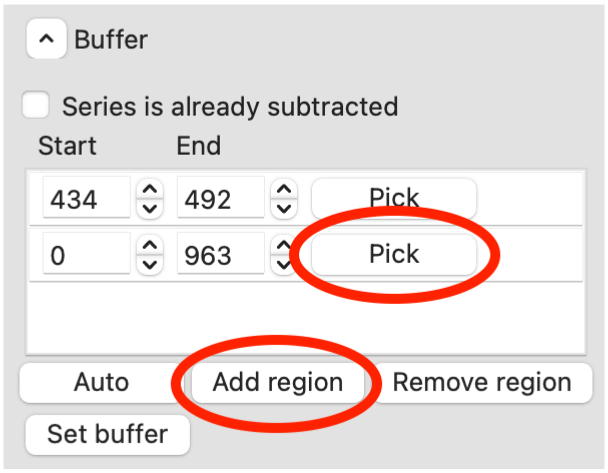
Possible correlations with frame number were detected in the following parameters (no correlation is expected for well-subtracted single-species data):
- Radius of gyration

Averaging some of the selected frames decreases signal to noise in the final profile. For the best overall signal to noise the following frames should not be included:
680, 681, 682, 683, 684, 685, 686

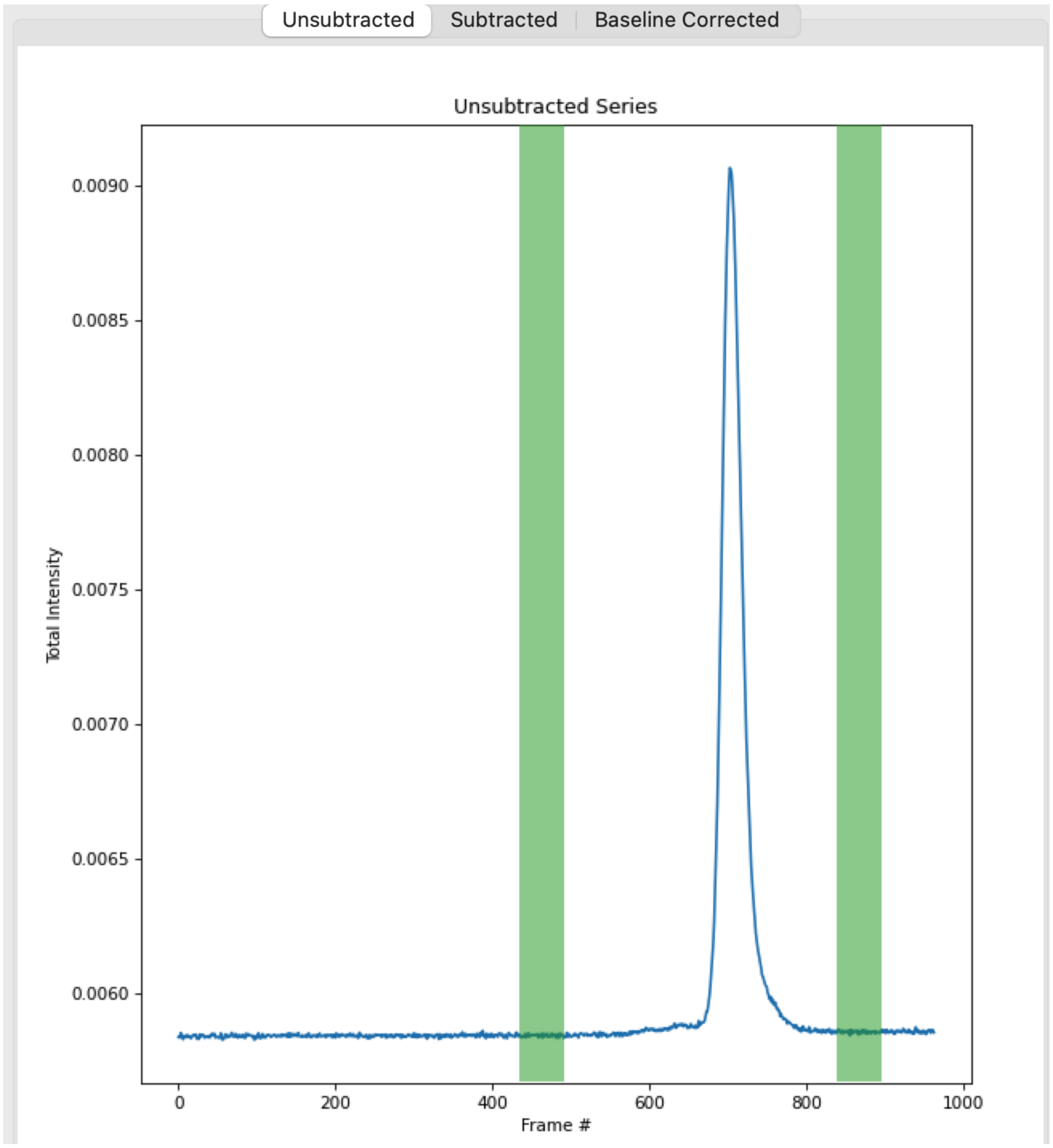
- *Note:* For buffer regions, RAW checks frame-wise similarity across the whole *q* range and at low and high *q*, correlations in intensity, and whether there are multiple singular values in the selected region.

  For sample regions, RAW checks frame-wise similarity across the whole *q* range and at low and high *q*, correlations in calculated values, whether there are multiple singular values in the selected region, and if some of the selected frames decrease the signal to noise of the average.

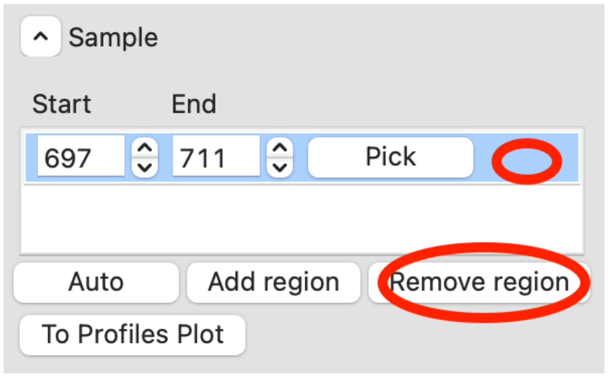
21. Click 'OK' to close the window and save your analysis results. In the Info panel above the Series control panel you should see information about the series, including the selected buffer and sample ranges. If you reopen the LC analysis window you will see the buffer and sample regions you selected are remembered.
22. Click on the Profiles plot tab and the Profiles tab. You should see one scattering profile, the buffer subtracted data set you sent to the Profiles plot. Carry out Guinier and MW analysis.
    - *Note:* The I(0) reference and absolute calibration will not be accurate for SEC-SAXS data, as the concentration is not accurately known.
    - *Question:* How does the R_g and MW you get from the averaged curve compare to what RAW found automatically for the peak?
    - *Tip:* Make sure your plot axes are Log-Lin or Log-Log. Make sure that both plots are shown by clicking the 1/2 button at the bottom of the plot window.
23. This particular dataset shows a small difference between initial and final buffer scattering profiles. A better scattering profile might be obtained by using buffer from both sides of the peak. To do so, start by reopening the LC Series Analysis panel.
24. Switch to showing the unsubtracted intensity by clicking on the 'Unsubtracted' plot tab.
25. Add a second buffer region by clicking the 'Add region' button.
26. For the second region, click the 'Pick' button.



27. Move your mouse across the plot. You will see a vertical green line moving with the mouse cursor. This represents the start of the buffer region. Click once to fix the start point where you click. Move the mouse further to the right and click again to fix the end point of the buffer region.



28. Once you are happy with the second buffer region, click 'Set buffer'. A range like ~840-896 is reasonable.
29. A warning window will pop up. In this case, we have purposefully chosen two buffer regions because they are different, so ignore the warning and click 'Continue'.
30. Remove the old sample region by clicking in the empty space to the right of the 'Pick' button to highlight it, and then clicking the 'Remove region' button.



31. Click the 'Auto' button to automatically find a new sample region. Click the 'To Profiles Plot' button to send that new region to the Profiles plot.

    - *Try:* You can see what the data subtracted by just the second buffer region looks like by removing the first buffer region, setting the buffer again, finding a new good sample region, and sending new region to the Profiles plot.

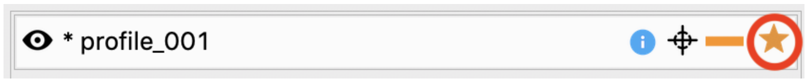
32. Cancel out of the LC Series analysis window. This will not save the changes you made to the buffer and sample regions.

33. Carry out the R_g and MW analysis on the new curve. How does the scattering profile compare to the one that you generated using only buffer from before the peak?

    - *Tip:* You should see subtle but noticeable differences in the Guinier fit.
    - *Note:* An alternative approach to using several buffer regions is to use a single buffer region and apply a baseline correction. Both approaches have advantages and disadvantages. If you want to do EFA deconvolution, it is best to not use a baseline correction, however in other cases it will be more accurate as it doesn't assume a single average buffer across the peak.

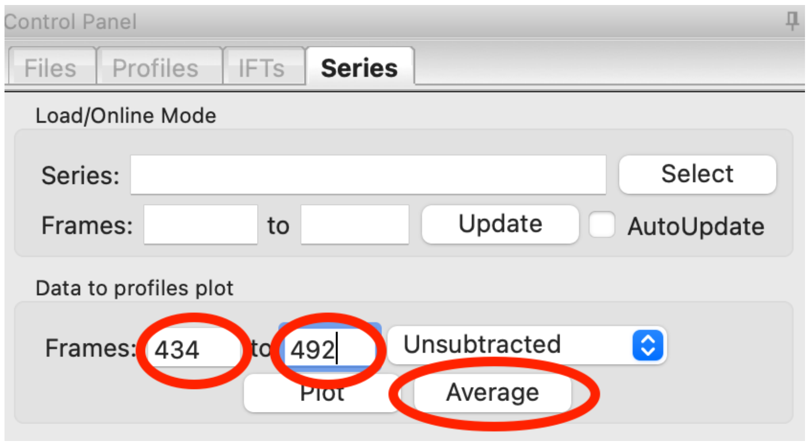
34. Return to the Series control and plot panels.

35. If you want to look at either individual profiles or the average of a range of profiles you can send profiles to the Profiles plot. To select which series curve to send profiles from, star the series curve of interest.



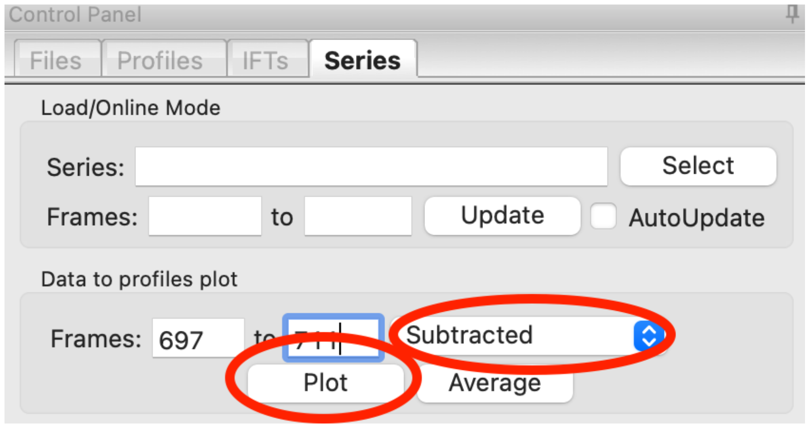
36. In the 'Data to Profiles plot' section enter the frame range of interest. For this dataset, try the buffer range you selected: 539 to 568. Then click the 'Average' button. That will send the average buffer to the Profiles plot.

    - *Try:* Send the average of the sample range you selected to the main plot (699 to 713), carry out the subtraction, and verify it's the same as the curve produced by the 'To Profiles Plot' button in the LC Series Analysis panel.
    - *Question:* When you send the sample average to the Profiles plot you will get a warning that the profiles are different. Why?



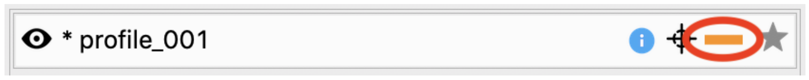
37. You can also send subtracted (or baseline corrected data) to the Profiles plot. For the selected sample range, select the 'Subtracted' frames and send each individual profile to the plot using the 'Plot' button.

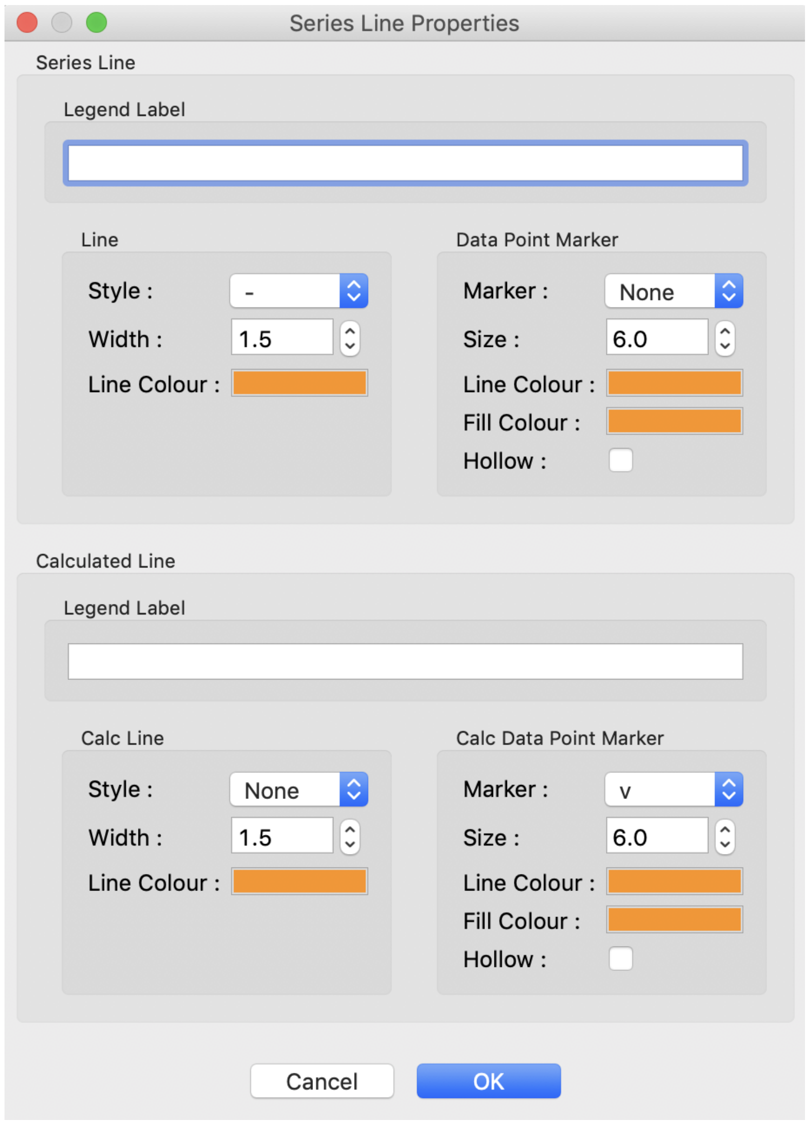
   - *Try:* Average these profiles and verify they match the subtracted profiles for this data set generated previously.



38. Click on the colored line next to the star in the Series control panel. In the line properties control panel this brings up, change the Calc Marker color to something different. Add a line to the Calc Markers by selecting line style '-' (solid), and adjust the line color to your liking.

   - *Tip:* You can do the same thing to adjust the colors of the scattering profiles and IFTs in the Profiles and IFT control tabs.




Series Line Properties

Series Line

Legend Label

Line

Style : -

Width : 1.5

Line Colour :

Data Point Marker

Marker : None

Size : 6.0

Line Colour :

Fill Colour :

Hollow :

Calculated Line

Legend Label

Calc Line

Style : None

Width : 1.5

Line Colour :

Calc Data Point Marker

Marker : v

Size : 6.0

Line Colour :

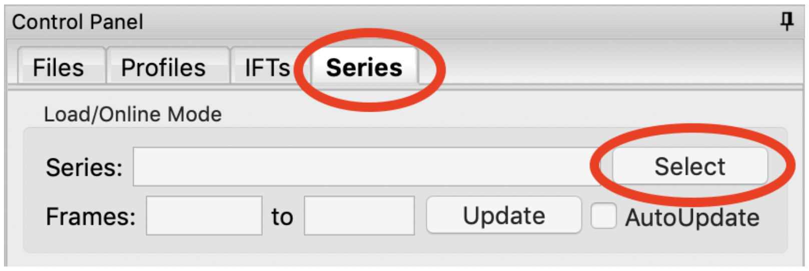
Fill Colour :

Hollow :

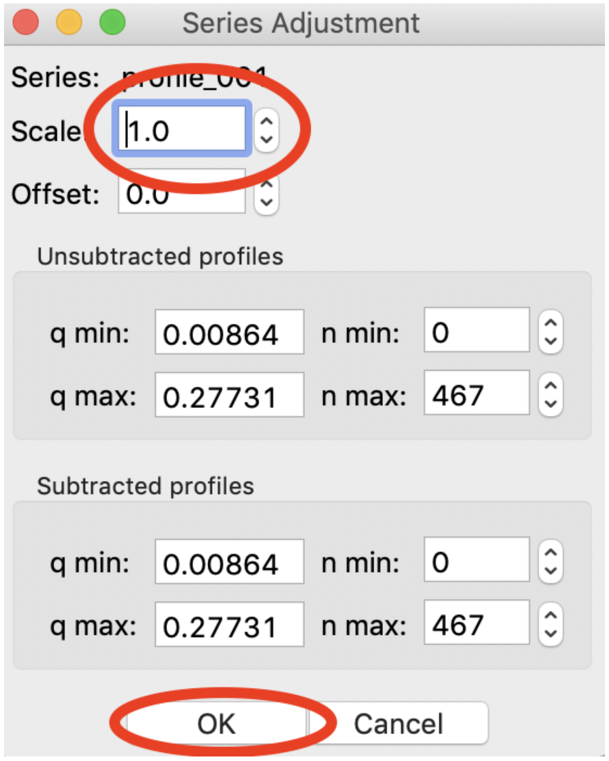
39. For certain beamlines (the BioCAT beamline at the APS and the MacCHESS BioSAXS beamline at CHESS), RAW can automatically load in series data from the series panel. This is typically used for online analysis while data is being collected, but can be used to load in series you have already collected as well.

40. We will load in the Bovine Serum Albumin (BSA) SEC-SAXS data contained in the **sec_sample_2** data folder using this automatic method. In the Series control panel, click the "Select" button. Navigate to the **Tutorial_Data/series_data/sec_sample_2** folder and select any of the **.dat** files in the folder.

    - *Troubleshooting:* If you get an error message, it means you don't have a configuration file loaded. Load the SAXS.cfg file referenced *earlier*.
    - The configuration file must be set to either BioCAT or MacCHESS beamlines for this method to work. Otherwise, RAW doesn't know how to create all the filenames in a series from a single filename.



41. The SEC-SAXS run will automatically load. Note that because SAXS data can be reported with an arbitrary intensity scale, the total intensity of this series is much larger than the previous series.
42. Right click on the **profile_001** series and select “Adjust scale, offset, q range”. This will open a window that allows you to adjust the overall scale and offset for your series data, as well as the q range used for each type of profile in the series data. Set the scale to 1800 and click “OK”. You should see that this scales the profiles_001 series to match the BSA series.
    - *Tip:* This applies the scale, offset, and q range settings to every profile in the series. So if you were now to send a profile to the Profiles plot, it would have an overall scale factor of 1800 applied to it.



Series: profile_001
Scale: 1.0
Offset: 0.0

Unsubtracted profiles
q min: 0.00864 n min: 0
q max: 0.27731 n max: 467

Subtracted profiles
q min: 0.00864 n min: 0
q max: 0.27731 n max: 467

43. Hide the first series (**profile_001**).
44. Select a good buffer region, and calculate the R_g and MW across the peak for the BSA.
    - *Tip:* If you hover your mouse cursor over the info icon, you will see the buffer range and window size used to calculate the parameters.
    - *Question:* Is the BSA peak one species?
45. Find the useful region of the peak (constant R_g/MW), and send the buffer and sample data to the Profiles plot. Carry out the standard R_g and MW analysis on the subtracted scattering profile. For BSA, we expect R_g ~28 Å and MW ~66 kDa.

46. In the Series control tab, right click on the name of BSA curve in the list. Select export data and save it in an appropriate location. This will save a CSV file with the frame number, integrated intensity, radius of gyration, molecular weight, filename for each frame number, and a few other items. This allows you to plot that data for publications, align it with the UV trace, or whatever else you want to do with it.

    - *Try:* Open the **.csv** file you just saved in Excel or Libre/Open Office Calc.

47. Select both items in the Series control panel list, and save them in the **series_data** folder. This saves the series data in a form that can be quickly loaded by RAW.

    - *Try:* Clear the Series data and then open one of your saved files from the Files tab using either the "Plot" or "Plot Series" button.

48. With both items selected, right click and select "Save report". Check the profiles assocaitated with the SEC-SAXS series, so that you're saving a report that includes both the series and the averaged subtracted profiles. Save the report as a pdf.

WAXS processing and merging

Several SAXS beamlines use two (or more) detectors to collect different q regions. For example, the MacCHESS G1 beamline used dual Pilatus detectors to measure SAXS and WAXS from q ~0.008 – 0.75 Å^{-1}. The SAXS detector has q ~< 0.25 Å^{-1} and the wide-angle scattering (WAXS) data has q >~ 0.25 Å^{-1}. All of the data that you have been working with so far has been SAXS data. Some experiments can make use of the WAXS data. In this part of the tutorial you will learn the basics of processing it.

A video version of this tutorial is available:

The written version of the tutorial follows.

1. Clear any data in RAW.
2. Navigate to the **standards_data** and load the **WAXS.cfg** file.
3. Plot the **lysbuff2 PIL3** and **lys2 PIL3** files. These are the images from the WAXS detector. Average these files and create a subtracted WAXS scattering profile.

    - *Tip:* Filenames should be **lysbuff2_<stuff>_PIL3_<stuff>.tiff** and **lys2_<stuff>_PIL3_<stuff>.tiff**.

4. Load the saved subtracted SAXS scattering profile for the lysozyme standards data.

    - *Note:* You should have saved it in the **standards_data** folder, and it is likely named **S_A_lys2_A9_17_001_0000.dat**.

5. Move the SAXS scattering profile you just loaded to the bottom plot by right clicking on it in the Profiles list and selecting "Move to bottom plot."
6. The WAXS data is not on the same scale as the SAXS data. For this data, the known scale factor to apply is 0.000014 to the WAXS data.

    - *Note:* The scale factor can be calculated as the ratio of solid angles subtended by the pixels on the SAXS and WAXS detectors, plus any scale factor for absolute calibration and normalization included for one curve but not the other.

7. Star the WAXS data. Right click on the SAXS data and select Other Operations -> Merge. This will create a new merged scattering profile. The new file will have the prefix **M_** to indicate it is a merged file.

    - *Tip:* If you can't see it, that's probably because it appeared on the upper plot, and is hidden by the very large intensities of the averaged WAXS files. Either try hiding those, or move the Merged curve to the lower plot.

A few additional tricks

Here are some additional tricks that may make your life easier while using RAW:

1. If you click on a scattering profile in the Profiles plot, the corresponding Profiles list item will be highlighted.
2. You can save the workspace by going to File->Save Workspace. This will save all of the scattering profiles, IFT curves, and Series curves. These will all be loaded again when you load the workspace.
   - *Note:* This does not save the settings!
3. If you go to Options -> Advanced Options -> Molecular weight, you can change the default type of molecule used in the MW estimation from the volume of correlation. This affects the default option selected in the MW window.
4. If you have the crossed arrows selected in the plot control bar to drag a plot, right clicking and dragging allows you to zoom a plot.
5. You can turn error bars on and off for scattering profiles using the error bar button in the plot control (to the right of the save button).
6. You can rename a curve by right clicking on the appropriate entry in the list and choosing rename.
7. You can view the history of a scattering profile by right clicking on it and selecting Show History. For a curve that has been processed from an image, this will show you processing parameters such as normalization and any corrections applied to the scattering intensity. For a curve that is processed (such as an averaged of subtracted curve) it will show you the steps used to make that curve. For example, for an averaged curve, it will show you all of the files that were averaged.
8. All of the main RAW plots read out the mouse coordinates at the bottom of the RAW window, just below all of the tool buttons.

5.3.3 Advanced processing

This section will guide you through advanced processing of files in RAW. It includes: pair-distance distribution analysis using GNOM and BIFT methods, ambiguity assessment for shape reconstructions, 3D reconstructions with bead models and electron density, aligning 3D reconstructions with PDB files, SEC-SAXS data deconvolution using singular value decomposition (SVD) and evolving factor analysis (EFA), baseline corrections for SEC-SAXS data, and regularized alternating least squares (REGALS) of SEC-SAXS and other SAXS data. It refers to the *RAW tutorial data*.

Select a section below to view the tutorial, or use the next and back buttons at the bottom of the page to navigate through it in order.

Pair-distance distribution analysis – GNOM in RAW

The first step in most advanced data processing is to calculate the P(r) function, the Fourier transform of I(q). This cannot be calculated directly from the scattering profile, so indirect Fourier transform (IFT) methods are typically used. The most common such method is implemented in the GNOM program from the ATSAS package. We will use RAW to run GNOM. Note that you need *ATSAS installed* to do this part of the tutorial.

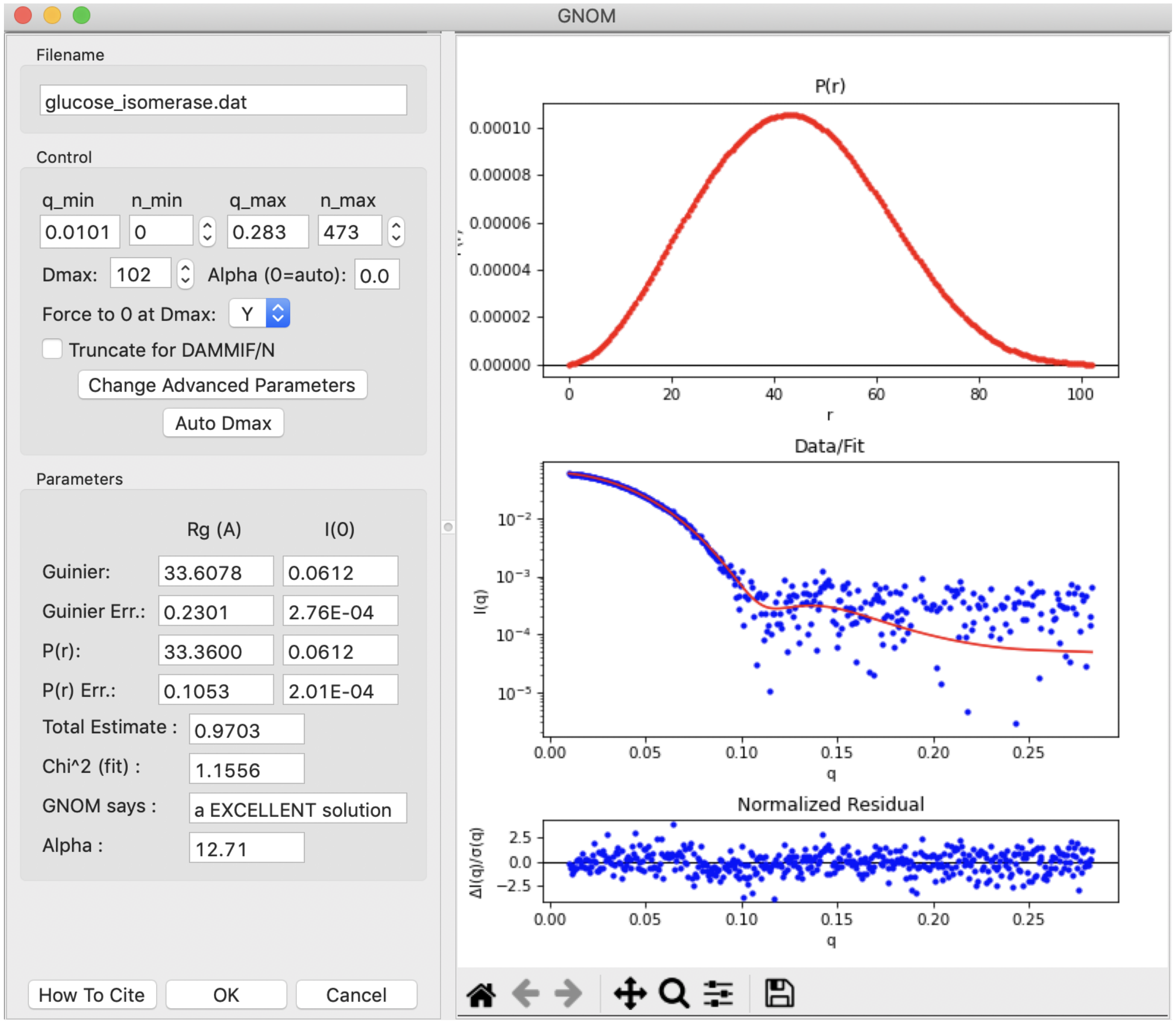
This tutorial covers how to use RAW for doing an IFT. This is not a tutorial on basic principles and best practices for doing an IFT or analysis of the resulting P(r) function. For that, please see the *SAXS tutorial*.

If you use RAW to run GNOM, in addition to citing the RAW paper, please cite the paper given in the GNOM manual.

A video version of this tutorial is available:

The written version of the tutorial follows.

1. Open RAW. The install instructions contain information on installing and running RAW. If RAW is already open, clear all loaded data in RAW.
2. Open the **glucose_isomerase.dat** file in the **Tutorial_Data/reconstruction_data** folder.
3. Right click on the glucose_isomerase profile in the Profiles list and select "IFT (GNOM)".
   - *Note:* RAW will automatically try to find an appropriate maximum dimension (D_{max}).
   - *Troubleshooting:* If you do not have the GNOM option in the right click menu, RAW does not know where your ATSAS programs are installed. If you installed the ATSAS programs after starting RAW, restart RAW and it will attempt to automatically find them. If that has failed, go to the Options->Advanced Options menu and choose the ATSAS settings ("ATSAS"). Uncheck the option to "Automatically find the ATSAS bin location", and specify the location manually.



GNOM

Filename

glucose_isomerase.dat

Control

q_min	n_min	q_max	n_max
0.0101	0	0.283	473

Dmax: 102 Alpha (0=auto): 0.0

Force to 0 at Dmax: Y

Truncate for DAMMIF/N

Parameters

	Rg (A)	I(0)
Guinier:	33.6078	0.0612
Guinier Err.:	0.2301	2.76E-04
P(r):	33.3600	0.0612
P(r) Err.:	0.1053	2.01E-04

Total Estimate : 0.9703

Chi^2 (fit) : 1.1556

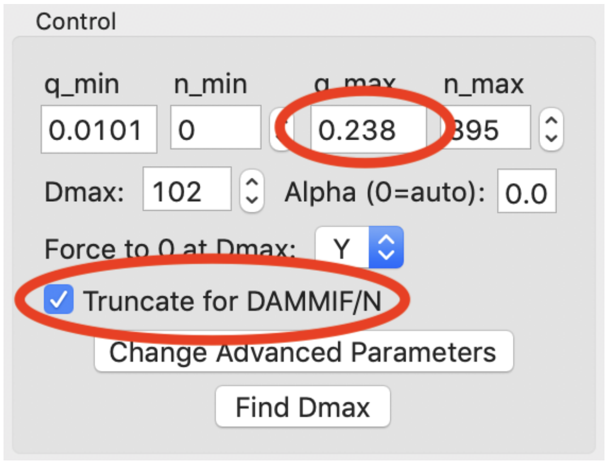
GNOM says : a EXCELLENT solution

Alpha : 12.71

4. The GNOM panel has plots on the right. These show the P(r) function (top panel), the data (middle panel, blue points) and the fit line (middle panel, red line), and the fit residual (bottom panel).
   - *Note:* The fit line is the Fourier transform of the P(r) function, and is also called the regularized intensity.
5. On the left of the GNOM panel are the controls and the resulting parameters. You can alter the data range used, the D_{max} value, whether the solution is forced to zero at D_{max}, and the alpha value used.
   - *Tip:* The Guinier and P(r) R_g and I(0) values should agree well for mostly rigid particles. The total estimate varies from 0 to 1, with 1 being ideal. GNOM also provides an estimate of the quality of the solution. You

want it to be at least a “REASONABLE” solution.

6. Try varying the D_{max} value up and down in the range of 80-110. Observe what happens to the P(r) and the quality of the solution.
   - *Note:* D_{max} is in units of Å.
7. Return the D_{max} value to that found by RAW by clicking the “Auto Dmax” button. D_{max} should be 102. By default, GNOM forces the P(r) function to zero at D_{max}. For a high quality data set and a good choice of D_{max}, P(r) should go to zero naturally. Change the “Force to 0 at Dmax” option to “N”.
   - *Try:* Vary D_{max} with this option turned off.
8. Reset it so that the P(r) function is again being forced to zero at D_{max}.
9. RAW makes it easy to truncate your data for bead model reconstructions with , DAMMIF/N by setting q_{max} to $8/R_g$ or 0.30, whichever is smaller. Check the “Truncate for DAMMIF/N” box to truncate the data.


Control

q_min: 0.0101 n_min: 0 q_max: 0.238 n_max: 395

Dmax: 102 Alpha (0=auto): 0.0

Force to 0 at Dmax: Y

✓ Truncate for DAMMIF/N

   - *Note:* The q_{max} goes from 0.283 to 0.238 when you check the box.
   - *Tip:* For electron density reconstruction with DENSS use the full available q range.
10. Click “OK” to exit the panel and save the IFT to the RAW IFTs panel and plot.
    - *Tip:* After exiting the IFT panel, note that data from the IFT shows up in the Information panel.
11. Click on the IFTs Control and Plot tabs. This will display the GNOM output you just generated. Save the **glucose_isomerase.out** item in the **reconstruction_data** folder.
    - *Note:* This saved file is all of the GNOM output, in the GNOM format. It can be used as input for any program that needs a GNOM **.out** file.

Pair-distance distribution analysis – BIFT in RAW

RAW has a built in method for determining the P(r) function using a Bayesian IFT method. This has the advantage of only have one possible solution. More information on this method can be found in the RAW paper and manual and references therein.

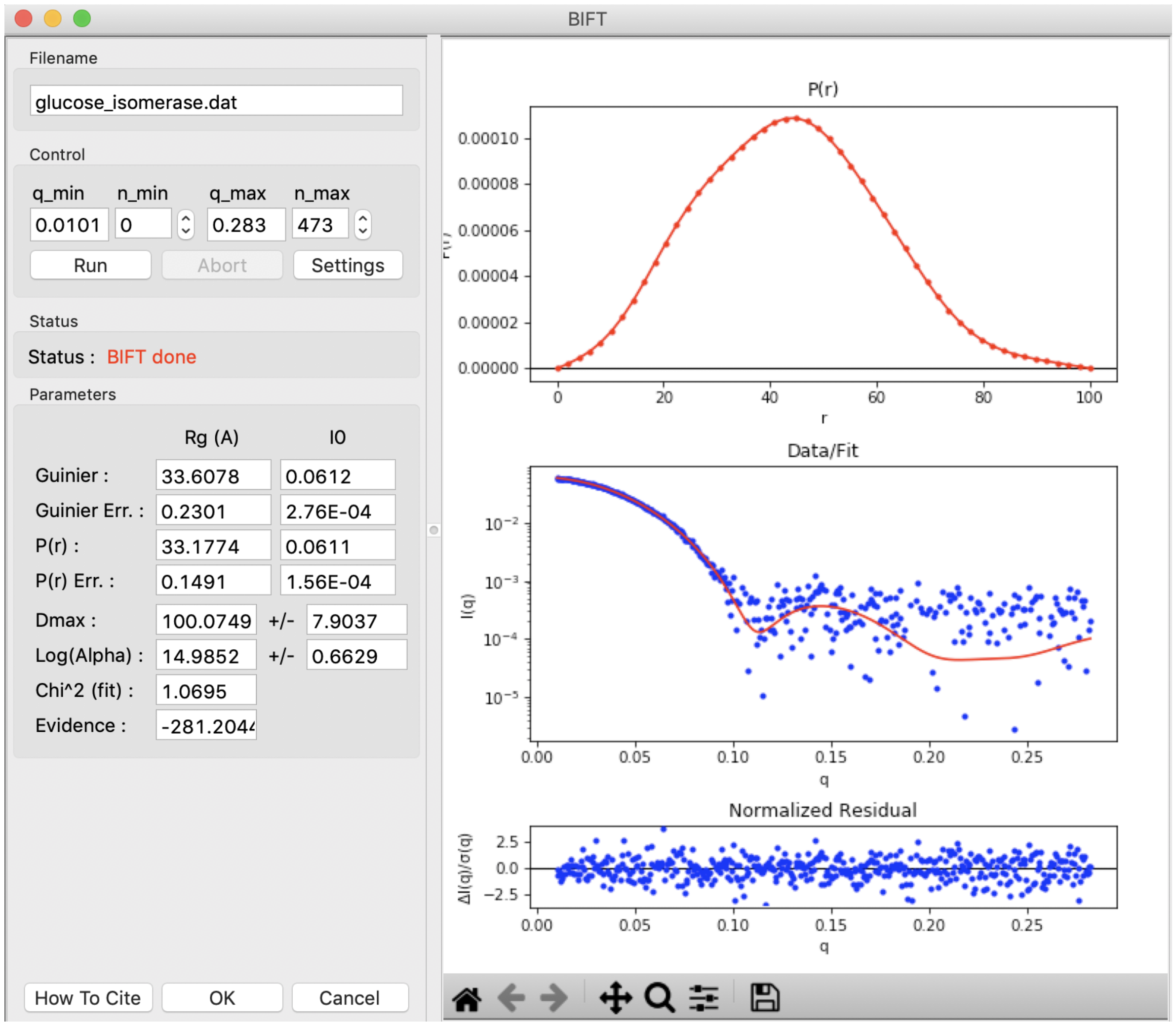
This tutorial covers how to use RAW for doing an IFT. This is not a tutorial on basic principles and best practices for doing an IFT or analysis of the resulting P(r) function. For that, please see the *SAXS tutorial*.

If you use RAW to run BIFT, in addition to citing the RAW paper, please cite the BIFT paper: Hansen, S. Journal of Applied Crystallography (2000) 33, 1415-1421. DOI: 10.1107/S0021889800012930

A video version of this tutorial is available:

The written version of the tutorial follows.

1. Right click on the glucose isomerase profile in the Profiles list you loaded *previously*. Select "IFT (BIFT)" from the resulting menu.



BIFT

Filename

glucose_isomerase.dat

Control

q_min	n_min	q_max	n_max
0.0101	0	0.283	473

Status

Status : BIFT done

Parameters

	Rg (A)		I0
Guinier :	33.6078		0.0612
Guinier Err. :	0.2301		2.76E-04
P(r) :	33.1774		0.0611
P(r) Err. :	0.1491		1.56E-04
Dmax :	100.0749	+/-	7.9037
Log(Alpha) :	14.9852	+/-	0.6629
Chi^2 (fit) :	1.0695		
Evidence :	-281.2044		

2. The BIFT panel has plots on the right. These show the P(r) function (top panel), the data (middle panel, blue points) and the fit line (middle panel, red line), and the fit residual (bottom panel).

3. On the left of the BIFT panel are the controls and the resulting parameters. Note that in this case you do not control the D_{max} value, the BIFT method finds that for you automatically. Because BIFT can take some time to run, if you change the *q* range for the data you have to click the 'Run' button to run BIFT again.

4. Note that for this dataset, BIFT has found a D_{max} value around 100, in good agreement with what we found from GNOM.

5. Click OK to exit the BIFT window. This saves the results into the IFTs panel.

6. Click on the IFTs Control and Plot tabs. This will display the BIFT output you just generated. Save the **glucose_isomerase.ift** item in the **reconstruction_data** folder.

Note: As of now, BIFT output from RAW is not compatible with DAMMIF or other ATSAS programs. However, it is compatible with electron density determination via DENSS.

Assessing ambiguity of 3D shape information - AMBIMETER in RAW

It is impossible to determine a provably unique three-dimensional shape from a scattering profile. This makes it important to determine what degree of ambiguity might be expected in our reconstructions. The program AMBIMETER from the ATSAS package does this by comparing the measured scattering profile to a library of scattering profiles from relatively simple shapes. The more possible shapes that could have generated the scattering profile, the greater ambiguity there will be in the reconstruction. We will use RAW to run AMBIMETER. Note that you need *ATSAS installed* to do this part of the tutorial.

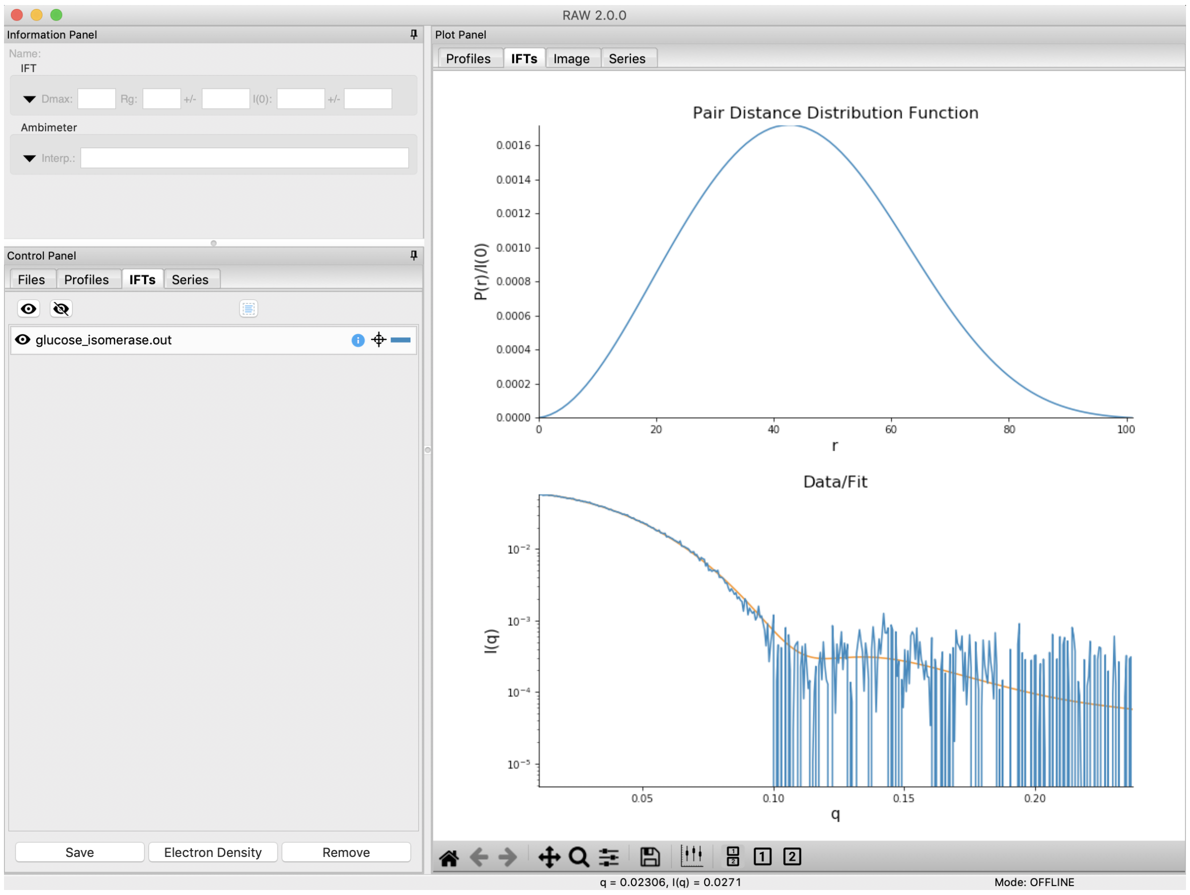
If you use RAW to run AMBIMETER, in addition to citing the RAW paper, please cite the paper given in the AMBIMETER manual.

A video version of this tutorial is available:

The written version of the tutorial follows.

1. Clear all of the data in RAW. Load the **glucose_isomerase.out** file that you saved in the **reconstruction_data** folder in a previous part of the tutorial.

   - *Note:* If you haven't done the previous part of the tutorial, or forgot to save the results, you can find the **glucose_isomerase.out** file in the **reconstruction_data/gi_complete** folder.

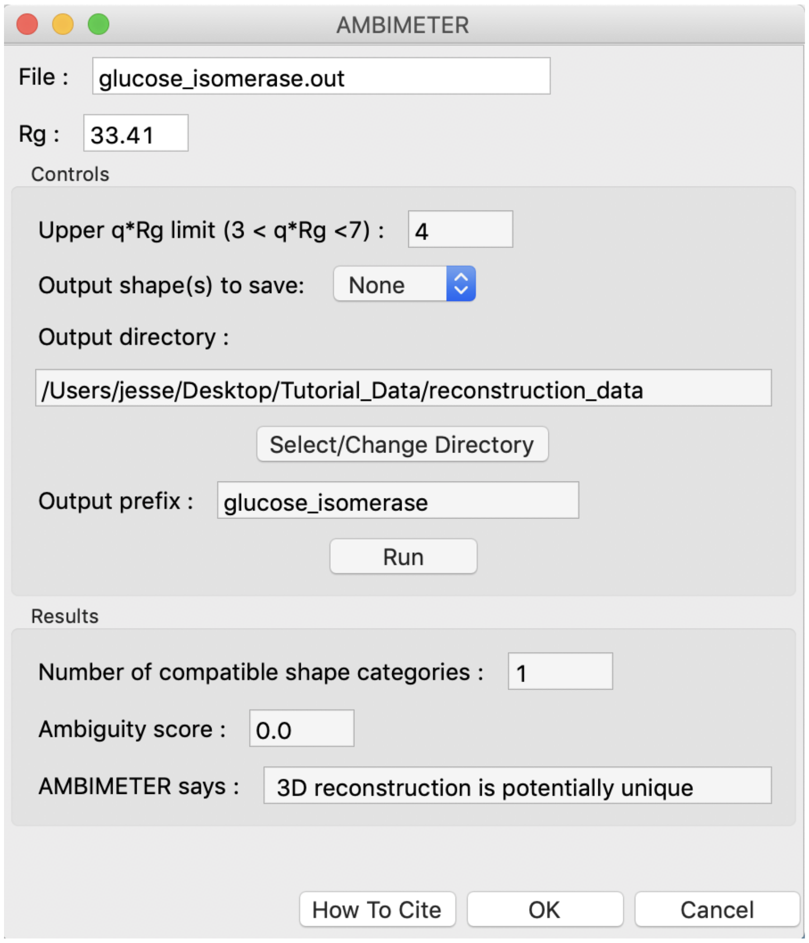


2. Right click on the **glucose_isomerase.out** item in the IFT list. Select the "AMBIMETER" option.

3. The new window will show the results of AMBIMETER. It includes the number of shape categories that are compatible with the scattering profile, the ambiguity score (a-score) (log base 10 of the number of shape categories), and the AMBIMETER interpretation of whether or not you can obtain a unique 3D reconstruction.

   - According to the original paper, "an a-score below 1.5 practically guarantees a unique ab initio shape determination, whereas when the a-score is in the range 1.5–2.5 care should be taken, perhaps involving cluster analysis, and for a-scores exceeding 2.5 unambiguous reconstruction without restrictions (for example, on symmetry and/or anisometry) is highly unlikely."

- *Note:* AMBIMETER can also save the compatible shapes (either all or just the best fit). You can do that by selecting the output shapes to save, giving it a save directory, and clicking run. We won't be using those shapes in this tutorial.



File : glucose_isomerase.out

Rg : 33.41

Upper q*Rg limit (3 < q*Rg <7) : 4

Output shape(s) to save: None

Output directory : /Users/jesse/Desktop/Tutorial_Data/reconstruction_data

Output prefix : glucose_isomerase

Number of compatible shape categories : 1

Ambiguity score : 0.0

AMBIMETER says : 3D reconstruction is potentially unique

4. Click "OK" to exit the AMBIMETER window.

   - *Tip:* After exiting the AMBIMETER window the results can be seen in the Information panel when the IFT is selected in the IFTs list.

3D reconstruction with bead models – DAMMIF/N and DAMAVER in RAW

Shape reconstruction in SAXS is typically done using bead models (also called dummy atom models, or DAMs). The most common program used to generate these shapes is DAMMIF (and, to a lesser degree, DAMMIN) from the ATSAS package. We will use RAW to run DAMMIF/N. Because the shape reconstruction is not unique, a number of distinct reconstructions are generated, and then a consensus shape is made from the average of these reconstructions. The program DAMAVER from the ATSAS package is the most commonly used program for building consensus shapes. Note that you need *ATSAS installed* to do this part of the tutorial.

This is not a tutorial on basic principles and best practices for doing bead model reconstructions. For that, please see the *SAXS tutorial*.

If you use RAW to run DAMMIF or associated programs, in addition to citing the RAW paper, please cite the papers given in the:

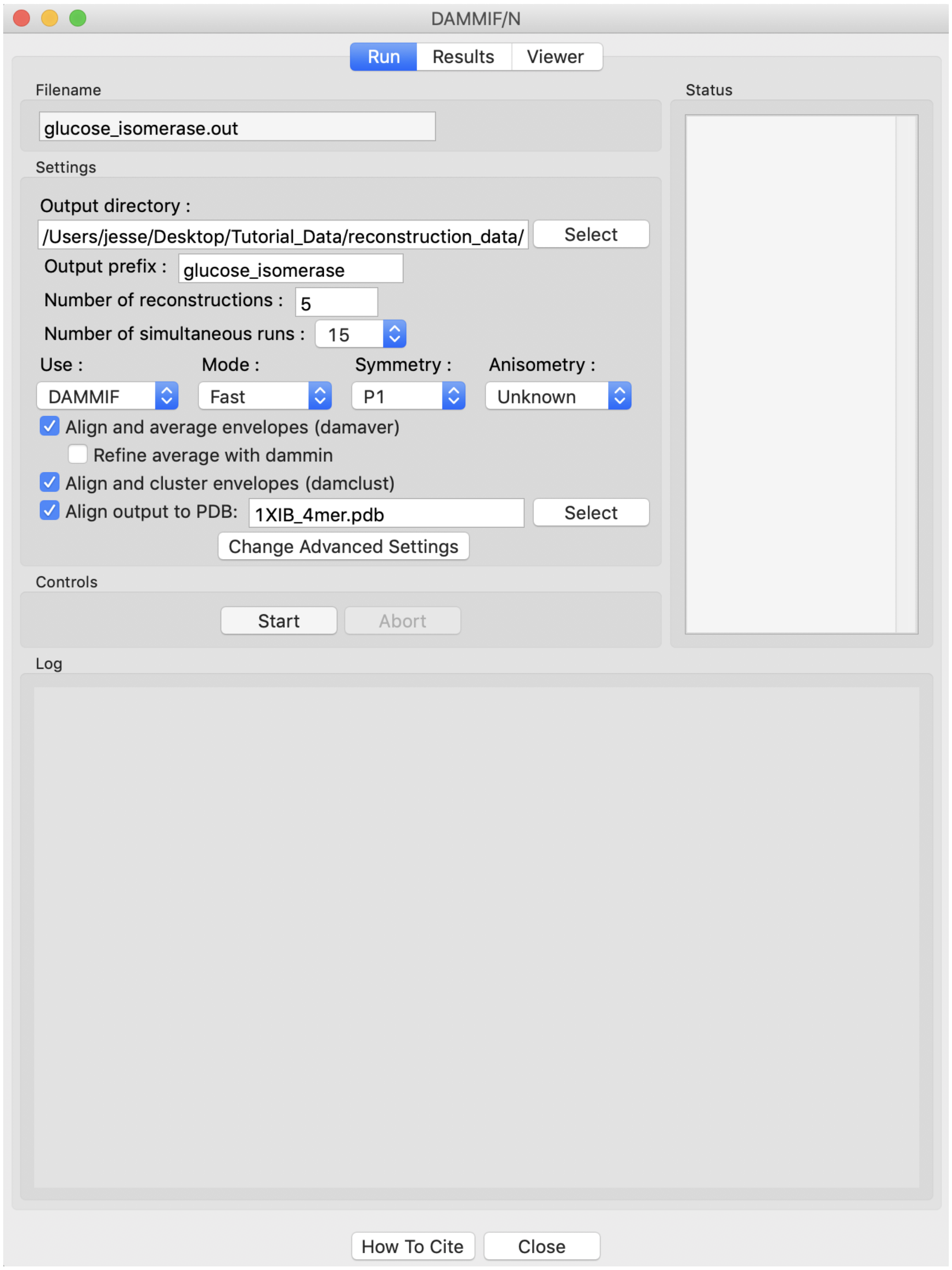
- DAMMIF manual
- DAMMIN manual
- DAMAVER manual
- DAMCLUST manual
- SASRES manual
- SUPCOMB manual

as appropriate

A video version of this tutorial is available:

The written version of the tutorial follows.

1. Clear all of the data in RAW. Load the **glucose_isomerase.out** file that you saved in the **reconstruction_data** folder in a previous part of the tutorial.
    - *Note:* If you haven't done the previous part of the tutorial, or forgot to save the results, you can find the **glucose_isomerase.out** file in the **reconstruction_data/gi_complete** folder.
2. Right click on the **glucose_isomerase.out** item in the IFT list. Select the "Bead Model (DAMMIF/N)" option.
3. Running DAMMIF generates a lot of files. Click the "Select" button for the output directory, make a new folder in the **reconstruction_data** directory called **gi_dammif** and select that folder.
4. Change the number of reconstructions to 5.
    - *Note:* It is generally recommended that you do 15-20 reconstructions. However, for the purposes of this tutorial, 5 are enough.
    - *Note:* For final reconstructions for a paper, DAMMIF should be run in Slow mode. For this tutorial, or for obtaining an initial quick look at results, Fast mode is fine.
5. Uncheck the "Refine average with dammin" checkbox.
    - *Note:* For final reconstructions for a paper, DAMMIN refinement should be done. However, it is quite slow, so for the purposes of this tutorial we won't do it.
6. If it's not already checked, check the "Align and cluster envelopes (damclust)" checkbox.
7. RAW can align the DAMMIF/N output with a PDB structure using SUPCOMB from the ATSAS package. To do so, check the 'Align output to PDB' box and select the **1XIB_4mer.pdb** file in the **reconstruction_data/gi_complete** folder.
    - *Tip:* If you're not sure if you selected the correct file, hovering your mouse over the filename will show the full path to the file.



Filename

glucose_isomerase.out

Settings

Output directory : /Users/jesse/Desktop/Tutorial_Data/reconstruction_data/

Output prefix : glucose_isomerase

Number of reconstructions : 5

Number of simultaneous runs : 15

Use : DAMMIF

Mode : Fast

Symmetry : P1

Anisometry : Unknown

☑ Align and average envelopes (damaver)

☐ Refine average with dammin

☑ Align and cluster envelopes (damclust)

☑ Align output to PDB: 1XIB_4mer.pdb

Controls

Status

Log

8. Click the "Start" button.

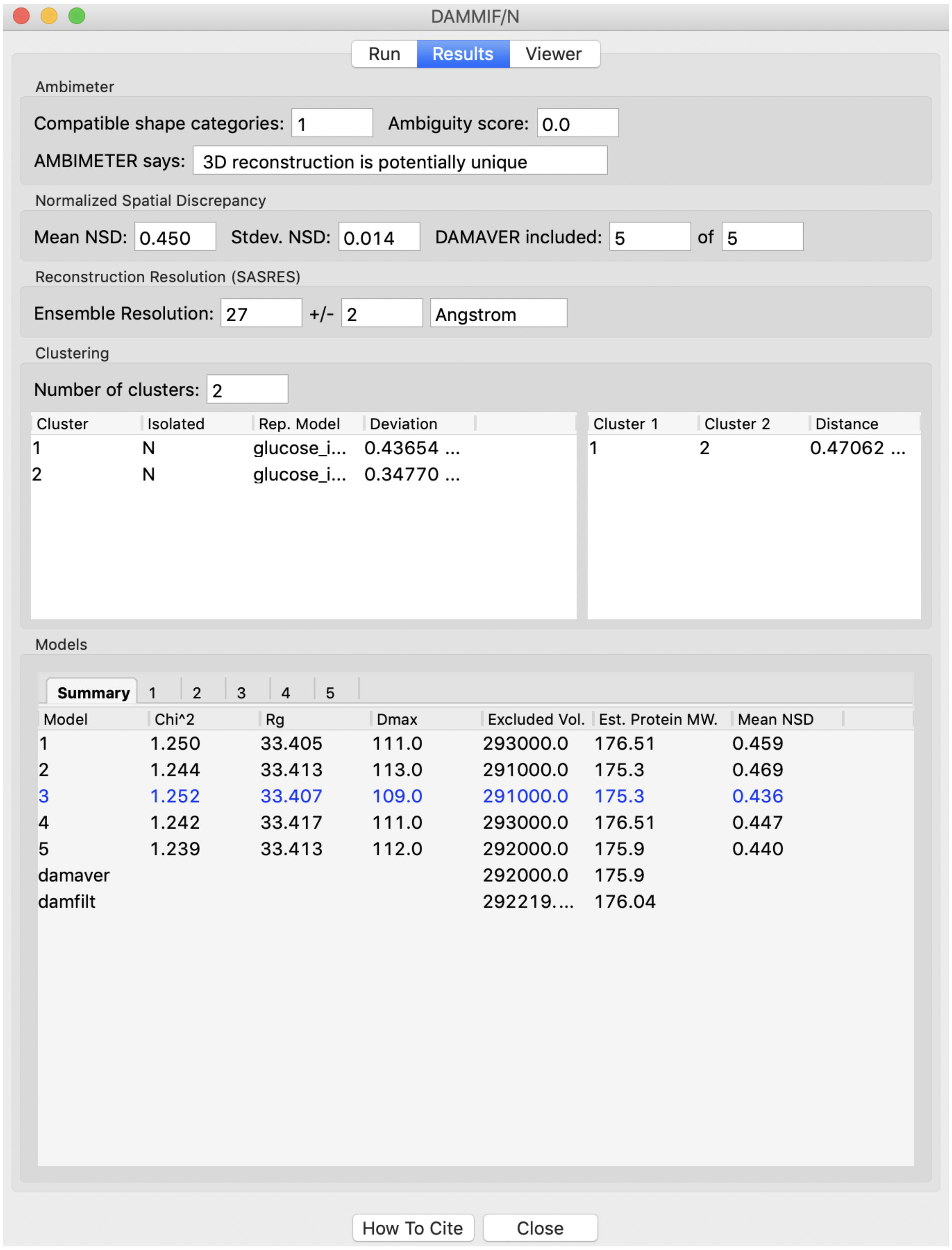
   • *Note:* The status panel will show you the overall status of the reconstructions. You can look at the detailed status of each run by clicking the appropriate tab in the log panel.

9. Note that by default the envelopes are aligned and averaged using DAMAVER, and then the aligned and averaged profile is refined using DAMMIN. Clustering analysis is also done by default.

   - Some settings are accessible in the panel, and all settings can be changed in the advanced settings panel.

10. Wait for all of the DAMMIF runs, DAMAVER, DAMCLUST, and alignment to finish. Depending on the speed of your computer this could take a bit.

11. Once the reconstructions are finished, the window should automatically switch to the results tab. If it doesn't, click on the results tab.



Ambimeter

Compatible shape categories: 1 Ambiguity score: 0.0

AMBIMETER says: 3D reconstruction is potentially unique

Normalized Spatial Discrepancy

Mean NSD: 0.450 Stdev. NSD: 0.014 DAMAVER included: 5 of 5

Reconstruction Resolution (SASRES)

Ensemble Resolution: 27 +/- 2 Angstrom

Clustering

Number of clusters: 2

Cluster	Isolated	Rep. Model	Deviation
1	N	glucose_i...	0.43654 ...
2	N	glucose_i...	0.34770 ...

Cluster 1	Cluster 2	Distance
1	2	0.47062 ...

Models

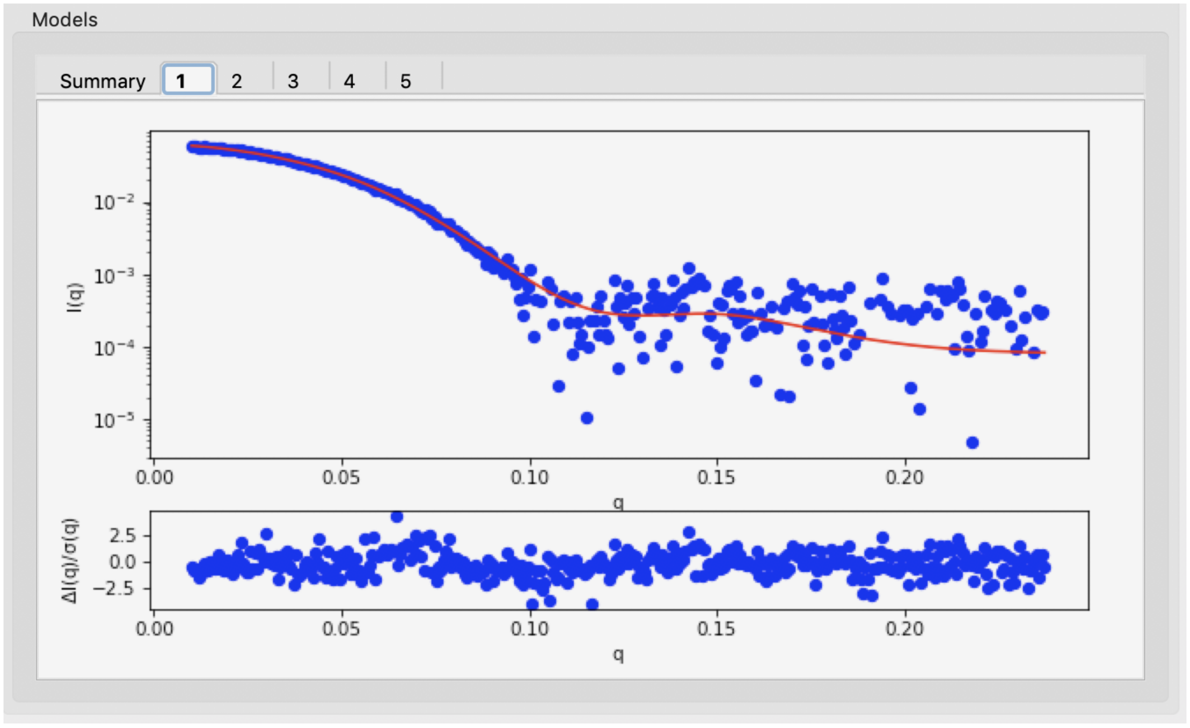
Summary 1 2 3 4 5

Model	Chi^2	Rg	Dmax	Excluded Vol.	Est. Protein MW.	Mean NSD
1	1.250	33.405	111.0	293000.0	176.51	0.459
2	1.244	33.413	113.0	291000.0	175.3	0.469
3	1.252	33.407	109.0	291000.0	175.3	0.436
4	1.242	33.417	111.0	293000.0	176.51	0.447
5	1.239	33.413	112.0	292000.0	175.9	0.440
damaver				292000.0	175.9	
damfilt				292219....	176.04	

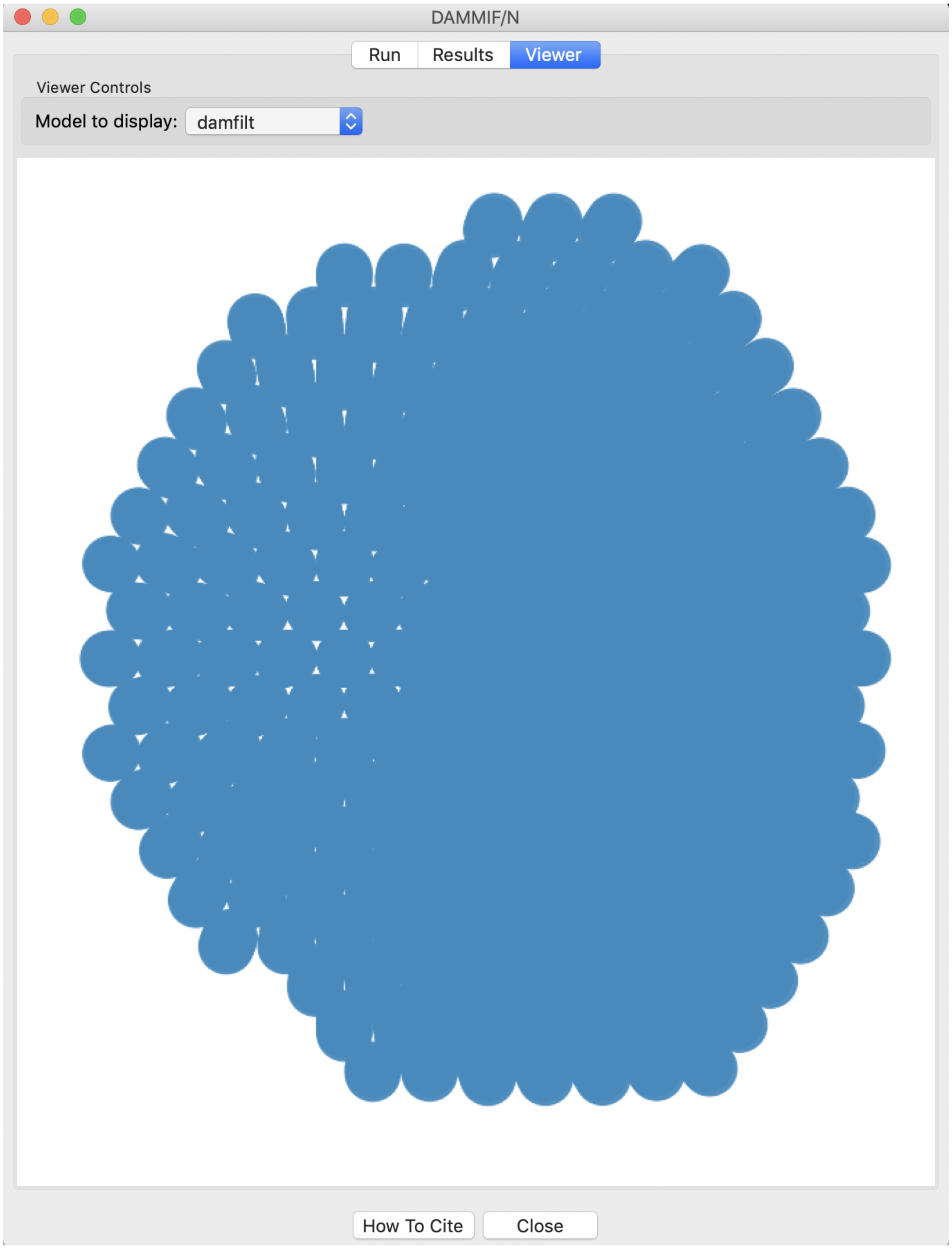
12. The results panel summarizes the results of the reconstruction run. At the top of the panel there is the AMBIMETER evaluation of how ambiguous the reconstructions might be (see previous tutorial section). If DAMAVER was run, there are results from the normalized spatial discrepancy (NSD), showing the mean and standard deviation of the NSD, as well as how many of the reconstructions were included in the average. If DAMAVER was

run on 3 or more reconstructions, and ATSAS >=2.8.0 is installed, there will be the output of SASRES which provides information on the resolution of the reconstruction. If DAMCLUST was run the number of clusters, information on each cluster, and the distance between each cluster is shown.

13. Information on each individual model is shown at the bottom. The summary tab gives the model chi squared, R_g, D_{max}, excluded volume, molecular weight estimated from the excluded volume, and, if appropriate, mean NSD of the model.
    - Any models rejected from the average by DAMAVER will be shown in red in the summary tab list.
    - *Tip:* The model highlighted in blue in the summary tab is the 'most probable' model, this can be used as your final bead model instead of doing a dammin refinement.
14. Also, each individual model has a tab which shows the data, the model fit, and the residuals.



15. The results summary shown in Summary tab is automatically saved as a **<prefix>_dammif_results.csv** csv file, e.g. for this data as **glucose_isomerase_dammif_results.csv**. All the plots shown on the individual model tabs are automatically saved as a multi-page pdf file with the same name.
16. Click on the Viewer tab to open the model viewer.
    - *Note:* The model viewer is intended for a fast first look at the results. It is not currently up to the standards of a program like pyMOL.



Model to display: damfilt

17. Click and drag the model to spin it.
    - Note: For glucose isomerase, it should look more or less like a flattened sphere.
18. Right click and drag the model to zoom in and out.

19. Use the "Model to display" menu in the Viewer Controls box to change which reconstruction is displayed.
20. Click the "Close" button when you are finished looking at the results and reconstructions.
21. The results from individual DAMMIF runs are saved in the selected output folder with the name **<prefix>_xx**, where *xx* is the run number: 01, 02, etc. For this tutorial, that would be **glucose_isomerase_01**, **glucose_isomerase_02**, and so on. The different files produced are described in the DAMMIF manual.
    - *Note:* Generally, the file of interest is the **-1.pdb** file, in this case **glucose_isomerase_01-1.pdb**, **glucose_isomerase_02-1.pdb**, etc.
22. If averaging was done with DAMAVER, the results are saved in the selected output folder with the given prefix, in this case **glucose_isomerase**. The output files generated are described in the DAMAVER manual.
    - *Note:* Generally, the files of interest are the generated pdbs: **<prefix>_damaver.pdb** and **<prefix>_damfilt.pdb**. For this tutorial, those would be **glucose_isomerase_damaver.pdb** and **glucose_isomerase_damfilt.pdb**.
23. If clustering was done with DAMCLUST, the results are saved in the selected output folder with the given prefix (for this tutorial, **glucose_isomerase**). The files generated are described in the DAMCLUST manual.
24. If refinement was done with DAMMIN, the results are saved in the selected output folder as **refine_<prefix>**, e.g. for this tutorial **refine_glucose_isomerase**. The files generated are described in the DAMMIN manual.
    - *Note:* Generally, the file of interest is the **-1.pdb** file, in this case **refine_glucose_isomerase-1.pdb**.
25. If alignment to a reference PDB was done with SUPCOMB, the files aligned depend on what other processing was done.
    - If refinement was done, then there will be a single file named **refine_<prefix>_-1_aligned.pdb**. For this tutorial, **refine_glucose_isomerase-1_aligned.pdb**.
    - If no refinement is done but averaging is done, then the damaver and damfilt results are aligned, as well as the most probable model (the blue highlighted model in the summary panel). The associated filenames would be **<prefix>_damaver_aligned.pdb**, **<prefix>_damfilt_aligned.pdb**, and **<prefix>_##_-1_aligned.pdb** where ## is the model number of the most probable model. For this tutorial, **glucose_isomerase_damaver_aligned.pdb**, **glucose_isomerase_damfilt_aligned.pdb**, and **glucose_isomerase_##-1_aligned.pdb**.
    - If no refinement is done but clustering is done, then the representative models of each cluster is aligned. The associated filenames would be **<prefix>_##-1_aligned.pdb** where ## is the model number of the representative model. For this tutorial, that is **glucose_isomerase_##-1_aligned.pdb**.
    - If no refinement, averaging, or clustering is done, then every calculated model is aligned. The associated filenames would be **<prefix>_##-1_aligned.pdb** where ## is the model number of a model. For this tutorial, that is **glucose_isomerase_##-1_aligned.pdb**.

3D reconstruction with electron density – DENSS in RAW

A new, exciting method for doing 3D shape reconstructions in SAXS yields actual electron density, rather than bead models. There are many potential advantages to this, but one significant one is easy handling of systems like RNA-Protein complexes or membrane proteins surrounded by lipids or detergents, which have more than one electron density. Bead models typically only have two (molecule and solvent) or three bead densities, and so typically fail to reconstruct these complex objects. DENSS has been fully implemented in RAW and will be used to reconstruct these electron densities.

If you use DENSS in RAW, in addition to citing the RAW paper, please cite the DENSS paper: T. D. Grant. Nature Methods (2018) 15, 191–193. DOI: 10.1038/nmeth.4581

A video version of this tutorial is available:

The written version of the tutorial follows.

1. Clear all of the data in RAW. Load the **glucose_isomerase.ift** file that you saved in the **reconstruction_data** folder in a previous part of the tutorial.

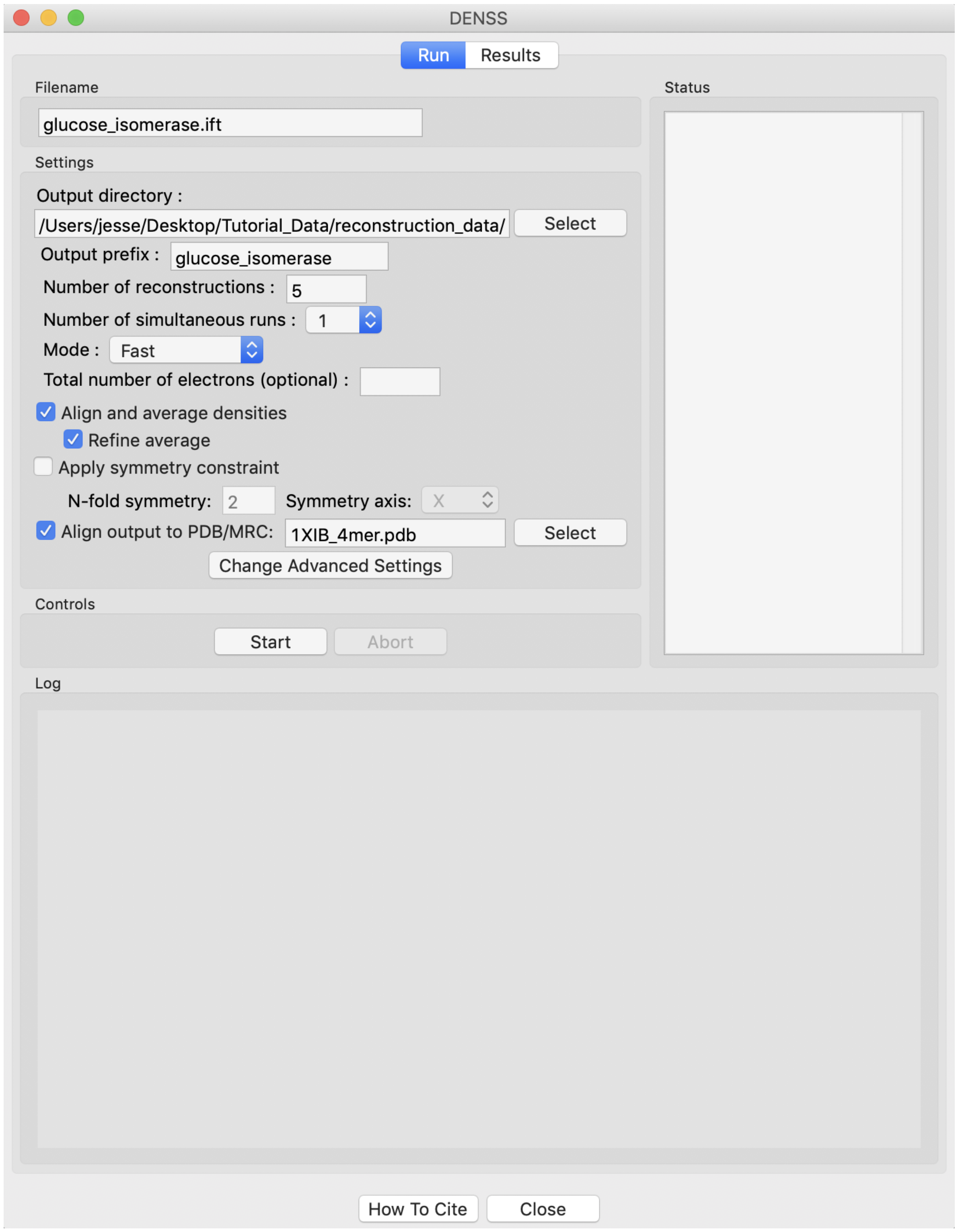
    - *Note:* If you haven't done the previous part of the tutorial, or forgot to save the results, you can find the **glucose_isomerase.ift** file in the **reconstruction_data/gi_complete** folder.
    - *Note:* You could run DENSS on the **glucose_isomerase.out** file that you used for *dammif*. However, that profile was truncated to a maximum q value of 8/Rg, ~0.23. For DENSS you want to use the full q range of the data. As the .ift file was generated using all the q range available, it is convenient to use it rather than generating another .out file.

2. Right click on the **glucose_isomerase.ift** item in the IFT list. Select the "Electron Density (DENSS)" option.
3. Running DENSS generates a lot of files. Click the "Select" button for the output directory, make a new folder in the **reconstruction_data** directory called **gi_denss** and select that folder.
4. Change the number of reconstructions to 5 and the mode to Fast.

    - *Note:* It is generally recommended that you do at least 20 reconstructions. However, for the purposes of this tutorial, 5 are enough.
    - *Note:* For final reconstructions for a paper, DENSS should be run in Slow mode. For this tutorial, or for obtaining an initial quick look at results, Fast mode is fine.

5. RAW can align the DENSS output with a PDB structure. To do so, check the Align output to PDB/MRC' box and select the **1XIB_4mer.pdb** file in the **reconstruction_data/gi_complete** folder.

    - *Tip:* If you're not sure if you selected the correct file, hovering your mouse over the filename will show the full path to the file.



Filename: glucose_isomerase.ift

Output directory : /Users/jesse/Desktop/Tutorial_Data/reconstruction_data/

Output prefix : glucose_isomerase

Number of reconstructions : 5

Number of simultaneous runs : 1

Mode : Fast

Total number of electrons (optional) :

[✓] Align and average densities

[✓] Refine average

[ ] Apply symmetry constraint

N-fold symmetry: 2 Symmetry axis: X

[✓] Align output to PDB/MRC: 1XIB_4mer.pdb

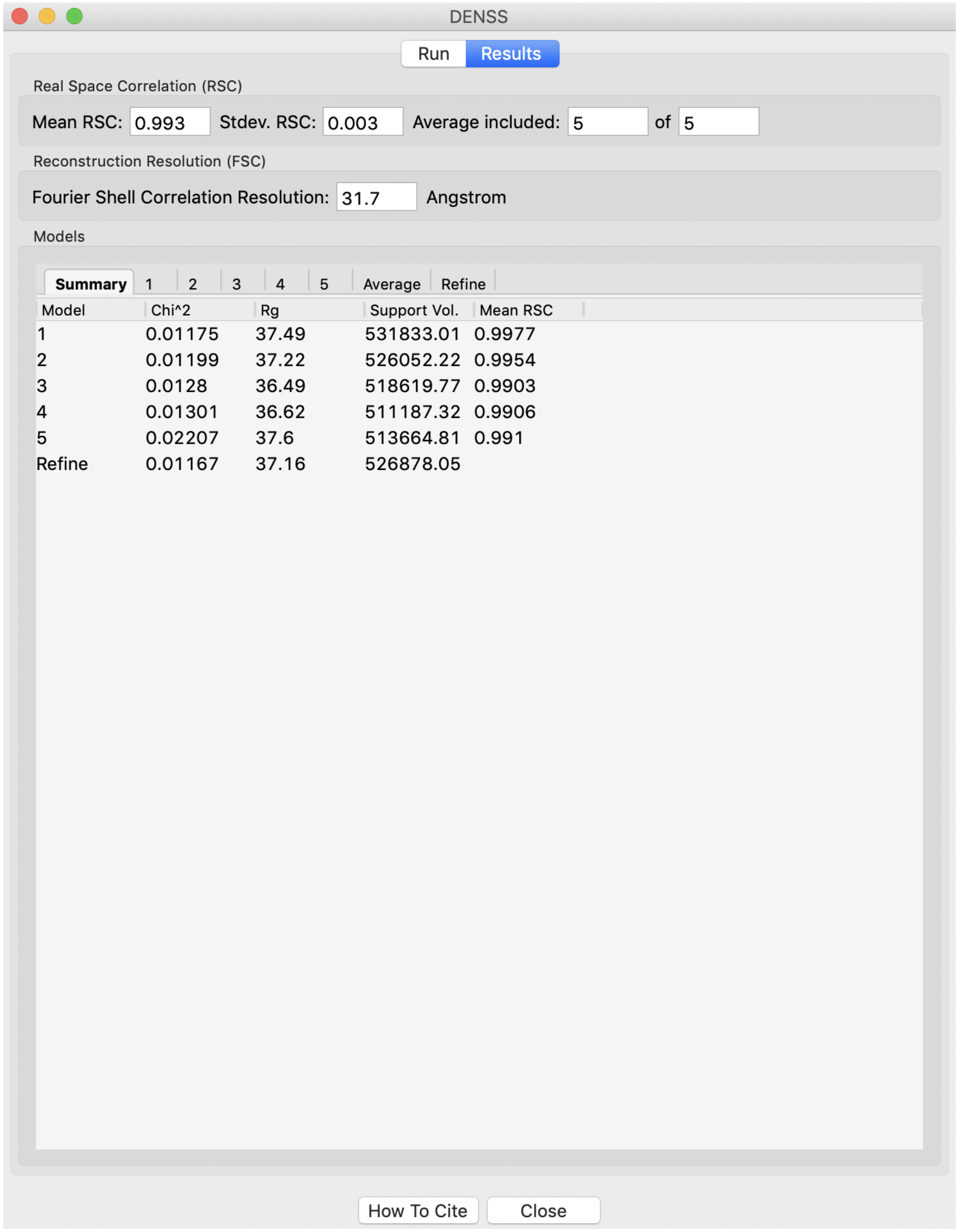
Change Advanced Settings

6. Click the "Start" button.

   - *Note:* The status panel will show you the overall status of the reconstructions. You can look at the detailed status of each run by clicking the appropriate tab in the log panel.

7. Note that by default the densities are aligned and averaged, including enantiomer filtering, and a refined density is created from the average.

8. Wait for all of the DENSS runs and averaging to finish. Depending on the speed of your computer this could take a bit.
9. Once the reconstructions are finished, the window should automatically switch to the results tab. If it doesn't, click on the results tab.



Real Space Correlation (RSC)

Mean RSC: 0.993 Stdev. RSC: 0.003 Average included: 5 of 5

Reconstruction Resolution (FSC)

Fourier Shell Correlation Resolution: 31.7 Angstrom

Models

Model	Chi^2	Rg	Support Vol.	Mean RSC
1	0.01175	37.49	531833.01	0.9977
2	0.01199	37.22	526052.22	0.9954
3	0.0128	36.49	518619.77	0.9903
4	0.01301	36.62	511187.32	0.9906
5	0.02207	37.6	513664.81	0.991
Refine	0.01167	37.16	526878.05	

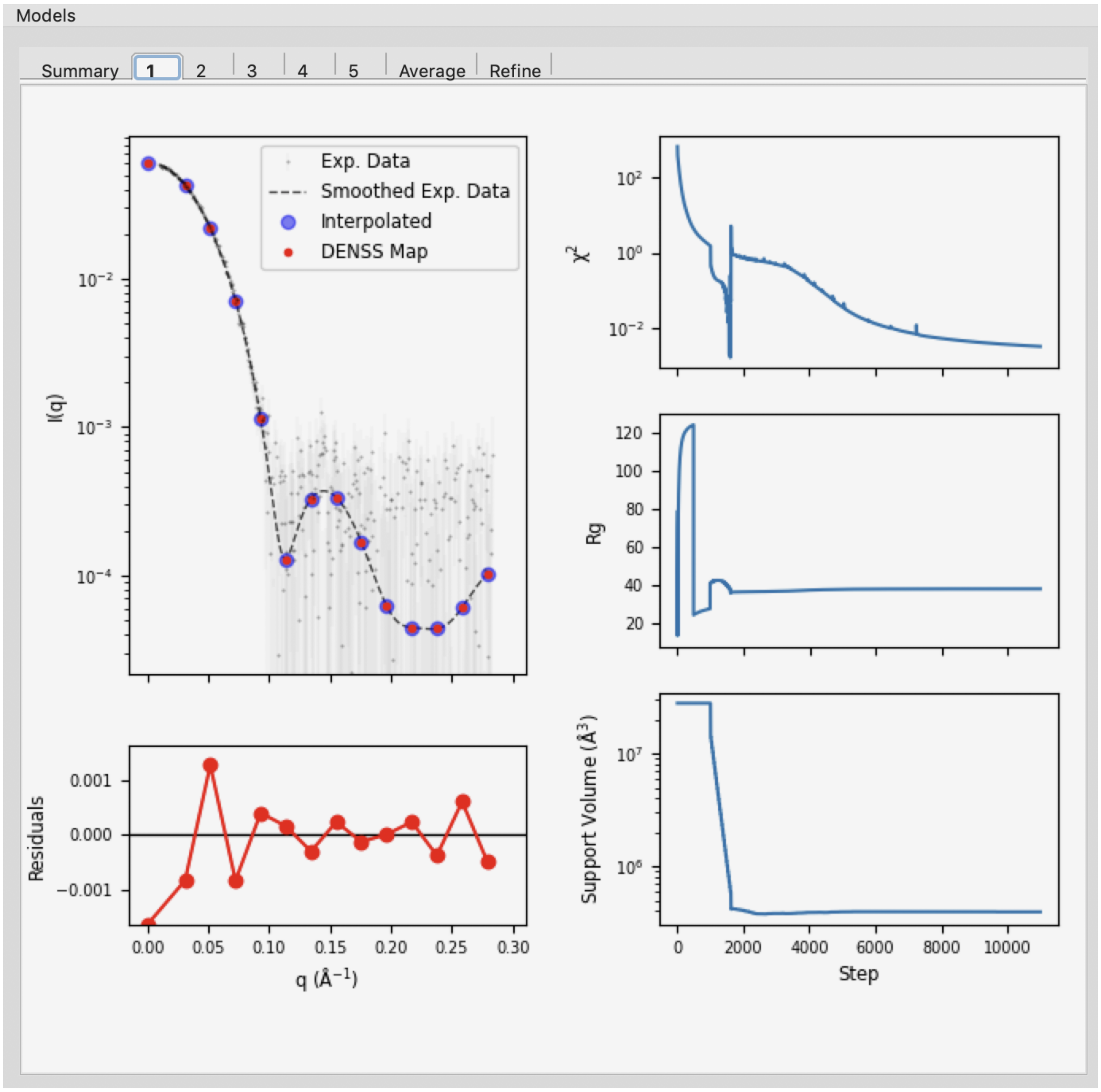
10. The results panel summarizes the results of the reconstruction runs. If you are using a .out file, then at the top of

the panel there is the ambimeter evaluation of how ambiguous the reconstructions might be (see earlier tutorial section). If averaging was run there is an estimate of the reconstruction resolution based on the Fourier shell correlation. In the models section there are several tabs. The summary tab shows the chi^2, R_g, support volume, and RSC to the reference model. If any model was not included in the averaging it is highlighted in red.

- Verify that the Rg is close to the expected value, and that the chi^2 and support volumes are relatively consistent between models.
- *Note:* Above we show the results for 20 runs instead of the 5 in the tutorial.

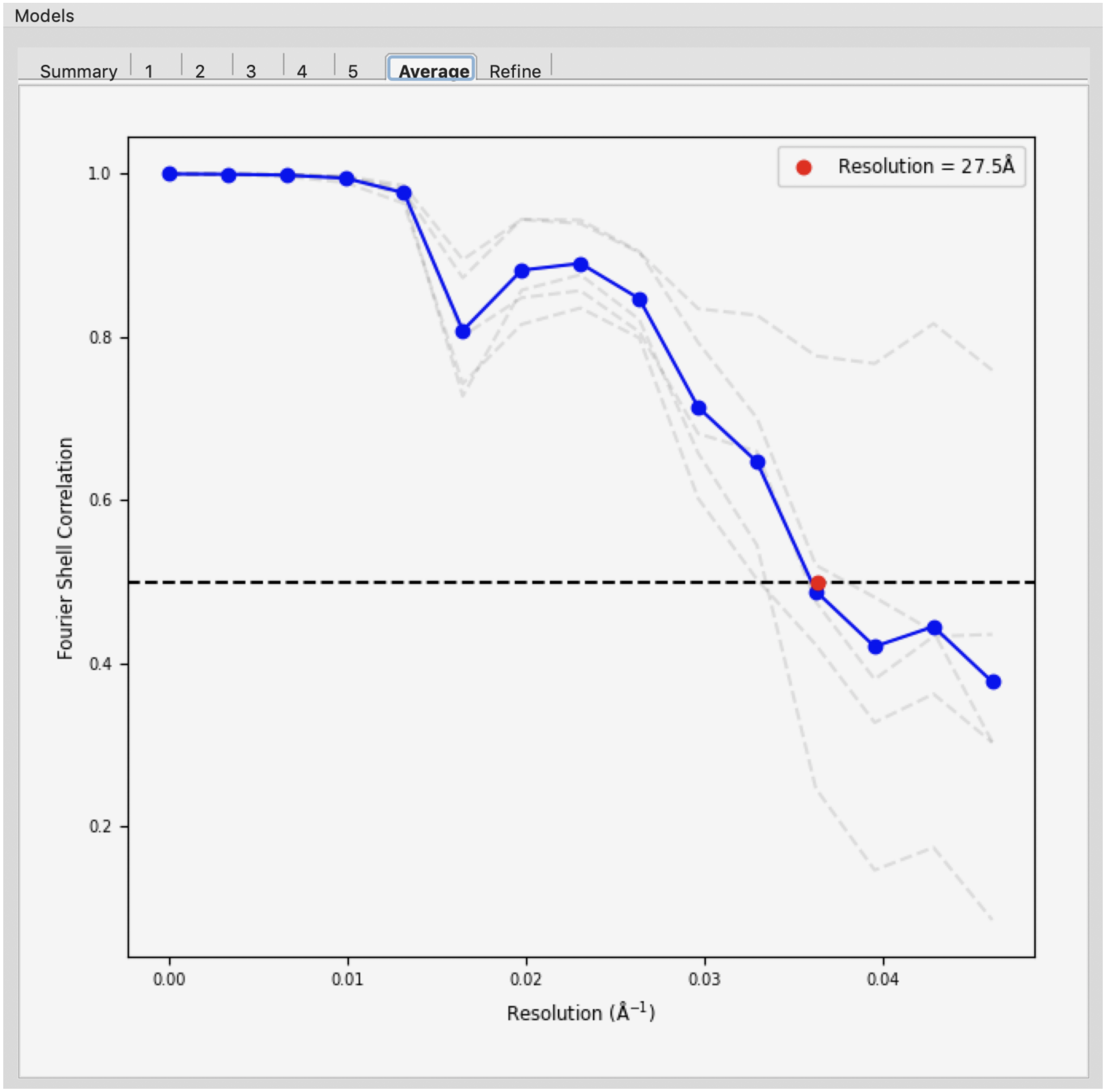
11. Individual model results are displayed in the numbered tabs. For each individual model there are plots of: the original data and the model data (scattering from density); the residual between the original data and the model data; and chi squared, R_g and support volume vs. refinement step.

    - Verify that the residual between the actual data and the model data is small.
    - Check that the chi squared, R_g, and support volume have all plateaued (converged) by the final steps.



12. If the densities were averaged, the average tab will display the Fourier shell correlation vs. resolution.

- *Note:* The reconstruction resolution is taken as the resolution in angstroms where the correlation first crosses 0.5.



13. The results summary shown in Summary tab is automatically saved as a **<prefix>_denss_results.csv** csv file, e.g. for this data as **glucose_isomerase_denss_results.csv**. All the plots shown on the individual model tabs are automatically saved as a multi-page pdf file with the same name.
14. Click the "Close" button when you are finished looking at the results and reconstructions.
15. The results from the individual DENSS runs are saved in the selected output folder as **<prefix>_xx.mrc** where *xx* corresponds to the run number: 01, 02, etc. For this tutorial that would be **glucose_isomerase_01.mrc**, **glucose_isomerase_02.mrc**, etc.
16. If averaging was done, final average density is saved in the selected output folder as **<prefix>_aver.mrc**. For this tutorial, that would be **glucose_isomerase_aver.mrc**.
17. If refinement was done, the final refined density is saved in the selected output folder as **<prefix>_refine.mrc**. For this tutorial that would be **glucose_isomerase_refine.mrc**.
18. If alignment to a reference model was done, the files aligned depend on what other processing was done.

- If refinement was done, then there will be a single file named **<prefix>_refine_aligned.mrc**. For this tutorial, **glucose_isomerase_refine_aligned.mrc**.
- If no refinement is done but averaging is done, then the averaged model is aligned. The associated filenames would be **<prefix>_average_aligned.mrc**. For this tutorial, **glucose_isomerase_averaged_aligned.mrc**.
- If no refinement or averaging is done, then every calculated model is aligned. The associated filenames would be **<prefix>_##_aligned.mrc** where ## is the model number of a model. For this tutorial, that is **glucose_isomerase_##_aligned.mrc**.

Note: **.mrc** files can be opened in Chimera and pyMOL. For tips about how to visualize the density and align it with known structures see the appropriate sections here: http://www.tdgrant.com/denss/tips/.

Aligning reconstructions with high resolution shapes

It is often important to align SAXS reconstructions, either bead models such as those from DAMMIF/N or electron density from DENSS, with high resolution structural models such as those from x-ray crystallography to see how well they agree. RAW can do that automatically when you generate the models, as described in the *DAMMIF/N* and *DENSS* sections of the tutorial. RAW also provides standalone windows for doing the alignment with already generated reconstructions.

If you use DENSS alignment in RAW, in addition to citing the RAW paper, please cite the DENSS paper: T. D. Grant. Nature Methods (2018) 15, 191–193. DOI: 10.1038/nmeth.4581

If you use RAW to run SUPCOMB, in addition to citing the RAW paper, please cite the paper given in the SUPCOMB manual.

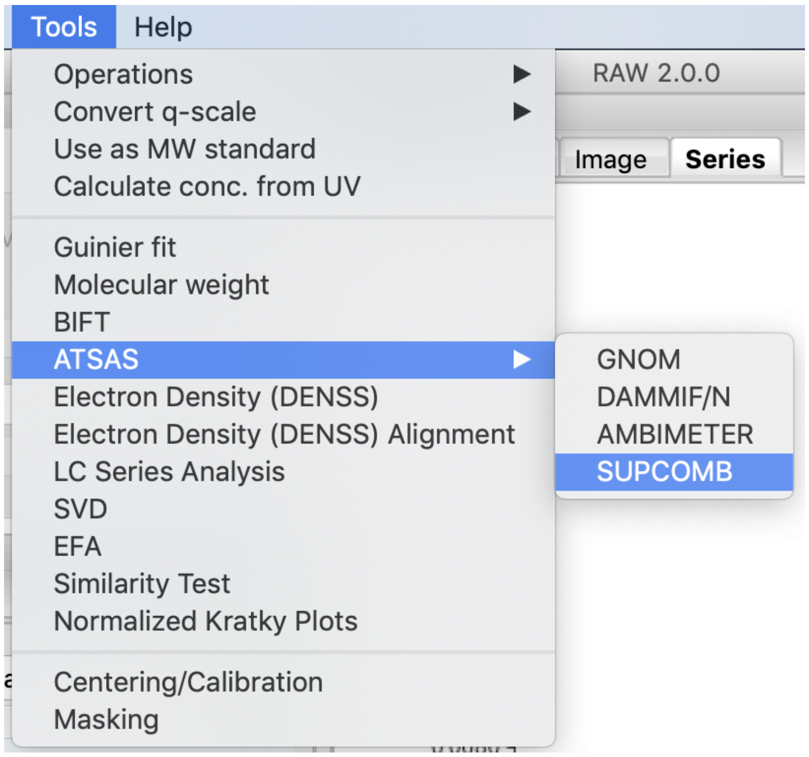
A video version of this tutorial is available:

The written version of the tutorial follows.

Bead models - SUPCOMB

SUPCOMB from the ATSAS suite can be used to align two PDB files. In this context, one model (the reference) is the high resolution structure while the other (the target) is the bead model reconstruction.

1. Open the SUPCOMB window by selecting Tools->ATSAS->SUPCOMB from the menu bar

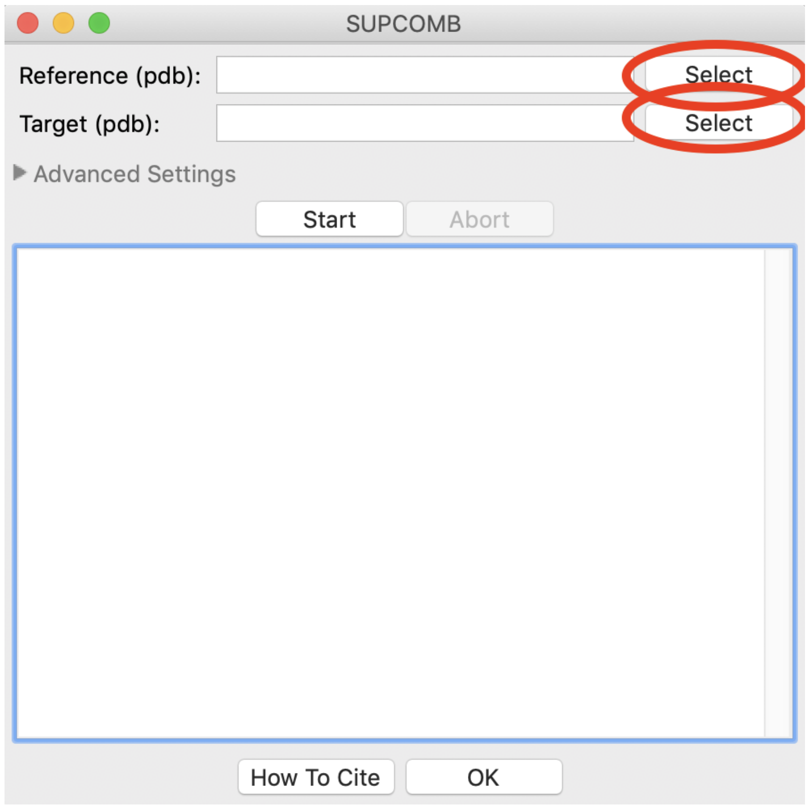


2. In the window that opens, 'Target' is the model that is aligned, where as 'Reference' is the model that the target is aligned to. In other words, the Reference model stays unchanged, while the Target model is moved to best align with the Reference.

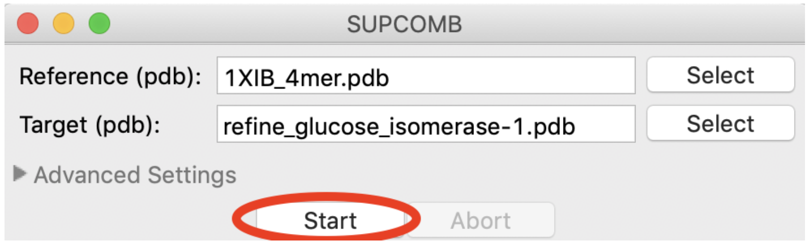
3. Use the Reference 'Select' button to select **reconstruction_data/gi_complete/1XIB_4mer.pdb** as the reference file.

   - *Tip:* Only the filename will show up in either the Reference or Target box. If you hover your mouse over the filename it will show the full path to the file.

4. Use the Target 'Select' button to select **reconstruction_data/gi_complete/gi_dammif/refine_glucose_isomerase-1.pdb**



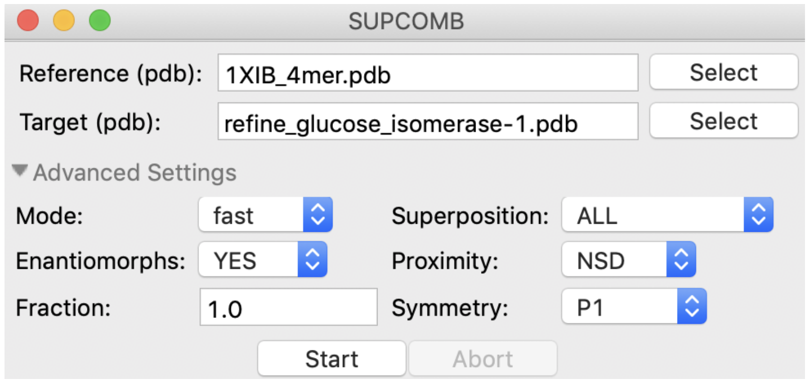
5. Click the start button. SUPCOMB will run.



Reference (pdb): 1XIB_4mer.pdb
Target (pdb): refine_glucose_isomerase-1.pdb

6. When SUPCOMB is finished, in the same folder as the target file you will see a **<target_name>_aligned.pdb** file, which is the target model aligned with the reference file.

7. Advanced settings can be accessed by clicking on the 'Advanced Settings' text to expand the section. These settings are described in the SUPCOMB manual.

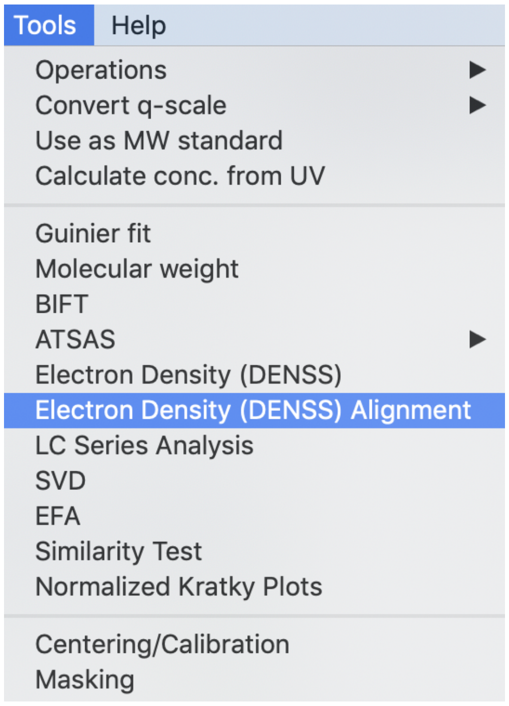


Reference (pdb): 1XIB_4mer.pdb
Target (pdb): refine_glucose_isomerase-1.pdb
Mode: fast
Superposition: ALL
Enantiomorphs: YES
Proximity: NSD
Fraction: 1.0
Symmetry: P1

Electron density

DENSS include a native tool for aligning two electron density files (.mrc) or an electron density to a PDB file. In this context, one model (the reference) is the high resolution

1. Open the Electron Density Alignment window by selecting Tools->Electron Density (DENSS) Alignment from the menu bar

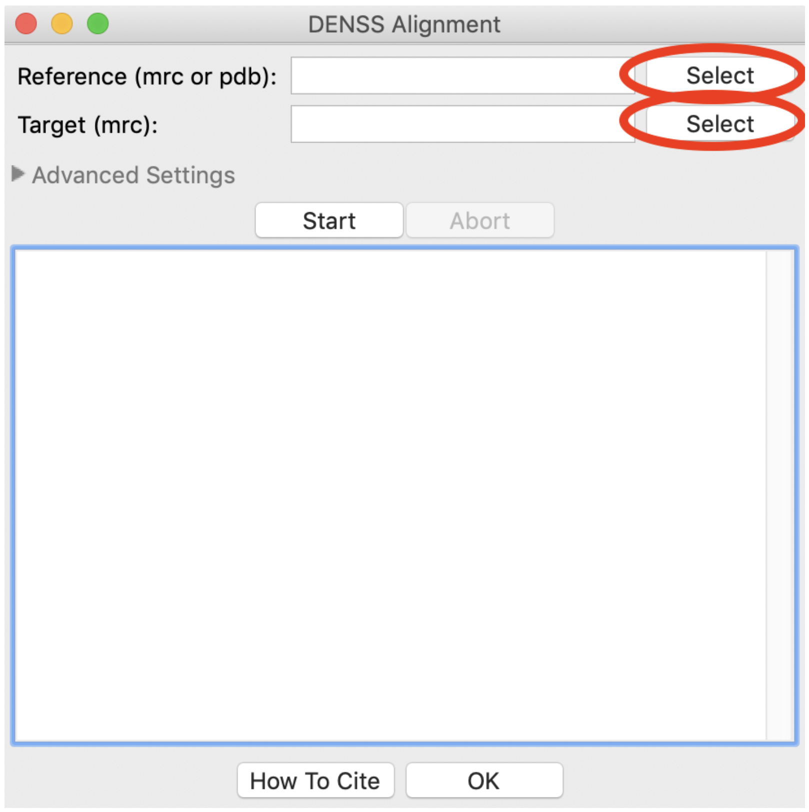


2. In the window that opens, 'Target' is the model that is aligned, where as 'Reference' is the model that the target is aligned to. In other words, the Reference model stays unchanged, while the Target model is moved to best align with the Reference.

3. Use the Reference 'Select' button to select **reconstruction_data/gi_complete/1XIB_4mer.pdb** as the reference file.

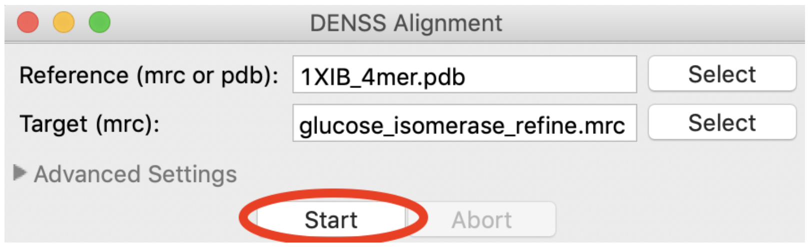
   - *Tip:* Only the filename will show up in either the Reference or Target box. If you hover your mouse over the filename it will show the full path to the file.

4. Use the Target 'Select' button to select **reconstruction_data/gi_complete/gi_denss/glucose_isomerase_refine.mrc**



5. Click the start button. DENSS alignment will run.

   - *Tip:* If there is already a file in the folder with the aligned output filename you will get a prompt to overwrite it.
   - *Tip:* By default, DENSS centers the Reference file. This writes out a file named **<reference_name>_centered.pdb** in the same folder as the reference file, which is what should be compared to the aligned file. You can turn this off in the Advanced Settings.



Reference (mrc or pdb): 1XIB_4mer.pdb

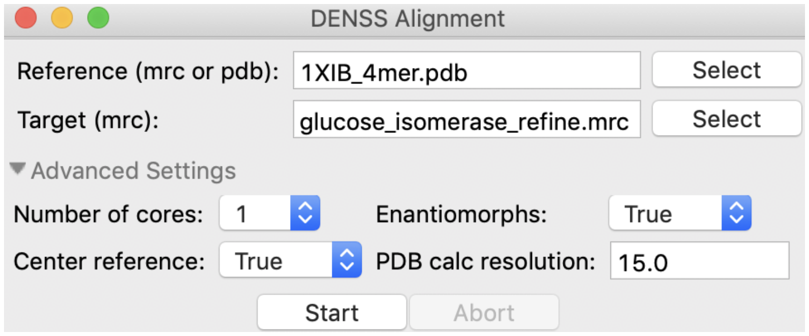
Target (mrc): glucose_isomerase_refine.mrc

6. When alignment is finished, in the same folder as the target file you will see a **<target_name>_aligned.mrc**. Compare this to the **<reference_name>_centered.pdb** file in reference file folder. In this case those names are **glucose_isomerase_refine_aligned.mrc** and **1XIB_4mer_centered.pdb**.

7. You can change the advanced settings by expanding the Advanced Settings section. These advanced settings are:

   - *Number of cores:* Number of cores to use during alignment
   - *Enantiomorphs:* Whether to generate enantiomorphs of the Target before doing the alignment.
   - *Center reference:* Whether to center the reference model at the origin. If used, this creates a **<reference_name>_centered.pdb** file in the same folder as the reference file.

- *PDB calc. resolution:* The resolution of the density map created from the Reference PDB model to compare with the Target model. This has no effect if the Reference is already a density.



Reference (mrc or pdb): 1XIB_4mer.pdb

Target (mrc): glucose_isomerase_refine.mrc

Number of cores: 1

Enantiomorphs: True

Center reference: True

PDB calc resolution: 15.0

Advanced SEC-SAXS processing – Singular value decomposition (SVD)

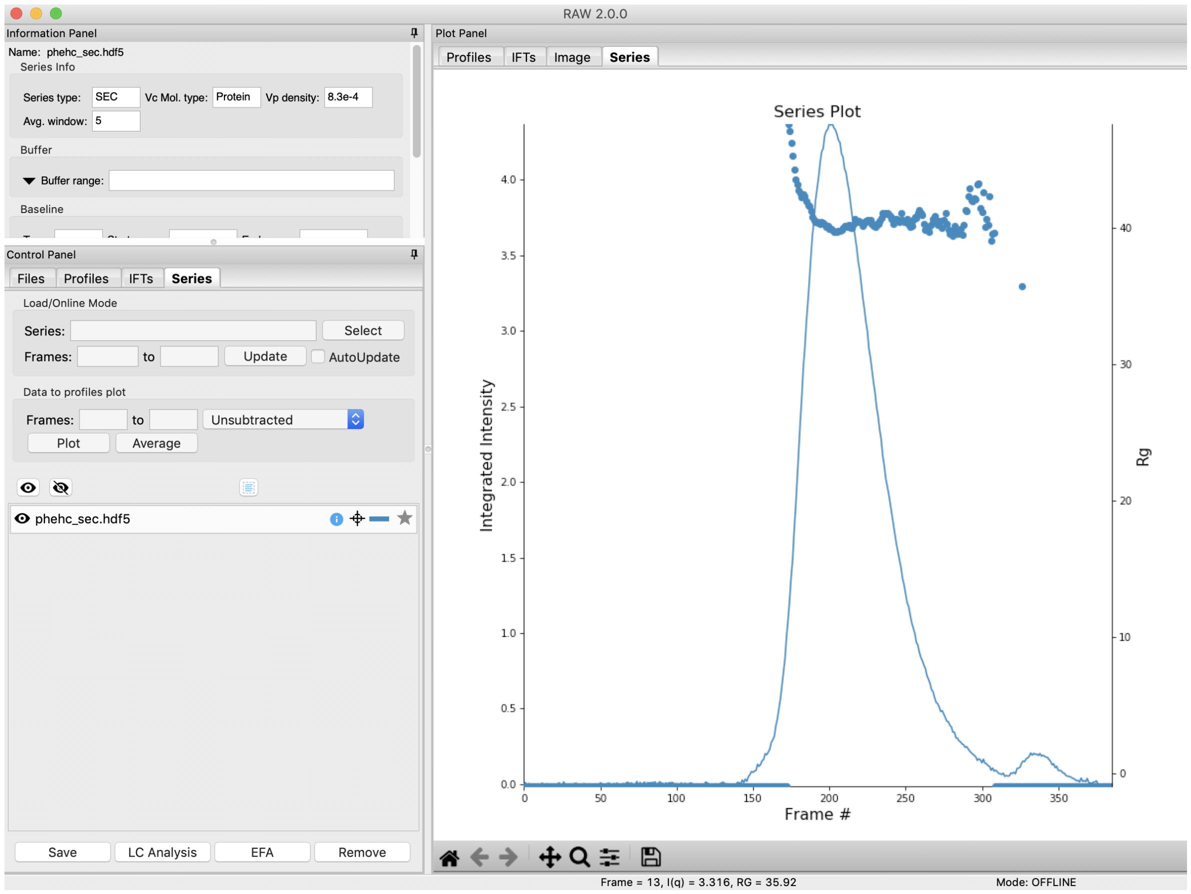
Sometimes SEC fails to fully separate out different species, and you end up with overlapping peaks in your SEC-SAXS curve. It is possible to apply more advanced mathematical techniques to determine if there are multiple species of macromolecule in a SEC-SAXS peak, and to attempt to extract out scattering profiles for each component in an overlapping peak. Singular value decomposition (SVD) can be used to help determine how many distinct scatterers are in a SEC-SAXS peak. *Evolving factor analysis (EFA)* is an extension of SVD that can extract individual components from overlapping SEC-SAXS peaks. Note that the first step of EFA is doing SVD, but that happens entirely within the EFA analysis window. The SVD window does not need to be opened before doing EFA. This tutorial covers SVD.

A video version of this tutorial is available:

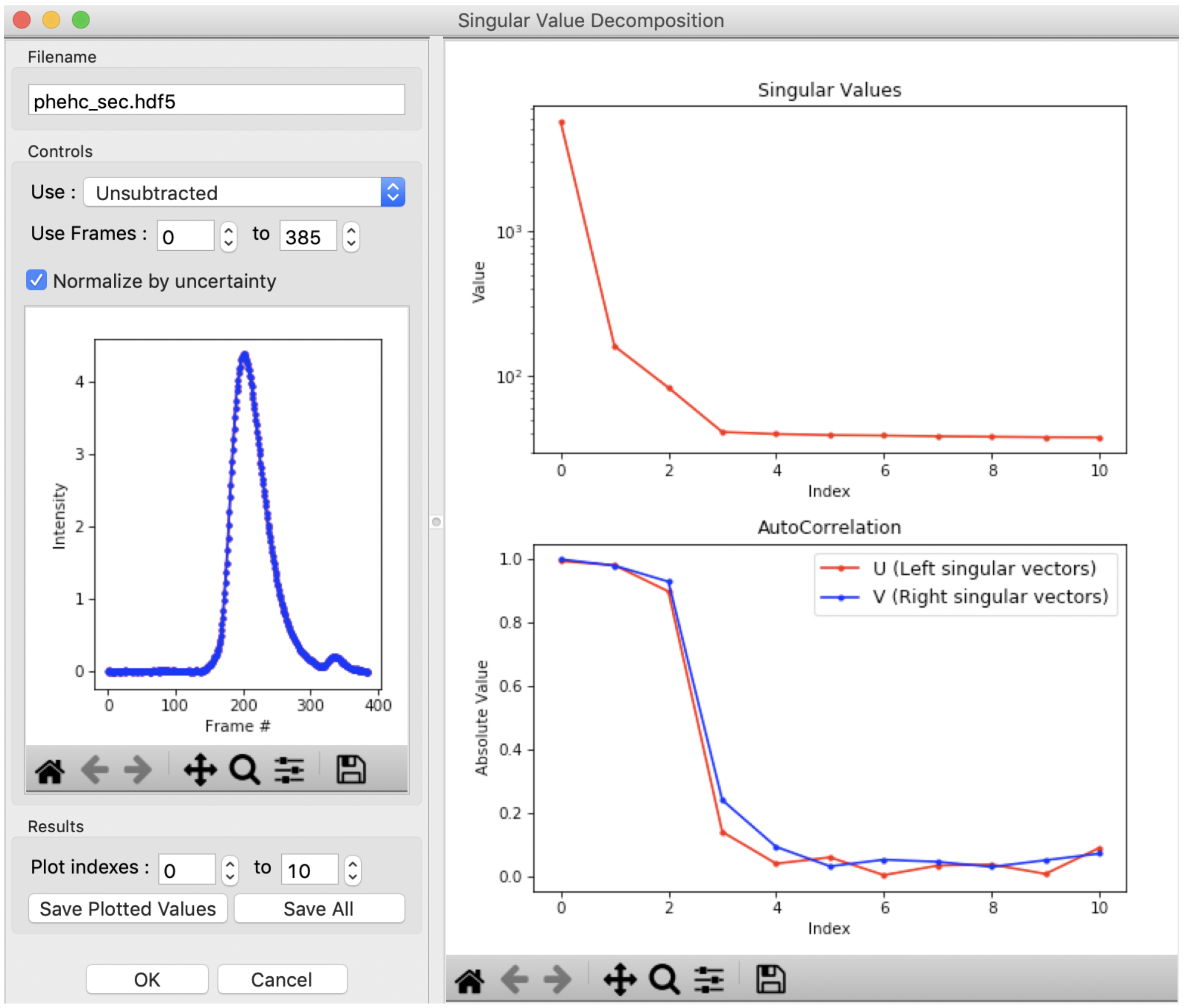
The written version of the tutorial follows.

1. Clear all of the data in RAW. Load the **phehc_sec.hdf5** file in the **series_data** folder.

    - *Note:* The data were provided by the Ando group at Cornell University and is some of the data used in the paper: *Domain Movements upon Activation of Phenylalanine Hydroxylase Characterized by Crystallography and Chromatography-Coupled Small-Angle X-ray Scattering*. Steve P. Meisburger, Alexander B. Taylor, Crystal A. Khan, Shengnan Zhang, Paul F. Fitzpatrick, and Nozomi Ando. Journal of the American Chemical Society 2016 138 (20), 6506-6516. DOI: 10.1021/jacs.6b01563



2. Right click on the **phehc_sec.hdf5** item in the Series list. Select the “SVD” option.
3. The SVD window will be displayed. On the left are controls, on the right are plots of the value of the singular values and the first autocorrelation of the left and right singular vectors.
   - *Note:* Large singular values indicate significant components. What matters is the relative magnitude, that is, whether the value is large relative to the mostly flat/unchanging value of high index singular values.
   - *Note:* A large autocorrelation indicates that the singular vector is varying smoothly, while a low autocorrelation indicates the vector is very noisy. Vectors corresponding to significant components will tend to have autocorrelations near 1 (roughly, >0.6-0.7) and vectors corresponding to insignificant components will tend to have autocorrelations near 0.


Singular Value Decomposition

Filename: phehc_sec.hdf5

Controls

Use : Unsubtracted

Use Frames : 0 to 385

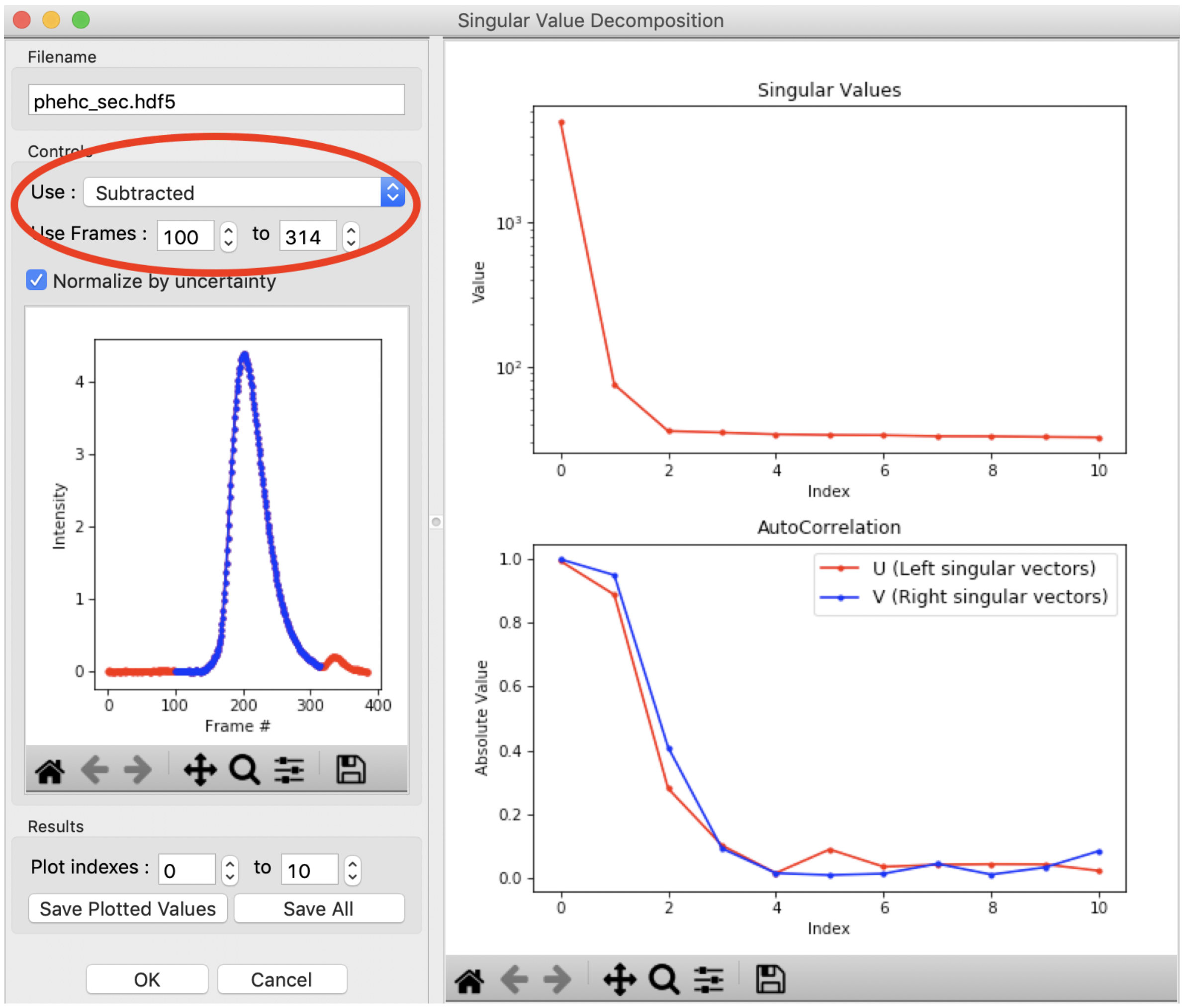
[✓] Normalize by uncertainty

Results

Plot indexes : 0 to 10

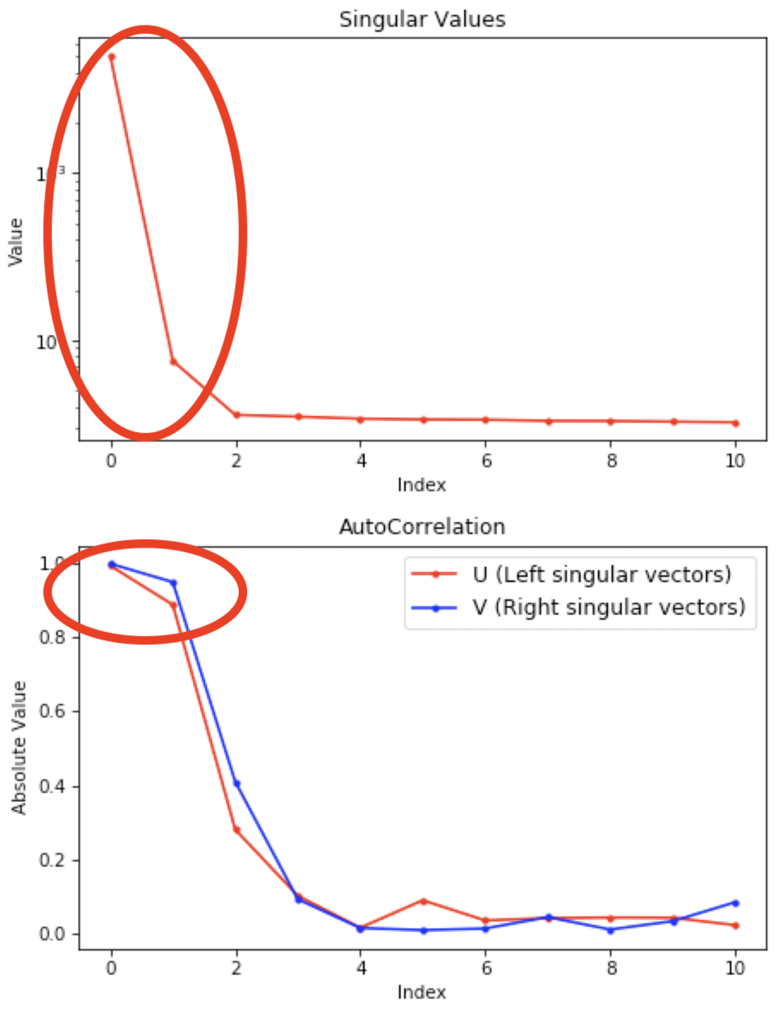
4. Adjust the starting frame number to 100, the ending frame number to near 300, and switch to using Subtracted data.

   - *Note:* The blue points are in the plot on the left are the region being used for SVD, while the red points show the rest of the SEC-SAXS curve.



5. We have now isolated the peak. Looking at the top plot, we see there are two singular values significantly above the baseline level, and from the autocorrelation we see two values with both left and right singular vectors autocorrelations near 1. This indicates that there are two scattering components in the peak, even though there are no obvious shoulders in the region we selected

   - *Try:* Adjust the starting and ending values and seeing how that changes the SVD results. Is there a region of the peak you can isolate that has just one significant component?
   - *Note:* Normally, changing between Unsubtracted and Subtracted SEC-SAXS profiles should remove one significant singular value component, corresponding to the buffer scattering. In this data, you will see no difference, as the profiles used to produce the SEC-SAXS curve were already background subtracted.
   - *Note:* You can save the SVD plots by clicking the Save button, as with the plots in the main RAW window. You can save the SVD results, either just the plotted values or all of the values, using the two Save buttons in the SVD panel.



6. Close the SVD window by clicking the OK button.

Advanced SEC-SAXS processing – Evolving factor analysis (EFA)

Sometimes SEC fails to fully separate out different species, and you end up with overlapping peaks in your SEC-SAXS curve. It is possible to apply more advanced mathematical techniques to determine if there are multiple species of macromolecule in a SEC-SAXS peak, and to attempt to extract out scattering profiles for each component in an overlapping peak. *Singular value decomposition (SVD)* can be used to help determine how many distinct scatterers are in a SEC-SAXS peak. Evolving factor analysis (EFA) is an extension of SVD that can extract individual components from overlapping SEC-SAXS peaks. Note that the first step of EFA is doing SVD, but that happens entirely within the EFA analysis window. The SVD window does not need to be opened before doing EFA. This tutorial covers EFA.

REGALS is a similar deconvolution technique, but can be applied in cases where there are components that are not strictly first-in-first-out and EFA would fail. EFA is recommended for standard SEC-SAXS data, but for more complex data, such as ion exchange chromatography, or time resolved or titration data you should use REGALS. REGALS can also handle deconvolution of SEC-SAXS data with a sloping baseline.

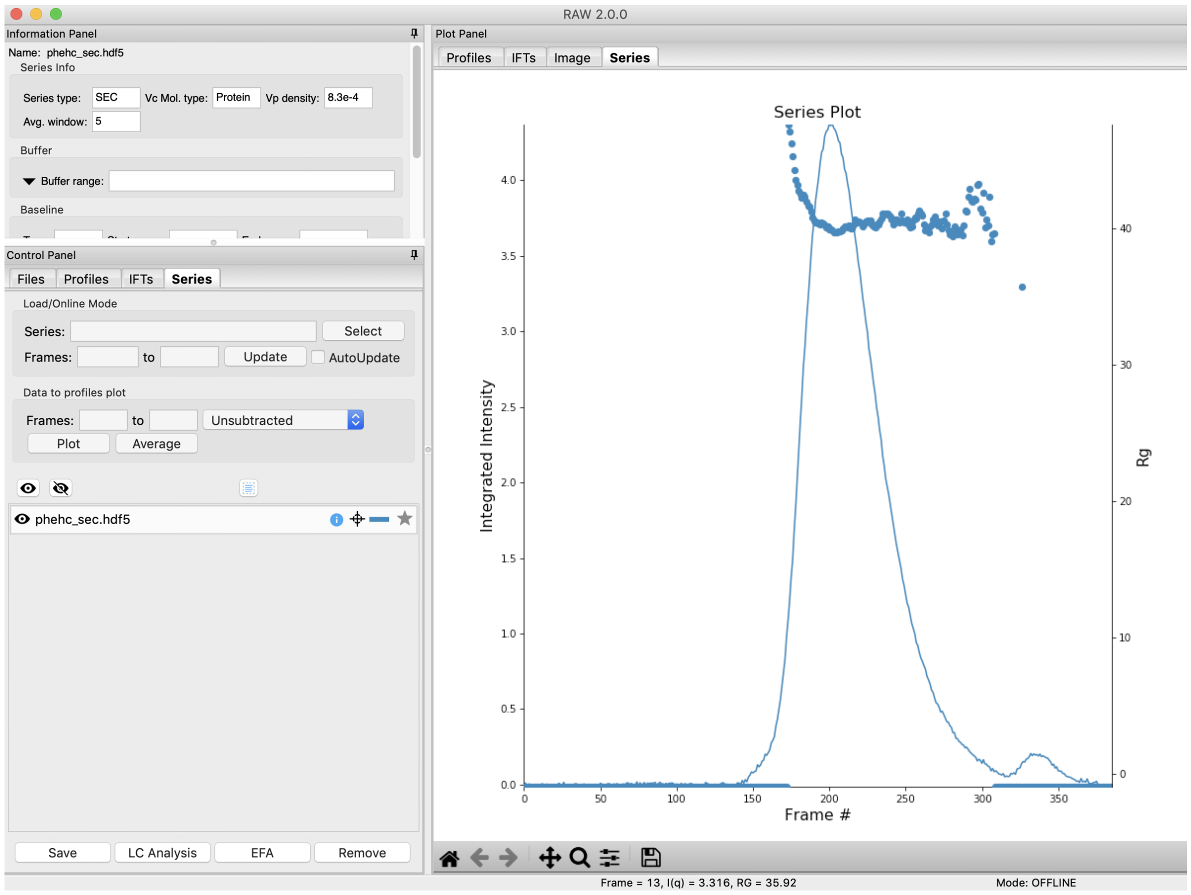
If you use EFA in RAW, in addition to citing the RAW paper, please cite the EFA paper: S. P. Meisburger, A. B. Taylor, C. A. Khan, S. Zhang, P. F. Fitzpatrick, N. Ando. Journal of the American Chemical Society (2016). 138(20), 6506-6516. DOI: 10.1021/jacs.6b01563

A video version of this tutorial is available:

The written version of the tutorial follows.

1. Clear all of the data in RAW. Load the **phehc_sec.hdf5** file in the **series_data** folder.

    - *Note:* The data were provided by the Ando group at Cornell University and is some of the data used in the paper: *Domain Movements upon Activation of Phenylalanine Hydroxylase Characterized by Crystallography and Chromatography-Coupled Small-Angle X-ray Scattering*. Steve P. Meisburger, Alexander B. Taylor, Crystal A. Khan, Shengnan Zhang, Paul F. Fitzpatrick, and Nozomi Ando. Journal of the American Chemical Society 2016 138 (20), 6506-6516. DOI: 10.1021/jacs.6b01563



RAW 2.0.0

Information Panel

Name: phehc_sec.hdf5

Series Info

Series type: SEC   Vc Mol. type: Protein   Vp density: 8.3e-4

Avg. window: 5

Buffer

Buffer range:

Baseline

Control Panel

Load/Online Mode

Series:

Frames: to   AutoUpdate

Data to profiles plot

Frames: to   Unsubtracted

phehc_sec.hdf5

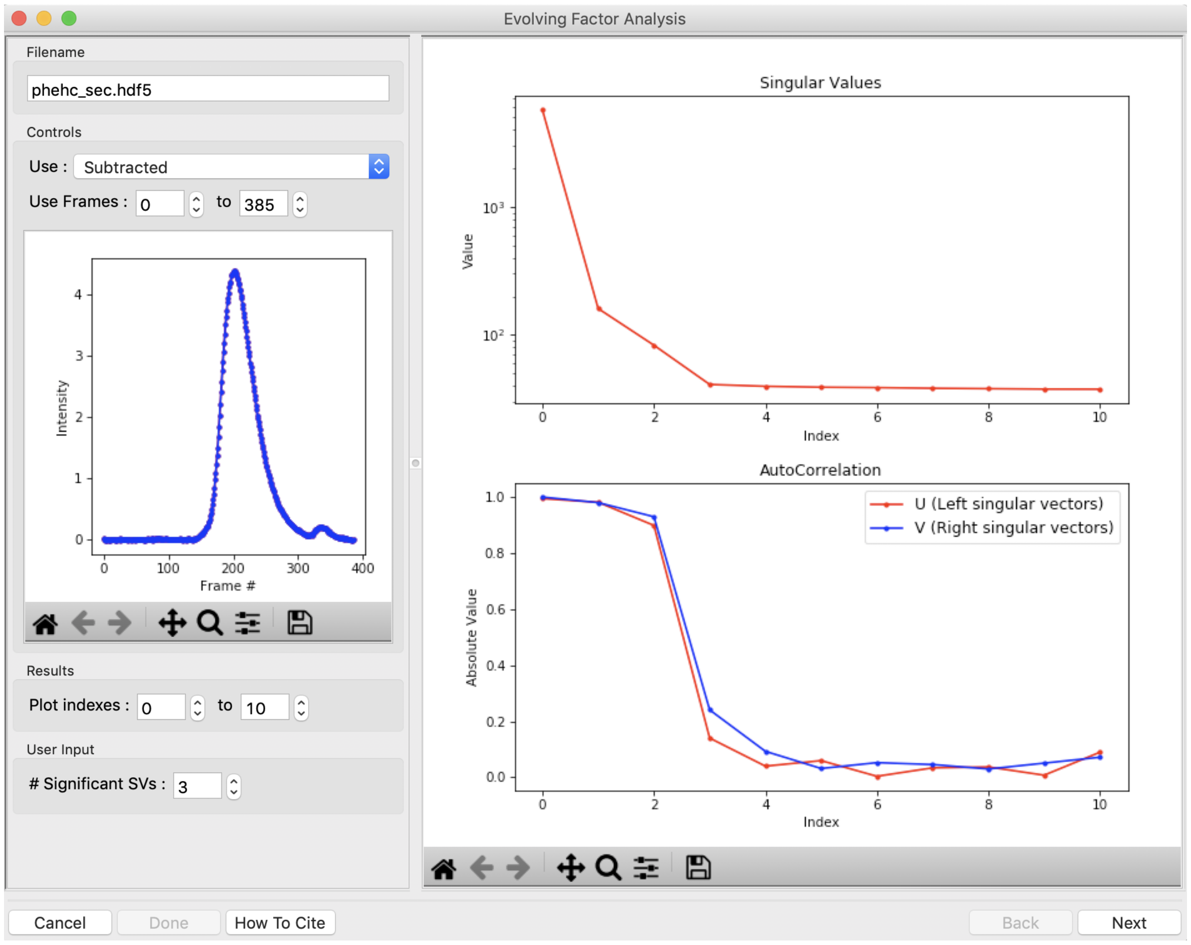
Frame = 13, I(q) = 3.316, RG = 35.92   Mode: OFFLINE

2. We will use EFA to extract out the two scattering components in the main peak in this data. Right click on the **phehc_sec.hdf5** item in the Series list. Select the "EFA" option.

3. The EFA window will be displayed. On the left are controls, on the right are plots of the value of the singular values and the first autocorrelation of the left and right singular vectors.

    - *Note:* Large singular values indicate significant components. What matters is the relative magnitude, that is, whether the value is large relative to the mostly flat/unchanging value of high index singular values.

    - *Note:* A large autocorrelation indicates that the singular vector is varying smoothly, while a low autocorrelation indicates the vector is very noisy. Vectors corresponding to significant components will tend to have autocorrelations near 1 (roughly, >0.6-0.7) and vectors corresponding to insignificant components will tend to have autocorrelations near 0.



Filename: phehc_sec.hdf5

Use : Subtracted

Use Frames : 0 to 385

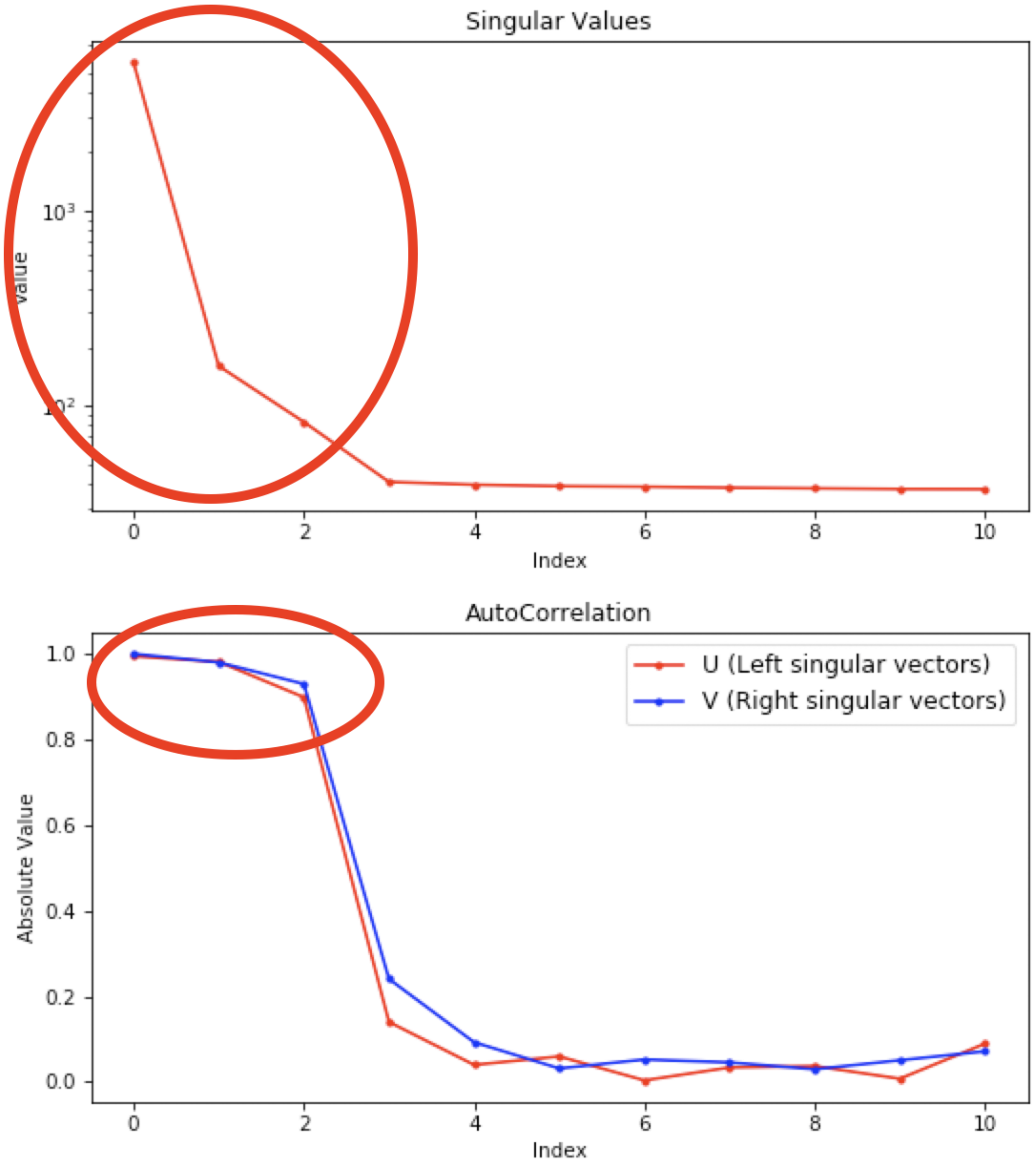
Plot indexes : 0 to 10

Significant SVs : 3

4. For successful EFA, you want to use Subtracted data, and you often want to have a buffer region before and after the sample. For this data set, using the entire frame range (from 0 to 385) is appropriate. With other data sets, you may need to change the frame range to, for example, remove other, well separated, peaks from the analysis.

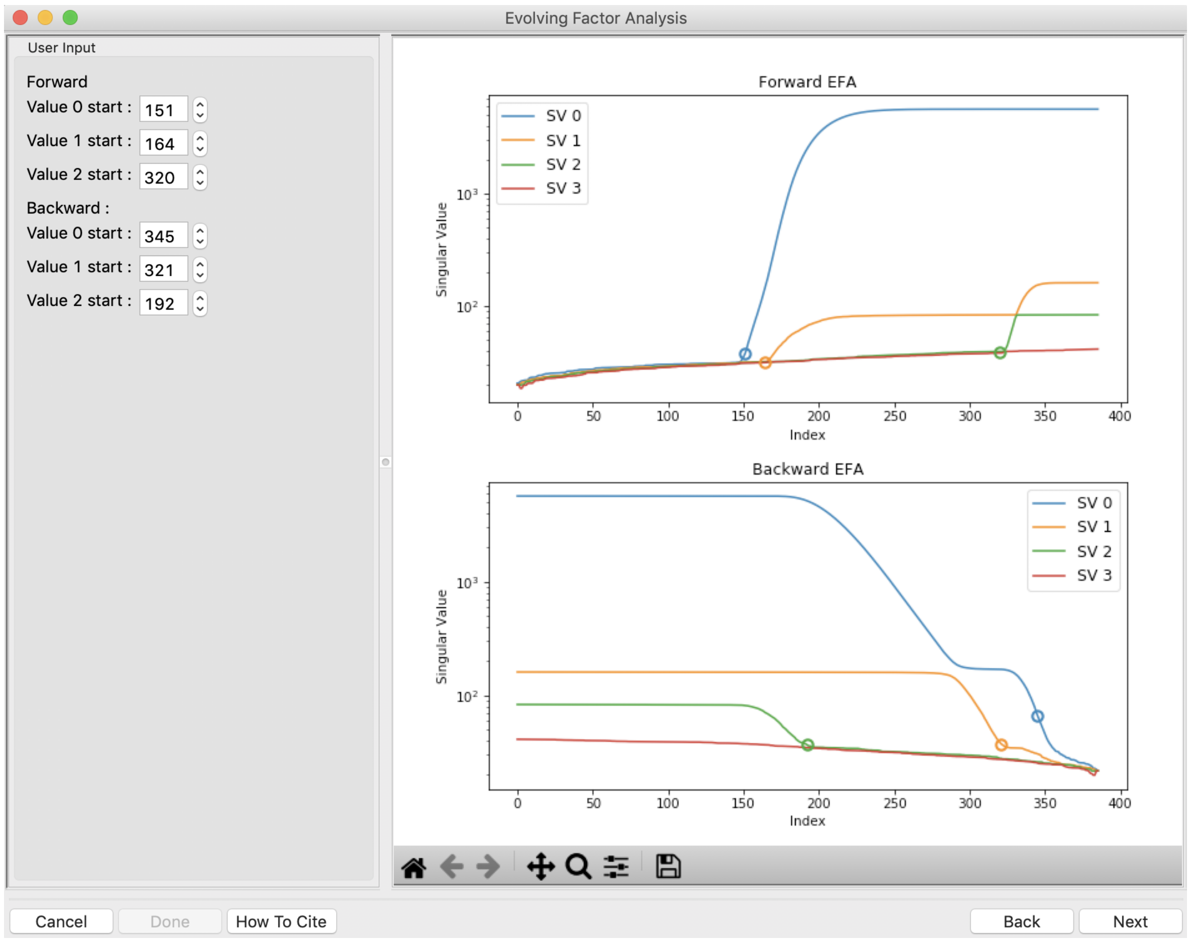
   - *Tip:* If you have a dataset where you have a large number of components, such as 4+, it can be useful to set the EFA range to isolate just 2-3 of those components. The more components you have, the harder it is to do the EFA. There is a trade off in the amount of data used (more is better), and the number of components in the deconvolution (less is better) that requires some experimentation to find the right balance for a given dataset.

5. RAW attempts to automatically determine how many significant singular values (SVs) there are in the selected range. This corresponds to the number of significant scattering components in solution that EFA will attempt to deconvolve. At the bottom of the control panel, you should see that RAW thinks there are three significant SVs (scattering components) in our data. For this data set, that is accurate. We evaluate the number of significant components by how many singular values are above the baseline, and how many components have both left and right singular vectors with autocorrelations near one. For this data there are three singular values above baseline, and three singular vectors with autocorrelations near 1 (see step 3).



- *Note:* Typically you want the number of significant singular values and the number of singular vectors with autocorrelations near 1 to be equal. If they aren't, it likely indicates a weak or otherwise poorly resolved component in the dataset. Try the deconvolution first with the lower then the higher number of components.
- *Note:* RAW can find the wrong number of components automatically. You will always want to double check this automatic determination against the SVD results in the plots. If you change the data range used (or data type), the number of components will not automatically update so you should check and update it if necessary.

6. Click the "Next" button in the lower right-hand corner of the window to advance to the second stage of the EFA analysis.
   - *Note:* It may take some time to compute the necessary values for this next step, so be patient.



7. This step shows you the "Forward EFA" and "Backward EFA" plots. These plots represent the value of the singular values as a function of frame.

   - *Note:* There is one more singular value displayed on each plot than available in the controls. This is so that in the following Steps you can determine where each component deviates from the baseline.

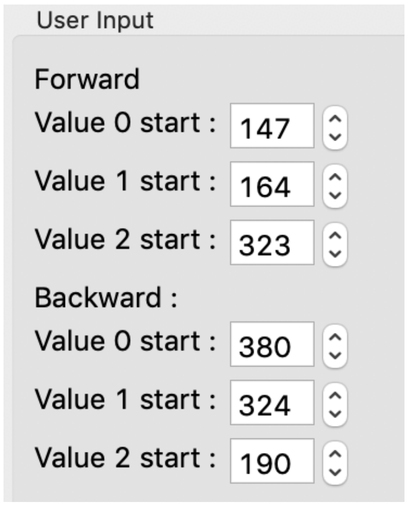
8. In the User Input panel, tweak the "Forward" value start frames so that the frame number, as indicated by the open circle on the plot, aligns with where the singular value first starts to increase quickly. This should be around 147, 164, and 323.

   - *Note:* For the Forward EFA plot, SVD is run on just the first two frames, then the first three, and so on, until all frames in the range are included. As more frames are added, the singular values change, as shown on the plot. When a singular value starts increasingly sharply, it indicates that there is a new scattering component in the scattering profile measured at that point. So, for the first ~150 frames, there are no new scattering components (i.e. just buffer scattering). At frame ~147, we see the first singular value (the singular value with index 0, labeled SV 0 on the plot) start to strongly increase, showing that we have gained a scattering component. We see SV 1 start to increase at ~164, indicating another scattering component starting to be present in the data.

9. In the User Input panel, tweak the "Backward" value start frames so that the frame number, as indicated by the open circle on the plot, aligns with where the singular value drops back to near the baseline. This should be around 380, 324, and 190.

   - *Note:* For the Backward EFA plot, SVD is run on just the last two frames, then the last three, and so on, until all frames in the range are included. As more frames are added, the singular values change, as shown on the plot. When a singular value drops back to baseline, it indicates that a scattering component is leaving the dataset at that point.

- *Note:* The algorithm for determining the start and end points is not particularly advanced. For some datasets you may need to do significantly more adjustment of these values



User Input

Forward

Value 0 start : 147

Value 1 start : 164

Value 2 start : 323

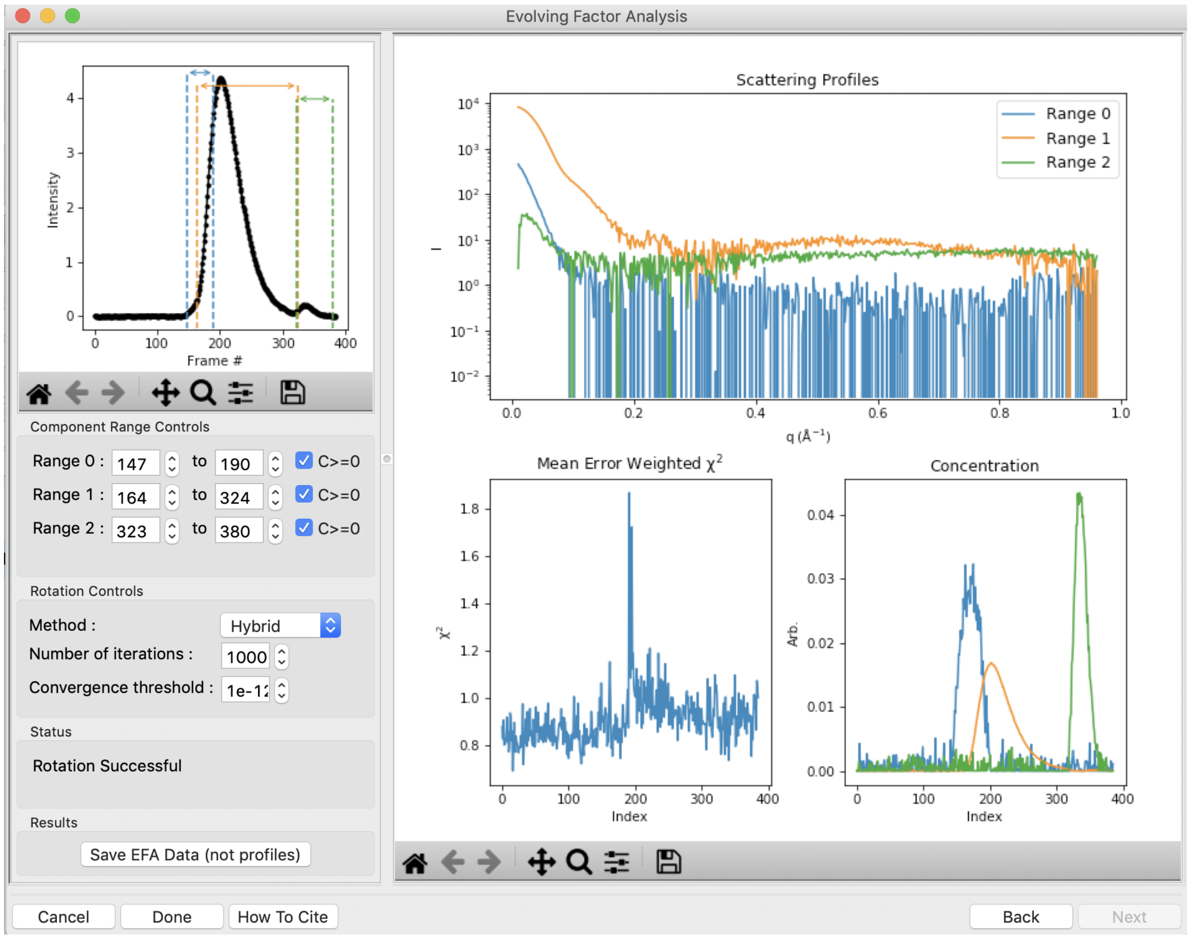
Backward :

Value 0 start : 380

Value 1 start : 324

Value 2 start : 190

10. Click the "Next" button in the bottom right corner to move to the last stage of the EFA analysis.



Evolving Factor Analysis

Component Range Controls

Range 0 : 147 to 190 ✓ C>=0

Range 1 : 164 to 324 ✓ C>=0

Range 2 : 323 to 380 ✓ C>=0

Rotation Controls

Method : Hybrid

Number of iterations : 1000

Convergence threshold : 1e-12

Status

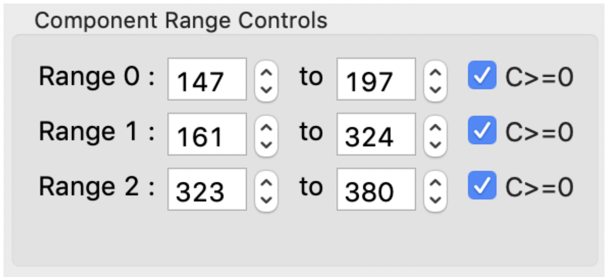
Rotation Successful

11. This window shows controls on the left and results on the right. In the controls area, at the top is a plot showing the SEC-SAXS curve, along with the ranges occupied by each scattering component, as determined from the

input on the Forward and Backward EFA curves in stage 2 of the analysis. The colors of the ranges correspond to the colors labeled in the Scattering Profiles plot on the top right and the Concentration plot in the lower right. This panel takes the SVD vectors and rotates them back into scattering vectors corresponding to real components.

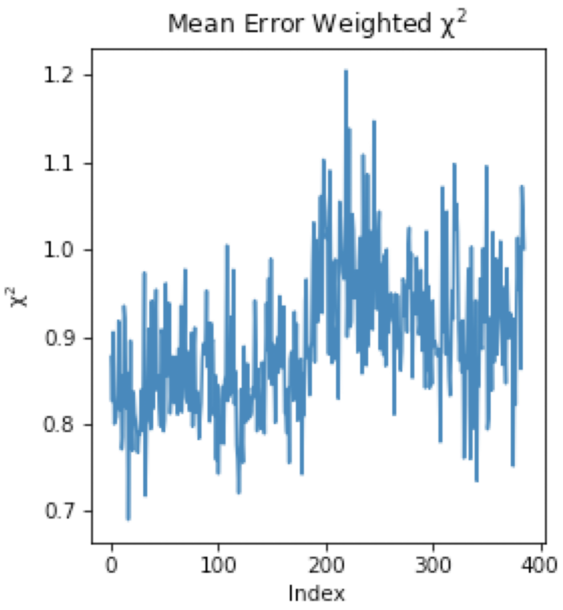
- *Note:* This rotation is not guaranteed to be successful, or to give you valid scattering vectors. Any data obtained via this method should be supported in other ways, either using other methods of deconvolving the peak, other biophysical or biochemical data, or both!

12. Fine tune the ranges using the controls in the "Component Range Controls" box. Adjust the starts and ends of Range 0 and the start of Range 1 by a few points until the spikes in the chi-squared plot go away. After these adjustments, Range 0 should be about 147 to 197, Range 1 from 161 to 324, and Range 2 from 323 to 380.



Component Range Controls

Range 0 :	147	to	197	✓ C>=0
Range 1 :	161	to	324	✓ C>=0
Range 2 :	323	to	380	✓ C>=0



13. To see these changes on the Forward and Backward EFA plots, click the "Back" button at the bottom right of the page. Verify that all of your start and end values are close to where the components become significant, as discussed in Steps 8 and 9.

14. Click the "Next" button to return to the final stage of the EFA analysis.

15. In the Rotation Controls box, you can set the method, the number of iterations, and the convergence threshold. As you can see in the Status window, the rotation was successful for this data. If it was not, you could try changing methods or adjusting the number of iterations or threshold.

16. Examine the chi-squared plot. It should be uniformly close to 1 for good EFA. For this data, it is.

17. Examine the concentration plot. You'll see three peaks, corresponding to the concentrations for the three components. In the Range Controls, uncheck the Range 0 C>=0 box. That removes the constraint that the concentration

must be positive. If this results in a significant change in the peak, your EFA analysis is likely poor, and you should not trust your results.

- *Note:* The height of the concentration peaks is arbitrary, all peaks are normalized to have an area of 1.

18. Uncheck all of the C>=0 controls.

    - *Question:* Do you observe any significant changes in the scattering profiles, chi-squared, or concentration when you do this? How about if you uncheck one and leave the others checked?

19. Recheck all of the C>=0 controls. You have now verified, as much as you can, that the EFA analysis is giving you reasonable results.

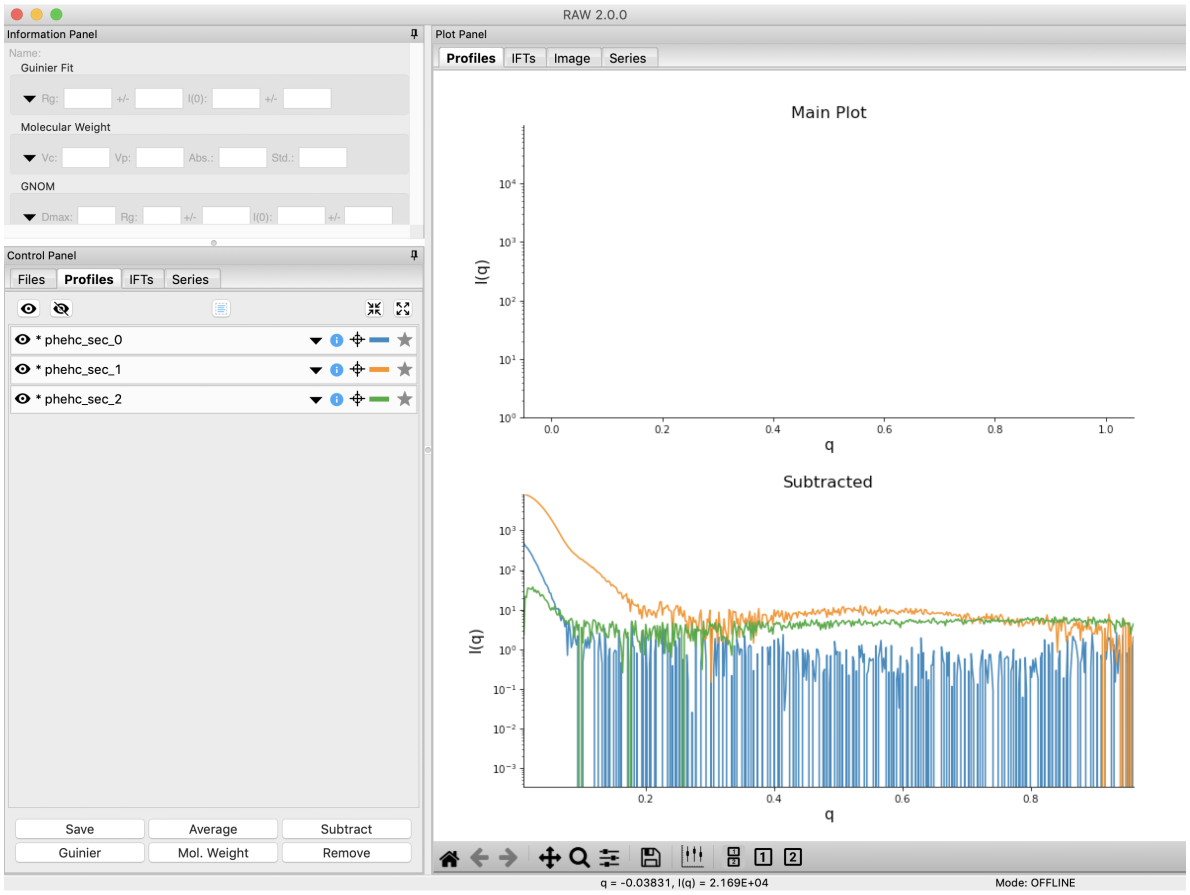
20. *Reminder:* Here are the verification steps we have carried out, and you should carry out every time you do EFA:

    1. Confirm that your selected ranges correspond to the start points of the Forward and Backward EFA values (Steps 12-13).
    2. Confirm that your chi-squared plot is close to 1, without any major spikes.
    3. Confirm that your concentrations are not significantly altered by constraining the concentration to be positive (Steps 17-19).

21. Click the "Save EFA Data (not profiles)" to save the EFA data, including the SVD, the Forward and Backward EFA data, the chi-squared, and the concentration, along with information about the selected ranges and the rotation method used.

22. Click the "Done" button to send the scattering profiles to the Profiles Plot.

23. In the main RAW window, go to the Profiles control tab and the Profiles plot. If it is not already, put the Profiles plot on a semi-Log or Log-Log scale.



24. The three scattering profiles from EFA are in the manipulation list. The labels _0, _1, and _2 correspond to the 0, 1, and 2 components/ranges.

    - *Note:* Regardless of whether you use subtracted or unsubtracted data, these scattering profiles will be buffer subtracted, as the buffer represents a scattering component itself, and so (in theory) even if it is present will be separated out by successful EFA.

Advanced Series processing – Regularized Alternating Least Squares (REGALS)

REGALS is an algorithm for deconvolution of mixtures in small angle scattering data. It can be applied to deconvolving overlapping peaks in SEC-SAXS, changing baseline and elution peaks in ion exchange chromatography SAXS, mixtures in time resolved SAXS data and equilibrium titration series data, and likely other cases. In practical application it can be thought of as an extension of the *evolving factor analysis (EFA)* technique to more complex conditions where components are not necessarily entering and exiting the dataset in a strict first-in-first-out approach like in SEC-SAXS. EFA is recommended for standard SEC-SAXS data, but for more complex data, as listed above, you should use REGALS. REGALS can also handle deconvolution of SEC-SAXS data with a sloping baseline.

To learn more about REGALS we recommend these resources:

- *REGALS: a general method to deconvolve X-ray scattering data from evolving mixtures* S. P. Meisburger, D. Xu, and N. Ando. IUCrJ (2021). 8(2), 225-237. DOI: 10.1107/S2052252521000555
- The source code for the REGALS algorithm on github.
- This talk by Dr. Steve Meisburger

If you use REGALS in RAW, in addition to citing the RAW paper, please cite the REGALS paper: S. P. Meisburger, D. Xu, and N. Ando. IUCrJ (2021). 8(2), 225-237. DOI: 10.1107/S2052252521000555

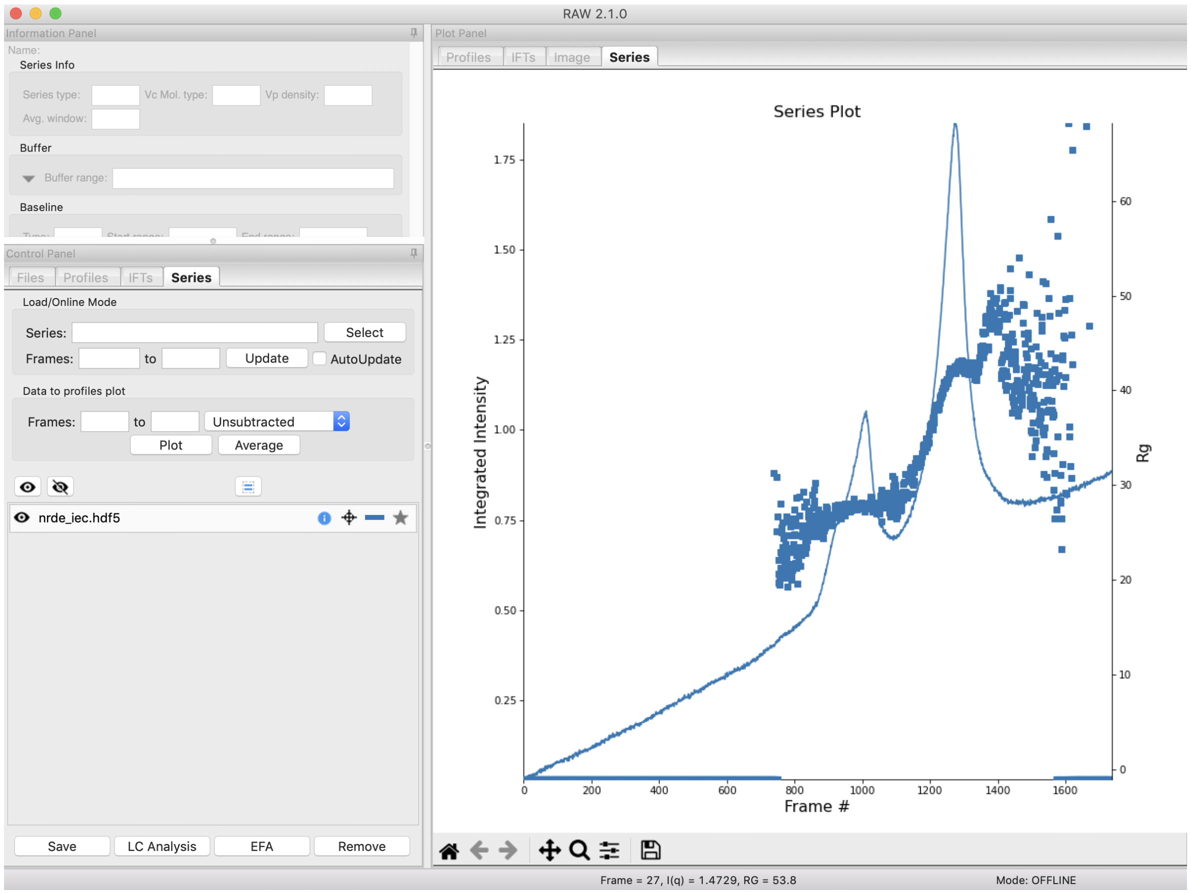
The written version of the tutorial follows.

Deconvolving ion exchange chromatograph coupled SAXS (IEC-SAXS) data

Ion exchange chromatography is similar to size exclusion chromatography, except that it uses a charged column media and a salt gradient to separate samples by charge rather than size. IEC-SAXS (sometimes called anion exchange or AEX-SAXS) can be useful in cases where there are multiple species in solution that are not separable by size. However, the use of a salt gradient causes a changing buffer background for the sample, making accurate buffer subtraction challenging. REGALS can be used to deconvolve the scattering of the eluted macromolecules from the changing scattering of the buffer.

1. Clear all of the data in RAW. Load the **nrde_iec.hdf5** file in the **series_data** folder.

    - *Note:* The data were provided by the Ando group at Cornell University, and were originally published in the paper: Parker, M. J. et al. (2018). *An endogenous dAMP ligand in Bacillus subtilis class Ib RNR promotes assembly of a noncanonical dimer for regulation by dATP.* Proc Natl Acad Sci USA 115, E4594–E4603. DOI: 10.1073/pnas.1800356115. It is also available from the REGALS github repository.

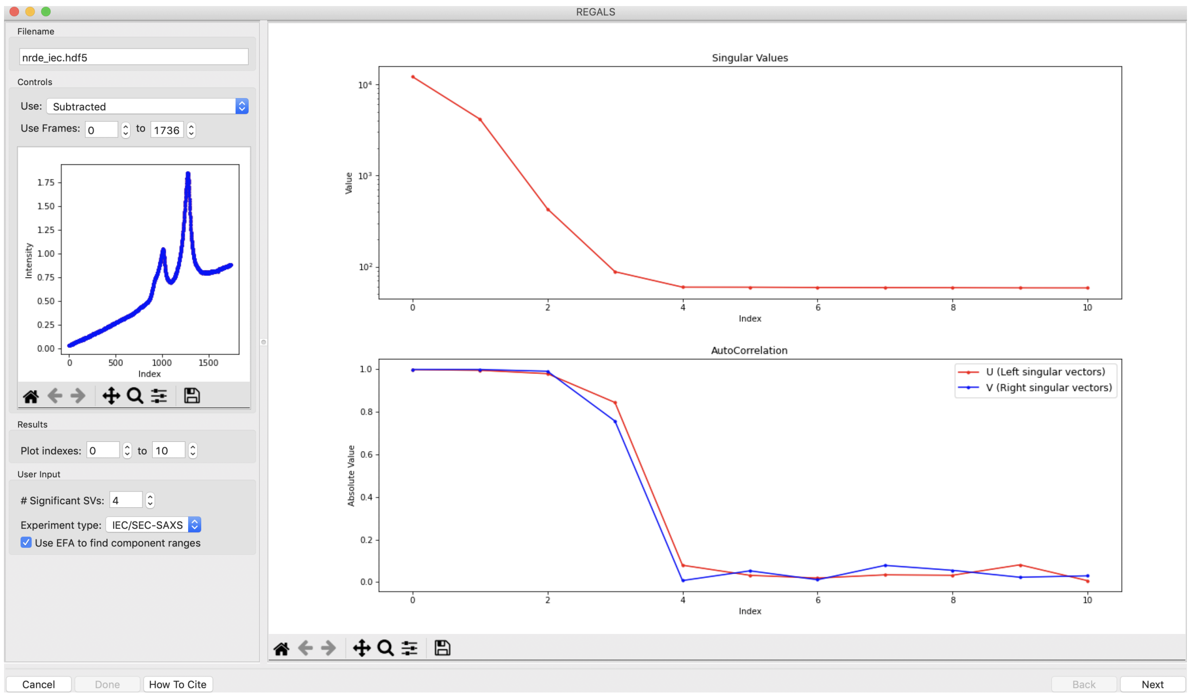


2. We will use REGALS to extract out the scattering components in both peaks of the data. Right click on the **nrde_iec.hdf5** item in the Series list. Select the “REGALS” option.

3. The REGALS window will be displayed. On the left are controls, on the right are plots of the value of the singular values and the first autocorrelation of the left and right singular vectors.

   - *Note:* Large singular values indicate significant components. What matters is the relative magnitude, that is, whether the value is large relative to the mostly flat/unchanging value of high index singular values.

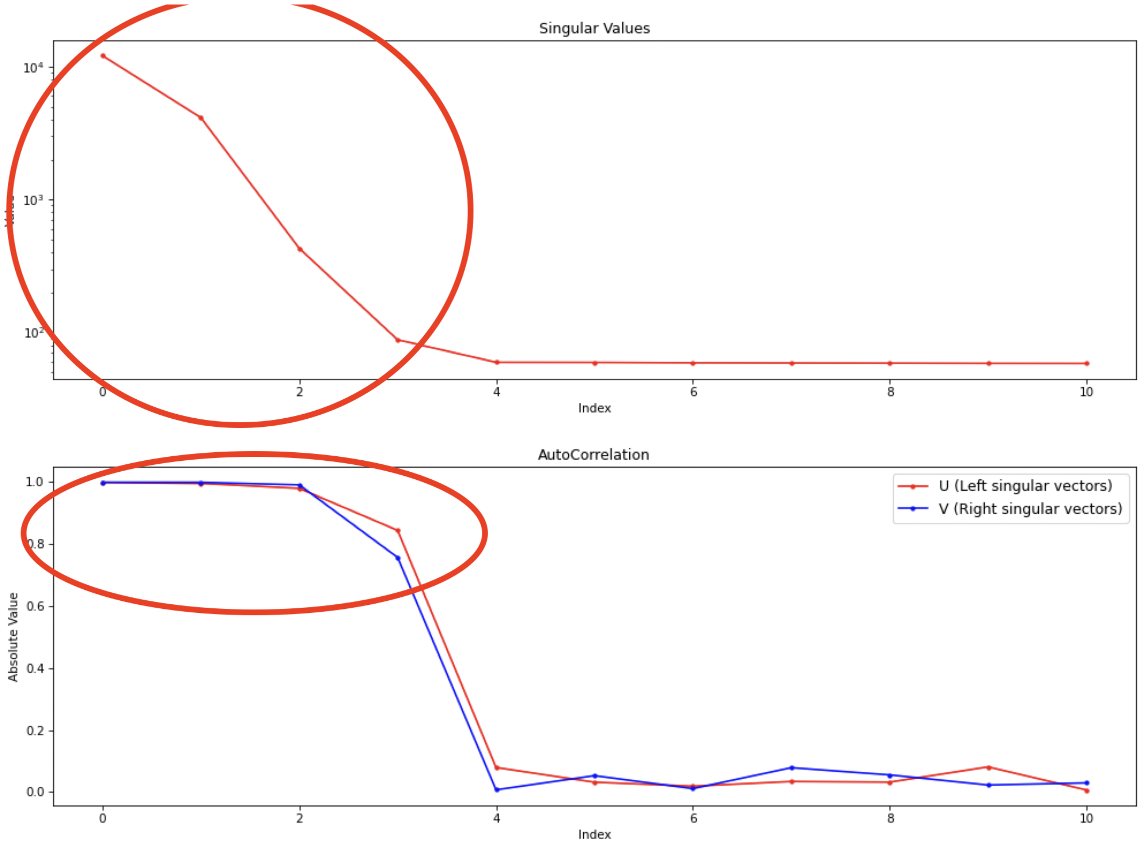
   - *Note:* A large autocorrelation indicates that the singular vector is varying smoothly, while a low autocorrelation indicates the vector is very noisy. Vectors corresponding to significant components will tend to have autocorrelations near 1 (roughly, >0.6-0.7) and vectors corresponding to insignificant components will tend to have autocorrelations near 0.



4. Nominally REGALS should work with either subtracted or unsubtracted data. Here, a buffer measured before the start of the dataset has been subtracted off of all the profiles in the dataset. This helps reduce the magnitude of the buffer components in the data, which can make it easier to refine the macromolecular components. We will use the full range of the dataset, but REGALS can also be done on restricted ranges.

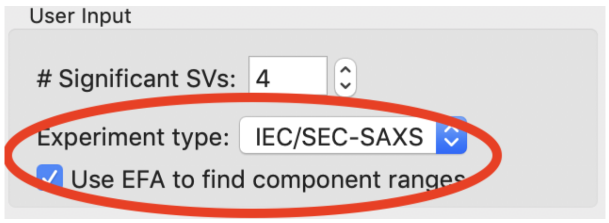
   - *Note:* Using a restricted range for REGALS might be useful to exclude one or more components of a complicated elution from the deconvolution. The more components you have the harder it is to do REGALS. There is a trade off in the amount of data used (more is better) and the number of components in the deconvolution (fewer is better) that can require some experimentation to find the right balance for a given dataset.

5. RAW attempts to automatically determine how many significant singular values (SVs) there are in the selected range. This corresponds to the number of significant scattering components in solution that REGALS will attempt to deconvolve. At the bottom of the control panel, you should see that RAW thinks there are four significant SVs (scattering components) in our data. For this data set, that is accurate. We evaluate the number of significant components by how many singular values are above the baseline, and how many components have both left and right singular vectors with autocorrelations near one. For this data there are four singular values above baseline, and four singular vectors with autocorrelations near 1 (see step 3).



- *Note:* Typically you expect the number of significant singular values and the number of singular vectors with autocorrelations near 1 to be equal. If they aren't, it likely indicates a weak or otherwise poorly resolved component in the dataset. Try the deconvolution first with the lower then the higher number of components.
- *Note:* RAW can find the wrong number of components automatically. You will always want to double check this automatic determination against the SVD results in the plots. If you change the data range used (or data type), the number of components will not automatically update so you should check and update it if necessary.

6. For REGALS there are two other settings you should check. First, set the experiment type. In this case, the default of 'IEC/SEC-SAXS' is correct. This sets default values later on in the deconvolution. Second specify whether you want to use the evolving factor plots to find initial guesses for the range of components in solution. This is useful in highly sampled datasets like IEC-SAXS, but may not be very accurate for more sparsely sampled data like a titration series, and so can be skipped. For this dataset leave it checked.


User Input

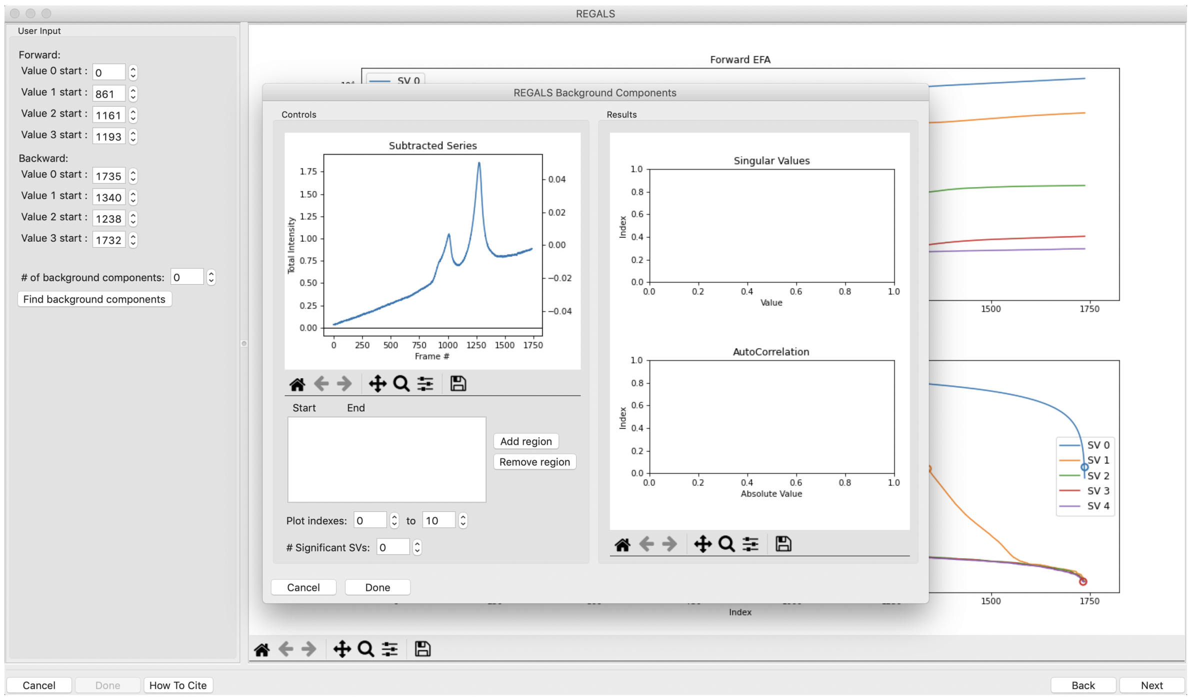
Significant SVs: 4

Experiment type: IEC/SEC-SAXS

✓ Use EFA to find component ranges

7. Click the "Next" button in the lower right-hand corner of the window to advance to the second stage of the REGALS analysis

   - *Note:* If, as in this case, you are going to use the EFA plots to find the component ranges, it will take some time to compute the necessary values for this next step, so be patient.



Forward:
Value 0 start : 0
Value 1 start : 861
Value 2 start : 1161
Value 3 start : 1193
Backward:
Value 0 start : 1735
Value 1 start : 1340
Value 2 start : 1238
Value 3 start : 1732
# of background components: 0

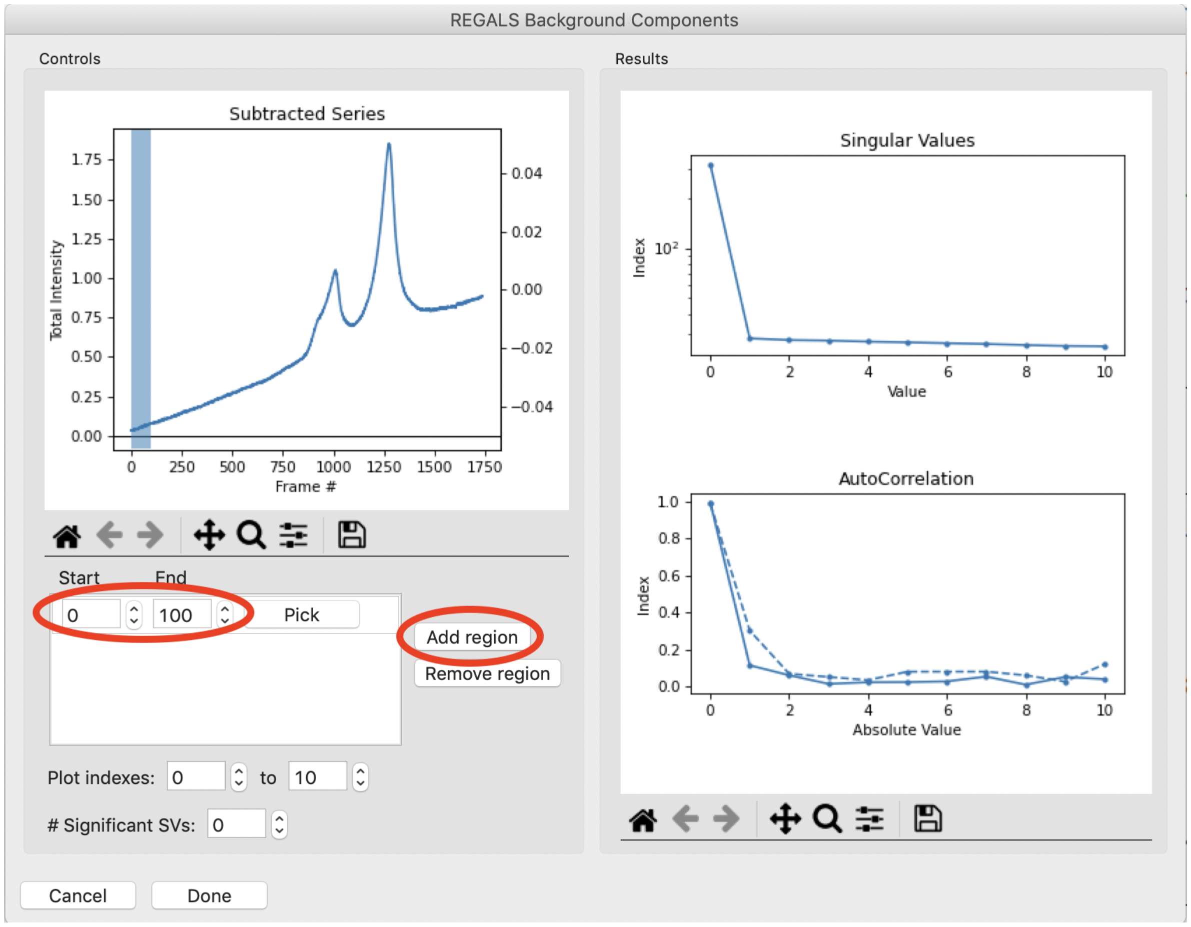
Start End

Plot indexes: 0 to 10
Significant SVs: 0

8. A new window will open on top of the current one when you click "Next". The bottom window is the main REGALS window, with the forward and backward evolving factor plots. On top of that is the REGALS Background Components window. We will first use the Background Components window to estimate how many of our components are background components.

   - *Note:* For the purposes of REGALS, a background component is a component that spans the full range of the dataset.
   - *Note:* If you select an experiment mode other than 'IEC/SEC-SAXS' the Background Components window will not open automatically, but can still be opened by clicking the "Find background components" button if desired.

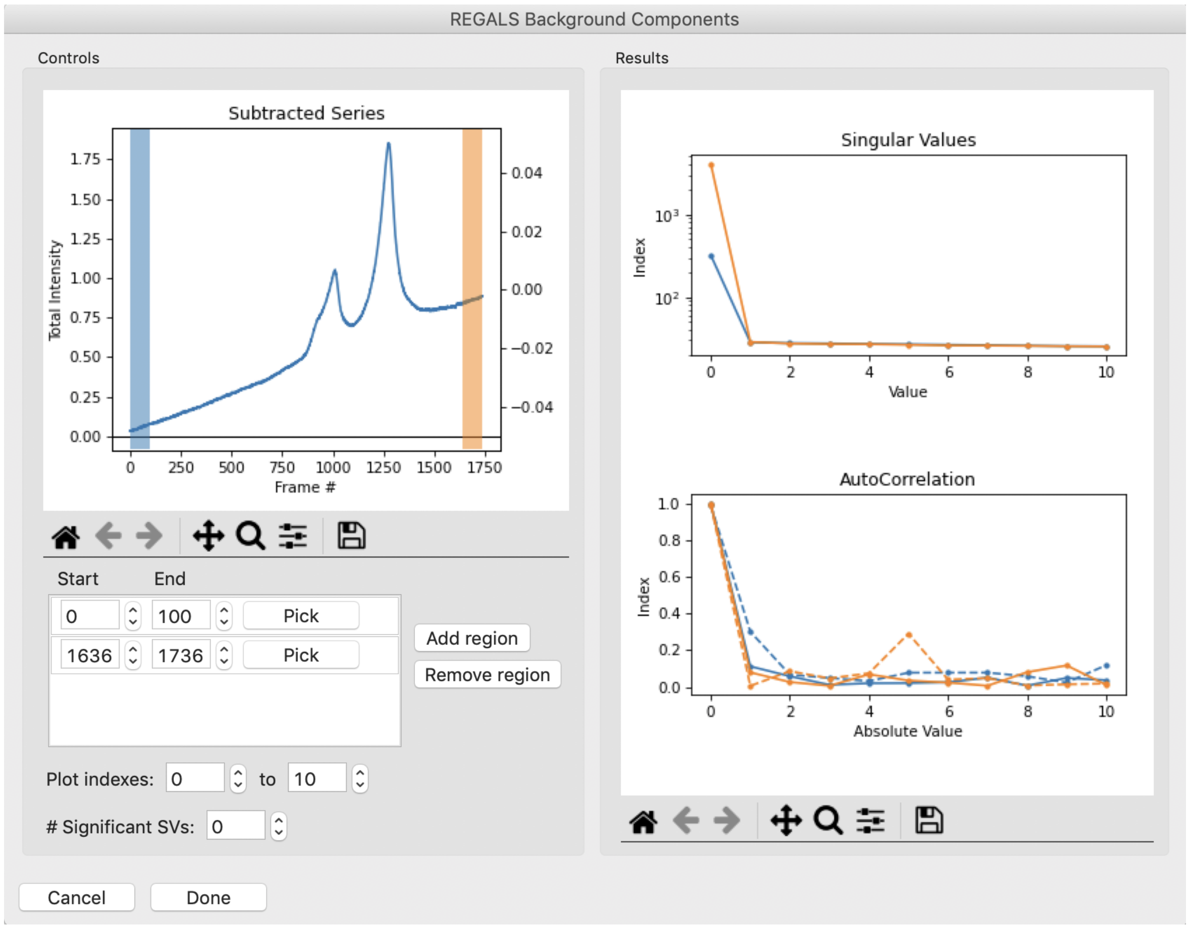
9. In the Background Components window, click "Add Region" to add a region to the plot. The SVD of the data in that region is shown in the plots on the right. Set the range of that region to 0 to 100. The SVD plots show that there is only one strong component in the dataset in this range.



Start: 0
End: 100
Plot indexes: 0 to 10
Significant SVs: 0

10. Add a region corresponding to the last 100 frames of the dataset. This will appear as a second colored range on the series plot, with corresponding second colored sets of lines in the SVD plots on the right. Notice that there is just one component in the last 100 frames of the dataset.

    - *Note:* The left autocorrelation is shown with the solid line, the right autocorrelation with the dashed line.



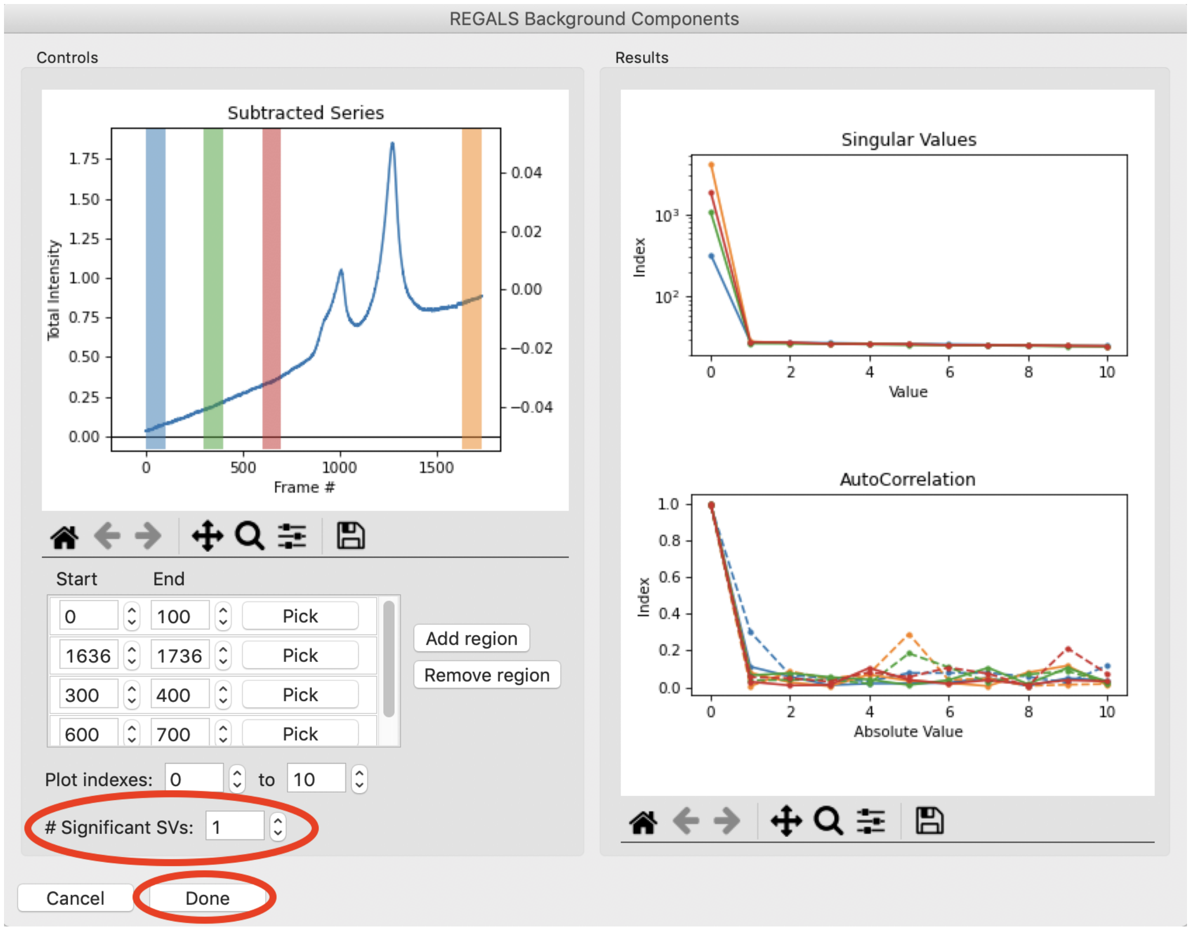
Start	End	
0	100	Pick
1636	1736	Pick

Plot indexes: 0 to 10

Significant SVs: 0

11. Add several more similar regions through the initial upward sloping buffer region. Notice that there appears to be just one strong value throughout most of the buffer region, both before and after the eluted peaks. So as a starting point we will set the number of significant background components to 1, by setting the "# Significant SVs" input to 1. Once that's done, click the "Done" button.

   - *Tip:* You can remove regions if they are not longer useful. To do so, select the region by clicking to the right of the pick button and then clicking the "Remove region" button.



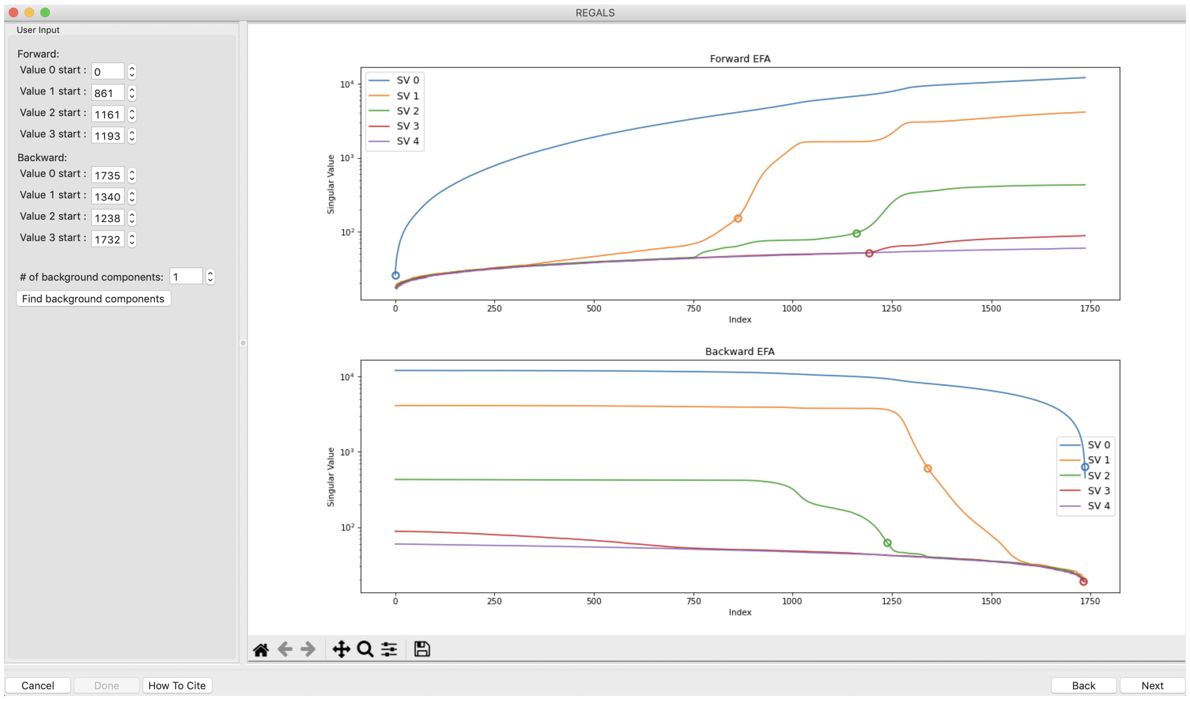
Start	End
0	100
1636	1736
300	400
600	700

Plot indexes: 0 to 10

Significant SVs: 1

12. The Background Components window will close and the number of background components shown in the main REGALS window will update. The main REGALS window will now be fully visible.

    - *Note:* You can reopen the Background Components window using the “Find background components” button if desired.



Forward:
Value 0 start : 0
Value 1 start : 861
Value 2 start : 1161
Value 3 start : 1193

Backward:
Value 0 start : 1735
Value 1 start : 1340
Value 2 start : 1238
Value 3 start : 1732

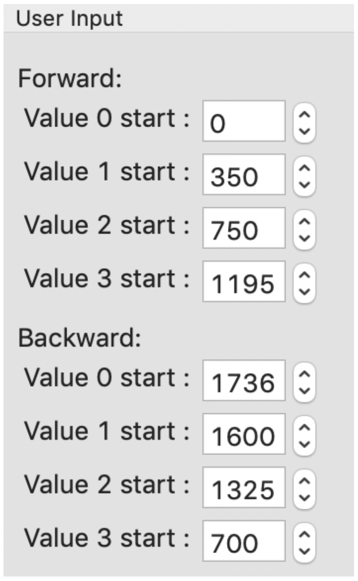
# of background components: 1

13. In the User Input panel, tweak the "Forward" value start frames so that the frame number, as indicated by the open circle on the plot, aligns with where the singular value first starts to rise above the baseline. This should be around 0, 350, 750, and 1195.

    - *Note:* For the Forward EFA plot, SVD is run on just the first two frames, then the first three, and so on, until all frames in the range are included. As more frames are added, the singular values change, as shown on the plot. When a singular value starts rising above the baseline, it indicates that there is a new scattering component in the scattering profile measured at that point. So, for the first ~350 frames, there is only one scattering components (i.e. just buffer scattering). At frame ~350, we see the second singular value (the singular value with index 1, labeled SV 1 on the plot) start to increase, showing that we have gained a scattering component.
    - *Note:* One component starts above baseline, indicating there is already a significant scattering component at the start of our dataset. In this case that is the sloping buffer gradient.

14. In the User Input panel, tweak the "Backward" value start frames so that the frame number, as indicated by the open circle on the plot, aligns with where the singular value drops back to near the baseline. This should be around 700, 1325, 1600, and 1736.

    - *Note:* For the Backward EFA plot, SVD is run on just the last two frames, then the last three, and so on, until all frames in the range are included. As more frames are added, the singular values change, as shown on the plot. When a singular value drops back to baseline, it indicates that a scattering component is leaving the dataset at that point.
    - *Note:* One component ends above baseline, indicating there is still a significant scattering component at the end of our dataset. In this case that is the sloping buffer gradient.



User Input

Forward:

Value 0 start : 0

Value 1 start : 350

Value 2 start : 750

Value 3 start : 1195

Backward:

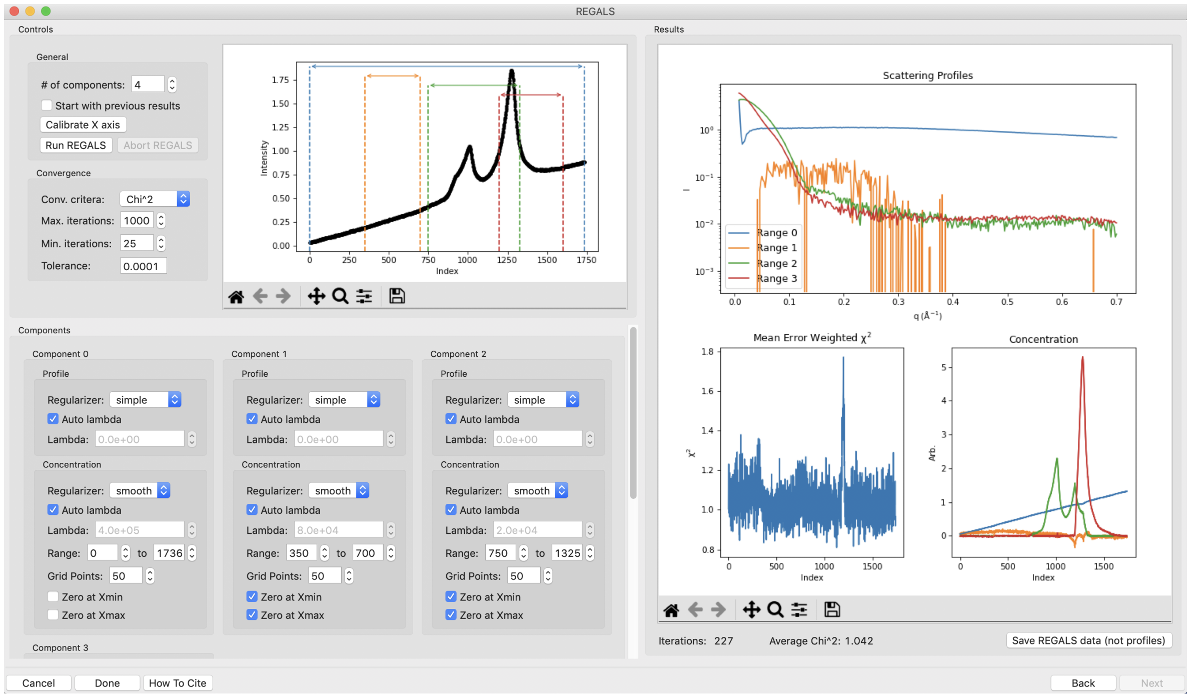
Value 0 start : 1736

Value 1 start : 1600

Value 2 start : 1325

Value 3 start : 700

15. Click the "Next" button in the bottom right corner to move to the last stage of the REGALS analysis.



REGALS

Controls

General

# of components: 4

Start with previous results

Convergence

Conv. critera: Chi^2

Max. iterations: 1000

Min. iterations: 25

Tolerance: 0.0001

Components

	Component 0	Component 1	Component 2
Profile Regularizer:	simple	simple	simple
Profile Auto lambda	✓	✓	✓
Profile Lambda:	0.0e+00	0.0e+00	0.0e+00
Concentration Regularizer:	smooth	smooth	smooth
Concentration Auto lambda	✓	✓	✓
Concentration Lambda:	4.0e+05	8.0e+04	2.0e+04
Range:	0 to 1736	350 to 700	750 to 1325
Grid Points:	50	50	50
Zero at Xmin		✓	✓
Zero at Xmax		✓	✓

Component 3

Results

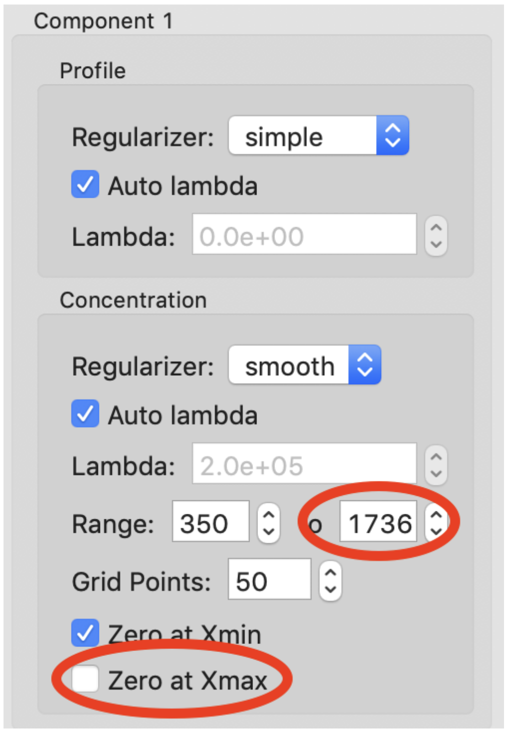
Iterations: 227 Average Chi^2: 1.042

16. This window shows controls on the left and results on the right. In the controls area at the top are general controls. You can adjust the number of components, calibrate the X axis, and set the convergence criteria for the REGALS algorithm. A plot of the ranges is also shown. On the bottom are the controls for each individual component. We won't go through all the possible permutations for each setting, so if you want to know more check out the links *at the top* of this tutorial.

  - *Note:* The ranges are automatically assigned based on the start and end points for components found in the EFA plots. Background components ranges are assigned on the principle of first-in last-out, whereas all the other component ranges are assigned by first-in first-out.
  - *Note:* Components are only shown three across. Scroll down in the component area to see more component settings.

17. Now we need to start refining our component ranges, and tweaking other settings in the components. Component 0 looks good for now, so we will start with component 1. On the concentration plot on the right, notice that despite the component 1 range being defined from only 350-700, the component shows features at a wide range of frame numbers (index). Looking at the scattergram with the ranges plotted, there's no obvious elution component in the range where component 1 is defined. This means that component 1 is likely a background component as well. Since the SVD for the background components and the EFA plots were not clear on where the component should end, we will fit it from the start point we found, 350, until the end of the dataset. To do this, change the endpoint of the concentration range to 1736. Also uncheck the 'Zero at Xmax' box for this component, as it may contribute to the buffer scattering at the end of the dataset.

    - *Tip:* If you return to the Background Components window, if you add a single range from 0 to 700 you'll see two components, indicating that there are likely multiple background components. However, if you look at the end region, assuming you avoid the tail of the peak (~1600-1736) there's only one component. So it's not clear where the first component might end and the second one start, and how much coexistence there is between these two background components. In this case, fitting both to end at the end of the dataset works well.


Component 1

Profile

Regularizer: simple

✓ Auto lambda

Lambda: 0.0e+00

Concentration

Regularizer: smooth

✓ Auto lambda

Lambda: 2.0e+05

Range: 350 to 1736

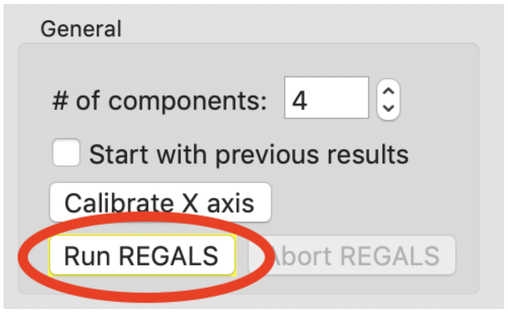
Grid Points: 50

✓ Zero at Xmin

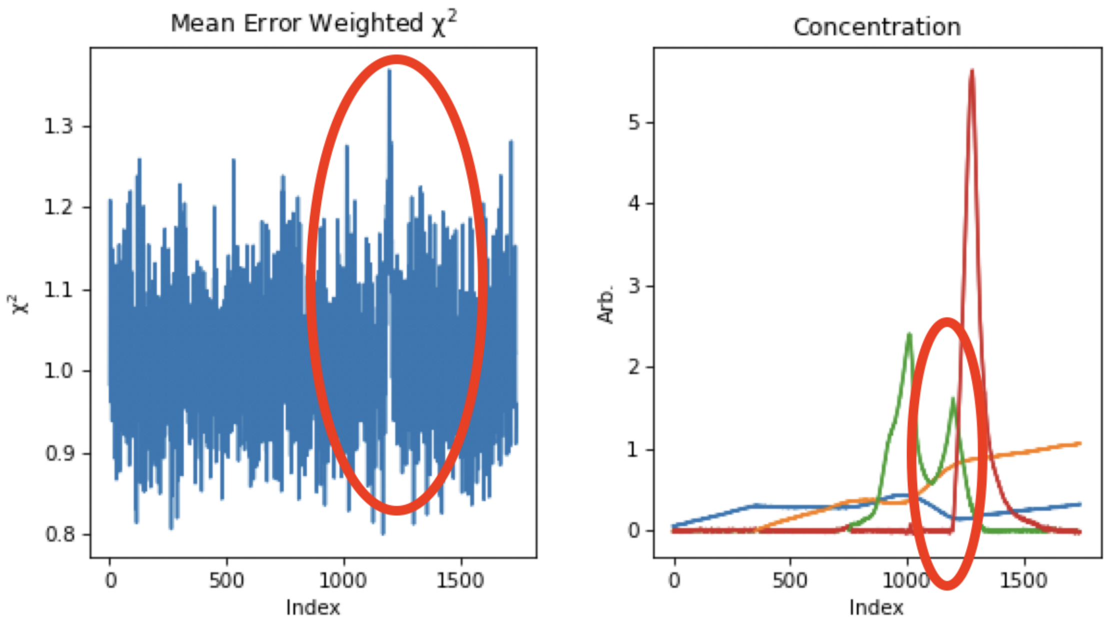
☐ Zero at Xmax

18. Because it can take a while to run, REGALS does not automatically update the results. To see how changing the range changed the deconvolution, click the "Run REGALS" button.

    - *Note:* If lambda is automatically updated for a component, this value in the GUI will not be updated until you run REGALS.
    - *Note:* When you have changes to your deconvolution settings and REGALS hasn't been run with those settings the "Run REGALS" button will have a yellow background.



19. There is a definite improvement in the REGALS results after rerunning the data. Next we will refine the protein components. We will first refine the ranges. On the concentration plot, notice that the component 3 (red) concentration is pulled down to zero pretty sharply on the left side. At the same point there's a peak in the component 2 (green) concentration right where the red one starts. There's also a small spike in the chi^2 plot around frame 1200, which is where component 3 starts. This indicates that we've restricted component 3 excessively and some of the scattering is going into component 2. We will change the range for component 3 to start at an earlier point and see how that affects the deconvolution. To do this, set the start of component 3 to 1150 and run REGALS.

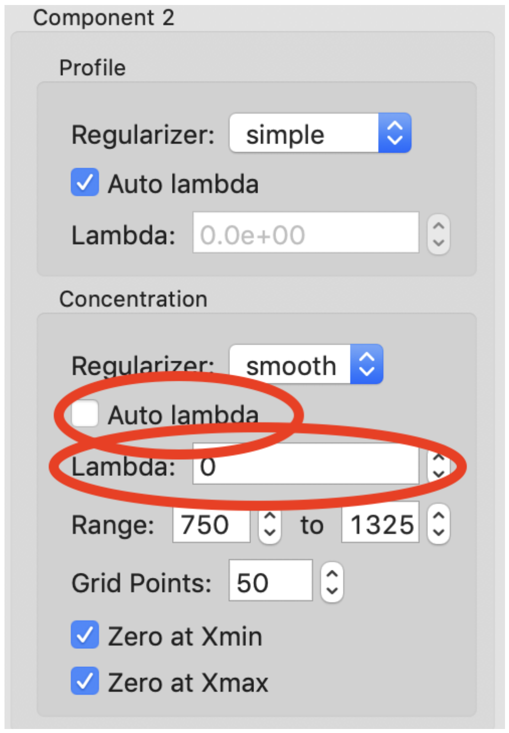


20. After running REGALS, notice that the chi^2 spike is completely gone, and that the range 3 concentration is coming back down to zero in a more natural way. Also notice that the spike in component 2 concentration is reduced. If you look closely at the component 2 concentration you can see a bit of a double peak around ~1150. There could be a little bit more component 3 there, so we will start the component 3 range earlier. Try 1125 and 1100 for the component 3 start.

21. As there's minimal change between 1100 and 1125, set the component 3 start back to 1125.

22. At this point the deconvolution is starting to look reasonable. There's nothing obviously wrong with the component ranges, so next we will adjust the lambda values for the components. These control the degree of smoothing in the deconvolution. For strong components with lots of measured profiles, like the peaks in this dataset, we don't need a large lambda. In fact, we might not need any lambda. For component 2 and 3 concentrations turn off "Auto lambda", and set the lambda value to 0. Then run REGALS again.

    - *Tip:* Make sure you're setting lambda for the concentration, not the profile, for each component.


Component 2

Profile

Regularizer: simple

✓ Auto lambda

Lambda: 0.0e+00

Concentration

Regularizer: smooth

Auto lambda

Lambda: 0

Range: 750 to 1325

Grid Points: 50

✓ Zero at Xmin

✓ Zero at Xmax

23. After running REGALS with the peak component concentration lambdas set to 0, the component 2 concentration (green) has a small negative dip at the end of the concentration range. This indicates that we should adjust the range for that component. In particular, we'll reduce the end point to try to eliminate that dip. Try 1300 and 1275 as endpoints for component 2 concentration.

    - *Note:* If we had set the lambdas particularly poorly, we would start to see the chi^2 plot deviate from ~1. Since we don't, we can conclude that our lambda values are probably okay.

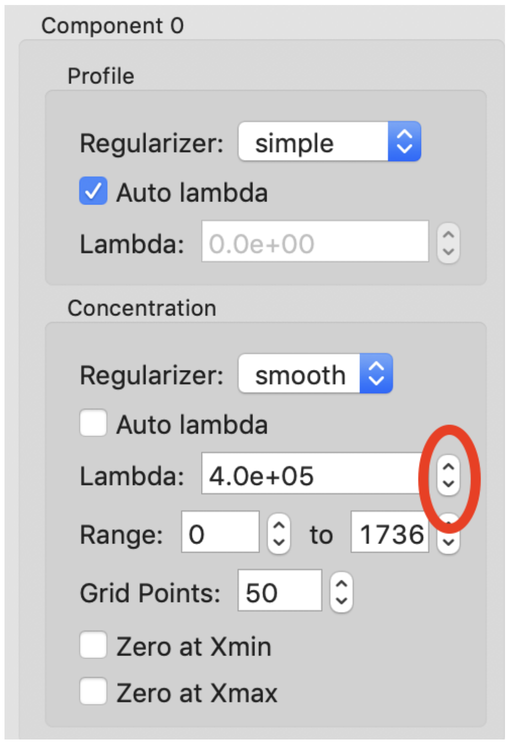
24. Notice that 1275 essentially eliminates the dip in the concentration. Set that as the endpoint for the component 2 concentration.

    - *Try:* You can try small tweaks near 1275 if you like, but you shouldn't see significant changes in the concentration shape. So we'll leave it set at 1275 for now.

25. The final thing we will adjust is the lambdas for the background component concentrations. Those concentration profiles are a bit wavy. We would expect the concentrations of the buffer to increase linearly, since a linear salt gradient was applied during elution. To increase the smoothness of the concentration, we will increase lambda. Turn off "Auto lambda" for components 0 and 1. Then use the up arrow next to the lambda box to increase each lambda by an order of magnitude. Run REGALS to see how this affects the deconvolution.

    - *Note:* Generally you want to adjust lambda by an order of magnitude or more. Smaller adjustments will have minimal effect on the deconvolution.

    - *Tip:* You can also type a new lambda value directly into the box.


Component 0

Profile

Regularizer: simple

[✓] Auto lambda

Lambda: 0.0e+00

Concentration

Regularizer: smooth

[ ] Auto lambda

Lambda: 4.0e+05

Range: 0 to 1736

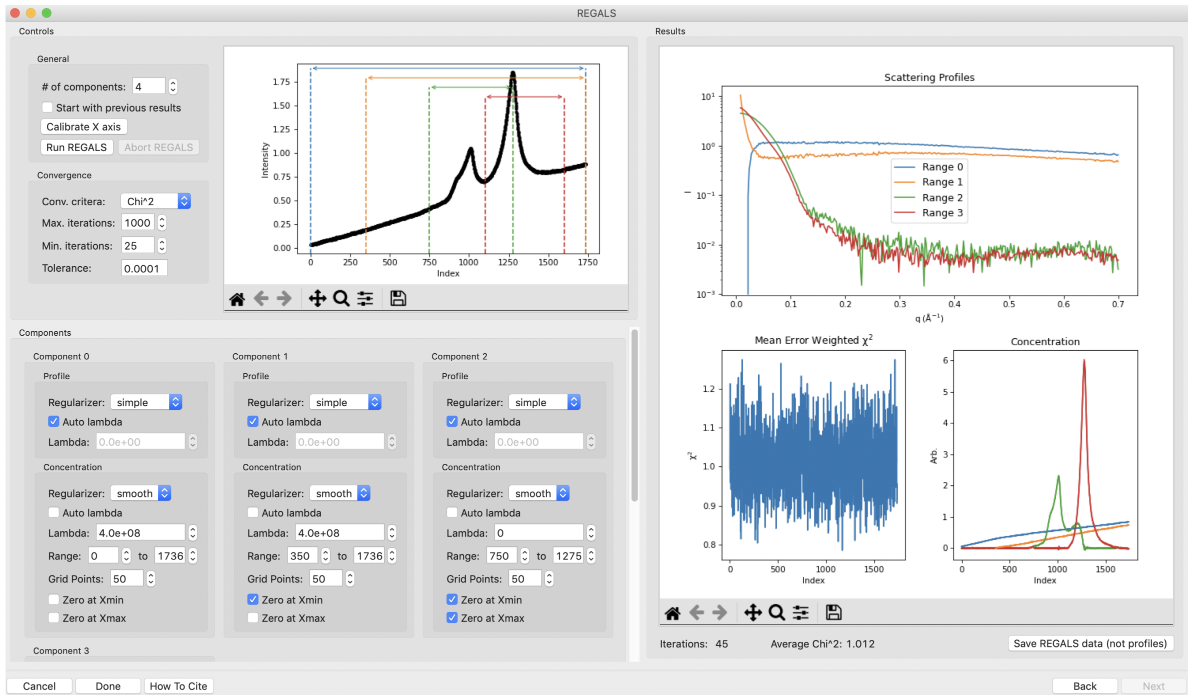
Grid Points: 50

[ ] Zero at Xmin

[ ] Zero at Xmax

26. There are two major changes with the new lambdas. First, the high q of the scattering profiles for ranges 2 and 3 will get more similar. Second, the concentration for ranges 0 and 1 will get smoother. The change in high q comes from a decrease in the high q values component 2, while component 3 stays mostly the same. This implies that more of the buffer scattering is getting picked out component 2. Keep increasing both buffer component lambdas by an order of magnitude until this stops changing. You will also see a sudden and dramatic change in the component 1 profile at some point.

27. Once you see the high q backgrounds match and the large change in the scattering profile for component 1, it means you're oversmoothing the buffer components. Reduce the lambda for both buffer components to the last good value, which would be ~4e8.



General

# of components: 4

Start with previous results

Convergence

Conv. critera: Chi^2

Max. iterations: 1000

Min. iterations: 25

Tolerance: 0.0001

Components

	Component 0	Component 1	Component 2
Profile Regularizer	simple	simple	simple
Auto lambda	✓	✓	✓
Lambda	0.0e+00	0.0e+00	0.0e+00
Concentration Regularizer	smooth	smooth	smooth
Auto lambda			
Lambda	4.0e+08	4.0e+08	0
Range	0 to 1736	350 to 1736	750 to 1275
Grid Points	50	50	50
Zero at Xmin		✓	✓
Zero at Xmax			✓

Component 3

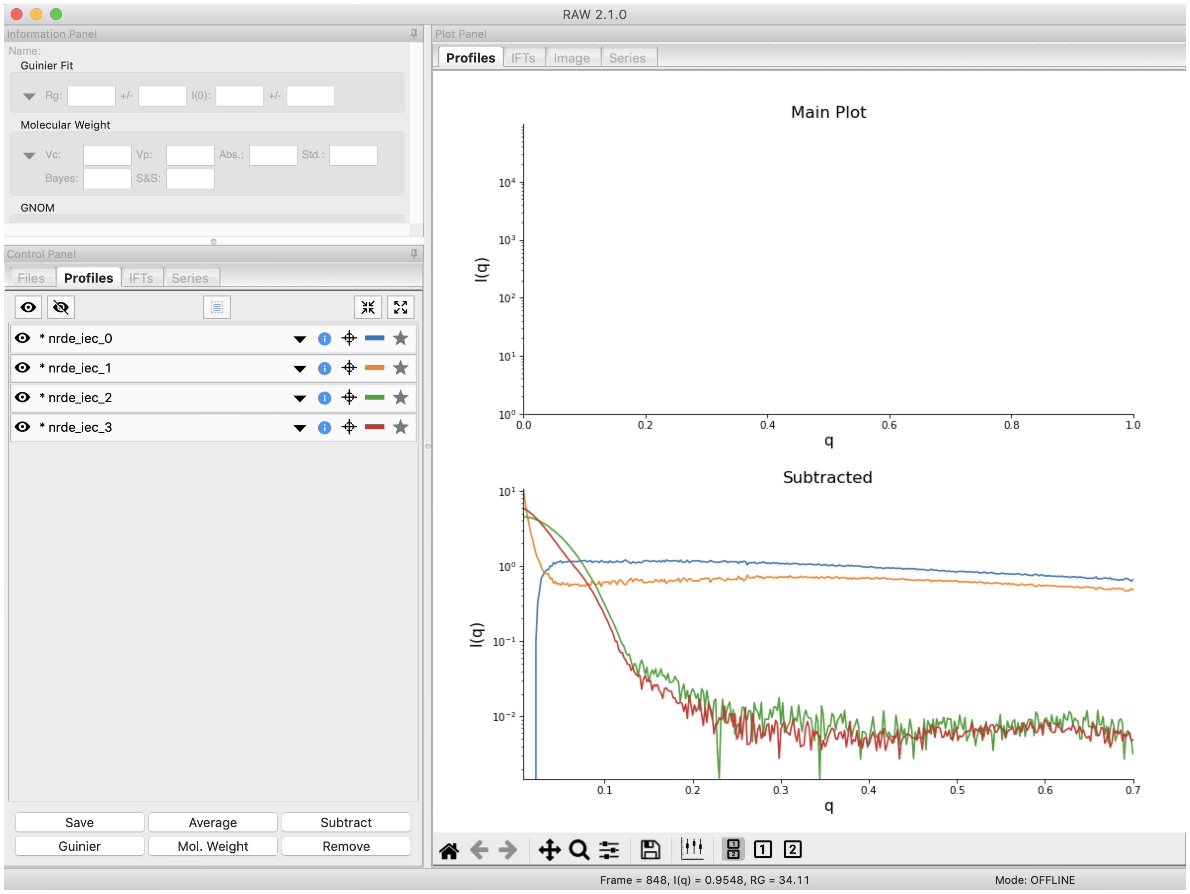
Iterations: 45

Average Chi^2: 1.012

28. This is a more or less optimum solution for REGALS deconvolution for this dataset. Since we're satisfied, we can now save the results. First, click on the "Save REGALS data (not profiles)" button in the bottom right corner and save. This saves a spreadsheet (.csv) file with all of the information from REGALS that isn't the scattering profiles, including the component settings and concentration and chi^2 vs. frame data.

29. Finally, click the "Done" button to close the REGALS window and send the deconvolved scattering profiles to the Profiles Plot.

30. In the main RAW window, go to the Profiles control tab and the Profiles plot. There you should see the deconvolved profiles. The labels _0, _1, _2, and _3 correspond to the 0, 1, 2, and 3 components from REGALS.



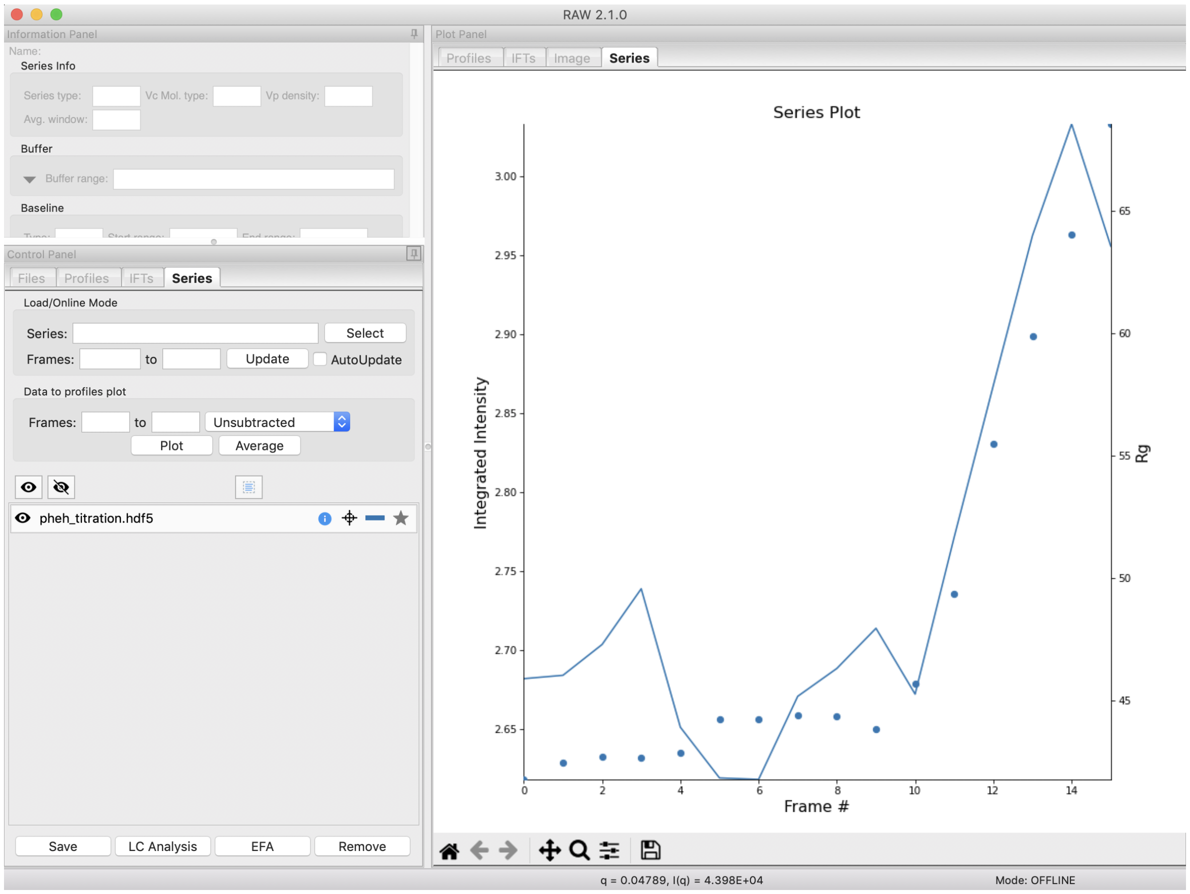
Deconvolving equilibrium titration series SAXS data

Titration series SAXS data looks at the change in scattering profile of a macromolecule as a substance (salt, substrate, another macromolecule, etc.) is titrated in or our of the solution. It is done in equilibrium, so buffers are prepared ahead of time and samples are equilibrated in the buffer, then measured. These measurements are typically done in batch mode SAXS (without in-line separation from SEC), and often involve transitions between different conformations or oligomeric states, and so the measured scattering at a given titration point is not from a homogeneous and monodisperse sample. We can use REGALS to deconvolve the different scattering components in the titration series to get pure profiles for each component.

1. Clear all of the data in RAW. Load the **pheh_titration.hdf5** file in the **series_data** folder.

   - *Note:* This data are a titration series of phenylalanine (L-phe) into a sample of phenylalanine hydroxylase (PheH). 16 different concentration points were collected, ranging from 0 to 80 mM L-phe. The data are already background subtracted, using buffers containing a matching concentration of L-phe. A conformational change has previously been observed on PheH binding L-phe. A small amount of aggregation was also observed at all concentrations, preventing the use of saturated endpoints in the titration series to completely determine each conformational state in batch mode experiments. We will deconvolve both conformations and the aggregate scattering from the titration series. Prior analysis of this data without the use of REGALS is published.

   - *Note:* The data were provided by the Ando group at Cornell University, and were originally published in the paper: Meisburger, S. P. et al. (2016). *Domain Movements upon Activation of Phenylalanine Hydroxylase Characterized by Crystallography and Chromatography-Coupled Small-Angle X-ray Scattering*. J Am Chem Soc 138, 6506–6516. DOI: 10.1021/jacs.6b01563. It is also available from the REGALS github repository.

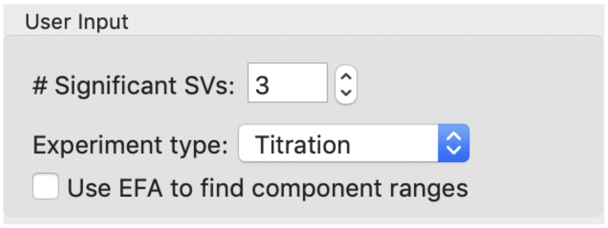


2. We will use REGALS to extract out the scattering of the individual conformers and the aggregate in the titration series. Right click on the **pheh_titration.hdf5** item in the Series list. Select the “REGALS” option.

3. Normally we use the plots of singular values and autocorrelations to determine the number of significant singular values (i.e. scattering components) in the dataset. Based on the plots of this titration series, there are ~4-5 significant singular values (SVs), for reference RAW’s automated method found 4 such values. However, prior knowledge indicates there are only two conformations we care about, and we suspect the other components are various aggregates. So we will fit the data to three components: each conformation and a single aggregate scattering component. Set the “# Significant SVs” to 3.

4. Set the experiment type to “Titration”. This affects the default settings later in the deconvolution.

5. Because we have so few data points, and we’re using fewer components than the actual number of significant SVs, trying to find the component ranges using the EFA plots will not be useful. Uncheck the “Use EFA to find component ranges” box.


User Input

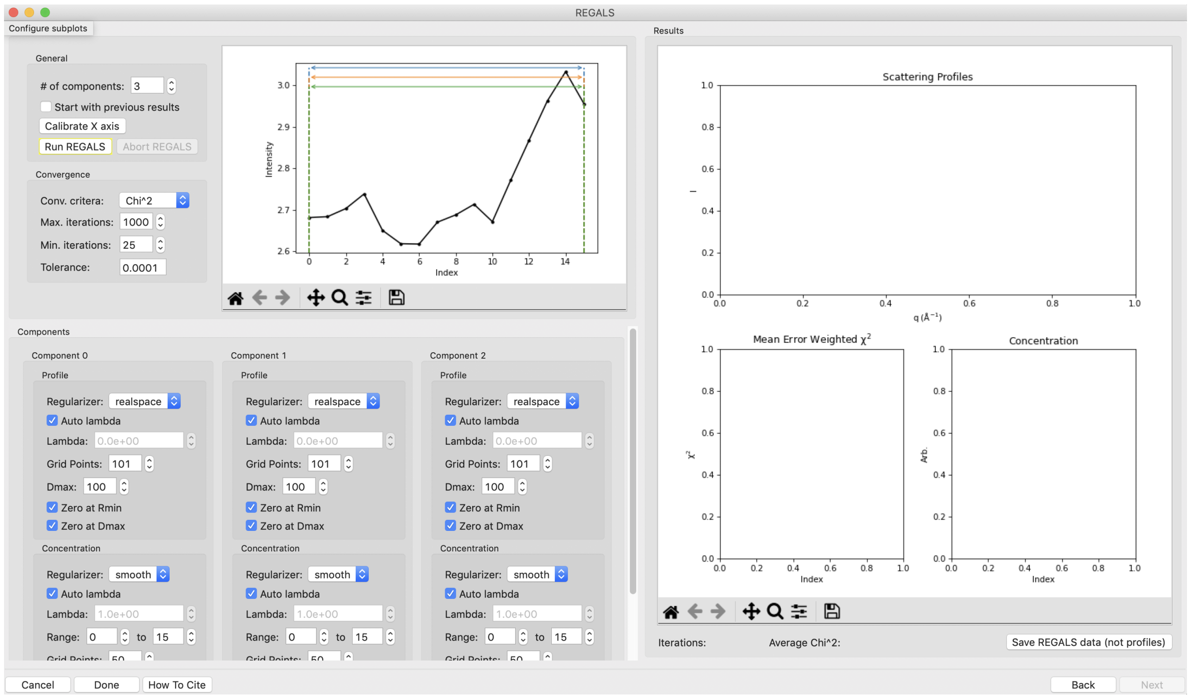
Significant SVs: 3

Experiment type: Titration

☐ Use EFA to find component ranges

6. Click the “Next” button in the lower right-hand corner of the window to advance to the final stage of the REGALS analysis.

   - *Note:* In this case, since we aren’t determining the component ranges with EFA, REGALS is not automatically run initially. We will have to set the component settings, then run REGALS.



General

# of components: 3

☐ Start with previous results

Convergence

Conv. critera: Chi^2

Max. iterations: 1000

Min. iterations: 25

Tolerance: 0.0001

Components

Component 0, Component 1, Component 2 (each):

Profile

Regularizer: realspace

✓ Auto lambda

Lambda: 0.0e+00

Grid Points: 101

Dmax: 100

✓ Zero at Rmin

✓ Zero at Dmax

Concentration

Regularizer: smooth

✓ Auto lambda

Lambda: 1.0e+00

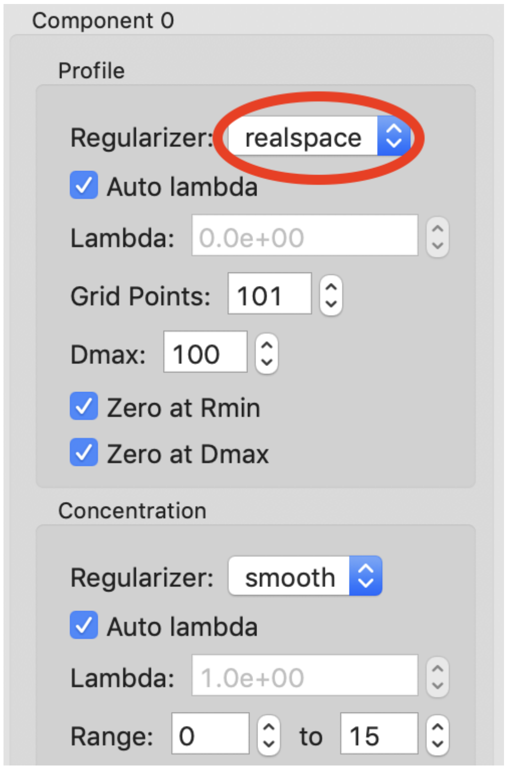
Range: 0 to 15

Grid Points: 50

Iterations:

Average Chi^2:

7. Notice that the profile part of each component has a "realspace" regularizer. This means that instead of constraining the data in q space via the scattering profile we will be constraining the data in real space using the P(r) function. So in addition to setting component ranges for the concentration and the lambda values, we will also set the D_{max} value for the P(r) function.



Component 0

Profile

Regularizer: realspace

✓ Auto lambda

Lambda: 0.0e+00

Grid Points: 101

Dmax: 100

✓ Zero at Rmin

✓ Zero at Dmax

Concentration

Regularizer: smooth

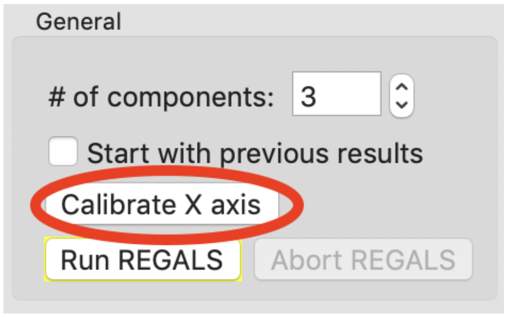
✓ Auto lambda

Lambda: 1.0e+00

Range: 0 to 15

8. The titration concentrations are not equally spaced from 0 to 80 mM, so we will calibrate the X axis appropri-

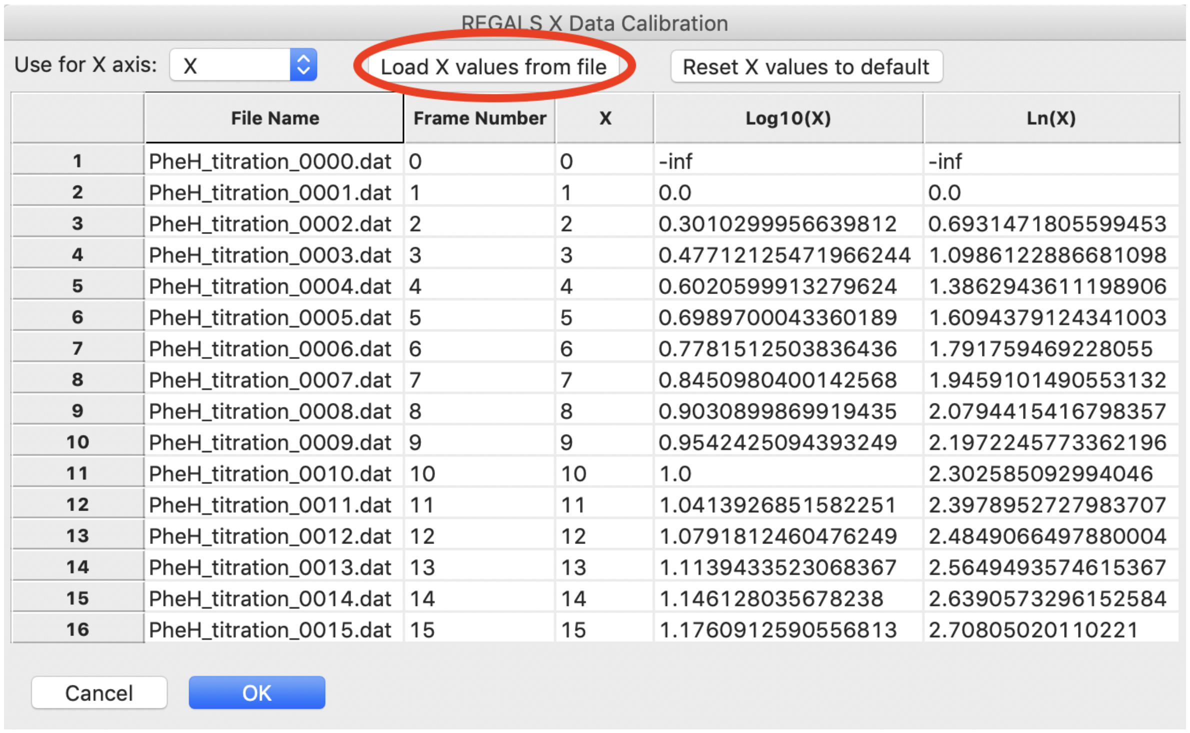
ately. Click the “Calibrate X axis” button.



# of components: 3

Start with previous results

9. In the window that opens click the “Load X values from file” button.



Use for X axis: X

	File Name	Frame Number	X	Log10(X)	Ln(X)
1	PheH_titration_0000.dat	0	0	-inf	-inf
2	PheH_titration_0001.dat	1	1	0.0	0.0
3	PheH_titration_0002.dat	2	2	0.3010299956639812	0.6931471805599453
4	PheH_titration_0003.dat	3	3	0.47712125471966244	1.0986122886681098
5	PheH_titration_0004.dat	4	4	0.6020599913279624	1.3862943611198906
6	PheH_titration_0005.dat	5	5	0.6989700043360189	1.6094379124341003
7	PheH_titration_0006.dat	6	6	0.7781512503836436	1.791759469228055
8	PheH_titration_0007.dat	7	7	0.8450980400142568	1.9459101490553132
9	PheH_titration_0008.dat	8	8	0.9030899869919435	2.0794415416798357
10	PheH_titration_0009.dat	9	9	0.9542425094393249	2.1972245773362196
11	PheH_titration_0010.dat	10	10	1.0	2.302585092994046
12	PheH_titration_0011.dat	11	11	1.0413926851582251	2.3978952727983707
13	PheH_titration_0012.dat	12	12	1.0791812460476249	2.4849066497880004
14	PheH_titration_0013.dat	13	13	1.1139433523068367	2.5649493574615367
15	PheH_titration_0014.dat	14	14	1.146128035678238	2.6390573296152584
16	PheH_titration_0015.dat	15	15	1.1760912590556813	2.70805020110221

10. Select the **pheh_titration_conc.txt** file in the **series_data** folder and load it in.

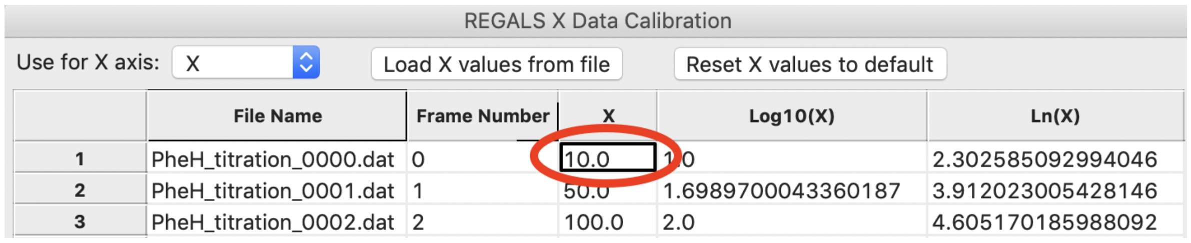
    - *Note:* A calibration file should consist of a single column of the concentration values, in the order that the profiles were loaded into the series. So the first line of the file has the concentration for the first profile in the series, and so on.

11. Upon loading you should see the values in the X column update.

    - *Note:* The concentrations are in uM, so you will see concentrations from 0 to 80000.

12. The concentration values are not linearly spaced, as is typical of a titration series. It is also most common to work with titration series on a logarithmic concentration scale. To do this, we need to define the first concentration as not zero (as log(0) is undefined). For this data we will set the concentration to 10 uM. Double click in the first box in the X column and enter 10.0.

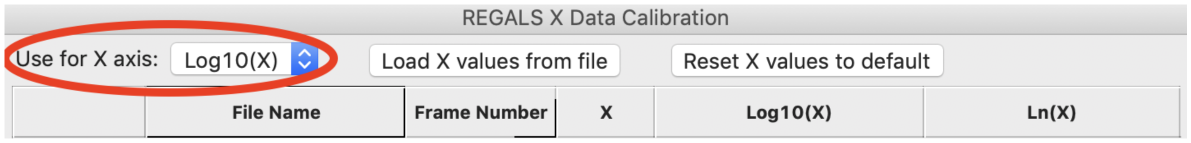
    - *Note:* You can enter all the concentrations manually if you want, you don’t have to create the concentration file we loaded in the previous steps. However, for a long series this can get a bit tedious.



Use for X axis: X

	File Name	Frame Number	X	Log10(X)	Ln(X)
1	PheH_titration_0000.dat	0	10.0	1.0	2.302585092994046
2	PheH_titration_0001.dat	1	50.0	1.6989700043360187	3.912023005428146
3	PheH_titration_0002.dat	2	100.0	2.0	4.605170185988092

13. As we want to work with our data in log space, select “Log10(X)” in the “Use for X axis” list. Then click the OK button to exit the window and save the X calibration.



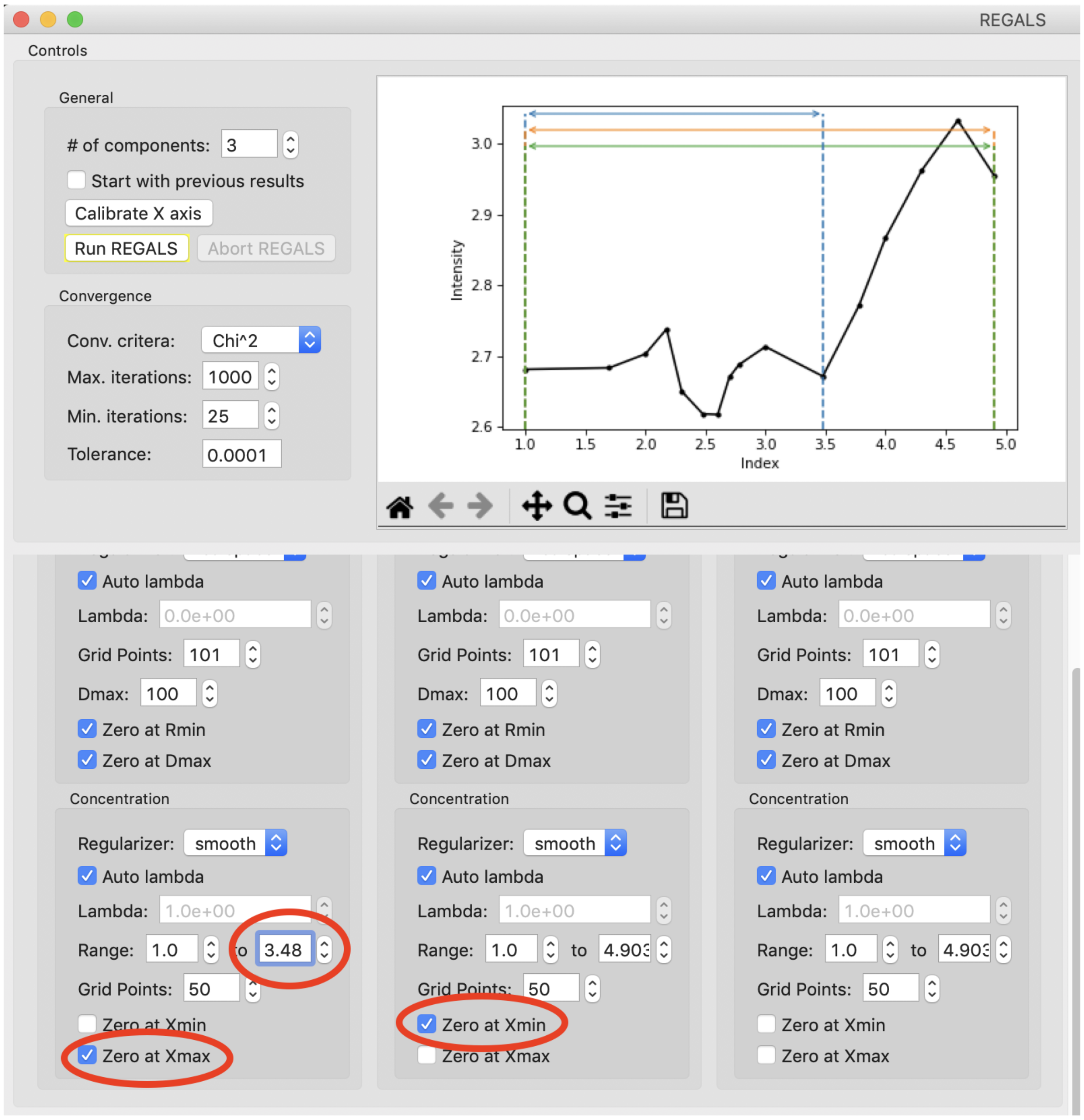
Use for X axis: Log10(X)

	File Name	Frame Number	X	Log10(X)	Ln(X)

14. In the main REGALS window, notice that the X axis is now calibrated as Log(concentration), and that the ranges for the concentration components have updated accordingly.

15. Next we will set the ranges for our components. We will use components 0 and 1 as the different conformations, resting and active respectively, and component 2 as the aggregate. Prior analysis of the system showed that of the two conformations, only the resting conformation, component 0 is present at 0 mM L-phe. So set the “Zero at Xmin” to True for component 1.

16. Prior analysis showed that features of the scattering profile associated with the active state saturated above 3 mM. So we will assume that there is no contribution to the scattering from the resting state at/above 3 mM (3.48 on the log10(X) axis). Set the end range for component 0 to 3.48, and apply a Zero at Xmax boundary condition to it.



# of components: 3

Start with previous results: unchecked

Calibrate X axis

Conv. critera: Chi^2

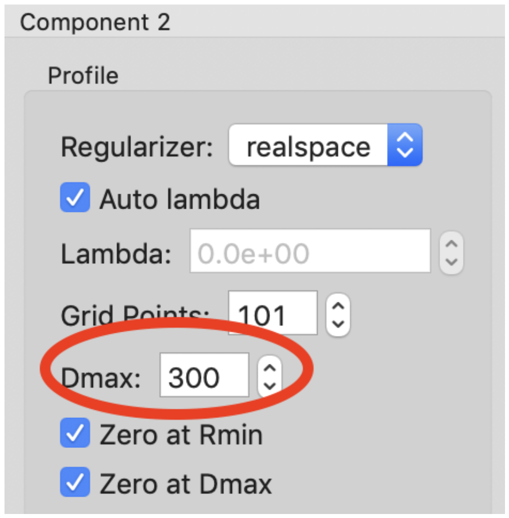
Max. iterations: 1000

Min. iterations: 25

Tolerance: 0.0001

✓ Auto lambda	✓ Auto lambda	✓ Auto lambda
Lambda: 0.0e+00	Lambda: 0.0e+00	Lambda: 0.0e+00
Grid Points: 101	Grid Points: 101	Grid Points: 101
Dmax: 100	Dmax: 100	Dmax: 100
✓ Zero at Rmin	✓ Zero at Rmin	✓ Zero at Rmin
✓ Zero at Dmax	✓ Zero at Dmax	✓ Zero at Dmax
Concentration	Concentration	Concentration
Regularizer: smooth	Regularizer: smooth	Regularizer: smooth
✓ Auto lambda	✓ Auto lambda	✓ Auto lambda
Lambda: 1.0e+00	Lambda: 1.0e+00	Lambda: 1.0e+00
Range: 1.0 to 3.48	Range: 1.0 to 4.90	Range: 1.0 to 4.90
Grid Points: 50	Grid Points: 50	Grid Points: 50
☐ Zero at Xmin	✓ Zero at Xmin	☐ Zero at Xmin
✓ Zero at Xmax	☐ Zero at Xmax	☐ Zero at Xmax

17. We will apply not constraints to the concentration range of the aggregate, so we are done with the concentration ranges. Next we need to set the D$_{max}$ values for the components. We'll start with the aggregate. Since we have no prior knowledge about the aggregate, we will assume it is a non-specific size. Because of the q range of the data, the largest dimension of an object that can be measured is ~300 Å, based on the Shannon limit of $D_{max} < \pi/q_{min}$. So we will set the D$_{max}$ value for the aggregate, component 2, to 300.



Component 2

Profile

Regularizer: realspace

☑ Auto lambda

Lambda: 0.0e+00

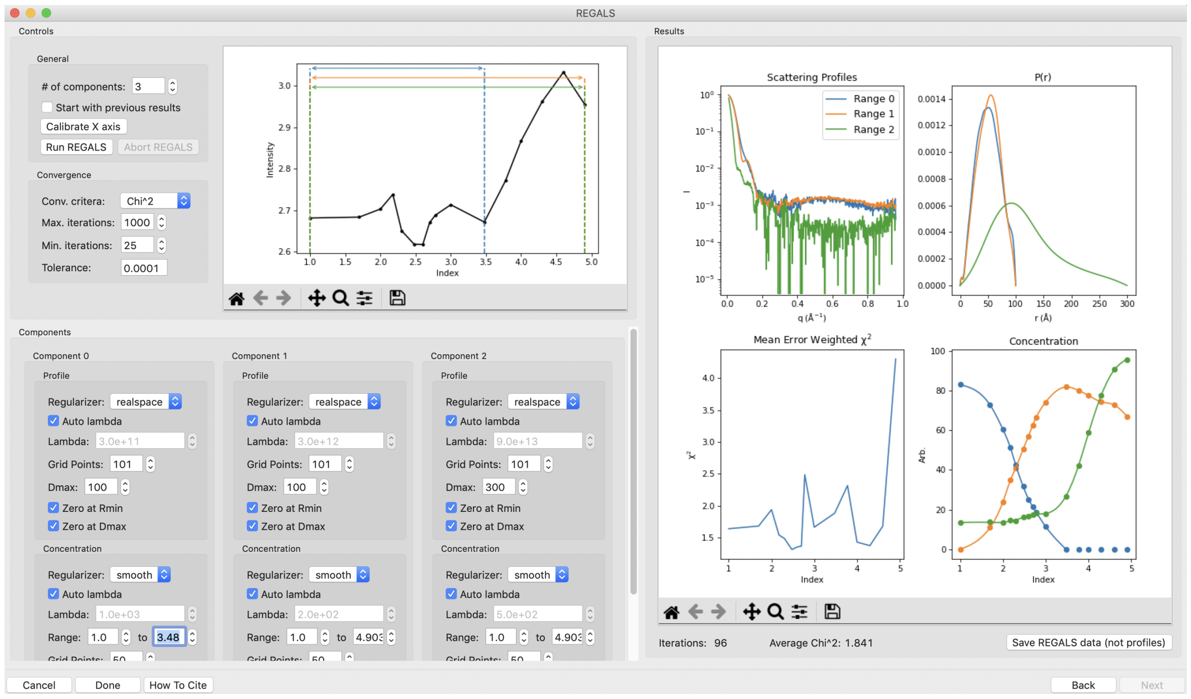
Grid Points: 101

Dmax: 300

☑ Zero at Rmin

☑ Zero at Dmax

18. We will now run REGALS to get an initial look at the deconvolution. Click the "Run REGALS" button.



REGALS

Controls

General

# of components: 3

☐ Start with previous results

Calibrate X axis

Run REGALS

Abort REGALS

Convergence

Conv. critera: Chi^2

Max. iterations: 1000

Min. iterations: 25

Tolerance: 0.0001

Components

Component 0	Component 1	Component 2
Profile	Profile	Profile
Regularizer: realspace	Regularizer: realspace	Regularizer: realspace
☑ Auto lambda	☑ Auto lambda	☑ Auto lambda
Lambda: 3.0e+11	Lambda: 3.0e+12	Lambda: 9.0e+13
Grid Points: 101	Grid Points: 101	Grid Points: 101
Dmax: 100	Dmax: 100	Dmax: 300
☑ Zero at Rmin	☑ Zero at Rmin	☑ Zero at Rmin
☑ Zero at Dmax	☑ Zero at Dmax	☑ Zero at Dmax
Concentration	Concentration	Concentration
Regularizer: smooth	Regularizer: smooth	Regularizer: smooth
☑ Auto lambda	☑ Auto lambda	☑ Auto lambda
Lambda: 1.0e+03	Lambda: 2.0e+02	Lambda: 5.0e+02
Range: 1.0 to 3.48	Range: 1.0 to 4.903	Range: 1.0 to 4.903

Results

Iterations: 96 Average Chi^2: 1.841

Save REGALS data (not profiles)

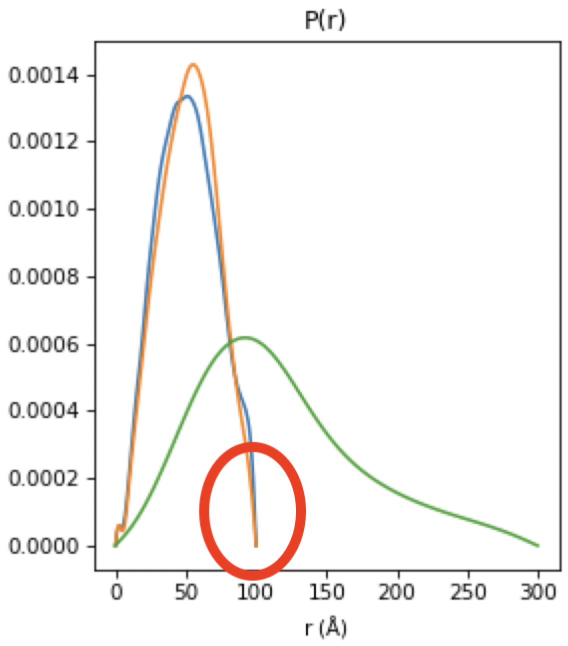
Cancel Done How To Cite Back Next

19. Looking at the results, it's already a reasonable deconvolution. The concentrations make sense with previous knowledge, e.g. that there's a transition from the resting (component 0) to active (component 1) with increasing L-phe concentration, and that there's a low level of aggregate throughout that increases significantly at higher concentrations of L-phe. The profiles and P(r) functions, while not completely correct, are at least reasonable shapes, and the chi^2 is mostly relatively low.

    - *Note:* On the concentration plot, the markers are the concentrations calculated at the titration points. The smooth lines are the concentration calculated at the regularlizer grid points, so it is effectively interpolated between the measured titration points. When you have more than 40 profiles in the series, only the concentration at the actual measured points is shown, and there it is shown as a continuous line, not individual points (as in the above IEC-SAXS example).

20. Next we will refine the D_{max} values for the resting and active states. On the P(r) plot, notice that for both components the P(r) function is forced to zero sharply, indicating that an underestimated D_{max} value. Increase

both D_{max} values to 110 and run REGALS again.



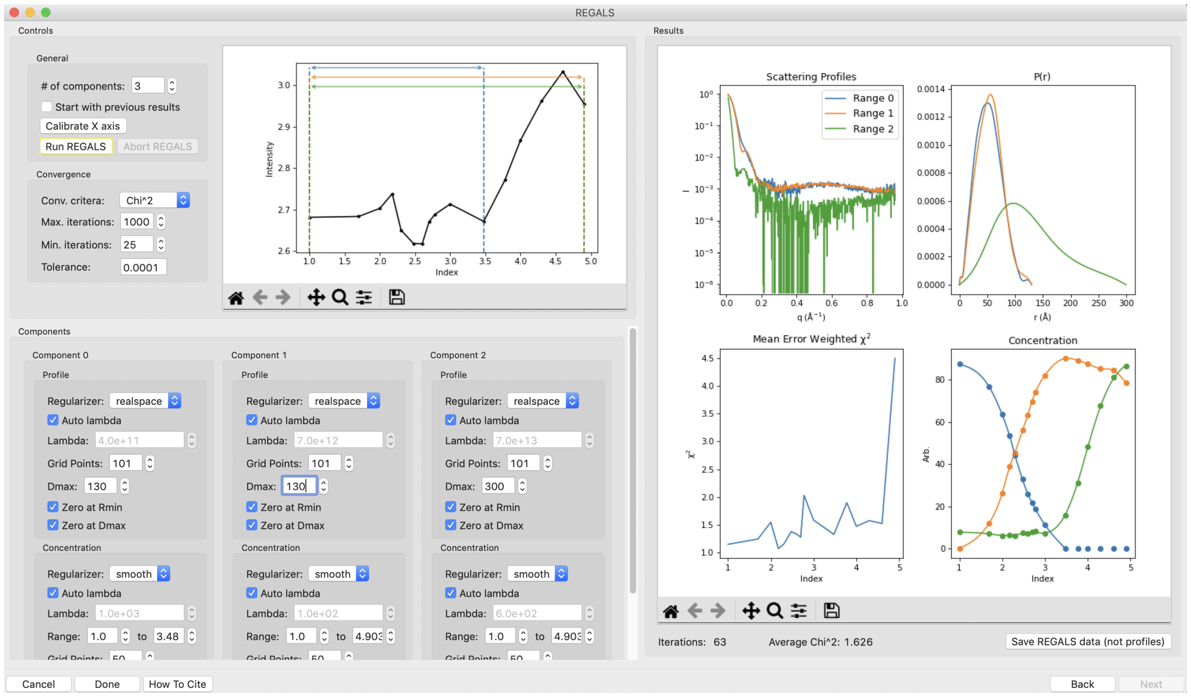
21. Notice that at a D_{max} of 110 the P(r) functions are still somewhat forced to zero, but the chi^2 is lower, so this has improved the deconvolution. Continue increasing D_{max} in steps of 10-20 until you reach 160. You should notice several things. First, there's a range from ~130-150 where the chi^2 is relatively stable, indicating D_{max} values in those ranges all provide relatively good fits to the data. Second, as you increase D_{max} the concentration of the aggregate decreases, particularly in the lower titration concentrations. This implies that having a larger D_{max} is letting that component take up some of the aggregate scattering. Third, after ~120 the P(r) functions stop looking as forced to zero. Fourth at 160 the chi^2 starts noticeably increasing, indicating that's too large for the D_{max} value. Based on this, we want to pick a D_{max} value near 130, to exclude as much of the aggregate as we can while still getting good P(r) functions and chi^2 values.

22. Set the D_{max} of both components to 130 and run REGALS.

    - *Note:* The D_{max} values found by previous analysis was ~130, so this validates our choice of D_{max}.
    - *Note:* The D_{max} values for different components won't necessarily agree. In this case they happen to.
    - *Tip:* If it takes a while to run REGALS every time you change a component, you can speed up the convergence by starting with the previous results. To do so, you would check the "Start with previous results" box. Then change the convergence criteria to "Iterations" and set the number of iterations to 10. This will allow you to quickly iterate on changes like D_{max}, as long as the magnitude of the change is relatively small. Just be sure to set the convergence criteria back to the default (not using previous results, and Chi^2 with 1000 iterations) to do your final REGALS run.

23. The REGALS deconvolution is now as optimized as we can make it. Notice that there's still a relatively high chi^2 value for the last frame. This indicates that the aggregate may be changing shape as the amount increases, and so a single component cannot fit all the data well. You can redo the deconvolution without the last data point if you want, the results are very similar albeit without the chi^2 spike for the last point. Since we're satisfied with these results we can now save them. First, click on "Save REGALS data (not profile)" and save.

    - *Note:* Among other things, this .csv file contains the P(r) functions, and the smoothed concentration curves (lines on the concentration plot). In this case the smoothed concentration curves are particularly useful because the protein changes shape but not size. As the P(r) functions are normalized to I(0), the concentration curves for both components are on the same overall scale. This means they differ by simply a uniform scale factor from the true concentrations (e.g. in mg/ml), and so could be useful to help characterize the two state transition. This will not always be true, such as if you're characterizing an oligomerization reaction.


REGALS

Controls

General

# of components: 3

Start with previous results

Calibrate X axis

Run REGALS | Abort REGALS

Convergence

Conv. critera: Chi^2

Max. iterations: 1000

Min. iterations: 25

Tolerance: 0.0001

Components

	Component 0	Component 1	Component 2
Profile			
Regularizer:	realspace	realspace	realspace
Auto lambda	✓	✓	✓
Lambda:	4.0e+11	7.0e+12	7.0e+13
Grid Points:	101	101	101
Dmax:	130	130	300
Zero at Rmin	✓	✓	✓
Zero at Dmax	✓	✓	✓
Concentration			
Regularizer:	smooth	smooth	smooth
Auto lambda	✓	✓	✓
Lambda:	1.0e+03	1.0e+02	6.0e+02
Range:	1.0 to 3.48	1.0 to 4.903	1.0 to 4.903
Grid Points:	50	50	50

Results

Iterations: 63 Average Chi^2: 1.626

Save REGALS data (not profiles)

Cancel | Done | How To Cite | Back | Next

24. Finally, click the "Done" button to close the REGALS window and send the deconvolved scattering profiles to the profiles plot.

General notes

1. It is okay to mix and match different types of regularizers (e.g. have both smooth and real space profile regularizers) for the same series.
2. You can change the regularizers away from the defaults for a given experiment type.
3. The REGALS examples shown here were chosen to demonstrate the features of the GUI. REGALS is not restricted in application to just IEC-SAXS and titration data. It has been successfully applied to time resolved SAXS data (similar to the titration series example), and we expect it will be applied to a range of other types of experiments as well.
4. As you saw in the PheH titration series example, it is quite useful, and sometimes necessary, to have additional information to input to the deconvolution, such as the range of the components, or a known maximum dimension. One of the advantages of REGALS is that it can incorporate these additional pieces of information to improve the deconvolution.

Advanced SEC-SAXS processing – Baseline correction

Sometimes SEC data shows a baseline drift. This can be due either to instrumental changes (such as beam drift), or changes in the measured system, such as capillary fouling. RAW provides the ability to correct for these forms of baseline drift using either a linear or integral baseline method. The linear baseline method is best for instrumental drifts, while the integral baseline method is best for capillary fouling. Both baseline methods apply a distinct correction for each q value.

If you use integral baseline correction in RAW, in addition to citing the RAW paper, please cite this paper: E. Brookes, P. Vachette, M. Rocco, and J. Pérez. Journal of Applied Crystallography (2016). 49, 1827-1841. DOI:

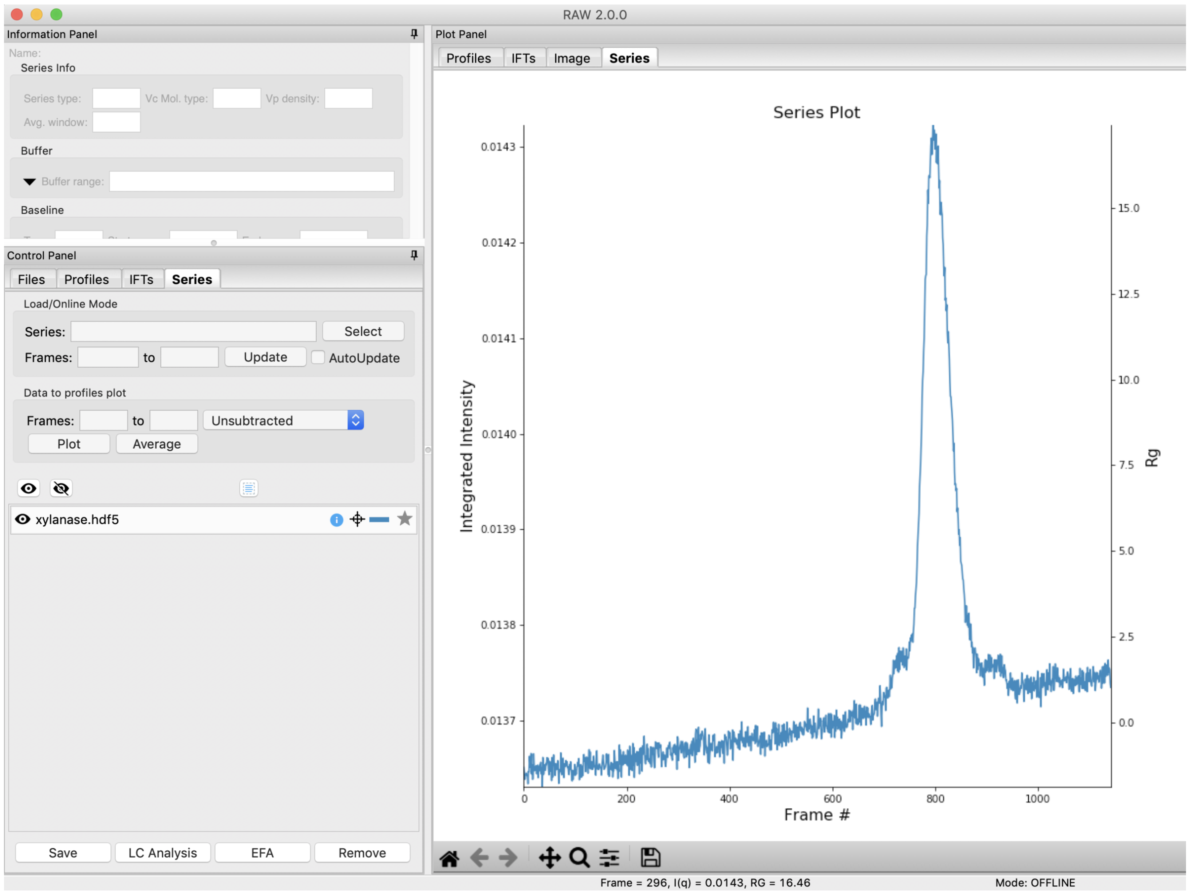
10.1107/S1600576716011201

A video version of this tutorial is available:

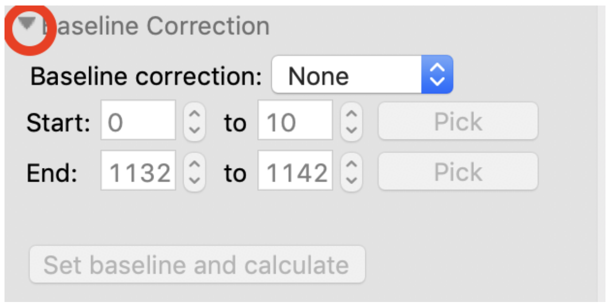
The written version of the tutorial follows.

Linear Baseline Correction

1. Clear all of the data in RAW. Load the **xylanase.hdf5** SEC data in the **series_data** folder.
2. When it loads in, you will see there is a distinct constant upward slope in the integrated intensity. This usually happens due to instrumental drift, and can often be mostly corrected for.



3. Open the LC Series analysis panel. Set a buffer range ('Auto' is fine). You'll still see the slope in the subtracted integrated intensity plot.
   - *Note:* To baseline correct data, you should only have buffer regions selected before the peak.
4. Use the triangle to expand the Baseline Correction section.
   - *Note:* On windows it will look like a button with '>>' after the name.


Baseline Correction

Baseline correction: None

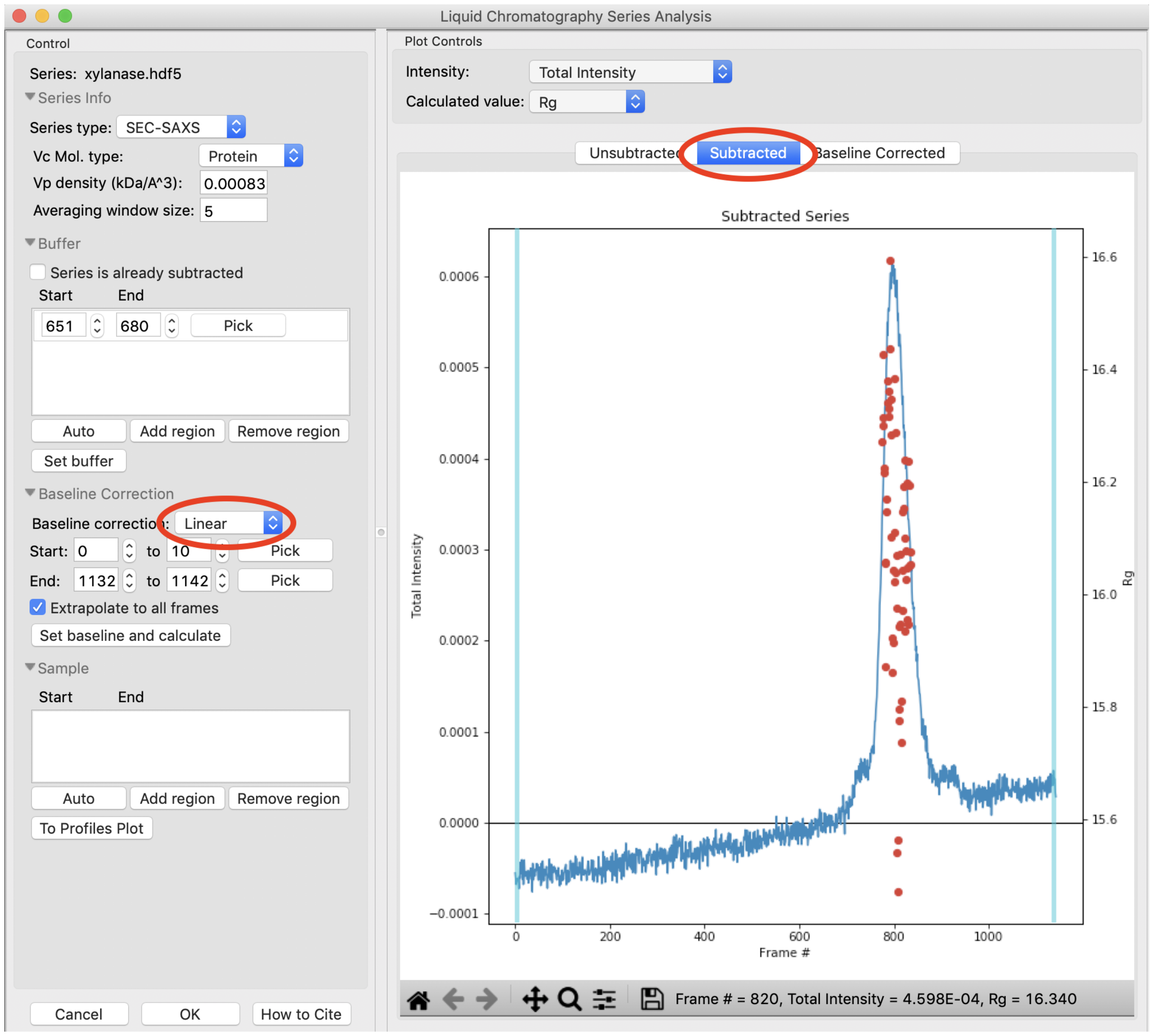
Start: 0 to 10 Pick

End: 1132 to 1142 Pick

Set baseline and calculate

5. Set baseline correction to 'Linear'. You should see the start and end controls become active, and two blue regions representing your selected start and end regions appear on the 'Subtracted' plot.

   - *Note:* This should automatically show the 'Subtracted' plot tab. If it doesn't, select the 'Subtracted' plot tab.


Liquid Chromatography Series Analysis

Control

Series: xylanase.hdf5

Series Info

Series type: SEC-SAXS

Vc Mol. type: Protein

Vp density (kDa/A^3): 0.00083

Averaging window size: 5

Buffer

Series is already subtracted

Start	End	
651	680	Pick

Auto · Add region · Remove region

Set buffer

Baseline Correction

Baseline correction: Linear

Start: 0 to 10 Pick

End: 1132 to 1142 Pick

✓ Extrapolate to all frames

Set baseline and calculate

Sample

Start End

Auto · Add region · Remove region

To Profiles Plot

Cancel · OK · How to Cite

Plot Controls

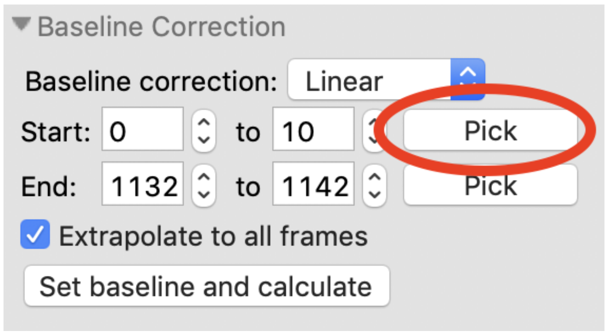
Intensity: Total Intensity

Calculated value: Rg

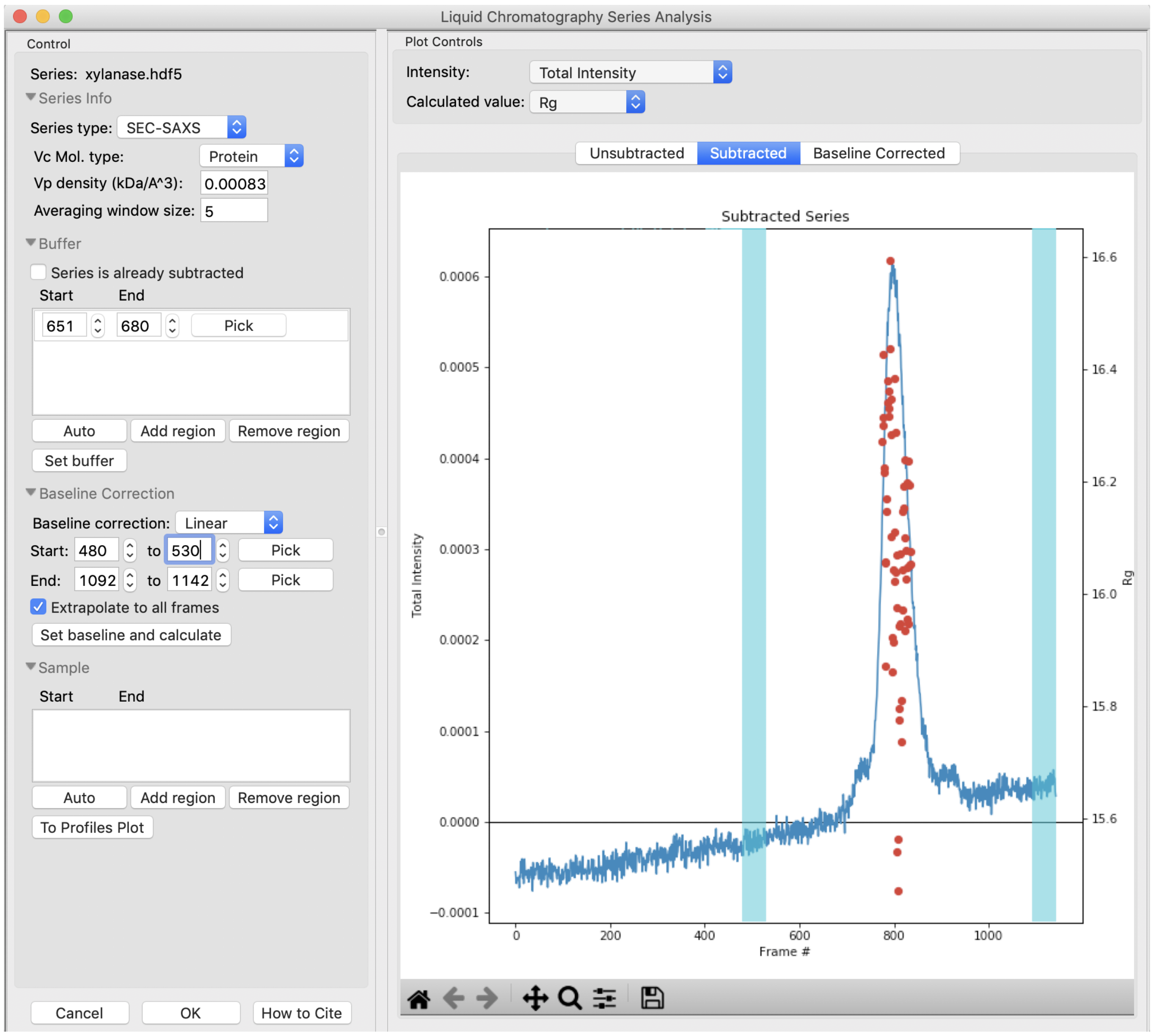
Unsubtracted · Subtracted · Baseline Corrected

Frame # = 820, Total Intensity = 4.598E-04, Rg = 16.340

6. If you look closely, it looks like the baseline may level off a little bit at the start of the series curve. So we will select a start range closer to the peak. Click the 'Pick' button for the baseline correction start region and select a start region about halfway to the peak. Roughly 30-50 frames is a good length for the start and end regions.


Baseline Correction
Baseline correction: Linear
Start: 0 to 10 Pick
End: 1132 to 1142 Pick
☑ Extrapolate to all frames
Set baseline and calculate

7. Expand the end range to be 50 frames long.

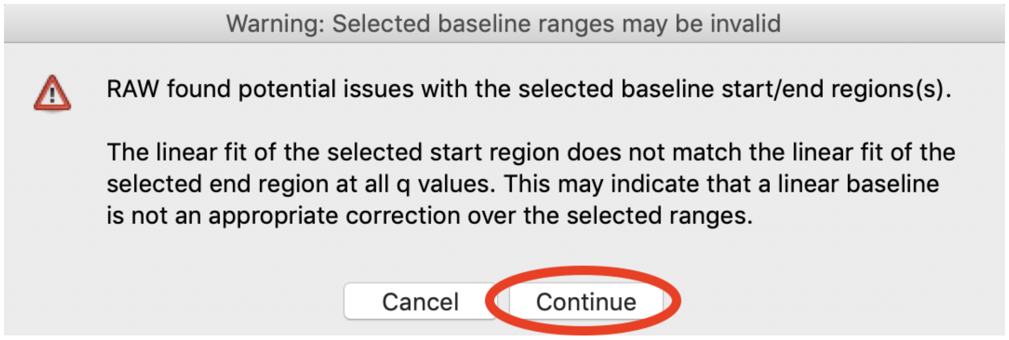

Liquid Chromatography Series Analysis

Control
Series: xylanase.hdf5
Series Info
Series type: SEC-SAXS
Vc Mol. type: Protein
Vp density (kDa/A^3): 0.00083
Averaging window size: 5
Buffer
☐ Series is already subtracted
Start End
651 680 Pick
Auto | Add region | Remove region
Set buffer
Baseline Correction
Baseline correction: Linear
Start: 480 to 530 Pick
End: 1092 to 1142 Pick
☑ Extrapolate to all frames
Set baseline and calculate
Sample
Start End
Auto | Add region | Remove region
To Profiles Plot
Cancel | OK | How to Cite

Plot Controls
Intensity: Total Intensity
Calculated value: Rg
Unsubtracted | Subtracted | Baseline Corrected

8. Click ‘Set baseline and calculate’. You may see a warning, click ‘Continue’.

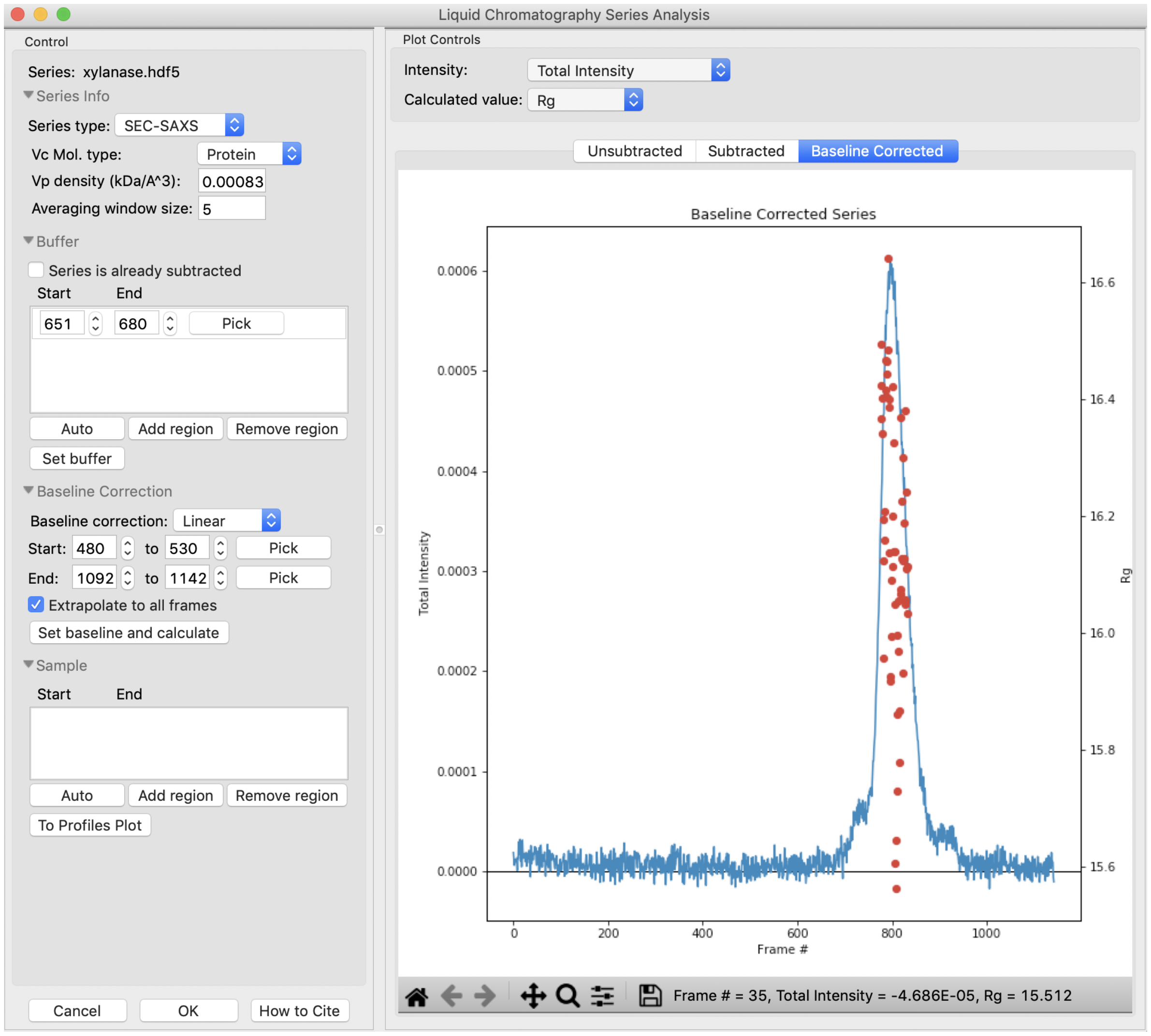
   • *Note:* The warning simply informs you that the slope of the linear correction is not the same in the start and end region at all q values. This is usually the case, and mostly can be ignored.


Warning: Selected baseline ranges may be invalid

RAW found potential issues with the selected baseline start/end regions(s).

The linear fit of the selected start region does not match the linear fit of the selected end region at all q values. This may indicate that a linear baseline is not an appropriate correction over the selected ranges.

9. The 'Baseline Corrected' plot should automatically show after you set the baseline region and calculate. If not, change to that plot tab. You can see that the upward drift is essentially gone.


Liquid Chromatography Series Analysis

Control

Series: xylanase.hdf5

Series type: SEC-SAXS
Vc Mol. type: Protein
Vp density (kDa/A^3): 0.00083
Averaging window size: 5

Buffer

Series is already subtracted

Start	End
651	680

Baseline Correction

Baseline correction: Linear
Start: 480 to 530
End: 1092 to 1142
Extrapolate to all frames ✓

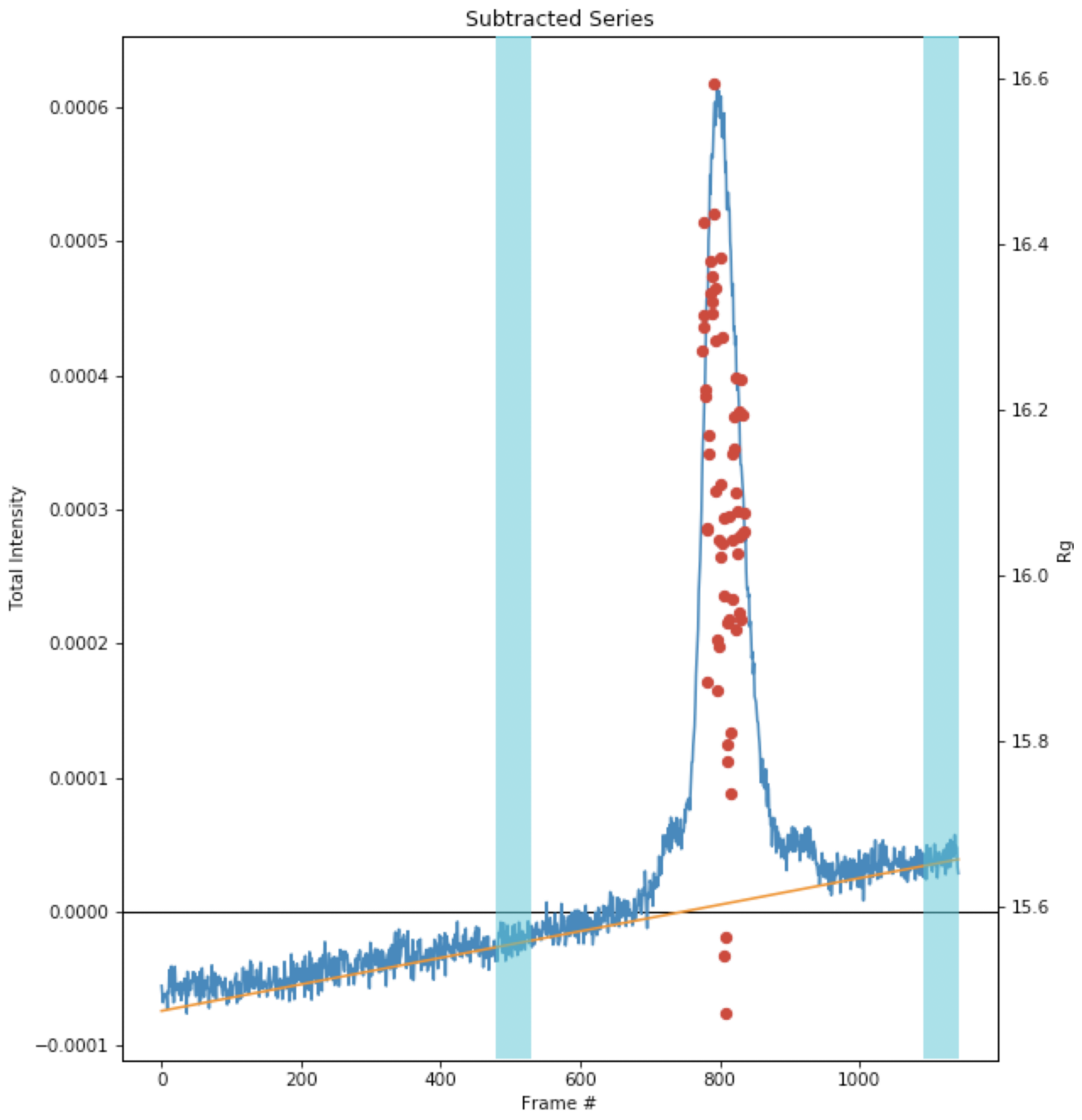
Sample

Plot Controls

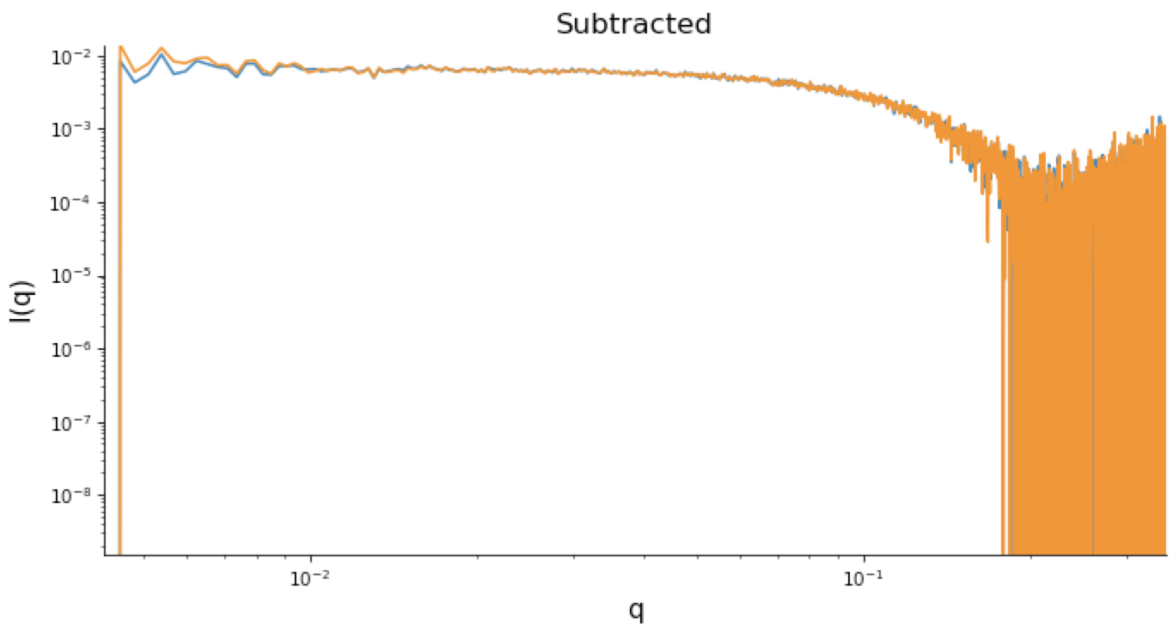
Intensity: Total Intensity
Calculated value: Rg

Frame # = 35, Total Intensity = -4.686E-05, Rg = 15.512

10. Switch back to the subtracted plot. You'll see the calculated baseline shown in orange.



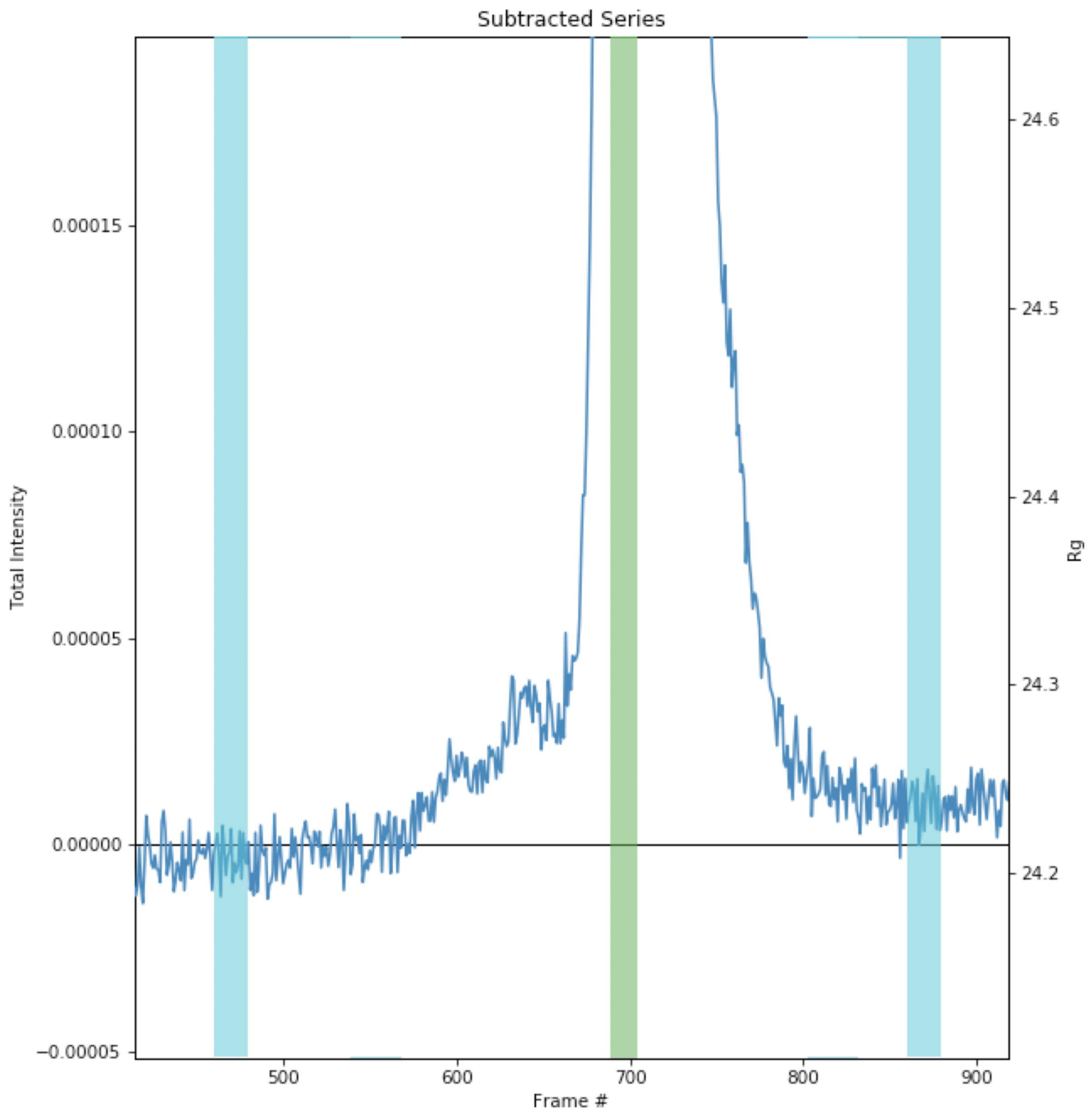
11. Switch back to displaying the Baseline Corrected plot.
12. Remove any existing sample region and find a new sample region using the 'Auto' button. Send that region to the Profiles plot.
13. You can remove the baseline correction by changing the 'Baseline correction' selection from 'Linear' to 'None'. Do this, then send the sample region to the Profiles plot again.
14. Change the 'Baseline correction' selection back to 'Linear'. Click 'Set Baseline and calculate' to redo the linear correction. Click 'OK' to exit the LC Series Analysis window. Now if you save series or reopen the window you will see your baseline correction.
15. Switch to the profiles plot. Put the subtracted plot on a Log-Log scale. You can see a difference in the profiles due to the baseline correction at low q.



Integral Baseline Correction

Integral baseline correction proceeds very similarly to linear baseline correction. Here we provide detail only where the procedures are different.

1. Clear all of the data in RAW. Load the **baseline.hdf5** SEC data in the **series_data** folder.
   - *Note:* This is the same as what you *previously saved* in an earlier part of the tutorial.
2. Open the LC Series analysis panel. Verify that your buffer regions are before the peak of interest.
3. Set baseline correction to 'Integral'.
4. Zoom in near the base of the subtracted peak. Pick a start region in the flat baseline area just before the start of the peaks.
5. Pick an end region in the flat region just after the end of the peaks.
   - *Note:* You should end up with regions ~460-480 and 860-880
   - *Try:* You can use the 'Auto' button to automatically find start and end regions. For this dataset, it ends up a little close to the peaks, so some manual adjustment is necessary.

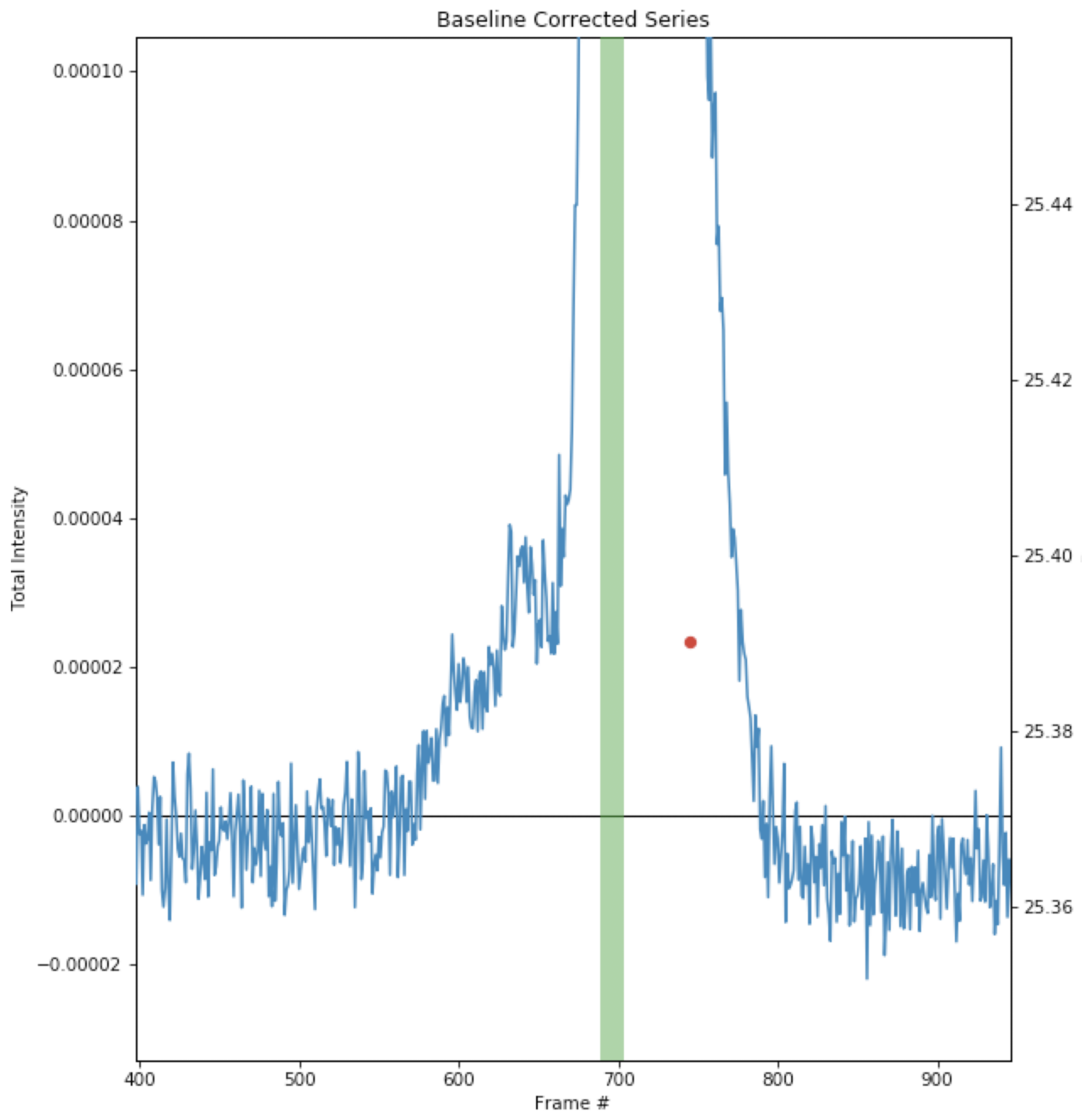


6. Click the 'Set baseline and calculate' button.

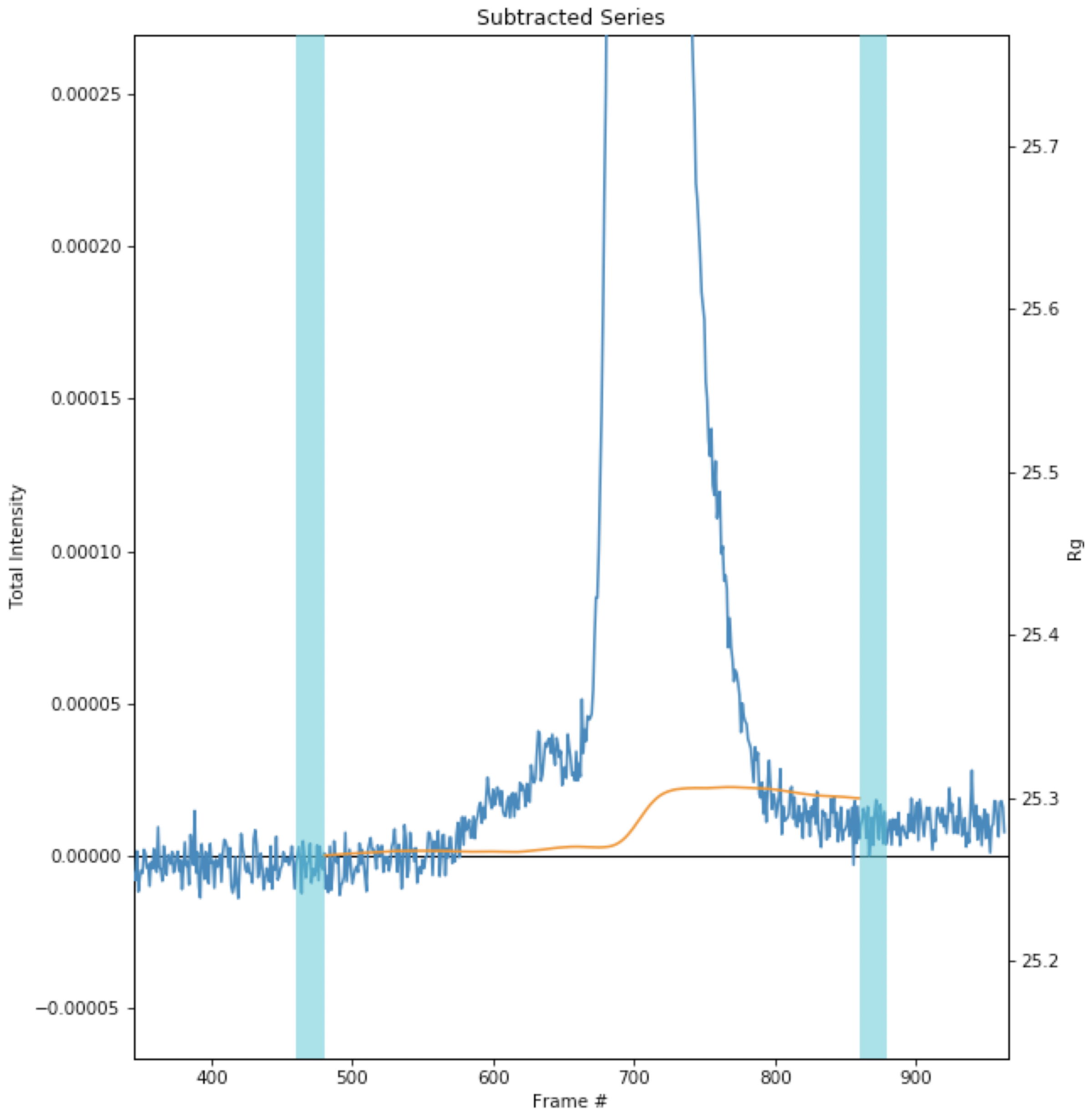
   - *Note:* The start and end points should be set in regions with no change in the baseline. If they aren't, RAW will give a warning. Try changing the end region to ~800-820 to see such a warning.

7. Zoom in on the base of the peak in the baseline corrected dataset. You should see that the baseline is actually a little overcorrected. This is because the integral baseline correction only allows for positive or no change in the baseline, so if some q values need a negative correction the total baseline ends up overcorrected, as the positive values are brought down but the negative values are not brought up.

   - *Note:* You can change the intensity display to individual q values or a q range and look at different points in q to try figure out which q values are causing the issue.



8. Switch back to the subtracted plot and zoom in on the base of the peak. You'll see the calculated baseline shown in orange.



9. Change the intensity display to 'Intensity in q range' and try several different q ranges. This will allow you to see what q values are getting the baseline overcompensated.
   - *Try:* Recommended regions to try for this dataset are 0.01-0.02, 0.05-0.06, 0.1-0.2, 0.2-0.27.
   - *Note:* You should find that it is the high *q* ranges that are being overcorrected. This may imply that the profiles are mostly noise in that range. If you examine the profiles and determine that is the case, you could truncate the profiles to lower *q* before doing the baseline correction.
10. Switch back to displaying the total intensity and the Baseline Corrected plot.
11. Remove any existing sample region and find a new sample region using the "Auto" button. Send that sample region to the main plot.

5.3.4 Creating a configuration file

This section will guide you through creating a configuration file for RAW that allows you to integrate 2D images into 1D scattering profiles. It refers to the *RAW tutorial data*.

Select a section below to view the tutorial, or use the next and back buttons at the bottom of the page to navigate through it in order.

Note: This tutorial assumes you are familiar with basic operations in RAW such as loading data. If that is not the case, please first do the RAW Tutorial Sections 1 and 2.

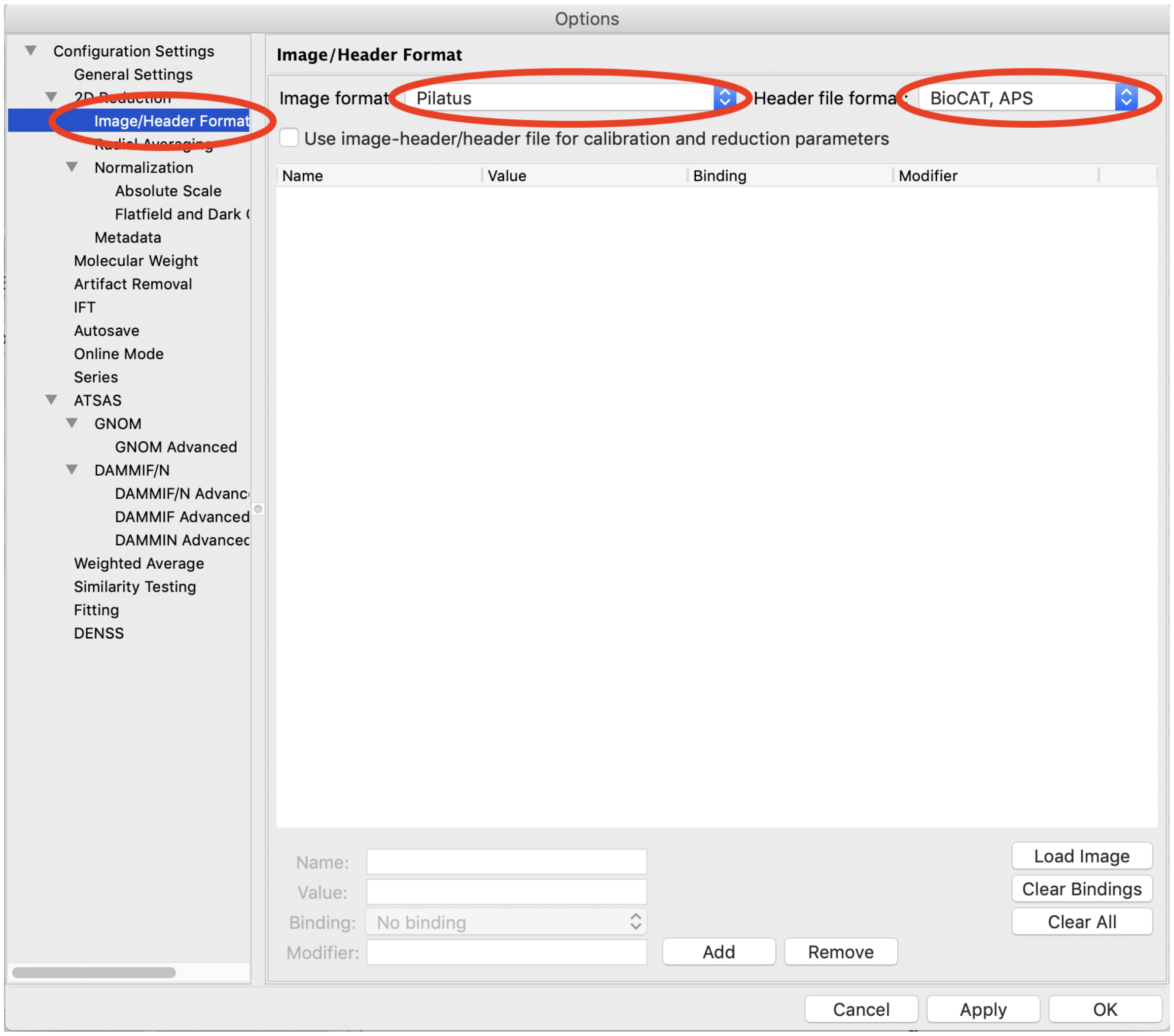
Masking

The first step in creating a calibraiton file is to to mask out unwanted portions of your image, such as the beamstop and bad detector pixels. Before this can be done, you have to set the image and file header type in the Options window.

A video version of this tutorial is available (please see the first part of the centering and calibration video for how to set the initial options):

The written version of the tutorial follows.

1. Open RAW. The install instructions contain information on installing and running RAW.
2. Open the Options window by going to the Options menu (top of the RAW window or in the system bar, depending on your OS) and selecting "Advanced Options"
3. In the options window, select the Image/Header Format section on the left.
4. In the area on the right, set the Image format drop-down menu to "Pilatus" and the Header file format to "BioCAT, APS".



Image/Header Format

Image format: Pilatus

Header file format: BioCAT, APS

☐ Use image-header/header file for calibration and reduction parameters

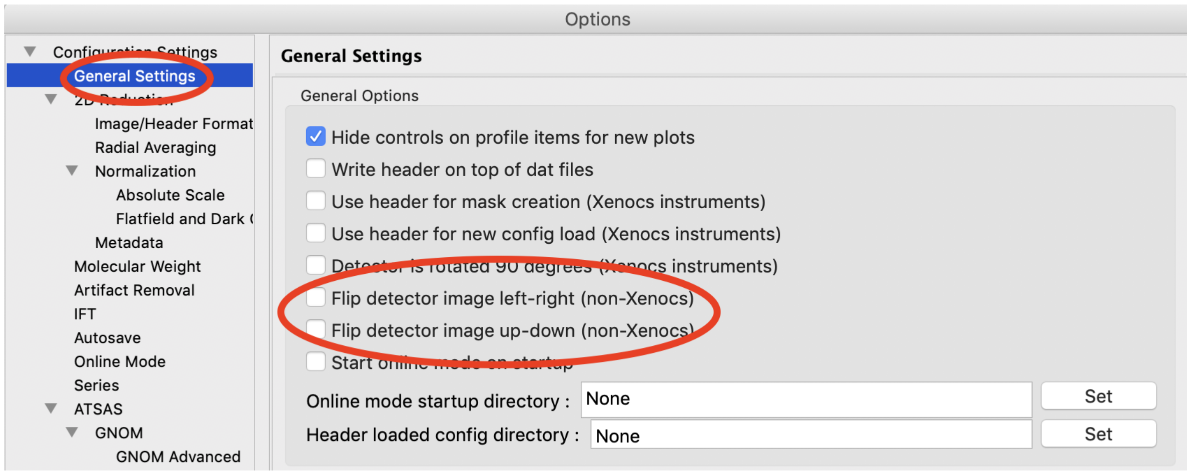
Name:

Value:

Binding: No binding

Modifier:

5. Go to the "General Settings" section. If necessary, uncheck both of the Flip Image checkboxes. These allow you to control the orientation of the displayed image, and make it match the physical layout of the instrument. For this beamline and image type, no modification is necessary.



General Settings

General Options

☑ Hide controls on profile items for new plots

☐ Write header on top of dat files

☐ Use header for mask creation (Xenocs instruments)

☐ Use header for new config load (Xenocs instruments)

☐ Detector is rotated 90 degrees (Xenocs instruments)

☐ Flip detector image left-right (non-Xenocs)

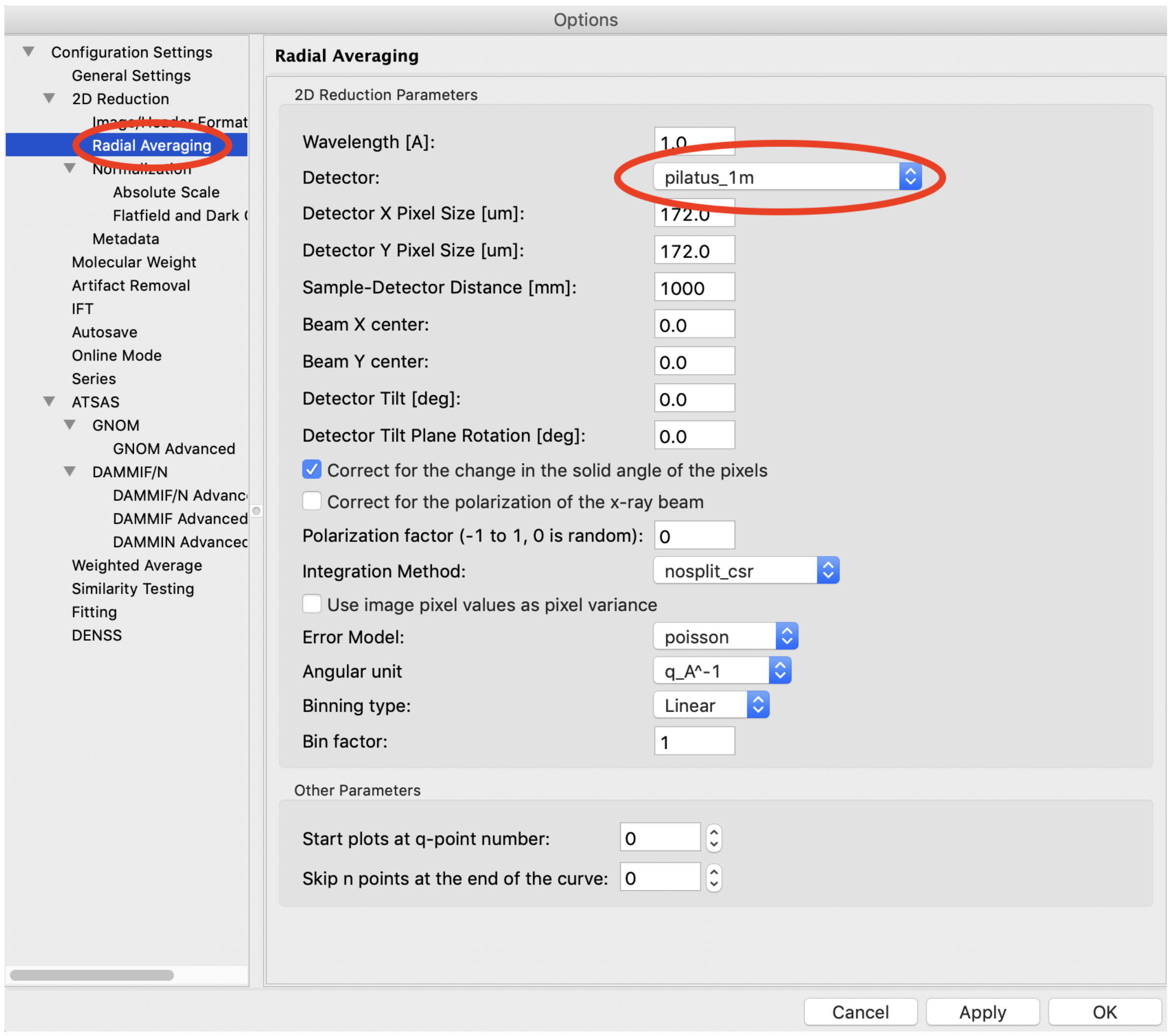
☐ Flip detector image up-down (non-Xenocs)

☐ Start online mode on startup

Online mode startup directory : None

Header loaded config directory : None

6. Go to the "Radial Averaging" section. Set the Detector to "pilatus_1m"



Radial Averaging

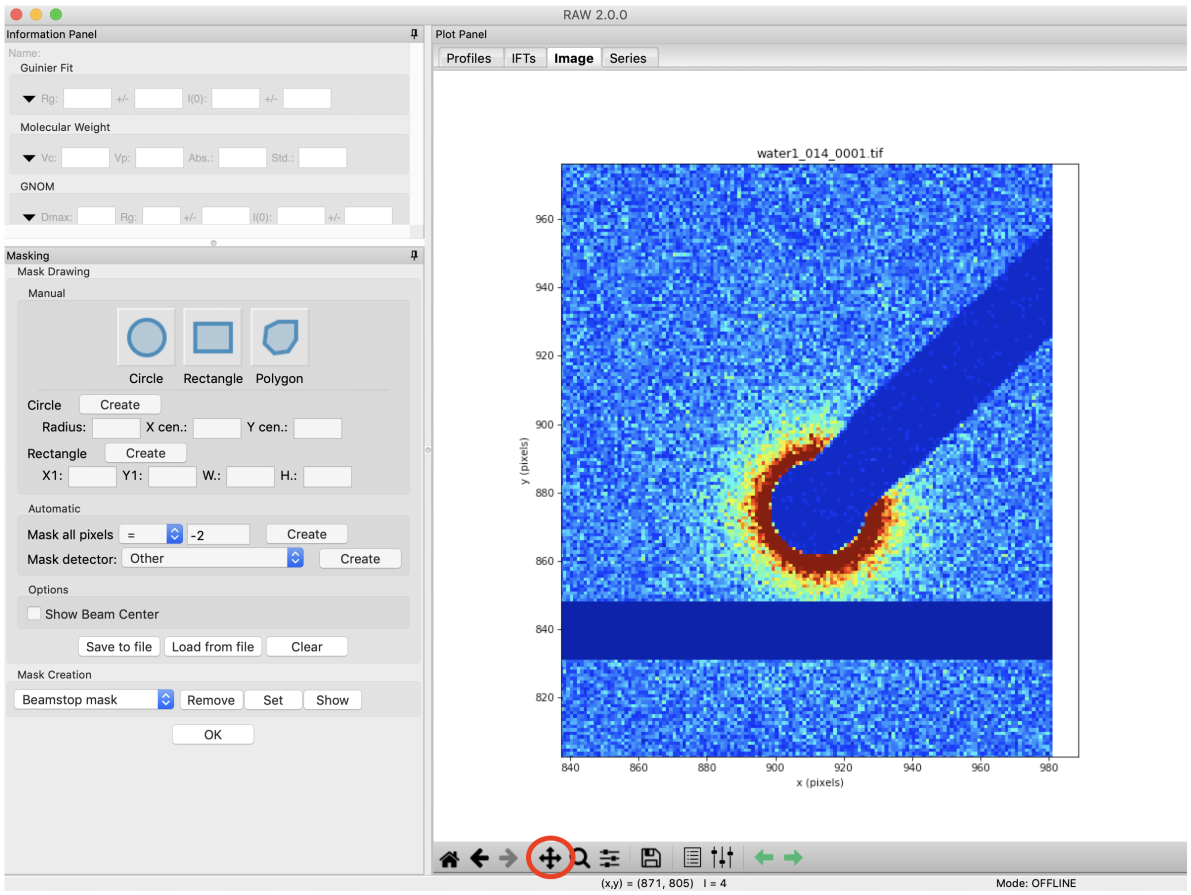
2D Reduction Parameters

Wavelength [A]: 1.0
Detector: pilatus_1m
Detector X Pixel Size [um]: 172.0
Detector Y Pixel Size [um]: 172.0
Sample-Detector Distance [mm]: 1000
Beam X center: 0.0
Beam Y center: 0.0
Detector Tilt [deg]: 0.0
Detector Tilt Plane Rotation [deg]: 0.0
[✓] Correct for the change in the solid angle of the pixels
[ ] Correct for the polarization of the x-ray beam
Polarization factor (-1 to 1, 0 is random): 0
Integration Method: nosplit_csr
[ ] Use image pixel values as pixel variance
Error Model: poisson
Angular unit q_A^-1
Binning type: Linear
Bin factor: 1

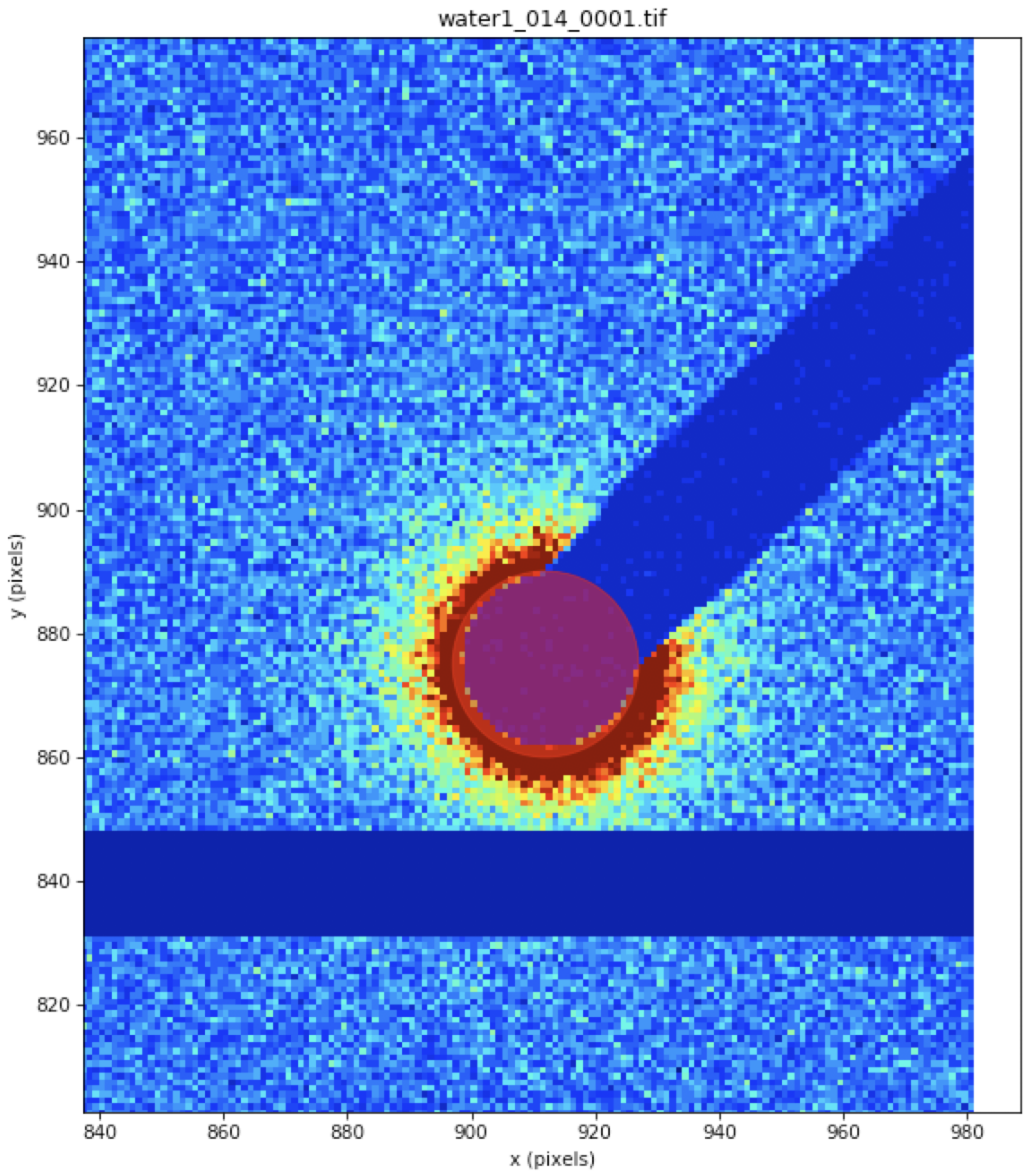
Other Parameters

Start plots at q-point number: 0
Skip n points at the end of the curve: 0

7. Click the OK button in the bottom right to close the window and save the changes to the settings.
8. In the files tab, navigate to the **Tutorial_Data/calibration_data** folder.
9. In the Files tab, select the **water1_014_0001.tif** file and click the show image button.
   - *Tip:* It's usually best to make a mask on a water or buffer image. If you do it on instrument background, which can be quite low, it can be hard to tell where the edges of the beamstop are. If you do it on a bright image, like a silver behenate or glassy carbon, you can get fooled by a small amount of bleed onto the pixels at the edge of the beamstop and not make a large enough mask.
10. Set the image upper limit to 30 on a linear scale.
11. Open the masking panel by clicking "Masking" in the Tools menu.
12. Zoom in around the beamstop.
    - *Note:* The beamstop is the blue bar in the upper right of the detector.
13. Select the Pan tool and left click and drag the image to the left until you can see a blank (white) canvas to the left of the beamstop.

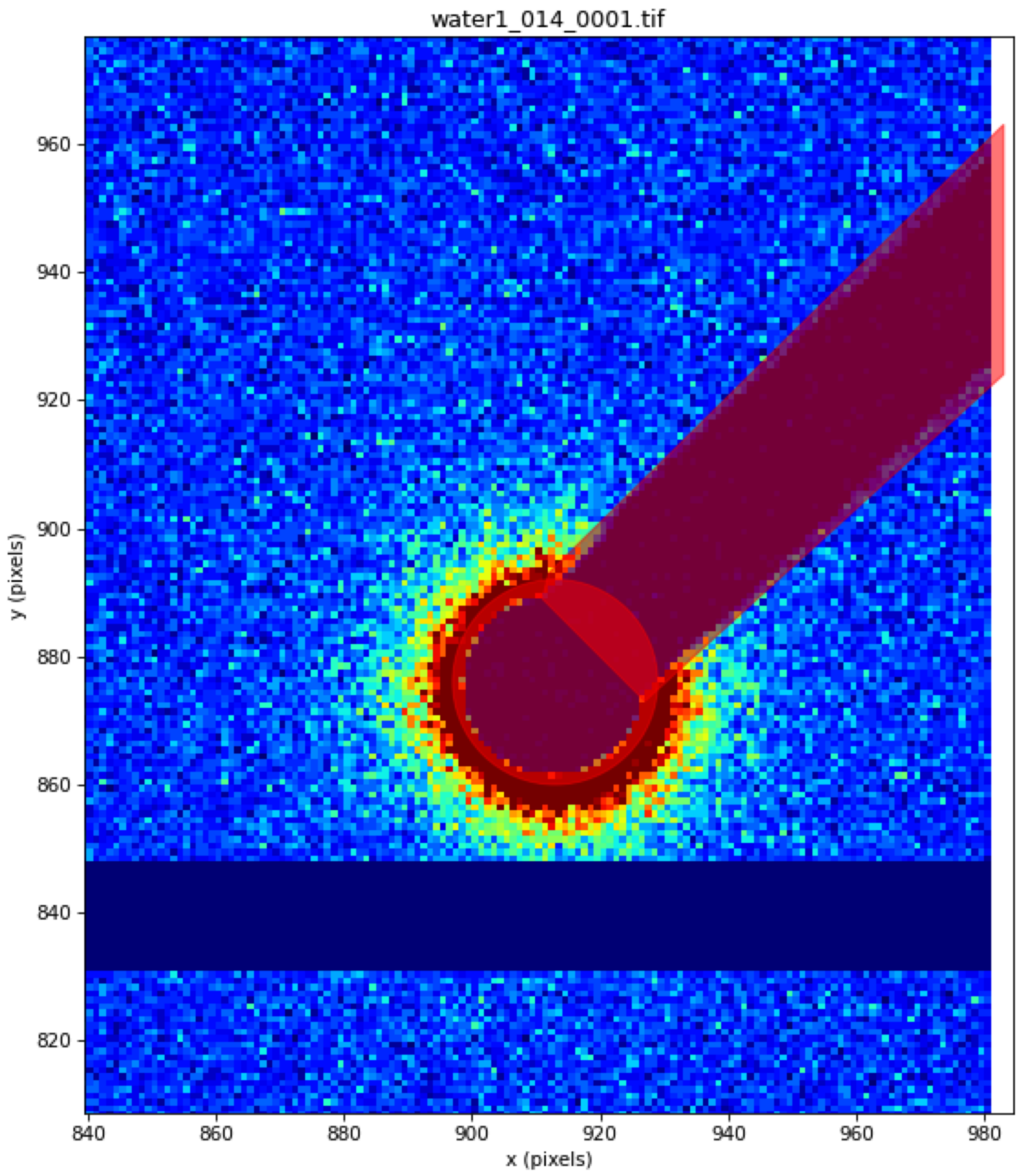


14. Click on the circle mask drawing button and click on a point at the center of the circular part of the beamstop.
15. Move the cursor until the red outline of the circle covers the beamstop. Click again to make the circle mask.
    - *Tip:* If you mess up drawing the mask, click on the masked region (shaded red) and click your backspace/delete key to remove it.
    - *Tip:* You can also resize a rectangle (or circle) mask my right clicking on it and selecting resize.
    - *Tip:* You can create a rectangle of defined position and size by entering the x and y coordinates of the bottom left corner and the width and height of the rectangle then clicking 'Create'.
16. If the mask is not perfectly centered, click and drag it until it's centered.
17. If the mask is too large or small once centered, right click and click resize. Then move the mouse pointer to resize the mask, and click once to set the new size.

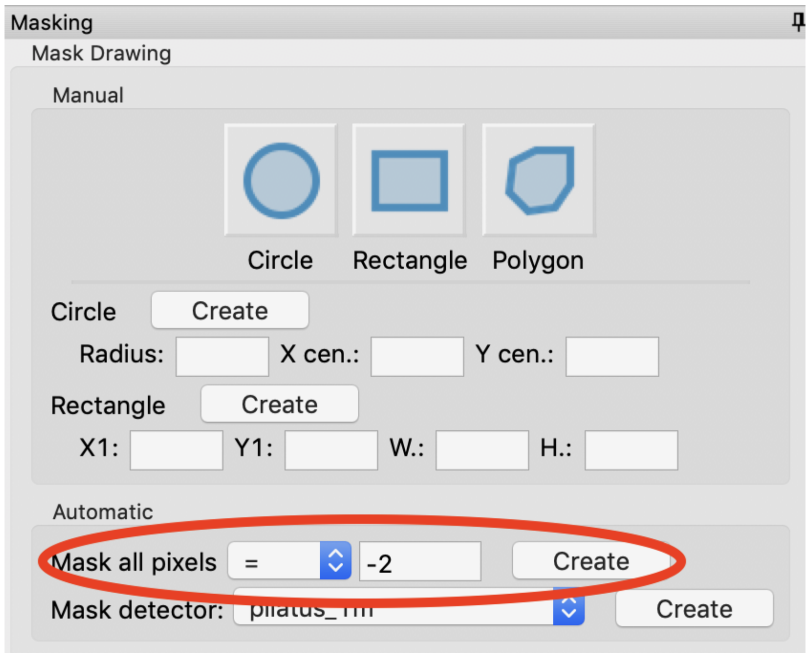


18. Click on the polygon mask drawing button. Click on a point circular part of the beamstop near the top right edge. Click again near the start of the bar. Click again just past the top of the bar in the white canvas. Click again in the white canvas just below the bottom of the bar. Click again near the start of the bar on the bottom side near the circular region. Click again in the circular part of the beamstop near the bottom edge. Right click to finish the map.

    - *Tip:* Each click in a polygon mask adds a new point, which defines a line segment between it and the previous point.

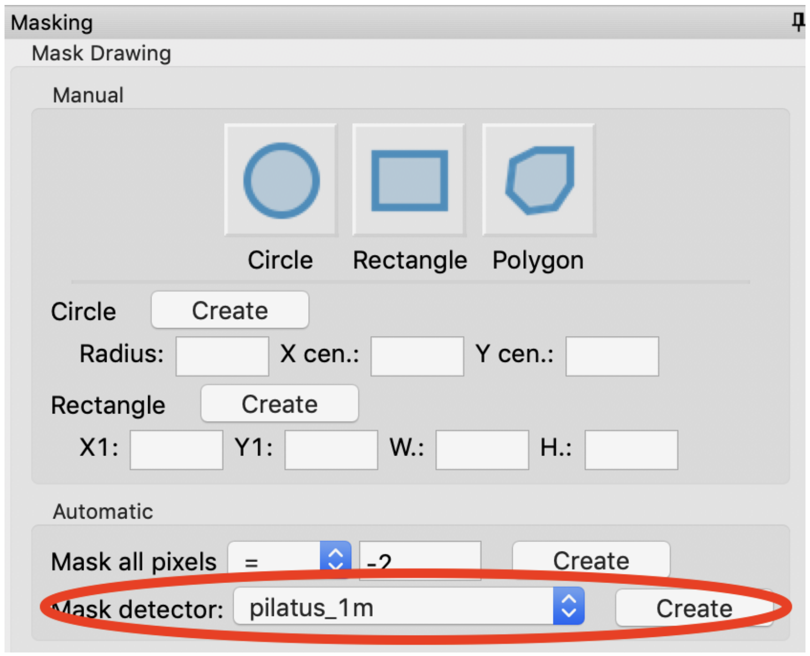


19. Zoom back out to the full extent of the image.
20. Next you need to mask out the bad pixels on the detector. On a Pilatus detector, bad pixels usually have a value of -2.
21. In the automatic section of the Mask Drawing controls, set Mask All Pixels = -2 and click 'Create'.
    - This control allows you to mask pixels at, above, or below a given threshold. It is particularly useful for things like bad pixels, where the value is known.

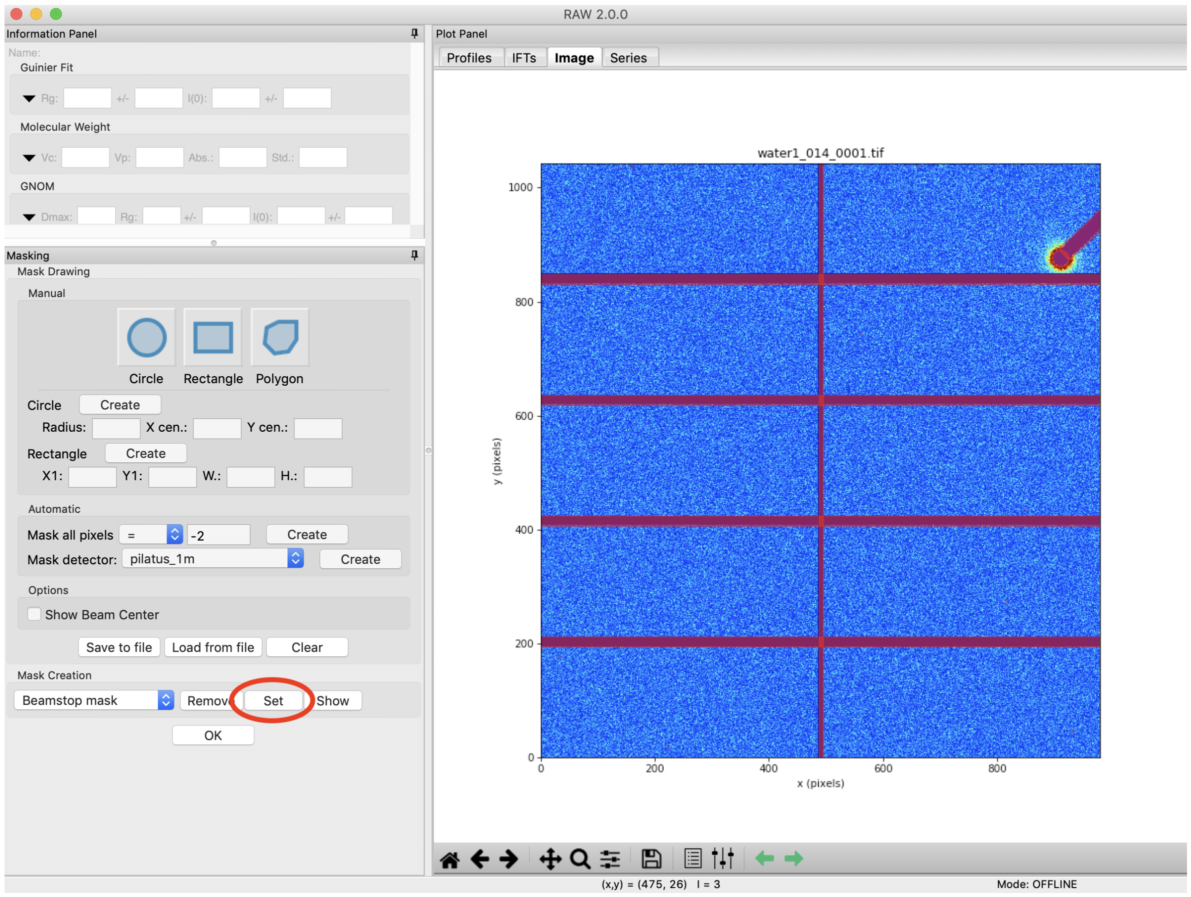


22. Next we need to mask the gaps between detector panels. Verify that the “pilatus_1m” detector is selected in the “Mask detector” control. Then click “Create”.

    - This control automatically creates masks of panel gaps for any known detector type.



23. In the masking panel, make sure that “Beamstop mask” is selected in the Mask Creation drop-down menu. Click the “Set” button to set the mask you just made as the beamstop mask.



24. Click the OK button to exit the masking panel.
25. If you wish to edit the current mask, reopen the Masking panel. Then click the "Show" button to show the current mask. From there you can make changes, then click "Set" again.

Additional Tips

1. Don't be confused by the "Save to File" and "Load from file" buttons. These save the mask to a separate file or load a mask from a separate file. These do not save or set the mask in RAW. To do that you need to use the "Set" button as described above. The mask is then saved with the settings.

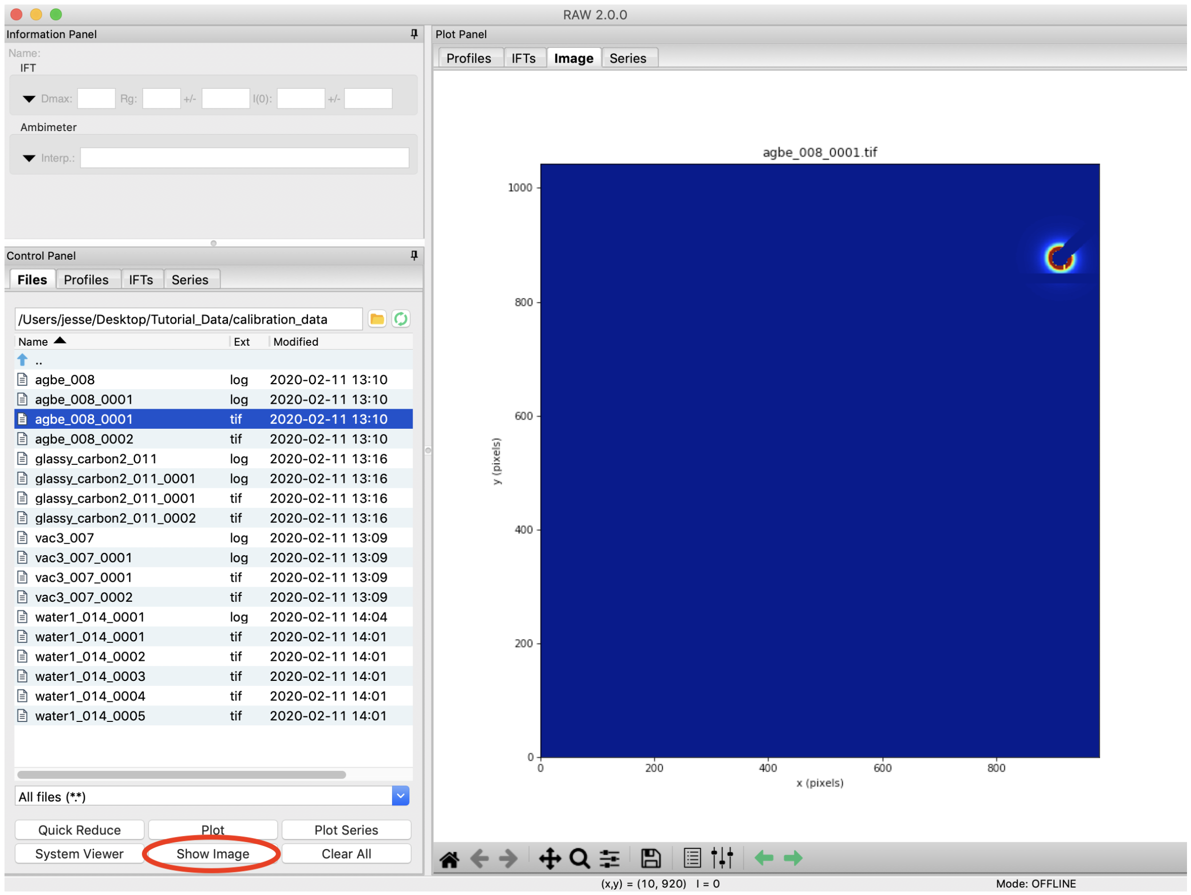
Centering and calibration

The next step is to set the beam center, x-ray wavelength, and sample to detector distance. Before this can be done, you have to set the image and file header type in the Options window. The best way to find the beam center and sample to detector distance is using the automated method in RAW. This tutorial assumes you have just done *Part 1*. If not, open RAW as in Step 1 and set your data folder as in Step 8 of *Part 1*.

A video version of this tutorial is available:

The written version of the tutorial follows.

1. We will use silver behenate to calibrate the sample to detector distance, the beam center on the detector, and the detector rotation. Show the silver behenate image by selecting the **agbe_008_0001.tif** file and clicking the show image button at the bottom of the File Control panel.



/Users/jesse/Desktop/Tutorial_Data/calibration_data

Name	Ext	Modified
..		
agbe_008	log	2020-02-11 13:10
agbe_008_0001	log	2020-02-11 13:10
agbe_008_0001	tif	2020-02-11 13:10
agbe_008_0002	tif	2020-02-11 13:10
glassy_carbon2_011	log	2020-02-11 13:16
glassy_carbon2_011_0001	log	2020-02-11 13:16
glassy_carbon2_011_0001	tif	2020-02-11 13:16
glassy_carbon2_011_0002	tif	2020-02-11 13:16
vac3_007	log	2020-02-11 13:09
vac3_007_0001	log	2020-02-11 13:09
vac3_007_0001	tif	2020-02-11 13:09
vac3_007_0002	tif	2020-02-11 13:09
water1_014_0001	log	2020-02-11 14:04
water1_014_0001	tif	2020-02-11 14:01
water1_014_0002	tif	2020-02-11 14:01
water1_014_0003	tif	2020-02-11 14:01
water1_014_0004	tif	2020-02-11 14:01
water1_014_0005	tif	2020-02-11 14:01

2. In the Image Plot Panel that is now showing, click on the Image Display Settings button (looks like vertical slider bars) at the bottom of the screen.

3. In the window that appears, set the scale to logarithmic and and click “OK”.

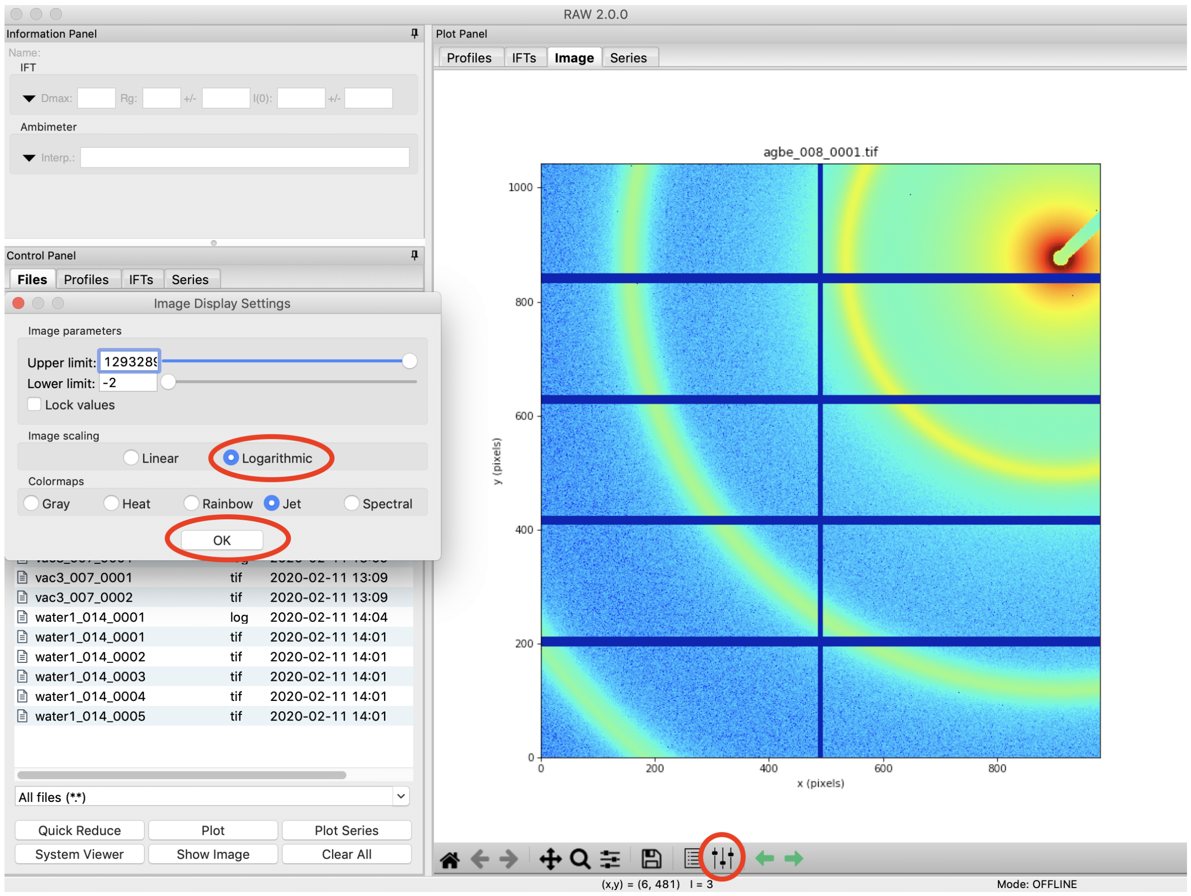


Image Display Settings

Image parameters

Upper limit: 129328
Lower limit: -2
Lock values

Image scaling: Linear [✓ Logarithmic]

Colormaps: Gray Heat Rainbow [✓ Jet] Spectral

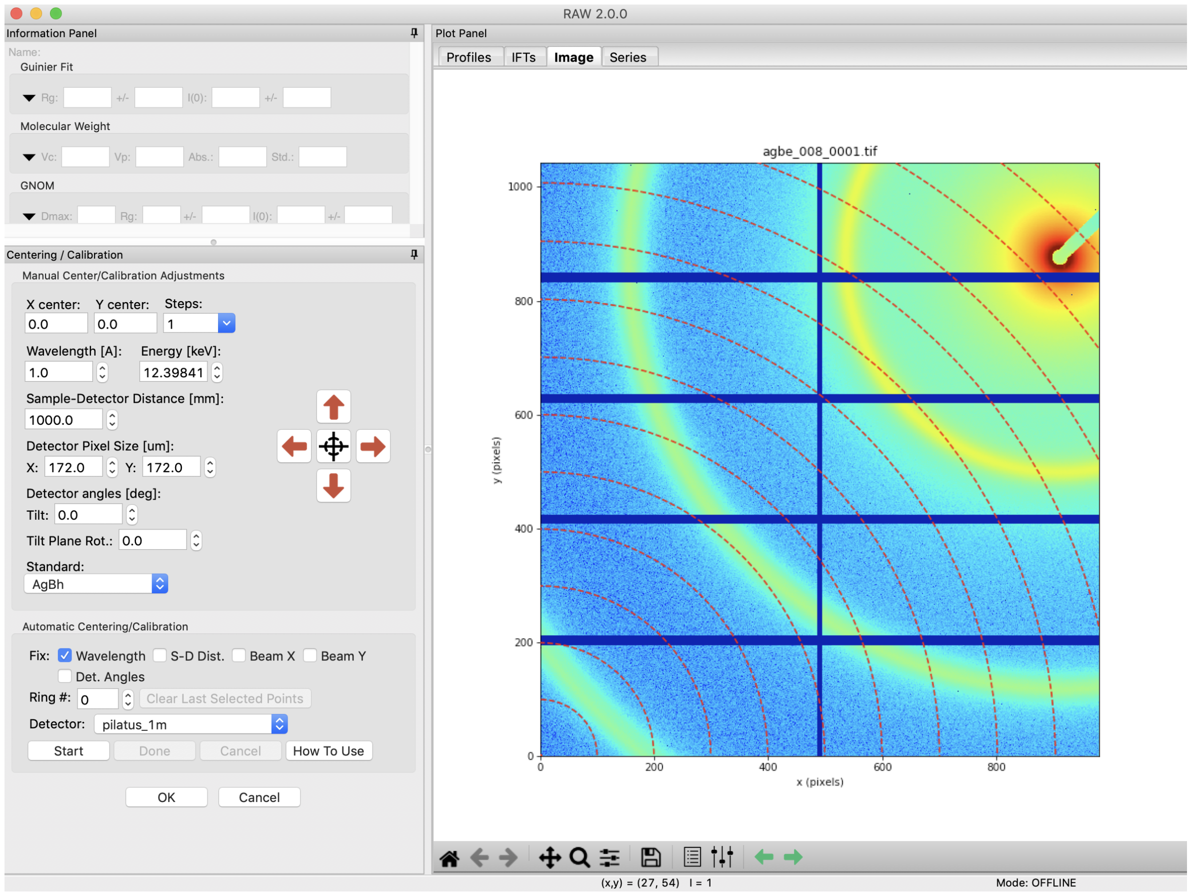
OK

vac3_007_0001	tif	2020-02-11 13:09
vac3_007_0002	tif	2020-02-11 13:09
water1_014_0001	log	2020-02-11 14:04
water1_014_0001	tif	2020-02-11 14:01
water1_014_0002	tif	2020-02-11 14:01
water1_014_0003	tif	2020-02-11 14:01
water1_014_0004	tif	2020-02-11 14:01
water1_014_0005	tif	2020-02-11 14:01

(x,y) = (6, 481) I = 3

Mode: OFFLINE

4. Open the Centering/Calibration panel by going to the Tools menu and selecting “Centering/Calibration”.


RAW 2.0.0

Information Panel

Name:

Guinier Fit

Rg: +/- I(0): +/-

Molecular Weight

Vc: Vp: Abs.: Std.:

GNOM

Dmax: Rg: +/- I(0): +/-

Centering / Calibration

Manual Center/Calibration Adjustments

X center: 0.0 Y center: 0.0 Steps: 1

Wavelength [A]: 1.0 Energy [keV]: 12.39841

Sample-Detector Distance [mm]: 1000.0

Detector Pixel Size [um]: X: 172.0 Y: 172.0

Detector angles [deg]:

Tilt: 0.0

Tilt Plane Rot.: 0.0

Standard: AgBh

Automatic Centering/Calibration

Fix: [✓ Wavelength] S-D Dist. Beam X Beam Y Det. Angles

Ring #: 0

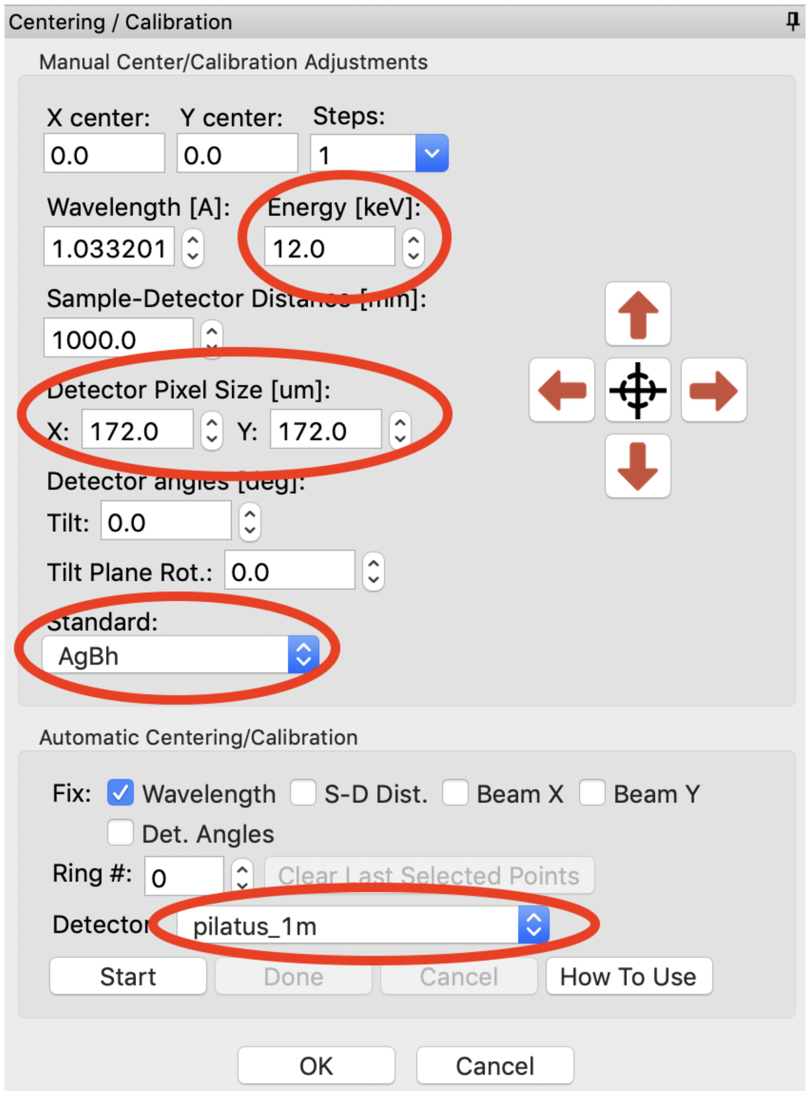
Detector: pilatus_1m

(x,y) = (27, 54) I = 1

Mode: OFFLINE

5. In the Centering/Calibration panel set the Energy to 12.0 keV. Verify that the Detector Pixel Size is 172.0 x 172.0 micron. Verify that the Detector is set to “pilatus_1m”. Verify that the standard is set to AgBh.

   - *Note:* The x-ray energy/wavelength is a previously measured value.
   - *Note:* If you set the detector properly in the radially averaging options panel the detector and detector pixel size values will fill in automatically (if the detector is not Other).


Centering / Calibration

Manual Center/Calibration Adjustments

X center: 0.0

Y center: 0.0

Steps: 1

Wavelength [A]: 1.033201

Energy [keV]: 12.0

Sample-Detector Distance [mm]: 1000.0

Detector Pixel Size [um]: X: 172.0 Y: 172.0

Detector angles [deg]:

Tilt: 0.0

Tilt Plane Rot.: 0.0

Standard: AgBh

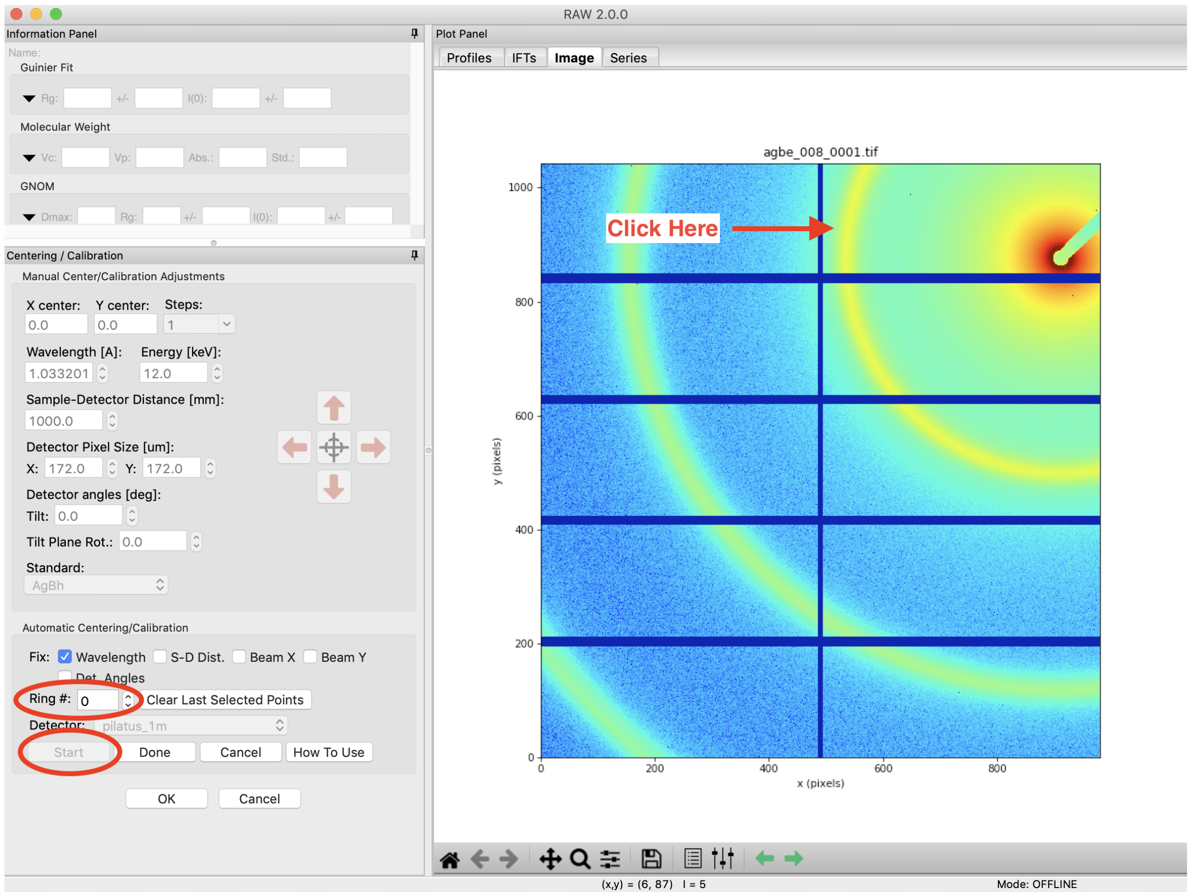
Automatic Centering/Calibration

Fix: [✓ Wavelength] S-D Dist. Beam X Beam Y Det. Angles

Ring #: 0

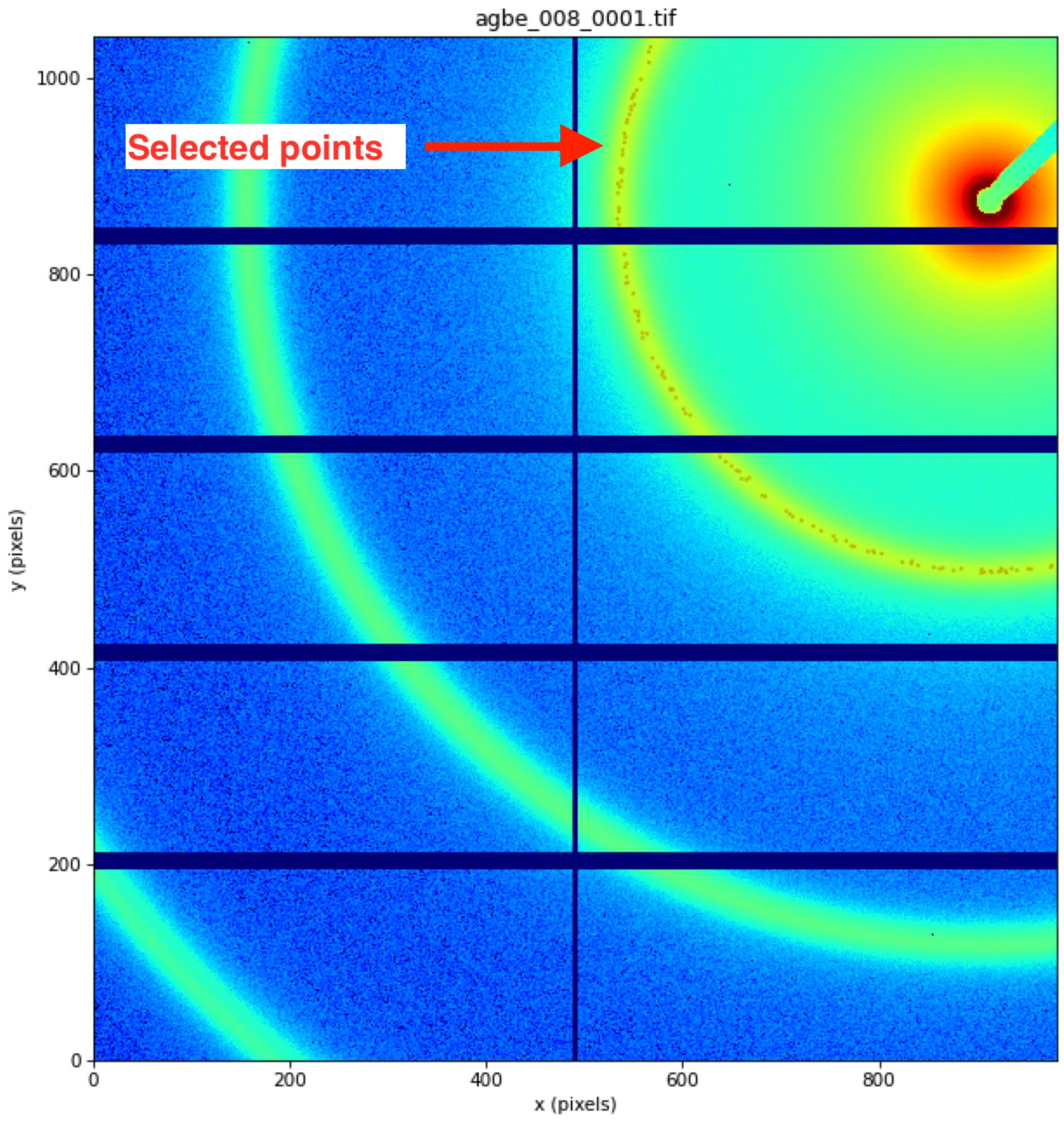
Detector: pilatus_1m

6. The goal of centering and calibration is to find a beam center position, sample to detector distance, and detector rotation that causes the calculated Silver Behenate ring pattern to match up with the rings on the image.
   - *Note:* The beamstop is the blue/green bar extending out from the top right edge of the detector
7. Click the “Start” button in the “Automatic Centering/Calibration” panel.
8. Make sure the “Ring #” is set to 0. Click on a point with strong intensity in the Silver Behenate ring nearest the beamstop.
   - *Note:* For some experimental setups, one or more of the largest d-spacing rings may not be visible on the detector. In this case, you need to figure out what the first visible ring on the detector is, and set the ring number to that. So, if the third ring was the first one on the detector, the Ring # would be set to 2 (the ring number is zero index, so 0 corresponds to the first ring, 1 to the second ring, and so on).

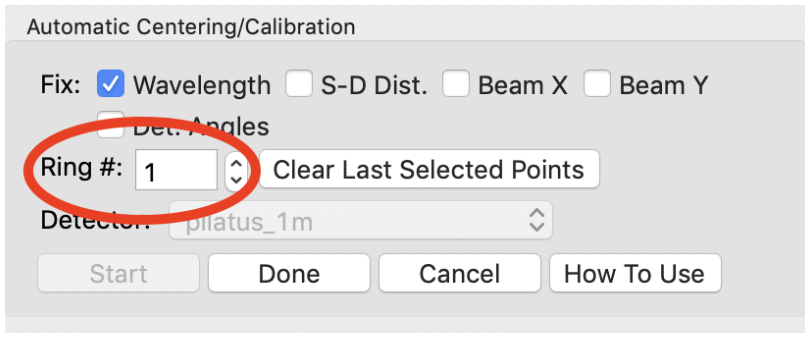


X center: 0.0
Y center: 0.0
Steps: 1
Wavelength [A]: 1.033201
Energy [keV]: 12.0
Sample-Detector Distance [mm]: 1000.0
Detector Pixel Size [um]: X: 172.0 Y: 172.0
Detector angles [deg]:
Tilt: 0.0
Tilt Plane Rot.: 0.0
Standard: AgBh
Fix: [✓ Wavelength] S-D Dist. Beam X Beam Y Det. Angles
Ring #: 0
Detector: pilatus_1m
(x,y) = (6, 87) I = 5
Mode: OFFLINE

9. The peak intensity points in that ring will be automatically found, and labeled with tan-ish dots.
   - *Tip:* If it didn't find very many points, try clicking again on another part of the ring, and it will add more points to your selection.
10. If needed, click on the other portions of the same ring that are on different detector panels to find out the rest of the points in this calibration ring.
    - *Note:* The autofind algorithm will only find peaks in contiguous regions. However, masked regions don't count, so if you've masked the panel gaps before calibration it should find points in all of the panels.
    - *Tip:* Due to the color map selected, the points may be hard to see. Try changing to the heat map or grayscale to see the selected points,.



11. Change the "Ring #" to 1.



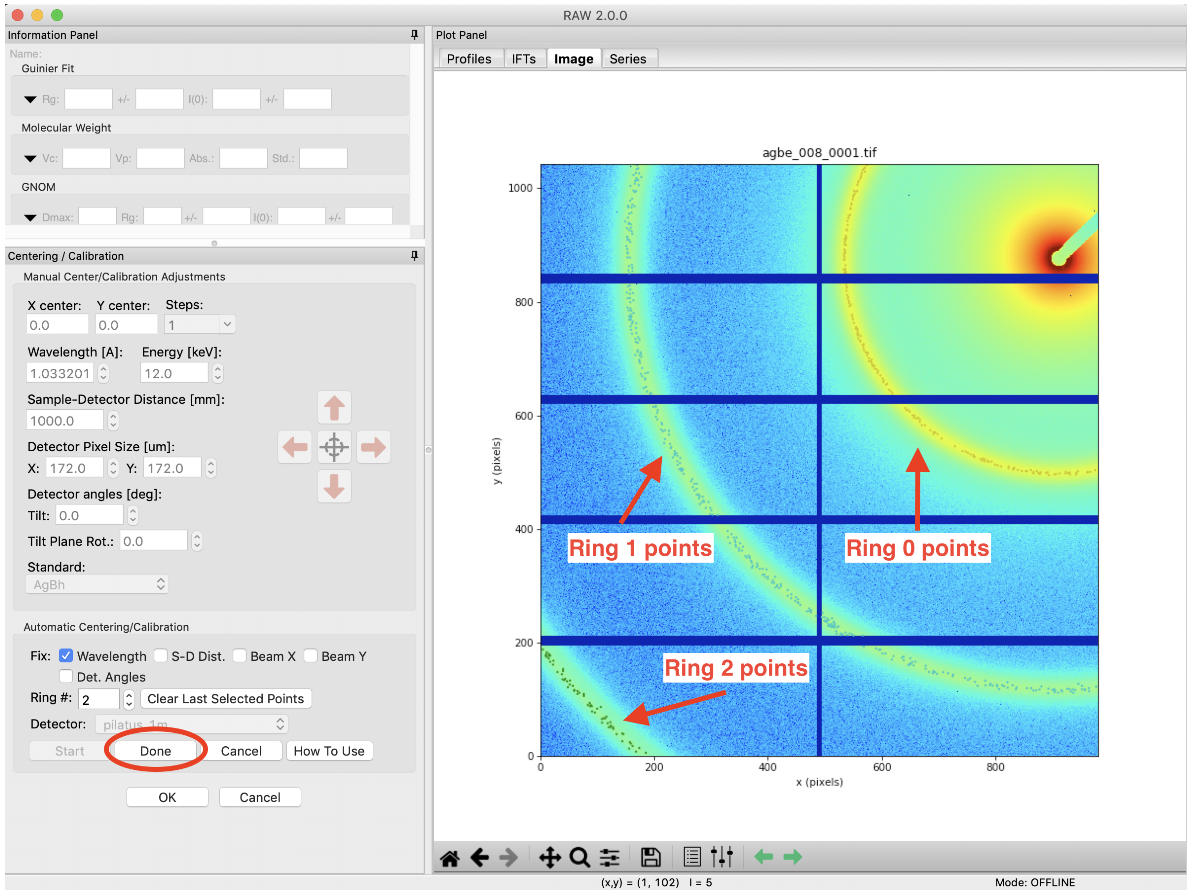
Fix: [✓ Wavelength] S-D Dist. Beam X Beam Y Det. Angles

Ring #: 1

Detector: pilatus_1m

12. Click on a peak intensity point of the second visible ring. Do this for all the sections of this ring in different detector modules.

13. Change the "Ring #" to 2. Click on a peak intensity point of the third visible ring. Points will be shown with green dots.
14. Click the "Done" button in the "Automatic Centering/Calibration" panel.



Centering / Calibration

Manual Center/Calibration Adjustments

X center: 0.0 Y center: 0.0 Steps: 1

Wavelength [A]: 1.033201 Energy [keV]: 12.0

Sample-Detector Distance [mm]: 1000.0

Detector Pixel Size [um]: X: 172.0 Y: 172.0

Detector angles [deg]: Tilt: 0.0 Tilt Plane Rot.: 0.0

Standard: AgBh

Automatic Centering/Calibration

Fix: [✓ Wavelength] S-D Dist. Beam X Beam Y Det. Angles

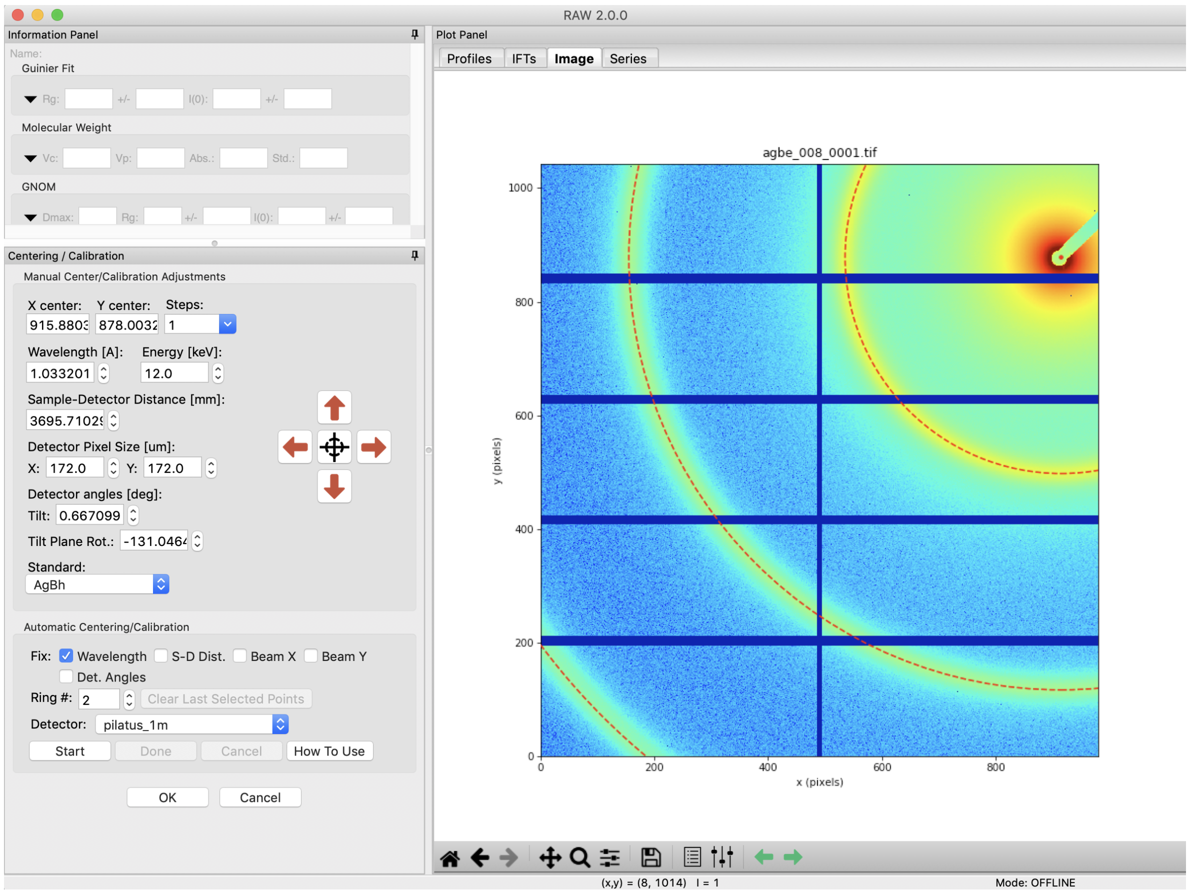
Ring #: 2

Detector: pilatus_1m

(x,y) = (1, 102) I = 5

Mode: OFFLINE

15. The beam position, sample to detector distance, and detector tilt angles will be calculated and filled in. Calculated rings will display on the plot as dashed red lines, based on the parameters found in the fit. The beam center is displayed as a red dot on the image. You can verify the validity of the fit based on how these calculated values match up with what is shown on the image.
    - *Note:* Calculated rings are displayed without detector tilt angles, so if the detector is significantly off beam normal the calculated rings will not match up with the measured rings.
    - *Note:* Image tilt plane rotation is an odd value. It represents motions of both X and Y around the Z axis of the detector. As such, it can take on large values (such as -131) for very small detector angles, which is just representing motion in both axes. In this case, all three detector angles are ~0.7 degrees or less.



Centering / Calibration

Manual Center/Calibration Adjustments

X center: 915.8803 Y center: 878.0032 Steps: 1

Wavelength [A]: 1.033201 Energy [keV]: 12.0

Sample-Detector Distance [mm]: 3695.7102

Detector Pixel Size [um]: X: 172.0 Y: 172.0

Detector angles [deg]: Tilt: 0.667099 Tilt Plane Rot.: -131.046

Standard: AgBh

Automatic Centering/Calibration

Fix: [✓ Wavelength] S-D Dist. Beam X Beam Y Det. Angles

Ring #: 2

Detector: pilatus_1m

(x,y) = (8, 1014) I = 1 Mode: OFFLINE

16. If necessary (such as if the autocentering routine fails), all of the calibration values can be adjusted manually. The beam center can either be typed into the appropriate boxes, or the red arrows can be used to nudge it by "Steps" pixels in any direction. The crosshairs can be used to pick the beam center position by hand, good for getting a rough alignment. The other parameters can be typed into their appropriate boxes. Manual centering is an iterative process:

    1. Enter rough values based on observation, measurement of actual sample detector distance.
    2. Use arrows to move beam center until you match up with the first ring.
    3. Adjust the distance until you match up with the second ring.
    4. Repeat the last two steps as necessary until you converge on a solution.

17. Click the OK button in the Centering/Calibration panel to save your settings and exit the panel.

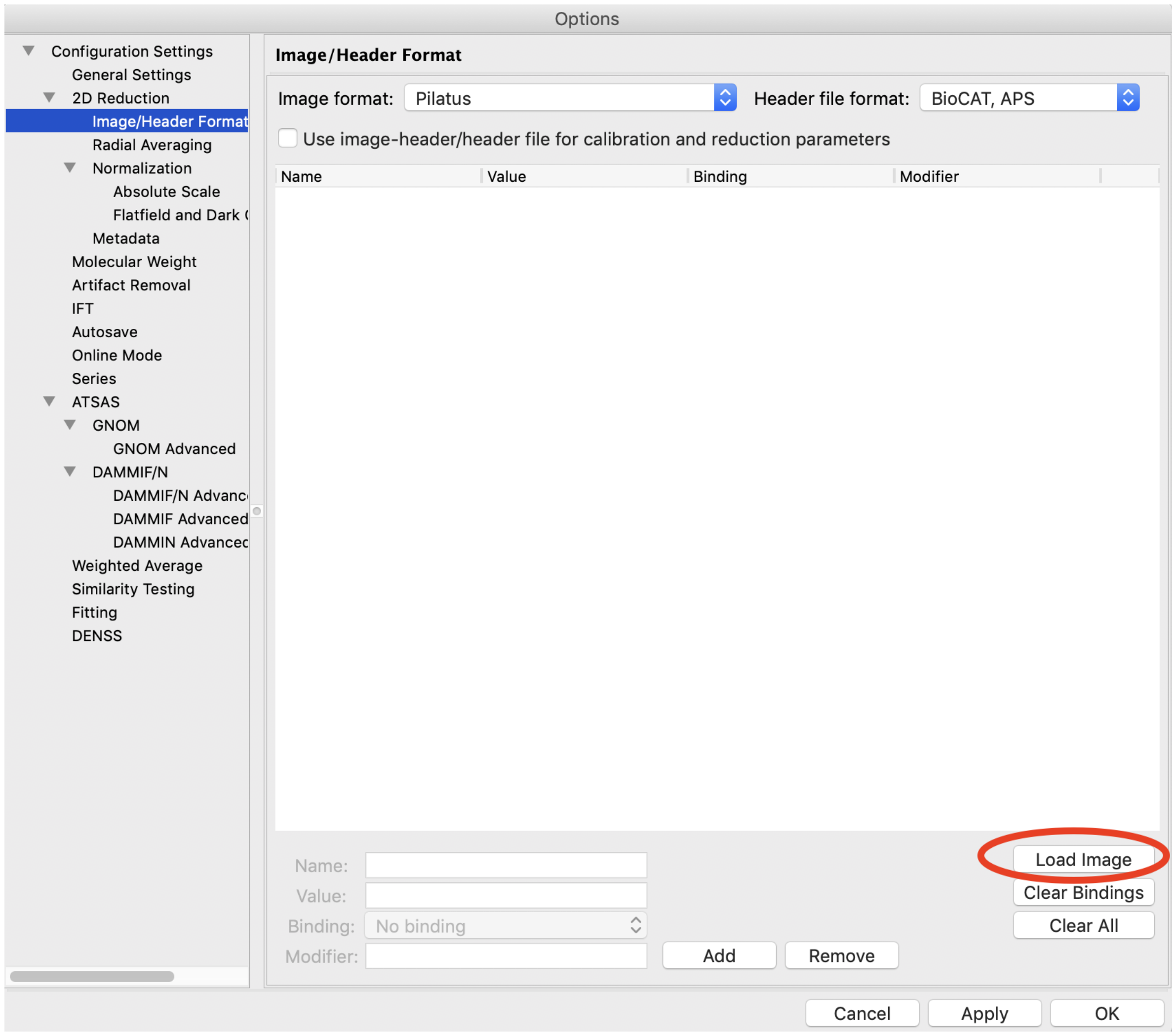
Setting normalization and other options

This section teaches you how to set up normalization by a beamstop counter, and other options. It assumes you have completed *Parts 1* and *2*.

A video version of this tutorial is available:

The written version of the tutorial follows.

1. Open the Options window by selecting "Advanced Options" in the Options menu.
2. In the window that shows up select the Image/Header Format section on the left. In the area on the right click the Load Image button.



Image/Header Format

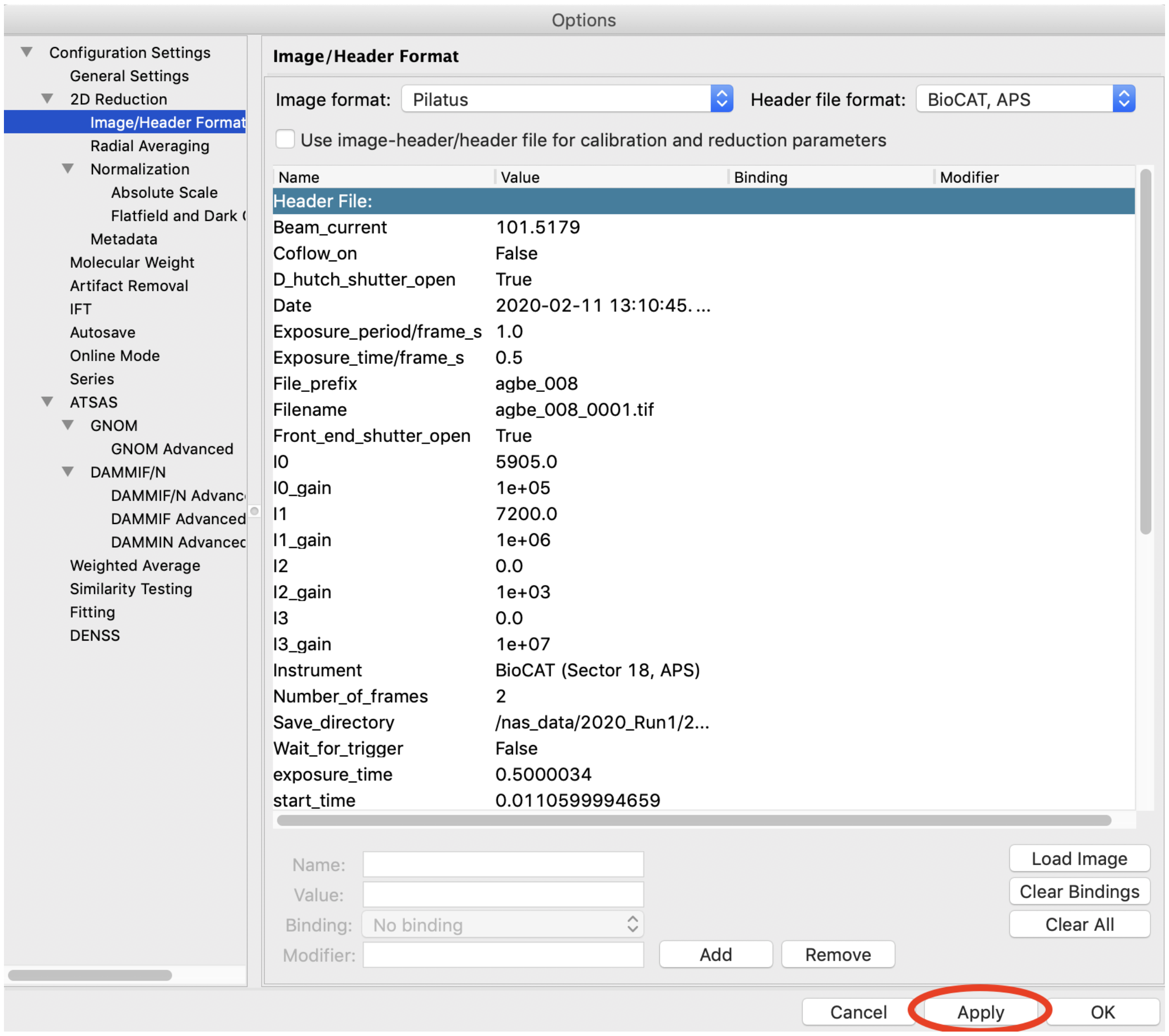
Image format: Pilatus   Header file format: BioCAT, APS

☐ Use image-header/header file for calibration and reduction parameters

Name	Value	Binding	Modifier

Name:
Value:
Binding: No binding
Modifier:

3. In the window that pops up, select the **agbe_008_0001.tif** file. Click the Open button.
   - *Note:* You can select any image of the appropriate type, not just the behenate.
4. In the Image/Header Format window you should now see header values loaded into the list. Click the Apply button at the bottom of the screen.



Image/Header Format

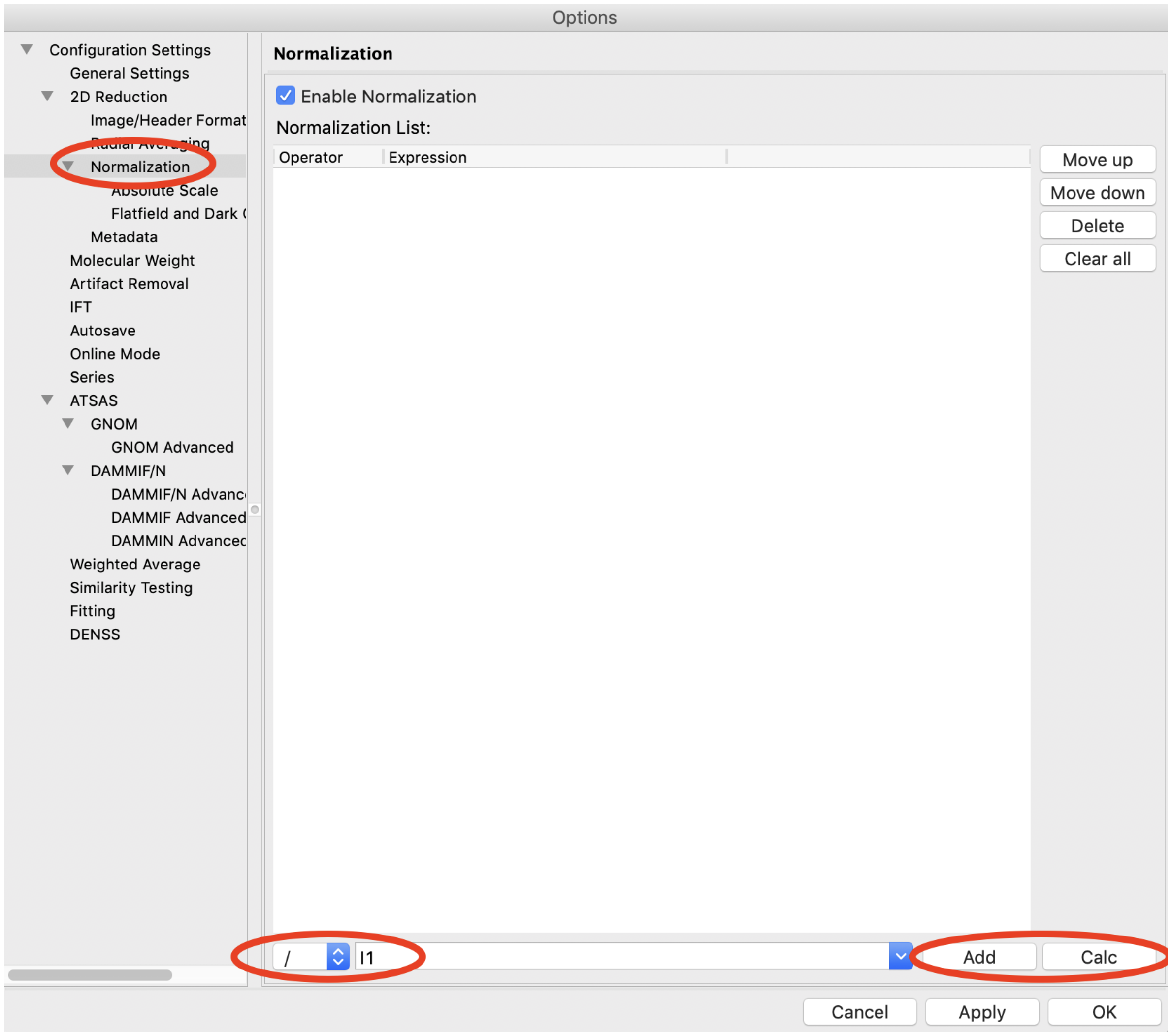
Image format: Pilatus   Header file format: BioCAT, APS

Use image-header/header file for calibration and reduction parameters

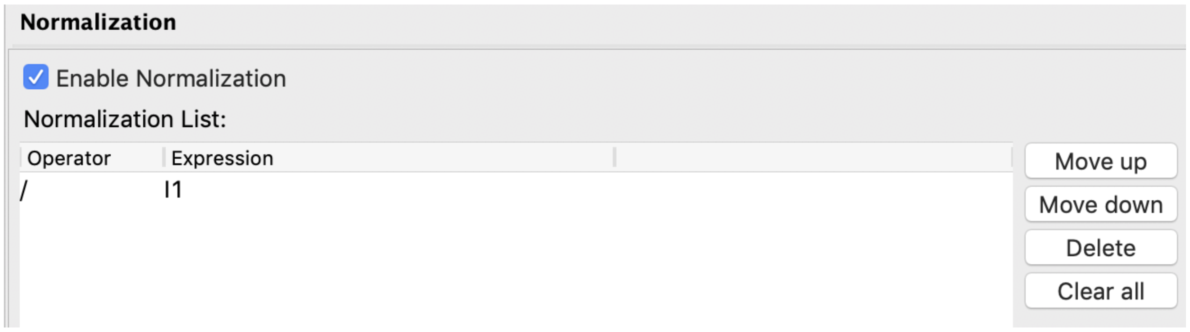
Name	Value	Binding	Modifier
Header File:			
Beam_current	101.5179		
Coflow_on	False		
D_hutch_shutter_open	True		
Date	2020-02-11 13:10:45....		
Exposure_period/frame_s	1.0		
Exposure_time/frame_s	0.5		
File_prefix	agbe_008		
Filename	agbe_008_0001.tif		
Front_end_shutter_open	True		
I0	5905.0		
I0_gain	1e+05		
I1	7200.0		
I1_gain	1e+06		
I2	0.0		
I2_gain	1e+03		
I3	0.0		
I3_gain	1e+07		
Instrument	BioCAT (Sector 18, APS)		
Number_of_frames	2		
Save_directory	/nas_data/2020_Run1/2...		
Wait_for_trigger	False		
exposure_time	0.5000034		
start_time	0.0110599994659		

Name:
Value:
Binding: No binding
Modifier:

5. Click on the Normalization section in the options list on the left.
6. In the fields at the bottom of the Normalization panel, make sure "/" is selected in the left drop-down menu, and enter I1 in the large field.
   - *Note:* It is typical in SAXS to normalize by the transmitted intensity. At the BioCAT beamline, the beamstop counter is name I1, which is why we are using that name in the normalization expression.
   - *Tip:* You can use the large field as a drop-down menu to see and select available normalization counters.
7. Click the Calc button to evaluate the expression for the counter values loaded in the Image/Header Format tab. You should get a value of 7200.0.
8. Click the Add button to add the expression to the normalization list.



9. You should now see the normalization in the normalization list. Make sure the “Enable Normalization” checkbox at the top of the page is checked.

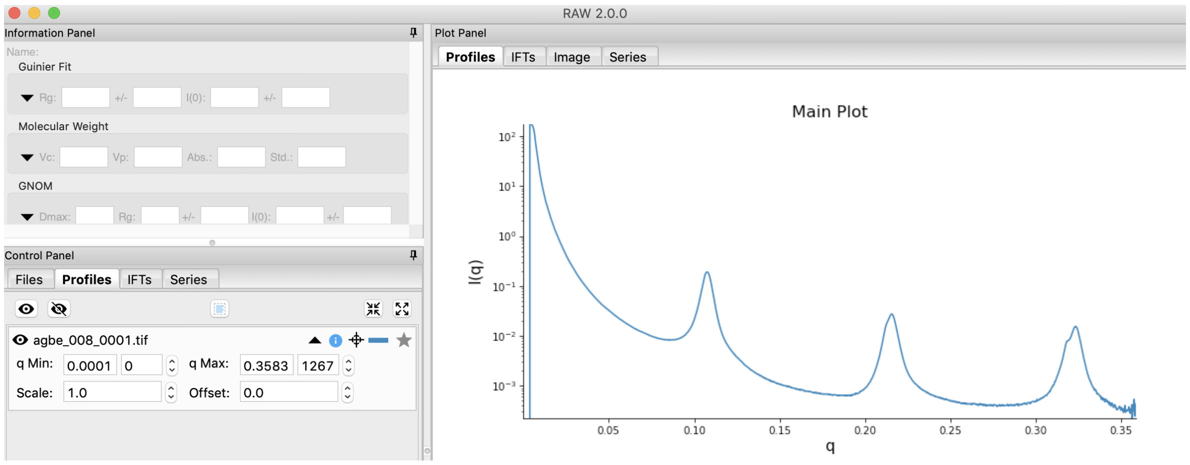


Operator	Expression
/	I1

10. Click OK to exit the options window.

11. In the file list, select the **agbe_008_0001.tif** file and click the Plot button. You will see a curve get plotted in the top panel of the Profiles plot.

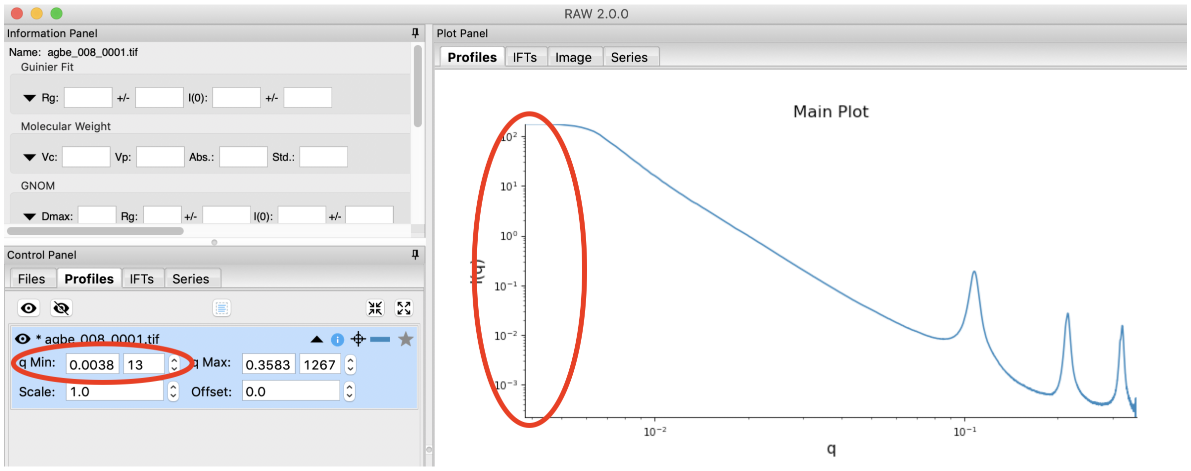
12. Click on the Profiles tab. You will see a profile loaded in the Profiles list.



agbe_008_0001.tif
q Min: 0.0001 0 q Max: 0.3583 1267
Scale: 1.0 Offset: 0.0

13. Adjust the start point for q Min to remove the points with zero value at the start of the curve (these are q points entirely in the mask). Set q Min so that the first point is the peak of the curve on the main plot. This should be around point 13 (depending on your mask).

    - *Tip:* It is easier to see the start point if you put the plot on a log-log scale.

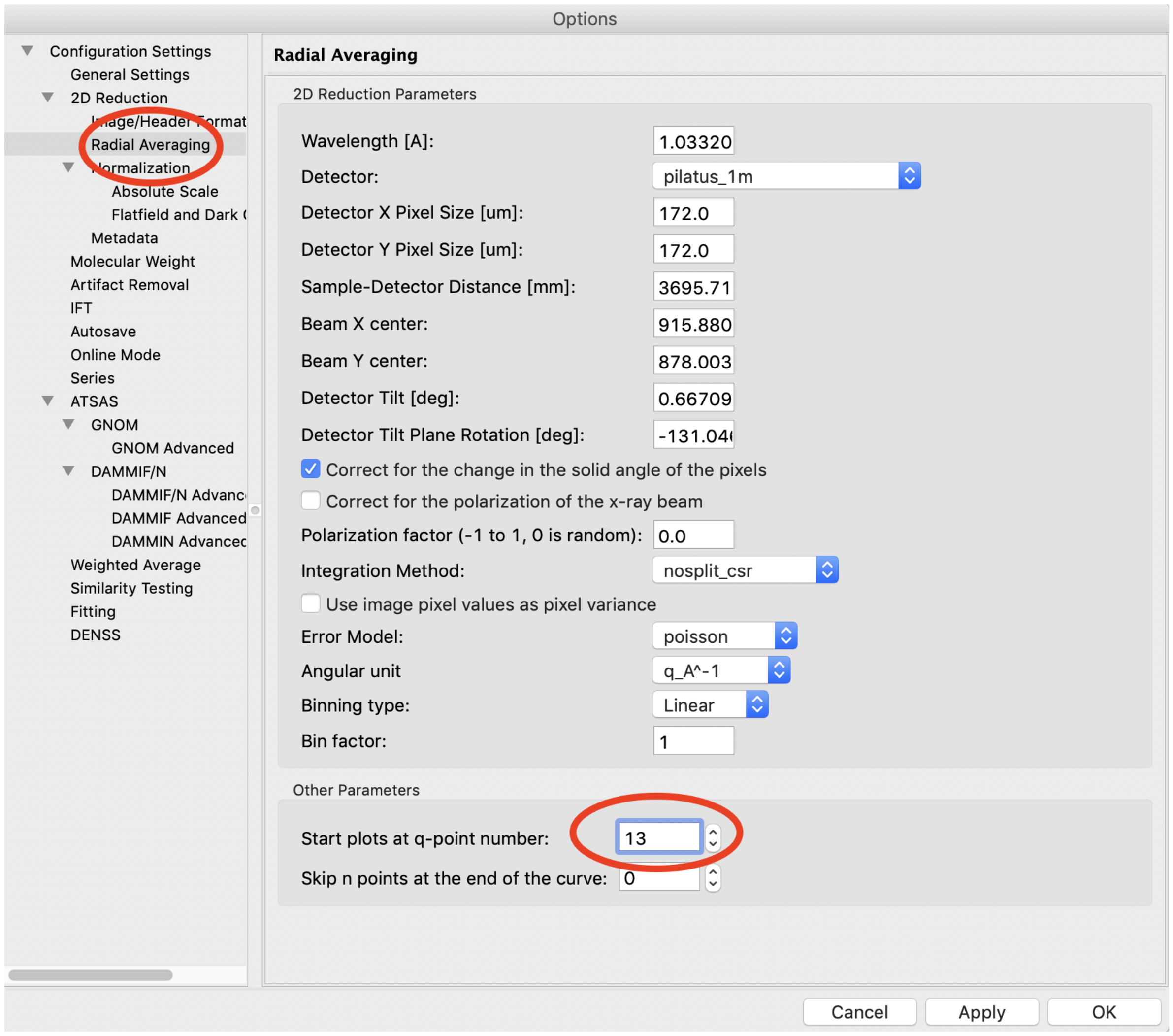


Name: agbe_008_0001.tif
* agbe_008_0001.tif
q Min: 0.0038 13 q Max: 0.3583 1267
Scale: 1.0 Offset: 0.0

14. Open the Options window as in Step 1.

15. Click on the Radial Averaging section in the options list on the left. Set "Start plots at q-point number" to the number you just found in Step 13.

    - *Note:* This makes it so that every curve loaded from now on will by default not display the first n points, which are covered by the beamstop.
    - *Tip:* You can do the same for the end point.



Radial Averaging

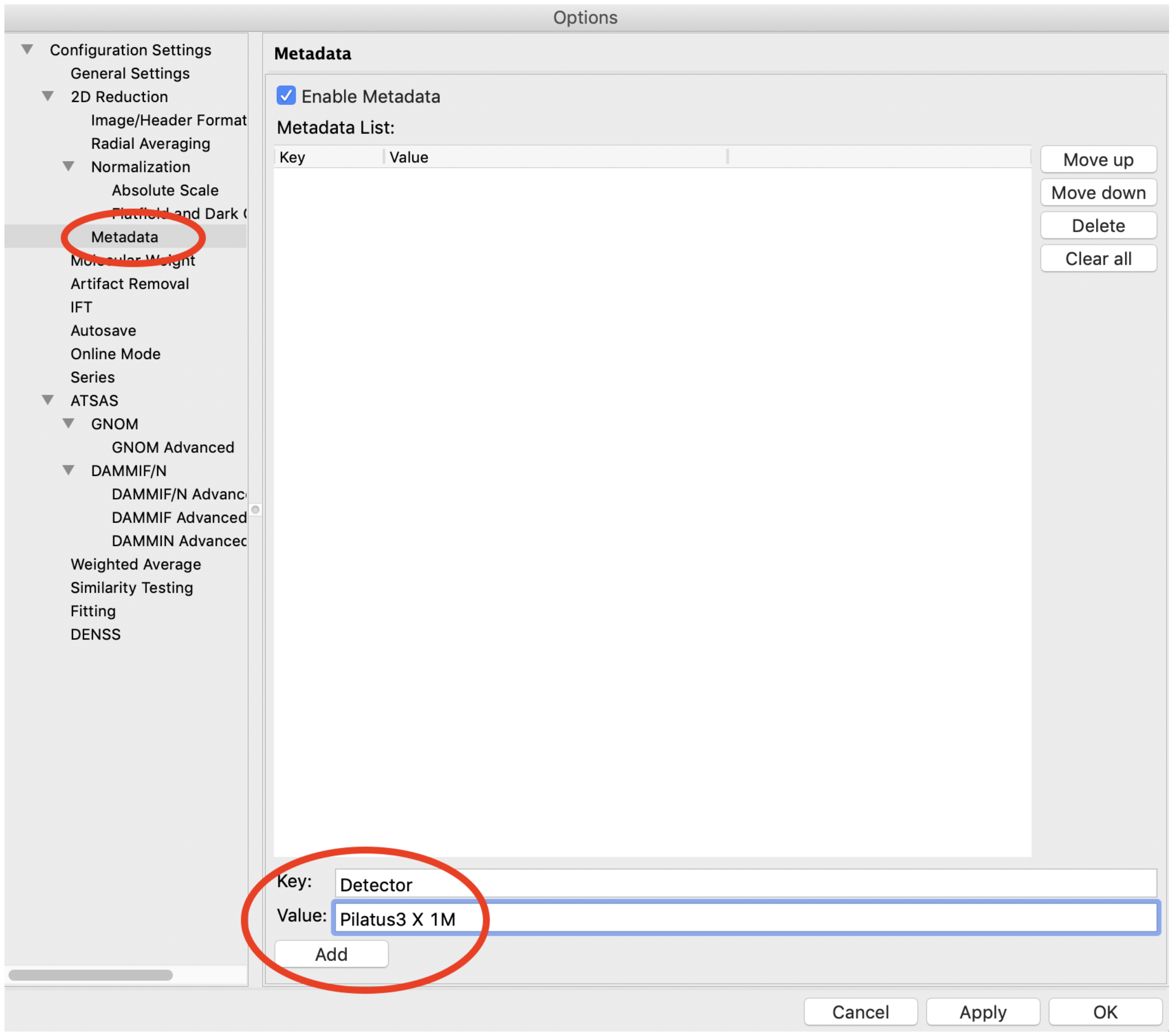
2D Reduction Parameters

Wavelength [A]:	1.03320
Detector:	pilatus_1m
Detector X Pixel Size [um]:	172.0
Detector Y Pixel Size [um]:	172.0
Sample-Detector Distance [mm]:	3695.71
Beam X center:	915.880
Beam Y center:	878.003
Detector Tilt [deg]:	0.66709
Detector Tilt Plane Rotation [deg]:	-131.04
Correct for the change in the solid angle of the pixels	✓
Correct for the polarization of the x-ray beam	☐
Polarization factor (-1 to 1, 0 is random):	0.0
Integration Method:	nosplit_csr
Use image pixel values as pixel variance	☐
Error Model:	poisson
Angular unit	q_A^-1
Binning type:	Linear
Bin factor:	1

Other Parameters

Start plots at q-point number:	13
Skip n points at the end of the curve:	0

16. RAW also allows you to add arbitrary metadata to your radially integrated files. One use case is to provide metadata keys for data deposition (e.g. in the SASBDB). To start, click on the Metadata section in the options list on the left of the Options window.
17. In the 'Key' field enter 'Detector'. In the 'Value' field enter 'Pilatus3 X 1M'.
18. Click the Add button.



☑ Enable Metadata

Key: Detector

Value: Pilatus3 X 1M

19. Click the OK button to exit the options window and save your changes.
20. You have configured everything necessary, and are now ready to save your settings. Go to the File menu and select "Save Settings".
21. Save the settings as **SAXS.cfg**.
22. These settings can now be used to process images, and can be reloaded when you open RAW by selecting "Load Settings" from the File menu.

Additional Tips

There are other settings you may find it useful to set.

1. In the "General Settings" section you can set "Hide controls on profile items for new plots". This means that when you load a profile item it starts with the extra controls (such as q min and q max) hidden.

Setting absolute scale with water

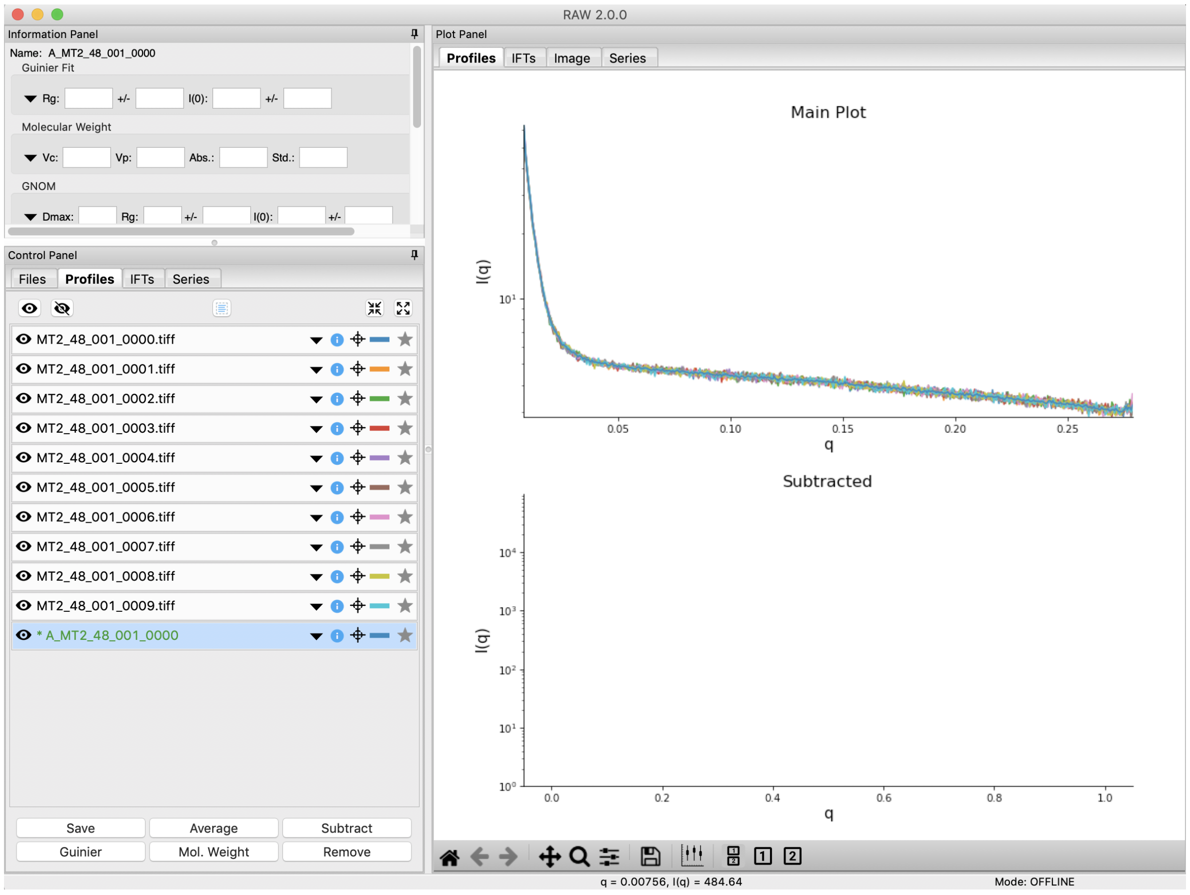
This section teaches you how to set up absolute scale using water as a reference. Note that you can use water or glassy carbon (*Part 5*) for absolute scale calibration in RAW. Glassy carbon is the more accurate approach, if available.

Note: The calibration dataset used for the first parts of this tutorial doesn't have the requisite data to use for this part. So we will use the data in the **calibration_data/extra** folder.

A video version of this tutorial is available:

The written version of the tutorial follows.

1. Load the **SAXS.cfg** file in the **calibration_data/extra** folder.
2. Plot all of the **MT2_48_001_000x.tiff** files, where x is 0-9, on the main plot.
   - *Tip: Section 1 Part 1* of this tutorial document teaches you how to do this.
3. Average the **MT2** files you just loaded. Save the average in the **calibration_data** folder.

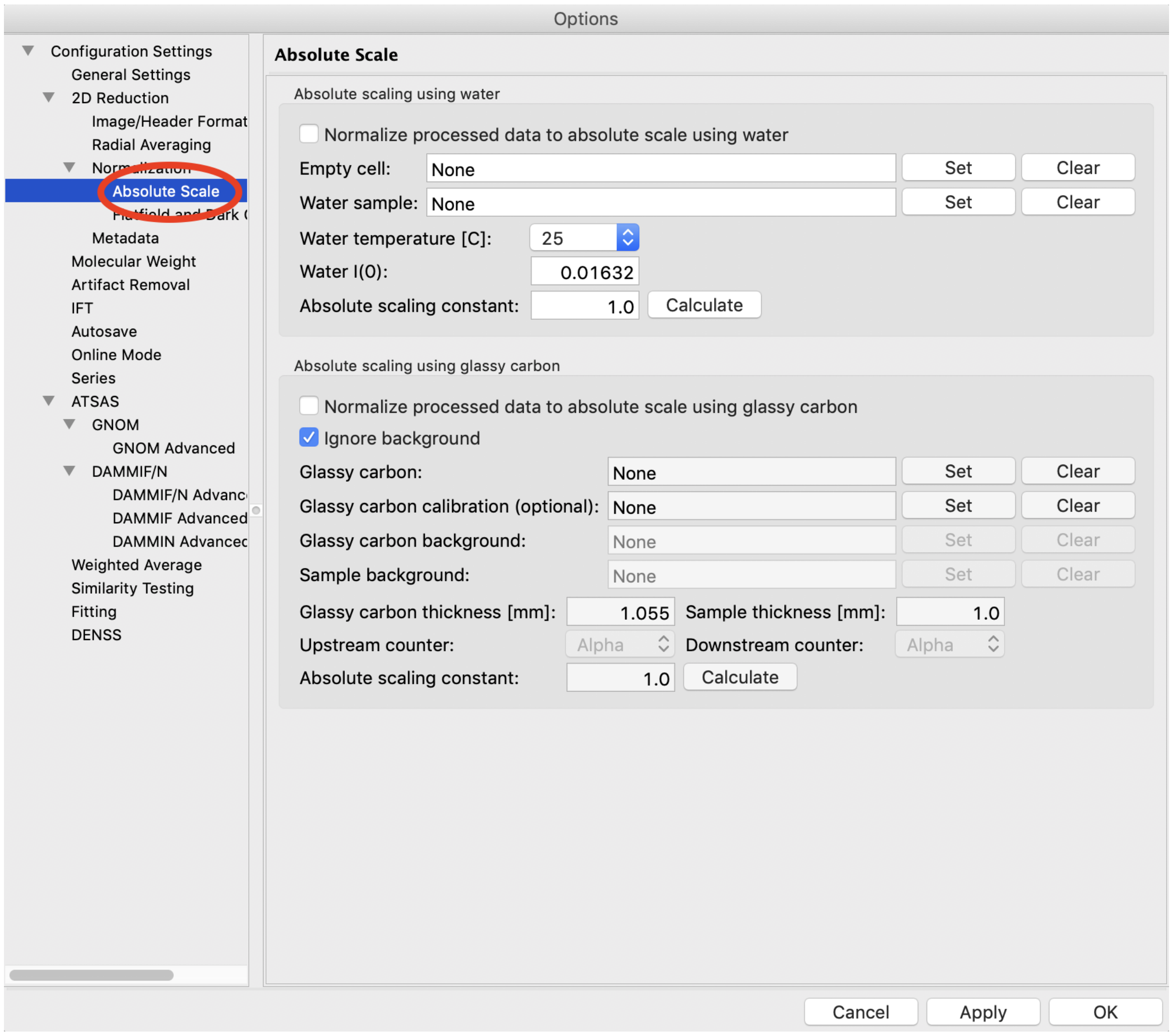


Name: A_MT2_48_001_0000

MT2_48_001_0000.tiff
MT2_48_001_0001.tiff
MT2_48_001_0002.tiff
MT2_48_001_0003.tiff
MT2_48_001_0004.tiff
MT2_48_001_0005.tiff
MT2_48_001_0006.tiff
MT2_48_001_0007.tiff
MT2_48_001_0008.tiff
MT2_48_001_0009.tiff
* A_MT2_48_001_0000

q = 0.00756, I(q) = 484.64

4. Repeat steps 1 and 2, plotting, averaging and saving, for the **water2_49_001_000x.tiff** files.
5. Open the Options window by selecting "Advanced Options" in the Options menu.
6. Click on the Absolute Scale section in the options list on the left.



Absolute Scale

Absolute scaling using water

- [ ] Normalize processed data to absolute scale using water

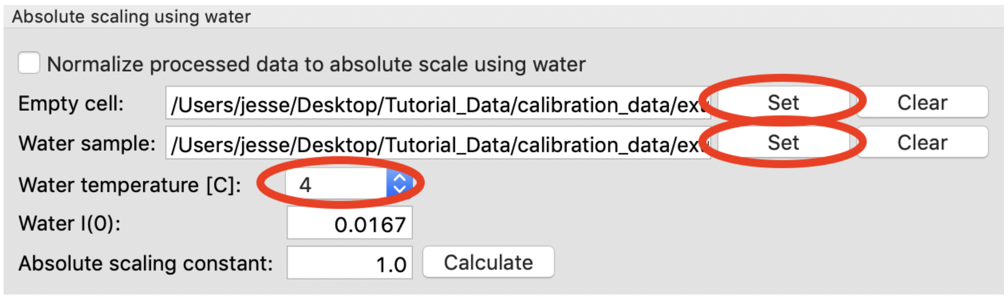
Empty cell: None (Set) (Clear)
Water sample: None (Set) (Clear)
Water temperature [C]: 25
Water I(0): 0.01632
Absolute scaling constant: 1.0 (Calculate)

Absolute scaling using glassy carbon

- [ ] Normalize processed data to absolute scale using glassy carbon
- [x] Ignore background

Glassy carbon: None (Set) (Clear)
Glassy carbon calibration (optional): None (Set) (Clear)
Glassy carbon background: None (Set) (Clear)
Sample background: None (Set) (Clear)
Glassy carbon thickness [mm]: 1.055 Sample thickness [mm]: 1.0
Upstream counter: Alpha Downstream counter: Alpha
Absolute scaling constant: 1.0 (Calculate)

7. Click on the Empty cell "Set" button and select the **A_MT2_48_001_0000.dat** file.
8. Click on the Water sample "Set" button and select the **A_water2_49_001_0000.dat** file.
9. Set the Water temperature to 4 C.


Absolute scaling using water

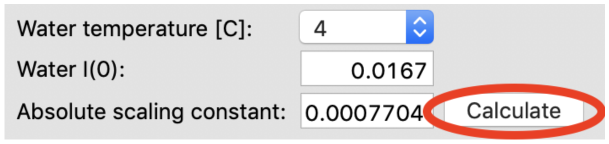
- [ ] Normalize processed data to absolute scale using water

Empty cell: /Users/jesse/Desktop/Tutorial_Data/calibration_data/ex (Set) (Clear)
Water sample: /Users/jesse/Desktop/Tutorial_Data/calibration_data/ex (Set) (Clear)
Water temperature [C]: 4
Water I(0): 0.0167
Absolute scaling constant: 1.0 (Calculate)

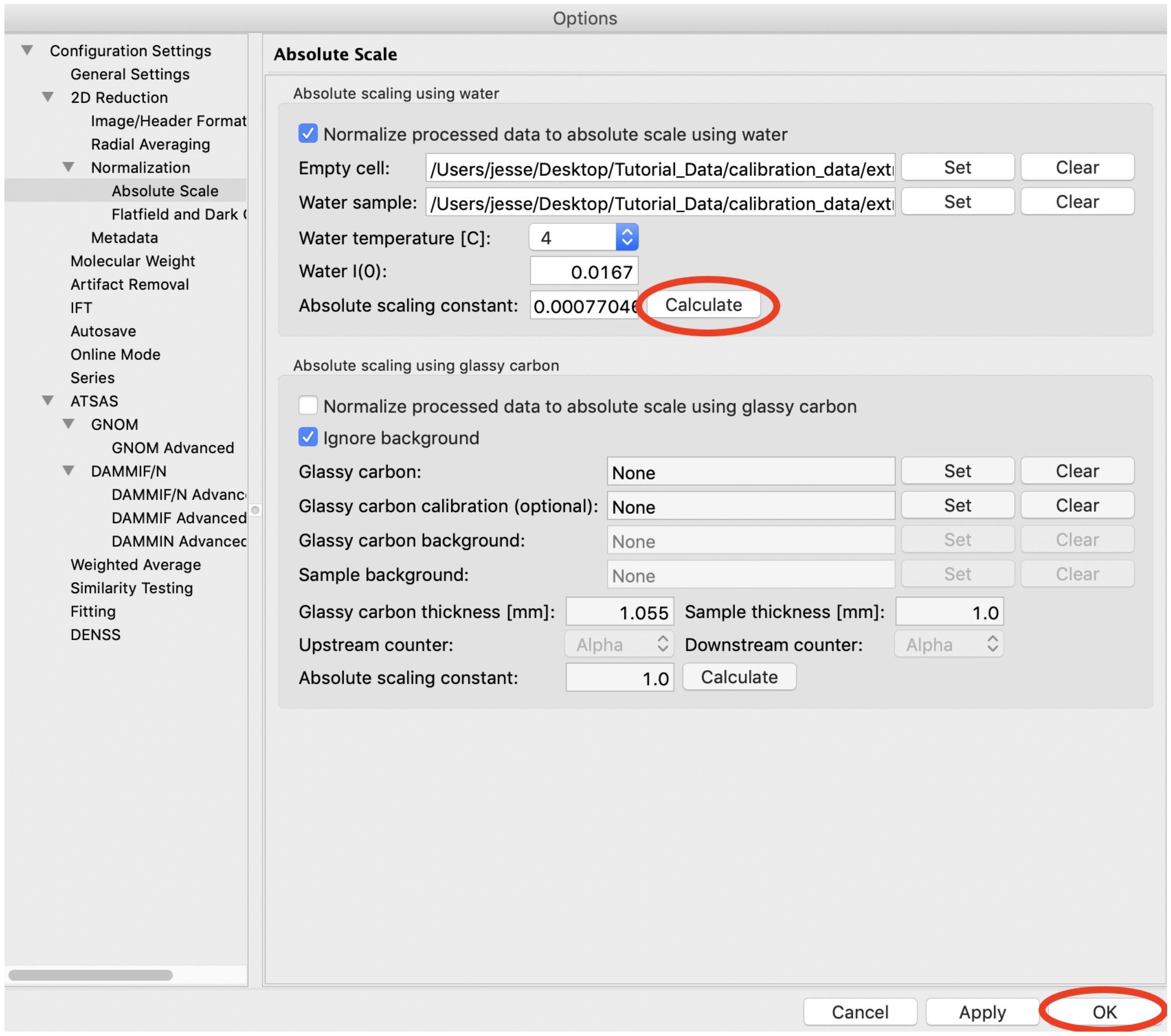
10. Click the Calculate button to calculate the Absolute Scaling Constant. You should get a value near 0.00077.

    - *Tip:* You can also use images to set the absolute scale. This may give worse results, as the signal to noise of the averaged file should be better than for a single image.
    - *Note:* It is important that you not change your normalization settings once you have set the absolute scaling constant. If you do, you will have to recalculate the absolute scaling constant. Also, make sure absolute

scale is turned off before you calculate the scale constant, otherwise you will get a bad scaling constant (see the manual for details).


Water temperature [C]: 4
Water I(0): 0.0167
Absolute scaling constant: 0.0007704

11. Check the "Normalize processed data to absolute scale" checkbox. Click "OK" to exit the advanced options window and save the changes.


Options

Absolute Scale

Absolute scaling using water

☑ Normalize processed data to absolute scale using water
Empty cell: /Users/jesse/Desktop/Tutorial_Data/calibration_data/ext
Water sample: /Users/jesse/Desktop/Tutorial_Data/calibration_data/ext
Water temperature [C]: 4
Water I(0): 0.0167
Absolute scaling constant: 0.00077046

Absolute scaling using glassy carbon

☐ Normalize processed data to absolute scale using glassy carbon
☑ Ignore background
Glassy carbon: None
Glassy carbon calibration (optional): None
Glassy carbon background: None
Sample background: None
Glassy carbon thickness [mm]: 1.055
Sample thickness [mm]: 1.0
Upstream counter: Alpha
Downstream counter: Alpha
Absolute scaling constant: 1.0

12. Save the settings for later use.

Setting absolute scale with glassy carbon

This section teaches you how to set up absolute scale using glassy carbon (NIST SRM 3600) as a reference. It assumes you have completed *Parts 1*, *2* and *3*. Note that you can use water (*Part 4*) or glassy carbon for absolute scale calibration in RAW. Glassy carbon is the more accurate approach, if available.

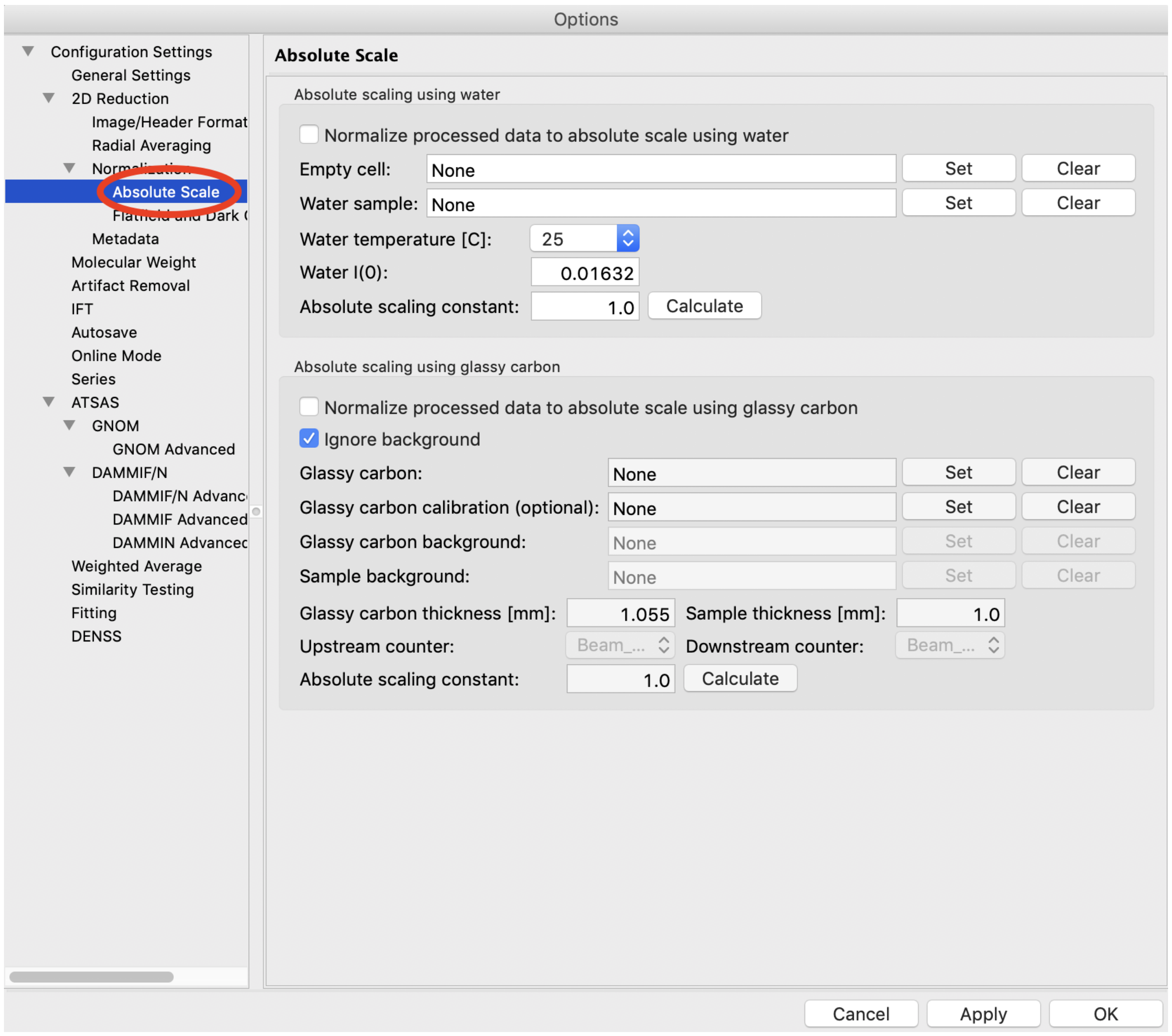
There are two ways to use glassy carbon as a standard in RAW. One way follows the NIST protocol, and will deliver the most accurate results. However, this method depends on all measurements having reliable flux measurements upstream and downstream of the sample. It also requires accurate measurements of the background of the glassy carbon measurement and the sample measurements. The second way is more similar to that used by water, in that it essentially ignores the background (assumes it to be small). This approach only requires regular normalization and a single measurement of the background for the glassy carbon sample.

A video version of this tutorial is available:

The written version of the tutorial follows.

Simple (ignoring background)

1. Load/use the settings from part 3 (without absolute scale set from water, part 4). If you haven't done those parts, the settings are saved available as **settings.cfg** in the **calibration_data** folder.
2. Plot both of the **glassy_carbon2_011_000x.tif.tif** files, where x is 1-2, on the main plot.
   - *Tip: Section 1 Part 1* of this tutorial document teaches you how to do this.
3. Average the **glassy_carbon** files you just loaded. Save the average in the **calibration_data** folder.
4. Open the Options window by selecting "Advanced Options" in the Options menu.
5. Click on the Absolute Scale section in the options list on the left.



Absolute Scale

Absolute scaling using water

- [ ] Normalize processed data to absolute scale using water

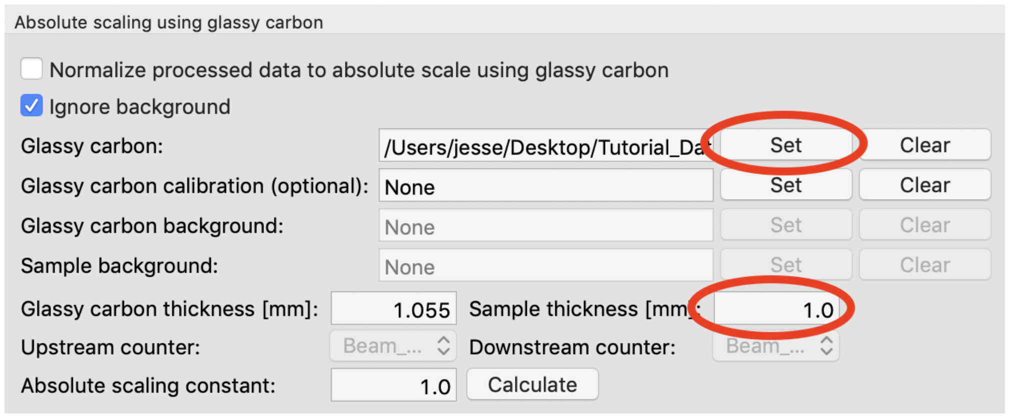
Empty cell: None
Water sample: None
Water temperature [C]: 25
Water I(0): 0.01632
Absolute scaling constant: 1.0

Absolute scaling using glassy carbon

- [ ] Normalize processed data to absolute scale using glassy carbon
- [x] Ignore background

Glassy carbon: None
Glassy carbon calibration (optional): None
Glassy carbon background: None
Sample background: None
Glassy carbon thickness [mm]: 1.055
Sample thickness [mm]: 1.0
Upstream counter: Beam_...
Downstream counter: Beam_...
Absolute scaling constant: 1.0

6. Click on the Glassy carbon "Set" button and select the **A_glassy_carbon2_011__0001.dat** file you just saved.
7. Set the Sample thickness to 1.0 mm.



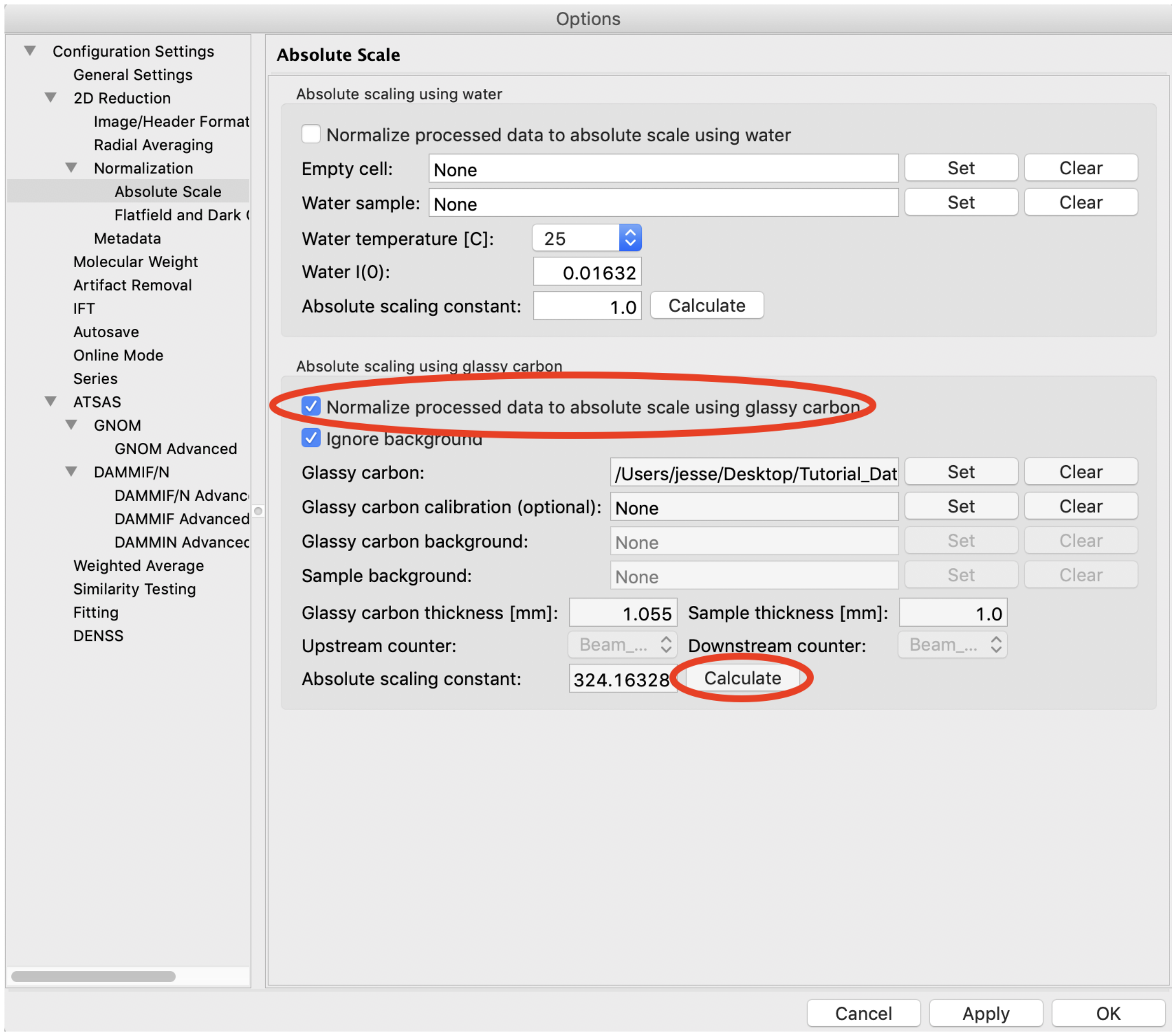
Absolute scaling using glassy carbon

- [ ] Normalize processed data to absolute scale using glassy carbon
- [x] Ignore background

Glassy carbon:	/Users/jesse/Desktop/Tutorial_Da	Set	Clear
Glassy carbon calibration (optional):	None	Set	Clear
Glassy carbon background:	None	Set	Clear
Sample background:	None	Set	Clear

Glassy carbon thickness [mm]: 1.055 Sample thickness [mm]: 1.0
Upstream counter: Beam_... Downstream counter: Beam_...
Absolute scaling constant: 1.0

8. Click "Calculate" button. You should get about 324.
   - *Note:* It is important that you not change your normalization settings once you have set the absolute scaling constant. If you do, you will have to recalculate the absolute scaling constant. Also, make sure absolute scale is turned off before you calculate the scale constant, otherwise you will get a bad scaling constant.
9. Check the "Normalize processed data to absolute scale using glassy carbon" checkbox.
10. Click "OK" to exit the advanced options panel, saving the changes.



Absolute Scale

Absolute scaling using water

☐ Normalize processed data to absolute scale using water

Empty cell: None

Water sample: None

Water temperature [C]: 25

Water I(0): 0.01632

Absolute scaling constant: 1.0

Absolute scaling using glassy carbon

✓ Normalize processed data to absolute scale using glassy carbon

✓ Ignore background

Glassy carbon: /Users/jesse/Desktop/Tutorial_Dat

Glassy carbon calibration (optional): None

Glassy carbon background: None

Sample background: None

Glassy carbon thickness [mm]: 1.055 Sample thickness [mm]: 1.0

Upstream counter: Beam_... Downstream counter: Beam_...

Absolute scaling constant: 324.16328

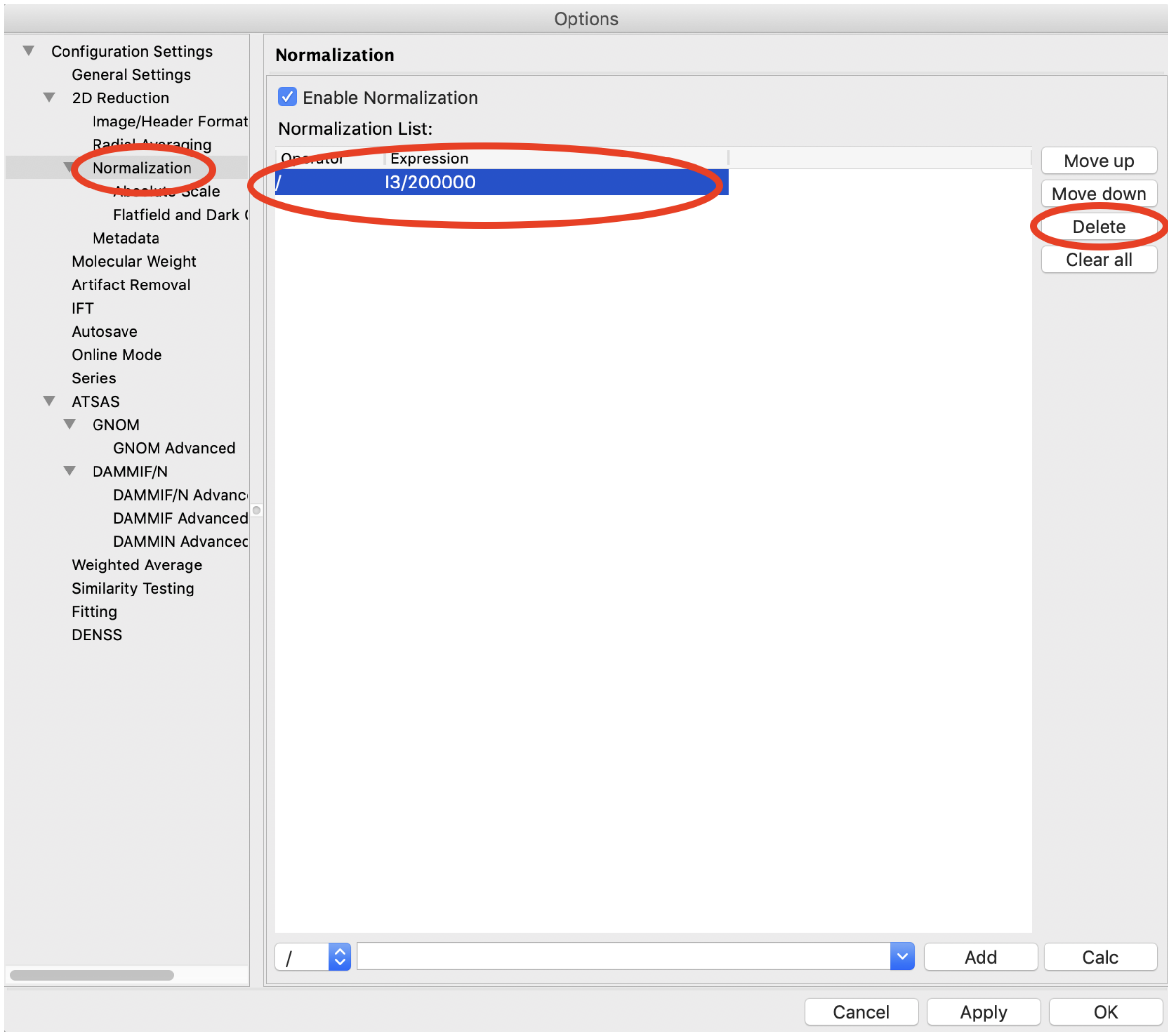
11. Save the settings for future use.

Full (NIST recommended)

Important note: All of the normalization (including flux, transmission, etc) happens through the absolute scale panel. You shouldn't have anything set in the Normalization panel (unless you are doing something like subtracting off a constant pedestal from the image).

Note: The calibration dataset used for the previous ('Simple') approach doesn't have the requisite data to use for the full approach. So we will use the data in the **calibration_data/extra** folder.

1. Load the **SAXS.cfg** file in the **calibration_data/extra** folder.
2. Open the Options window by selecting "Advanced Options" in the Options menu.
3. Click on the Normalization section in the options list on the left.
4. Remove any/all items in the Normalization List by highlighting them in the list and clicking the "Delete" button.



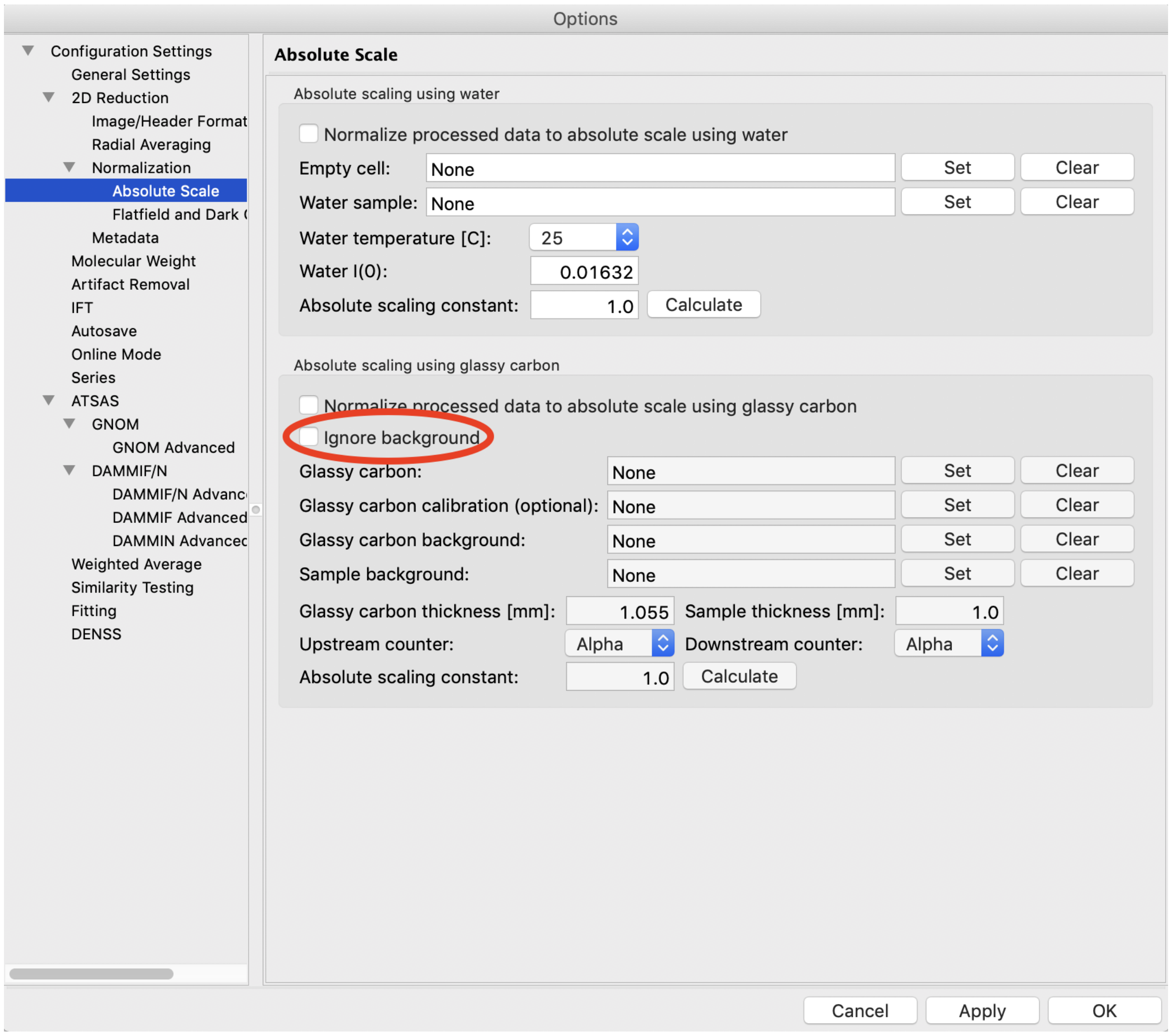
Normalization

✓ Enable Normalization

Normalization List:

Operator	Expression
/	I3/200000

5. Go to the Absolute Scale options section and turn off any absolute scaling already in place.
6. Click "OK" to exit the advanced options window and save the changes.
7. Plot the **glassy_carbon_41_001_0000.tiff** file.
   - *Tip: Section 1 Part 1* of this tutorial document teaches you how to do this.
8. Save the **glassy_carbon** profile in the **calibration_dat/extra** folder.
9. Plot and save the **vac_37_001_0000.tiff** and ** MT2_48_001_0000.tiff ** profiles.
10. Open the Options window and select the Absolute Scale section.
11. Uncheck the Ignore background checkbox.



Absolute Scale

Absolute scaling using water

Normalize processed data to absolute scale using water

Empty cell: None

Water sample: None

Water temperature [C]: 25

Water I(0): 0.01632

Absolute scaling constant: 1.0

Absolute scaling using glassy carbon

Normalize processed data to absolute scale using glassy carbon

Ignore background

Glassy carbon: None

Glassy carbon calibration (optional): None

Glassy carbon background: None

Sample background: None

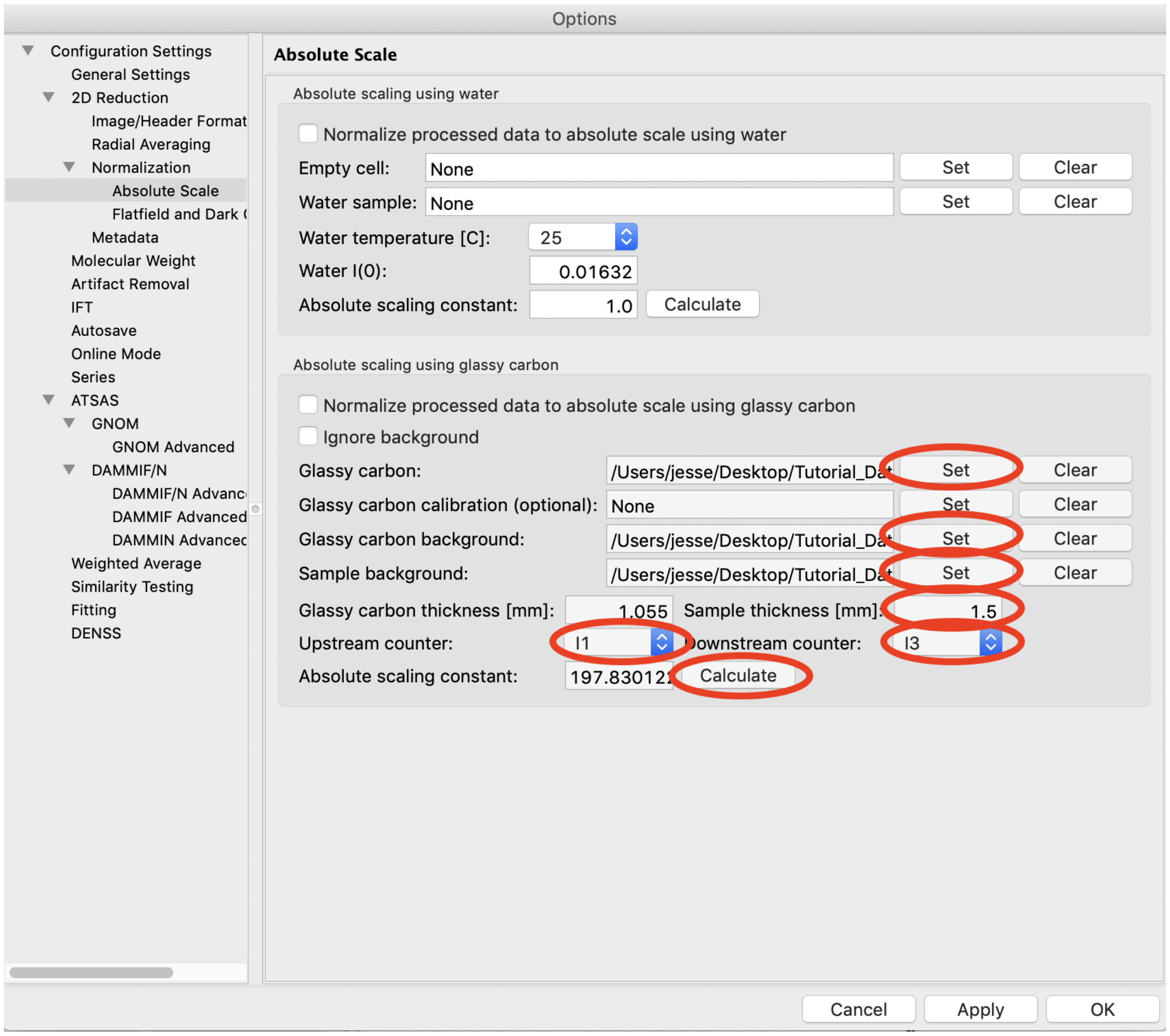
Glassy carbon thickness [mm]: 1.055 Sample thickness [mm]: 1.0

Upstream counter: Alpha Downstream counter: Alpha

Absolute scaling constant: 1.0

12. Click the Glassy carbon “Set” button and select the **glassy_carbon_41_001_0000.dat** file.
13. Click the Glassy carbon background “Set” button and select the **vac_37_001_0000.dat** file.
    - *Tip:* This is the instrument background from when the glassy carbon images were taken.
14. Click the Sample background “Set” button and select the **MT2_48_001_0000.tiff** file.
    - *Tip:* This is the instrument background from when sample images were taken, including the empty sample cell.
15. Set the Sample thickness to 1.5 mm.
16. Set the Upstream counter to I1.
17. Set the Downstream counter to I3.
18. Click the “Calculate” button. You should get an absolute scaling constant near 198.
    - *Note:* This approach will only work if the .dat files you select for the glassy carbon, glassy carbon background, and sample background contain the upstream and downstream counter values. This happens automatically with RAW. Otherwise, you should use images, which will have more noise, but should allow RAW to find all of the appropriate counter values.
    - *Note:* It is important that you not change your normalization settings once you have set the absolute scaling

constant. If you do, you will have to recalculate the absolute scaling constant. Also, make sure absolute scale is turned off before you calculate the scale constant, otherwise you will get a bad scaling constant (see the manual for details).



Absolute Scale

Absolute scaling using water

- [ ] Normalize processed data to absolute scale using water

Empty cell: None
Water sample: None
Water temperature [C]: 25
Water I(0): 0.01632
Absolute scaling constant: 1.0

Absolute scaling using glassy carbon

- [ ] Normalize processed data to absolute scale using glassy carbon
- [ ] Ignore background

Glassy carbon: /Users/jesse/Desktop/Tutorial_Da
Glassy carbon calibration (optional): None
Glassy carbon background: /Users/jesse/Desktop/Tutorial_Da
Sample background: /Users/jesse/Desktop/Tutorial_Da
Glassy carbon thickness [mm]: 1.055
Sample thickness [mm]: 1.5
Upstream counter: I1
Downstream counter: I3
Absolute scaling constant: 197.830122

19. Check the "Normalize processed data to absolute scale using glassy carbon" checkbox.
20. Click "OK" to exit the advanced options panel, saving the changes.
21. Save the settings for future use.

Comparison note:

We find that for the example data in the calibratin_data/extras folder, the two methods of glassy carbon calibration agree within ~1.5%. The best approach depends on how strong your background scattering is relative to the rest of the scattering in the system.

Setting a molecular weight standard

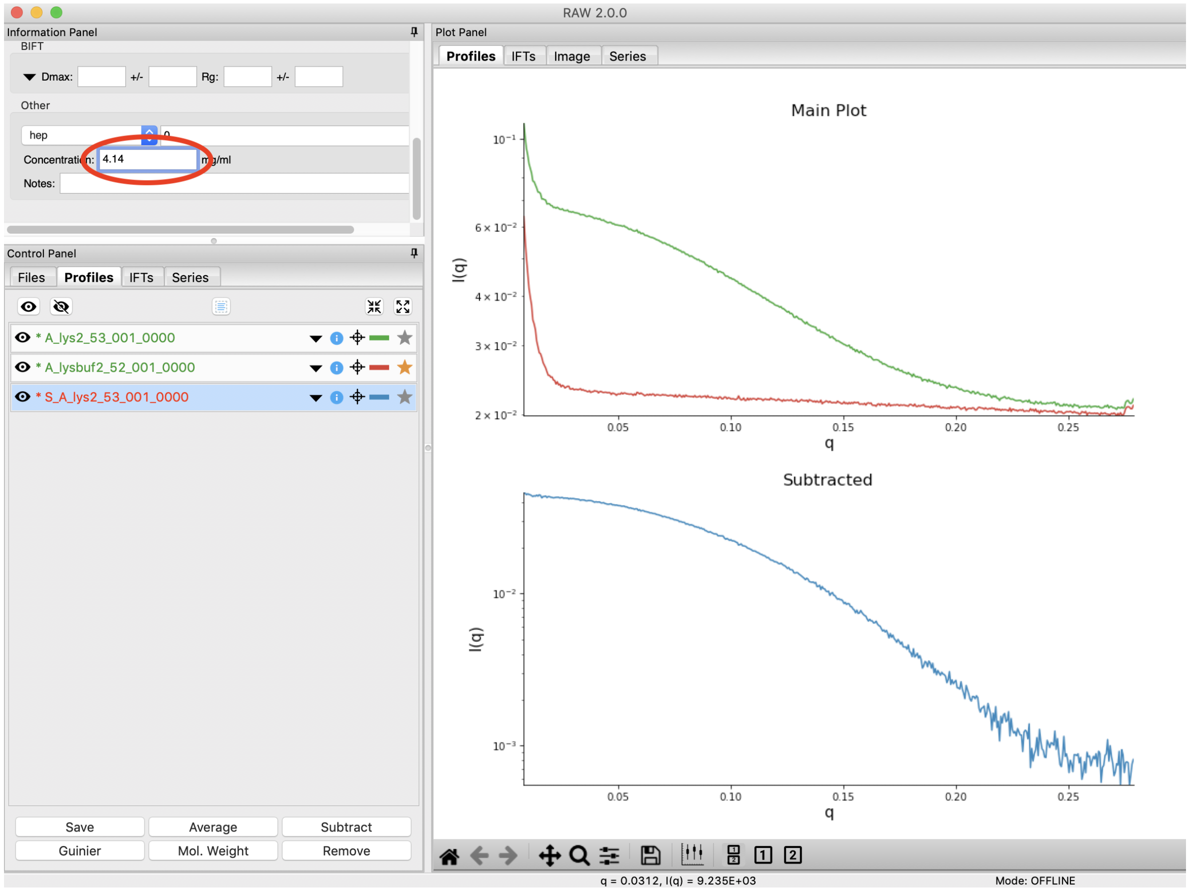
One method for determining molecular weight from a scattering profile is comparison to a known scattering profile with known molecular weight. This part will teach you how to set that known standard in RAW.

Note: The calibration dataset used for the first parts of this tutorial doesn't have the requisite data to use for this part. So we will use the data in the **calibration_data/extra** folder.

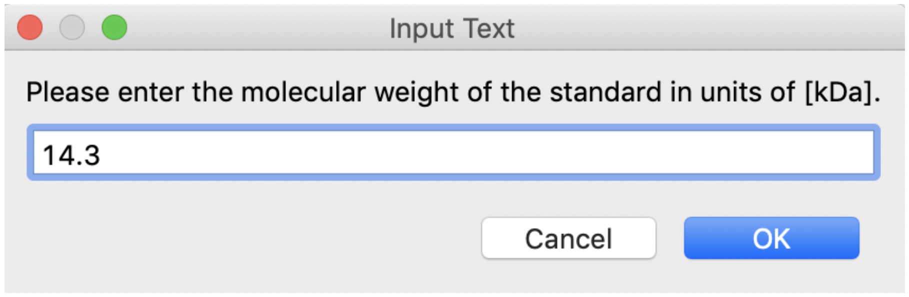
A video version of this tutorial is available:

The written version of the tutorial follows.

1. Load the **SAXS.cfg** file in the **calibration_data/extra** folder.
2. Plot all of the **lysbuf2_52_001_000x.tiff** files, where x is 0-9, on the main plot.
   - *Tip: Section 1 Part 1* of this tutorial document teaches you how to do this.
3. Average the **lysbuf2** files you just loaded.
4. Repeat steps 2-3 for the **lys2_52_001_000x.tiff** files.
5. Subtract the averaged buffer profile (**lysbuf2**) from the averaged sample profile (**lys2**).
   - *Tip: Section 1 Part 1* of this tutorial document teaches you how to do this.
6. Select the subtracted profile by clicking on it. In the information panel, set the concentration to 4.14 (this is concentration in mg/ml).
   - *Tip:* You will have to scroll down to the bottom of the Information panel to find the Concentration.



7. Perform a Guinier fit on the subtracted profile.
   - *Tip: Section 1 Part 2* of this tutorial document teaches you how to do this.
8. Right click on the subtracted profile and select the "Other Operations->Use as MW Standard" option.
9. Enter the molecular weight of the standard in kDa in the box that appears. For this lysozyme sample, the molecular weight is 14.3 kDa.



Please enter the molecular weight of the standard in units of [kDa].

14.3

10. Click "OK" to save the molecular weight standard.
11. Save the settings for future use.

5.4 SAXS Tutorial

5.4.1 Introduction

Overview

This tutorial covers basic principles and best practices in SAXS data processing, including:

- *Guinier analysis*
- *Molecular weight calculation*
- *Indirect Fourier transforms and P(r) functions*
- *Bead model (dummy atom) reconstructions*

This is not a tutorial on how to use RAW for this type of analysis. For that, please see the *RAW tutorial*.

Other useful materials

1. Video lectures from BioCAT's Everything BioSAXS workshops, which can help you learn more about best practices for SAXS data processing.

Notes

These tutorials depends on user feedback to get better. If you have questions, find bugs, or think a part of this tutorial is unclear, *please let the developers know.*

5.4.2 Guinier analysis

This tutorial covers basic principles and best practices for doing a Guinier analysis. This is not a tutorial on how to use RAW for this type of analysis. For that, please see the *RAW tutorial*.

Overview

Guinier's approximation states that at low-q the scattering profile can be approximated as follows:

$$I(q) \approx I(0)e^{-q^2R_g^2/3}$$

where R_g is the radius of gyration and I(0) is the intensity at zero scattering angle (q=0). Because of the exponential in the Guinier approximation, R_g and I(0) can be determined by performing a linear fit to a plot of $\ln(I)$ vs. q^2, called the Guinier plot. The region chosen for the linear fit is called the "Guinier region."

The Guinier fit provides information on the overall size of the molecule, and about the quality of your data.

Why do we do a Guinier fit?

The Guinier fit is done for two reasons. First, you get the R_g and I(0) parameters. The R_g tells you about the overall size of the molecule, while I(0) depends on the molecular weight times the concentration. These parameters are useful characterizations of your molecule, and are also needed to calculate other information from the SAXS data, including molecular weight and volume.

Second, and perhaps most importantly, many of the problems that can affect your SAXS data will show up in the low q region, including:

- Aggregation
- Radiation damage
- Interparticle interactions
- Buffer mismatch (in some cases)

These data quality issues cause deviations from linearity in the Guinier region. For this reason, having a good Guinier fit is one of the primary ways we assess the quality of SAXS data. A good Guinier is a strong indicator that your data is from a monodisperse sample and is otherwise free of artifacts. If you cannot obtain a good Guinier fit, or you can only obtain a good Guinier fit by excluding a significant amount of data at the lowest q values, then your data probably has one or more of the problems listed above and usually should not be used for further analysis.

How do we do a Guinier fit?

The Guinier approximation only holds when the exponential $\exp(-q^2R_g^2/3)$ is small. This means that in order to do a good Guinier fit, we need qR_g to be sufficiently small. The qR_g value at which the Guinier approximation starts to fail for a given scattering profile depends on the overall shape of the scatterer. Below is a figure showing the Guinier approximation (black), and the scattering intensity for a sphere, thin rod, and thin disc (all with the same R_g).

As you can see, the scattering intensity for the rod only agrees with the Guinier approximation until $qR_g \sim 1.0$, the sphere until $qR_g \sim 1.3$, and the disc until $qR_g \sim 1.7$. Thus, depending on the overall particle shape, you should fit different amounts of the low q data to have a good Guinier fit. (Note: the plot's x axis is $(qR_g)^2$, so $qR_g \sim 1.3$ appears as the dashed line at 1.69)

In practice, we fit both globular (sphere-like and disc-like) objects until $qR_g \sim 1.3$. while we fit highly extended (rod-like) objects until $qR_g \sim 1.0$. These values were chosen to have <10% error resulting from the deviation of actual shape from the Guinier approximation. The reason we accept that much deviation is that you also get uncertainty from fitting fewer points in your data, so there is a trade-off between how well the approximation works (fitting to smaller maximum qR_g) and how well you can fit your data (fitting to larger maximum qR_g).

The range of the Guinier fit is thus ideally from the earliest available q value until a maximum qR_g of 1.0 or 1.3. However, given that R_g is derived from the Guinier fit, how do you determine the correct maximum q value for the end of the fit? The answer is that the Guinier fit is done iteratively:

1. Guess a starting maximum q value for the fit.
2. Calculate the Guinier fit and get R_g.
3. If $q_{max}R_g > 1.3$ (or 1.0), reduce the maximum q. If $q_{max}R_g < 1.3$ (or 1.0), increase the maximum q.
4. Repeat steps 2 and 3 until you converge on a final maximum q.

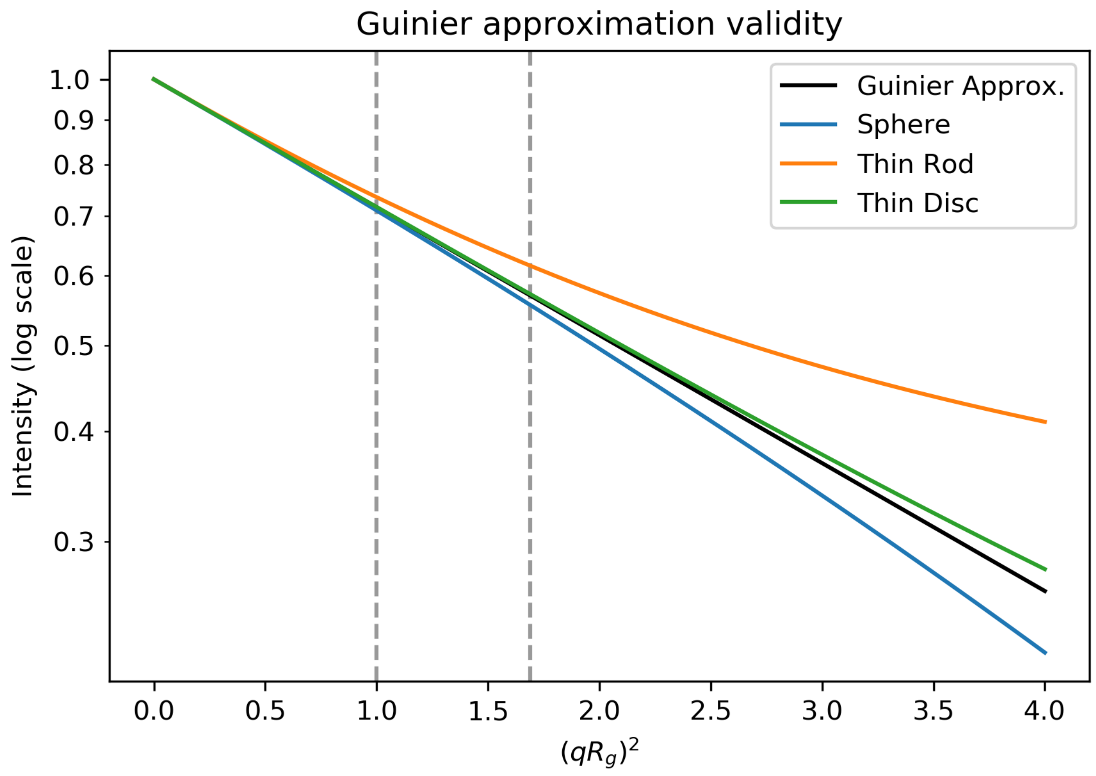


Fig. 1: Plot based on Figure 3.3 in [1]. Intensity for the geometric shapes from Table 3.4 in [1]. Dashed lines are at qR_g of 1.0 and 1.3 ($(qR_g)^2$ of 1.0 and 1.69).

Most software these days will do this iterative search for you, and for good quality data will provide you with a reasonable maximum q value that may need just a bit of manual refinement.

The minimum q value of a Guinier fit is usually determined by the minimum available q value in your data, which is set by the instrument on which you make the measurement. However, it is important to have a small enough minimum q to have a reasonable range for the Guinier fit. Typically, the minimum qR_g value should be $qR_g \leq 0.65$, though for globular systems it can be okay to have $qR_g \leq 1.0$. This means that the minimum q value required depends on the size of the system measured. In some cases, with particularly large systems, you may have to deliberately seek out an instrument that can measure to sufficiently low q.

If your data has quality issues at low q, which can be caused by the problems listed above, you may find that excluding those data from the fit can improve the quality of the fit. While this can be acceptable, you should proceed with caution when doing that, and always show the full data range on plots. The most acceptable case for this to happen is when the first few points are either too high or too low, but the rest of the range fits perfectly (see below for criteria for a good fit). In that case, those couple of points closest to the beamstop may have poor statistics or higher instrumental background scattering, and can usually be safely ignored.

Criteria for a good Guinier fit

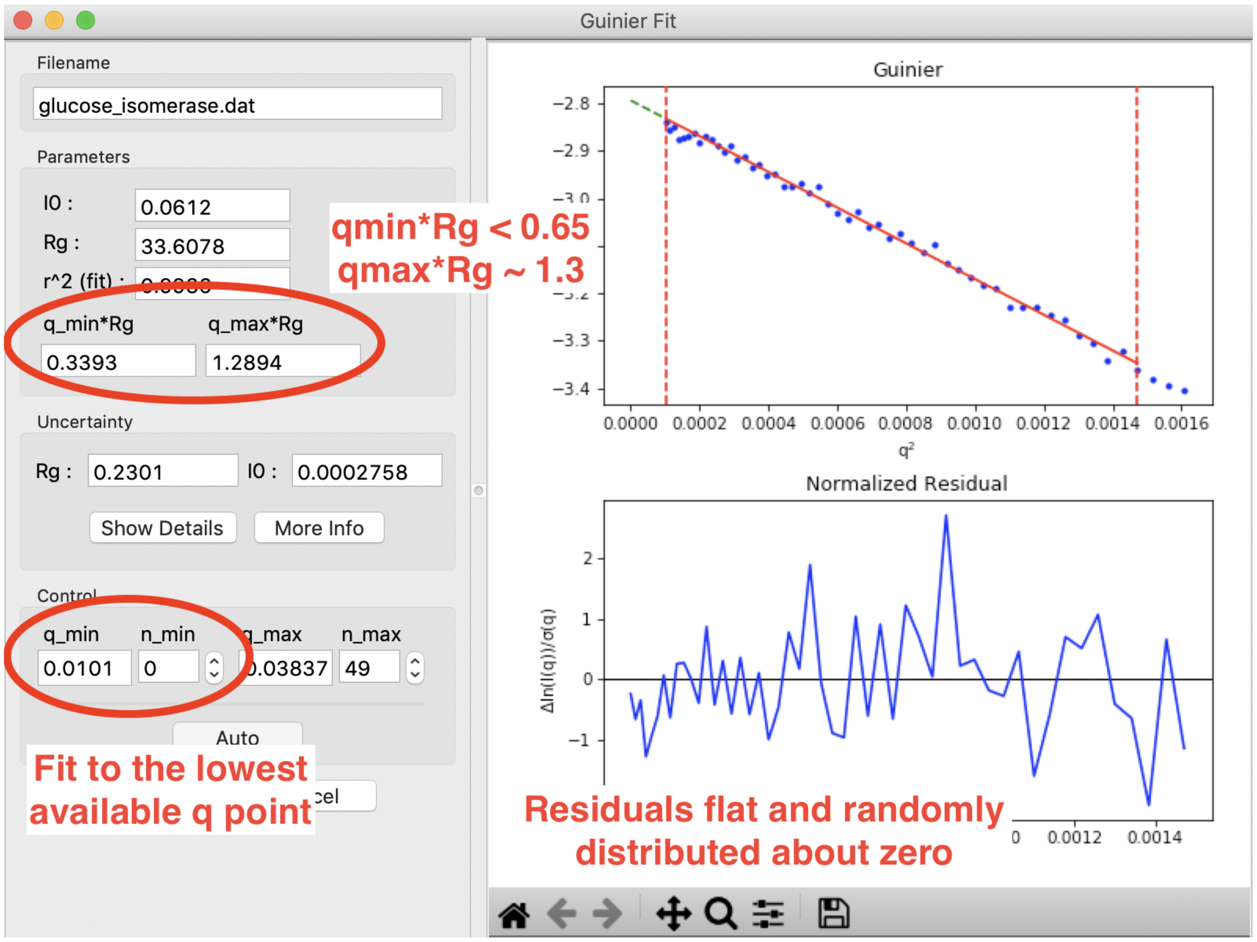
You are looking for four essential components in your Guinier fit:

1. $\mathbf{q_{min}R_g < 0.65}$ **.**
    - The minimum q of your fit, q_{min}, times the R_g of your fit should be less than 0.65. This criteria ensures you have enough q range to properly calculate the R_g and I(0) values. For globular particles (sphere- or disk-like), you can get away with $q_{min}R_g < 1.0$.
2. $\mathbf{q_{max}R_g \sim 1.3}$ **(globular) or** $\mathbf{q_{max}R_g \sim 1.0}$ **(extended).**
    - The maximum q of your fit, q_{max}, times the R_g of your fit should be less than 1.3 for globular (sphere- and disc-like) particles and less than 1.0 for extended (rod-like) particles. This ensures you remain in the linear range of the Guinier approximation for the fit.
3. **The Guinier fit residuals should be flat and randomly distributed about zero.**
    - If your residuals have a 'smile' (above zero near start and end of fit, below in the middle), or a 'frown' (below zero near start and end of fit, above in the middle), it indicates you have non-ideal data. The 'smile' is characteristic of aggregation, the 'frown' characteristic of interparticle repulsion.
4. **The fit extends to the lowest available q point.**
    - You shouldn't have to excluded very many points at the start of the fit. A few is generally fine, as the points nearest the beamstop can be noisy (depending on the exact details of the measurement). Having to exclude more than 3-5 points at the low q may indicate a problem with your data.

Having a good Guinier fit is a major quality check, and a good sign that your data is from a monodisperse sample with no interparticle interactions.

What is a bad Guinier fit, and what does it mean?

Non-linearities in your Guinier fit are indicative or problems in your sample. The type of non-linearity can indicate what the problem may be. The figure below gives a quick summary off the most common pathologies, more detail is available in the sections below.



Filename: glucose_isomerase.dat

I0 : 0.0612

Rg : 33.6078

q_min*Rg: 0.3393

q_max*Rg: 1.2894

Uncertainty: Rg : 0.2301 I0 : 0.0002758

q_min: 0.0101 n_min: 0 q_max: 0.03837 n_max: 49

Fig. 2: A Guinier fit done in RAW for glucose isomerase (available in the RAW Tutorial data). This shows what a good Guinier fit looks like. It has $q_{min}R_g < 0.65$, $q_{max}R_g \sim 1.3$, the normalized fit residual (bottom plot) is flat and randomly distributed about zero, and the fit extends to the lowest q point available.

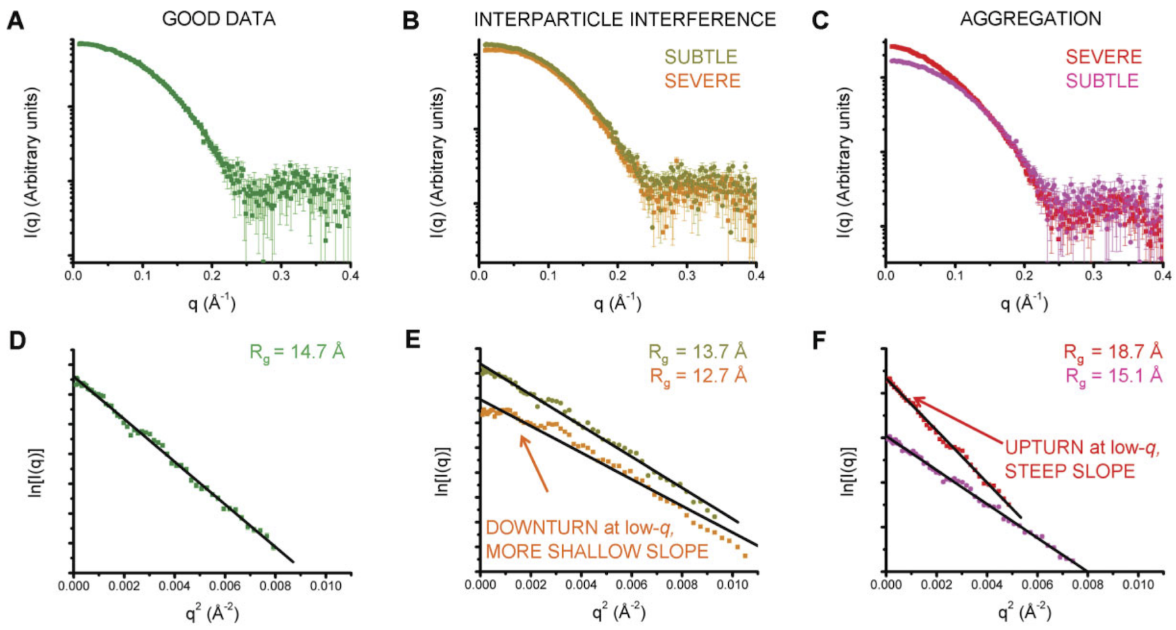


Fig. 3: Figure 3 from [2]. A and D show a good (monodisperse) scattering profile and Guinier fit. B and E show scattering profiles with varying degrees of interparticle interference. C and F show scattering profiles with varying degrees of aggregation.

Aggregation

Aggregation causes a characteristic upturn at low q. This can either be caused by aggregates initially present in your sample, or by radiation induced aggregation (radiation damage). The figure below shows what that might look like in your data.

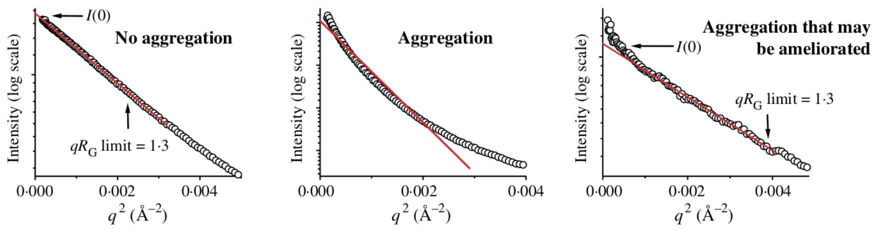


Fig. 4: Figure 23 from [3], showing no (left panel), large (middle panel) and small (right panel) amounts of aggregates seen in the Guinier fit.

The effect of aggregation can also be clearly seen in the fit residual, where it shows up as a 'smile', with the residual above zero near the start and end of the fit and below zero in the middle. The figure below illustrates this.

Repulsion

Repulsive interparticle interactions result in a structure factor that causes a downturn in the scattering profile at low q. This is typically caused by electrostatic interactions, and can often be remedied by either reducing the sample concentration or adding more salt to the buffer. These effects are also clearly seen in the fit residual, where it shows up as a 'frown', with the residual below zero near the start and end of the fit and above zero in the middle. This is shown in the figure below.

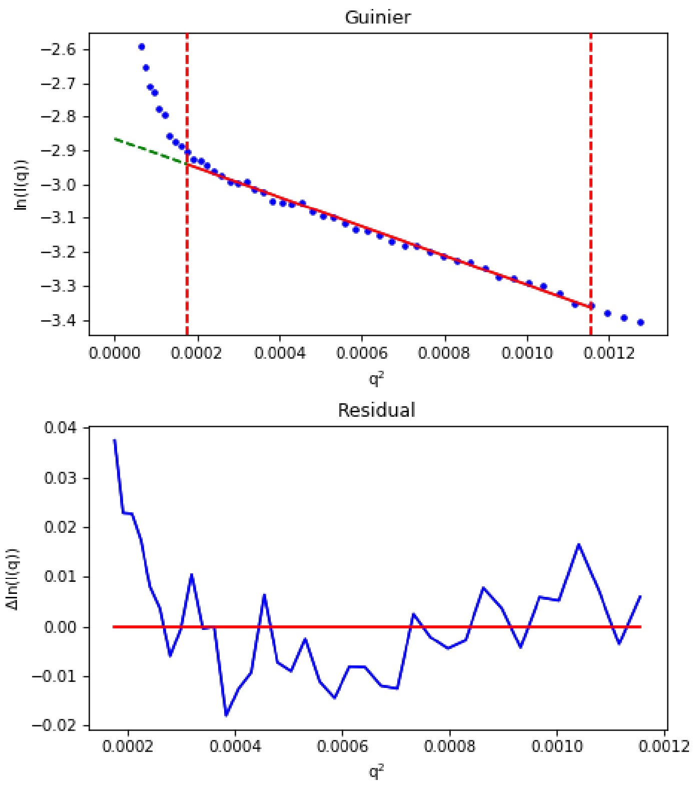


Fig. 5: For an aggregated sample, even with the worst low *q* values cut off from the fit, the residual (bottom plot) show the characteristic 'smile', rather than being flat and randomly distributed about zero.

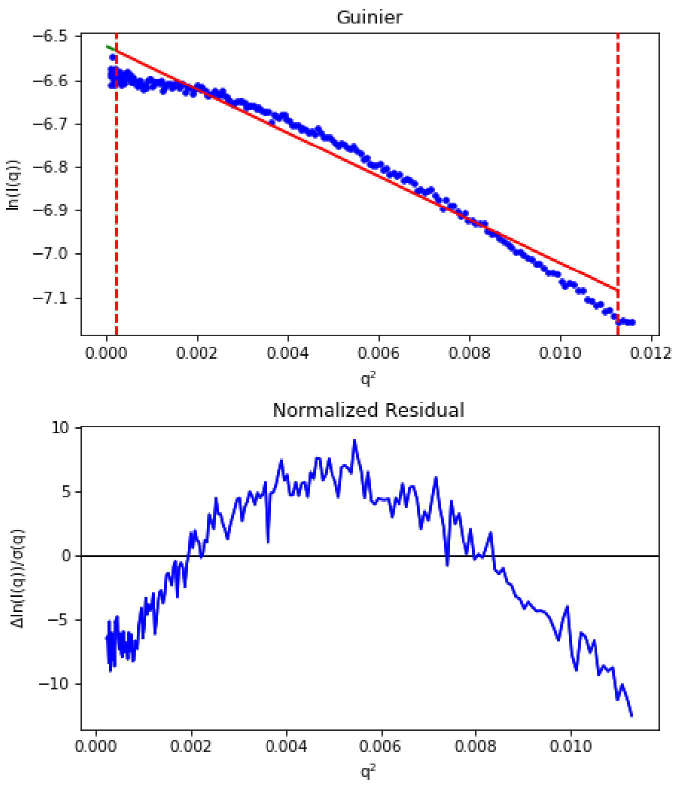


Fig. 6: For a repulsive sample, the residual (bottom plot) show the characteristic ‘frown’, rather than being flat and randomly distributed about zero.

Note that the above figure is an extreme example of repulsion, the downward curve may not be that obvious.

Bad buffer subtraction

Good SAXS data depends on subtracting away all scattering from the buffer and instrument background. If this subtraction is not good, you can end up with a downturn at low q (over subtraction) or an upturn at low q (under subtraction). This will look similar to aggregation or repulsion in the Guinier fit.

FAQ

What if I don't know my particle shape, should I fit to a maximum qR_g 1.3 or 1.0?

Very often, you don't know what your particle shape is before making a SAXS measurement (in fact, this is often one of the purposes of a SAXS measurement). In that case, start out by fitting to a maximum qR_g of 1.3. If that has a non-flat residual, reduce the fitting range to a maximum qR_g of 1.0. If the residual becomes flat upon reducing the maximum qR_g, then your particle is likely more extended than globular, and you should keep the maximum qR_g at 1.0. If reducing the maximum qR_g still leaves a non-flat residual, your data is showing signs of aggregation, repulsion, or some other issue.

My Guinier fit isn't great, can I still use my data?

If your Guinier fit isn't great, typically you shouldn't use the data. Even small amounts of aggregation (<1%) can affect things like the measured maximum dimension, and three dimensional reconstructions. While data with imperfect Guinier fits can be used in some specific cases, my general recommendation is to collect the data again.

My Guinier fit is a bit off, how can I fix it?

If your Guinier fit isn't ideal, typically the only way to fix this is to collect the data again, improving your sample/solution conditions. Below are a few things you can try to fix these issues.

For inherent aggregation:

1. Spin down your sample in a centrifuge at high speeds (~16000 g) for 5-10 minutes before data collection. In some cases, ultracentrifugation may help.
2. Use size exclusion chromatography coupled to SAXS instead of batch mode SAXS for in-line sample purification.
3. Reduce the concentration of your sample.
4. Re-purify your sample using size exclusion or ion exchange chromatography immediately before the SAXS measurement.

For radiation damage (often aggregation):

1. Add 1-5% glycerol.
2. Increase the flow/oscillation speed of your sample.
3. Reduce the exposure time and/or number of exposures.
4. Add radical scavengers like DTT.
5. Attenuate the incident x-ray beam.

For repulsion:

1. Add salt to the buffer to reduce repulsion.
2. Reduce the concentration of your sample.
3. Change the pH of your buffer.

For bad buffer subtraction:

1. Prepare matching buffer using dialysis.
2. Exchange buffer across a sizing column or desalting column.

References

1. Feigin, L., Svergun, Structure Analysis by Small-Angle X-ray and Neutron Scattering (1987).
2. Jacques, D. A. & Trewhella, J. (2010). Protein Sci. 19, 642–657. DOI: 10.1002/pro.35
3. Putnam, C. D., Hammel, M., Hura, G. L. & Tainer, J. a (2007). Q. Rev. Biophys. 40, 191–285. DOI: 10.1017/S0033583507004635

5.4.3 Molecular weight calculation

This tutorial covers basic principles and best practices for calculating molecular weight from SAXS data. This is not a tutorial on how to use RAW for this type of analysis. For that, please see the *RAW tutorial*.

Overview

There are a number of ways to calculate molecular weight from SAXS data. RAW supports four of the most common methods natively:

1. Molecular weight from absolute scaled I(0) [1].
2. Molecular weight by comparing to a reference standard [1].
3. Molecular weight from the Porod volume [2].
4. Molecular weight from the volume of correlation [3].

There are two additional methods supported in the ATSAS software, which RAW will show if ATSAS is installed:

1. Molecular weight estimation by comparison of scattering to known structures [4].
2. Molecular weight calculation by Bayesian inference from the other molecular weight methods [5].

All of these methods have advantages and disadvantages, and which to use and trust depends to a certain extent on the details of your data. Regardless, calculating molecular weight is important to verify the oligomeric state of your sample.

Why do we do calculate molecular weight?

SAXS molecular weight calculations are not terribly accurate, a usual rule of thumb is ~10% uncertainty (or more). For this reasons, SAXS should not be used to determine the molecular weight of your sample, something like multi-angle light scattering (MALS) is going to be much more accurate. The main reason to calculate molecular weight from SAXS data is to determine the oligomeric state of the protein in solution. Particularly if you're studying a system that can form homodimers or higher order oligomers, the SAXS molecular weight calculations should be accurate enough to tell what oligomeric state you're measuring. For this reason, the molecular weight is an important diagnostic for verifying that what you think is in solution is actually what's in solution.

How do we do calculate molecular weight?

Because there are so many methods for calculating molecular weight from SAXS data, we will only present a short summary of the methods listed above here. Generally speaking the methods can be broken up into two categories: concentration dependent and concentration independent. The concentration dependent methods require knowing the concentration of the sample in the SAXS cell, which often means they are incompatible with SEC-SAXS measurements. The concentration independent methods do not require knowledge of the concentration, and so are useful for SEC-SAXS measurements.

Molecular weight from absolute scaled I(0)

SAXS data is often put on an absolute scale, so that the intensity values have units (by convention of cm^{-1}). This means that the I(0) value can be directly related to the number of electrons in the sample and the concentration and contrast of the sample [6]. If the concentration and contrast of the sample is known, the total number of electrons can be calculated, and then that can be used to determine the molecular weight of the sample. The accuracy of this approach can be improved by measuring or calculating (for example, using MULCh) the protein partial specific volume. If necessary, adjusting the electron densities of the sample and buffer will also improve the accuracy. Molecular weight is calculated as:

$$MW = \left(\frac{N_A I(0)}{c\Delta\rho_M^2}\right)$$

where *MW* is the molecular weight, *c* is the concentration, N_A is Avagadro's constant, and $\Delta\rho_M$ is the scattering contrast per mass.

The accuracy of this method was assessed in [1] where they found it to be <~10% for most proteins.

Molecular weight by comparing to a reference standard

The scattering at zero angle, I(0) is proportional to the molecular weight of the macromolecule, and the concentration and contrast of the macromolecule in solution. If a reference sample of known molecular weight and concentration is measured, it can be used to calibrate the molecular weight of any other scattering profile with known concentration (assuming constant contrast between reference and sample, and a monodisperse sample) [1]. Molecular weight is calculated as:

$$MW_m = \left(\frac{I(0)_m}{c_m}\right)\left(\frac{MM_{st}}{\left(\frac{I(0)_{st}}{c_{st}}\right)}\right)$$

where *MW* is the molecular weight, *c* is the concentration, and the subscripts *m* and *st* designate quantities from the macromolecule of interest and the standard respectively.

Molecular weight from Porod volume

The Porod volume is nominally the excluded volume of the macromolecule in solution. It can be calculated directly from the scattering profile, first by calculating the Porod invariant:

$$Q_p = \int_0^\infty q^2 I(q) dq$$

where Q_p is the Porod invariant. The Porod volume is then calculated as:

$$V_p = \frac{2\pi^2 I(0)}{Q_p}$$

where *I(0)* is the scattering at zero angle. Once this volume is determined, the molecular weight is determined by simply multiplying by the macromolecule's density.

Because Q_p is determined by an integral from 0 to ∞, the actual integration must be approximated in some way. There are various approaches, including the the 'Porod' and 'Qp' methods implemented in the ATSAS datmw program. RAW uses the SAXSMoW 2 approach described in [2], which applies a correction factor to the Porod volume based on the available range of data in the scattering profile. Then an average protein density is used to calculate the molecular weight. The accuracy of this method can be improved if a more accurate density for the macromolecule is known.

In [2] they found a median uncertain of 12% for calculated molecular weight from globular proteins. By contrast, in [5] they found using a large test set of simulated curves that the median molecular weight of this method was 3% high, and the median absolute deviation from this method was 5%. The uncertainty was higher for data with simulated noise. It seems reasonable to say the uncertainty in molecular weight is ~10% for most systems, though there are outliers.

Molecular weight from volume of correlation

In [3] they defined the volume of correlation as

$$V_c = \frac{I(0)}{\int_0^\infty qI(q)dq}$$

They empirically found the ratio V_c^2/R_g is logarithmically proportional to molecular weight, with the following formula:

$$MW = \left(\frac{V_c^2/R_g}{c}\right)^k$$

where *c* and *k* are empirically determined constants via fitting results from theoretical scattering profiles. They found different constants for proteins and RNA. For proteins, $c = 0.1231$ and $k = 1$ while for RNA $c = 0.00934$ and $k = 0.808$ (note: *c* and *k* are defined slightly differently in the original paper).

In [3] they found a molecular weight uncertainty of ~5% from theoretical profiles and ~10% from experimental profiles. By contrast, in [5] they found using a large test set of simulated curves that the median molecular weight of this method was 2% low, and the median absolute deviation from this method was 7%. The uncertainty was higher for data with simulated noise. It seems reasonable to say the uncertainty in molecular weight from this is ~10% for most systems, though there are outliers.

Molecular weight by comparison to known structures

In [4] they describe a machine learning method that categorizes SAXS data into shape categories based on comparison with a catalog of known structures from the PDB. By finding the nearest structures in shape and size (also the name of the method: Shape&Size), they can obtain estimates for the molecular weight of the sample.

In [4] they found that, for the theoretical scattering profiles used for testing, the method calculated molecular weights within 10% of the expected value for 90% test data. In [5] they found that for the test dataset the median molecular weight was correct and the median absolute deviation was 4%. Again, it seems reasonable to say that the uncertainty in molecular weight from this method is ~10% for most systems, though there are outliers.

Molecular weight from Bayesian inference

In [5] they describe a method for calculating a molecular weight using Bayesian inference with the molecular weight calculations from the Porod volume, volume of correlation, and comparison to known structures methods as the evidence. Essentially, it takes a large test dataset of theoretical scattering profiles, calculates the molecular weight for each using each method, then creates a probability distribution for each method that describes the probability of obtaining a particular calculated molecular weight given the true molecular weight. These probabilities are combined across all the methods, and the most likely molecular weight is thus estimated.

They found that for the theoretical scattering profiles used, the median molecular weight from this method was accurate and the median absolute deviation was 4%. Overall, they reported that it was more accurate than any individual method. It may be that the uncertainty in this method is usually closer to ~5% than 10% for the other methods.

What are the strengths and weaknesses of different MW methods?

Each method has distinct strengths and weaknesses, and tend to be better with certain types of data. Every method requires a good determination of I(0), and all of the concentration independent methods require R_g, which generally means a good Guinier fit is required in all cases. Also, in [5] it is reported that all of the concentration independent methods struggle with flat and ring-shaped proteins.

Molecular weight from absolute scaled I(0)

Advantages:

- Can be highly accurate when all parameters are well known.
- With correct parameters can be used for proteins or RNA/DNA.

Disadvantages:

- Requires accurate sample concentration.
- Requires accurate absolute calibration.
- Best when the scattering contrast of the macromolecule is well known.
- Best when partial specific volume is well known.

Molecular weight by comparing to a reference standard

Advantages:

- Can be highly accurate for similar standards and samples under the same conditions.
- With correct standards can be used for proteins or RNA/DNA.

Disadvantages:

- Requires accurate sample concentration.
- The reference standard should have the same scattering contrast as the sample (i.e.. is in a similar buffer).
- The standard and sample should be similar shapes (i.e. the same partial specific volume).

Molecular weight from Porod volume

Particular to the MoW method described in [2].

Advantages:

- Accurate for most molecule shapes [5].
- More accurate than the volume of correlation method when the signal to noise level of the data is reasonable [5].

Disadvantages:

- Should struggle when the macromolecule is flexible or extended in solution (though [5] found this was not always the case).
- May need to have the protein density adjusted in some cases (default: 0.83 kDa/^{3})
- Will fail if the macromolecule is not a protein.
- Sensitive to subtraction errors.

Molecular weight from volume of correlation

Advantages:

- More accurate than other methods when signal to noise is low [5].
- More accurate than other methods when there are subtraction errors [5].
- Should be accurate for flexible or extended extended macromolecules [3] (though [5] found that was not always the case).
- Works for both proteins and RNA/DNA.

Disadvantages:

- Less accurate than other methods for high signal to noise data [5].
- Less accurate than the Porod volume MoW method for extended macromolecules [5].
- Large uncertainty for macromolecules less than ~15-20 kDa (based on experience, and the fact that the empirical coefficients were generated from size ranges 20 kDa and larger).
- Doesn’t work for protein nucleic acid complexes.
- The integral of *qI(q)* needs to have converged (see figure below).

Molecular weight by comparison to known structures

Advantages:

- Most accurate individual concentration independent method except for low signal to noise data [5].
- Relatively accurate when there are subtraction errors.

Disadvantages:

- Provides no result for flexible systems.
- Only works for proteins.

Molecular weight from Bayesian inference

Advantages:

- More accurate than individual concentration independent methods in most cases [5].

Disadvantages:

- Struggles with significant subtraction errors.
- Only works for proteins.

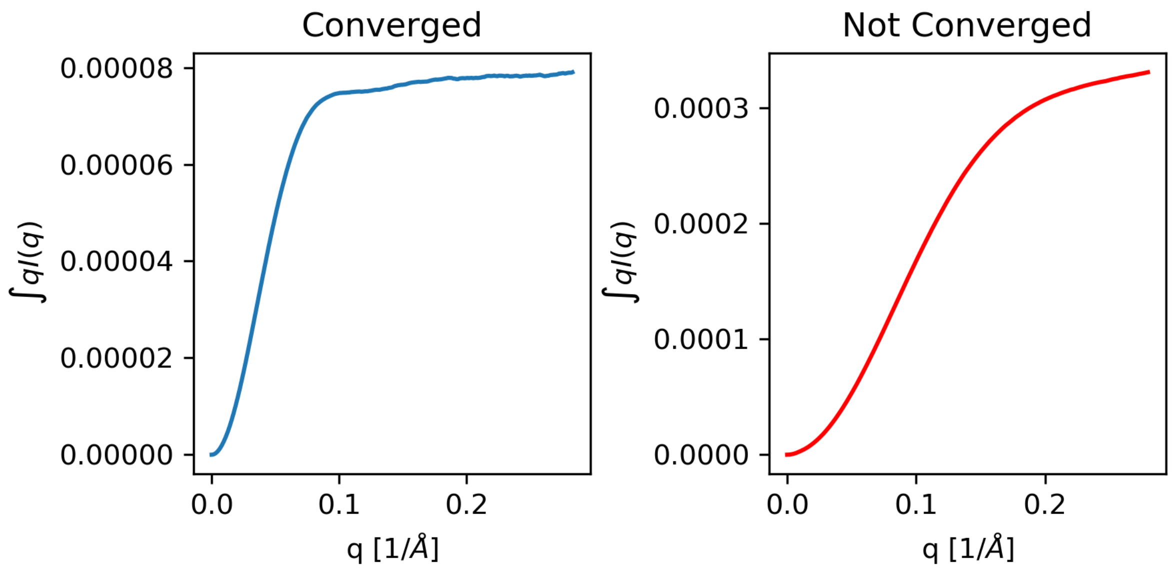


Fig. 7: Both plots show the integral of *qI(q)* as a function of *q*. The plot on the left shows data where the integrated value has converged, i.e. it is essentially unchanging at high q as q increases. The plot on the right shows data where the integrated value has not converged, i.e. it is increasing at high q as q increases. The data on the right will not give an accurate molecular weight by the volume of correlation method.

FAQ

I don't get the expected molecular weight from my SAXS data, what do I do?

Molecular weight from good SAXS data has relatively large uncertainties (often ~10%), and for low signal to noise data can be significantly worse. What you need to do in the case where it's wrong depends on what you're trying to determine.

If you know your sample is stable in solution (not prone to aggregation/degradation), or you have evidence it was all in one state (such as elution in a single sharp peak in a SEC-SAXS experiment), if your MW is a bit off that's okay. In this case you're just trying to determine the oligomeric state of the sample. If you can clearly make the distinction, then you're fine. If not, you need to measure the molecular weight with a different method.

If your sample is unstable in solution (prone to aggregation/degradation), you need to measure the molecular weight of your sample with another method. Good methods include multi-angle light scattering (MALS) or analytical ultra centrifugation (AUC). The best approach is to do a SEC-MALS-SAXS experiment, where MALS data is collected on the same elution as the SAXS data, removing any question about changes in the sample between the MALS and SAXS measurements.

I need to determine if *<small molecule>* is bound to *<big molecule>*. Or I want to determine the binding stoichiometry. Can I do that with SAXS?

It depends to a certain extent on the relative sizes of your molecules. However, if you have something that's small, say ~20 kDa, and something much larger, say ~250 kDa, SAXS data is unlikely to be reliable enough to accurately determine the difference between bound and unbound (250 kDa or 270 kDa), or between 1:1 and 2:1 binding (270 kDa or 290 kDa). In this case you should pursue additional characterization of the molecular weight. The best approach is to do a SEC-MALS-SAXS experiment, where MALS data is collected on the same elution as the SAXS data.

References

1. Mylonas, E. & Svergun, D. I. (2007). J. Appl. Crystallogr. 40, s245–s249. DOI: 10.1107/S002188980700252X
2. Piiadov, V., de Araújo, E. A., Oliveira Neto, M., Craievich, A. F. & Polikarpov, I. (2018). Protein Sci. 2–22. DOI: 10.1002/pro.3528
3. Rambo, R. P. & Tainer, J. A. (2013). Nature. 496, 477–481. DOI: 10.1038/nature12070
4. Franke, D., Jeffries, C. M. & Svergun, D. I. (2018). Biophys. J. 114, 2485–2492. DOI: 10.1016/j.bpj.2018.04.018
5. Hajizadeh, N. R., Franke, D., Jeffries, C. M. & Svergun, D. I. (2018). Sci. Rep. 8, 7204. DOI: 10.1038/s41598-018-25355-2
6. Orthaber, D., Bergmann, A. & Glatter, O. (2000). J. Appl. Crystallogr. 33, 218–225. DOI: 10.1107/S0021889899015216

5.4.4 Indirect Fourier Transform (IFT) and the P(r) function

This tutorial covers basic principles and best practices for doing an Indirect Fourier Transform (IFT) to get a P(r) function. This is not a tutorial on how to use RAW for this type of analysis. For that, please see the RAW tutorial for *GNOM* and *BIFT*.

Overview

The SAXS scattering profile is measured in reciprocal distance space, as I(q) where q has units of one over distance (usually 1/Angstrom or 1/nm). We can apply a Fourier transform to the data to get information in real space about the macromolecule, as:

$$P(r) = \frac{r^2}{2\pi^2}\int_0^\infty q^2 I(q)\frac{\sin(qr)}{qr}dq$$

This produces the P(r) function, also called the pair distance distribution function. Essentially, the P(r) function is the r^2 weighted histogram of all possible pairs of electrons in the sample.

Why do we do an IFT?

The P(r) function contains valuable information about the shape and size of a macromolecule. First, in doing the P(r) function we get an estimate of the maximum dimension of the macromolecule (D_{max}). It also provides another, potentially more accurate, way to calculate the R_g and I(0). The shape of the P(r) function can also be directly interpreted in terms of the shape of the macromolecule, providing information about the overall shape, such as globular vs. rod-like, or whether the macromolecule contains multiple domains. Also, the P(r) function and derived parameters such as D_{max} are required for many advanced analysis techniques including ab-initio reconstructions of the shape.

Additionally, the P(r) function is sensitive to data quality issues, particularly aggregation and interparticle interference. Thus, doing an IFT is another quality check on your data. If you cannot obtain a good P(r) function, then your data probably has one of those issues and usually should not be used for further analysis.

How do we do an IFT?

The equation above cannot be used to directly calculate the P(r) function. The finite extent of our measurement (and measurement noise) means that a direct Fourier transform of I(q) will distort the true P(r) function by introducing truncation artifacts. Instead, the typical approach is to fit the P(r) function against the data. First, you generate the scattering intensity for a given P(r) function as:

$$I(q) = 4\pi \int_0^{D_{max}} P(r)\frac{\sin(qr)}{qr}dr$$

Using this equation you generate the P(r) function that yields the best fit to the data. The fitting criteria include both the actual fit, usually as measured by χ^2, and regularization parameters. These regularization parameters allow you to add back in information to improve the P(r) function. Typical examples of the regularization parameters include perceptual criteria such as:

- Smoothness of the P(r) function.
- Positivity of the P(r) function.
- Whether the solution changes significantly when changing the weighting of the regularization parameters.

Determining a P(r) function in this way thus requires determining three things:

1. The maximum dimension, D_{max}, of the sample, as that determines the upper bound of the integral above.
2. The weighting parameter, usually called α, that determines the relative contribution of χ^2 and the perceptual criteria to the overall fit quality.
3. The P(r) function that yields the best fit to the data, given the particular D_{max} and α values.

This is a complicated problem, and there are a number of different programs out there for finding a P(r) function. These programs all typically involve searching a set of possible D_{max} values and α values to find which yields the best overall fit. RAW natively supports one method for finding the P(r) function, the Bayesian Indirect Fourier Transform (BIFT) [1], which provides a completely automated determination of D_{max} and α.

The most popular method for determining the P(r) function is the GNOM [2] software in the ATSAS package, which RAW provides an interface to if ATSAS is installed. Below we discuss how to determine a good P(r) function using GNOM, after discussing the criteria for a good P(r) function.

Criteria for a good P(r) function

We employ the following criteria to determine if a P(r) function is good:

1. **The P(r) function falls gradually to zero at D_{max}.**
2. **The P(r) function fits the measured scattering profile.**
3. **The P(r) function goes to zero at $r = 0$ and $r = D_{max}$.**

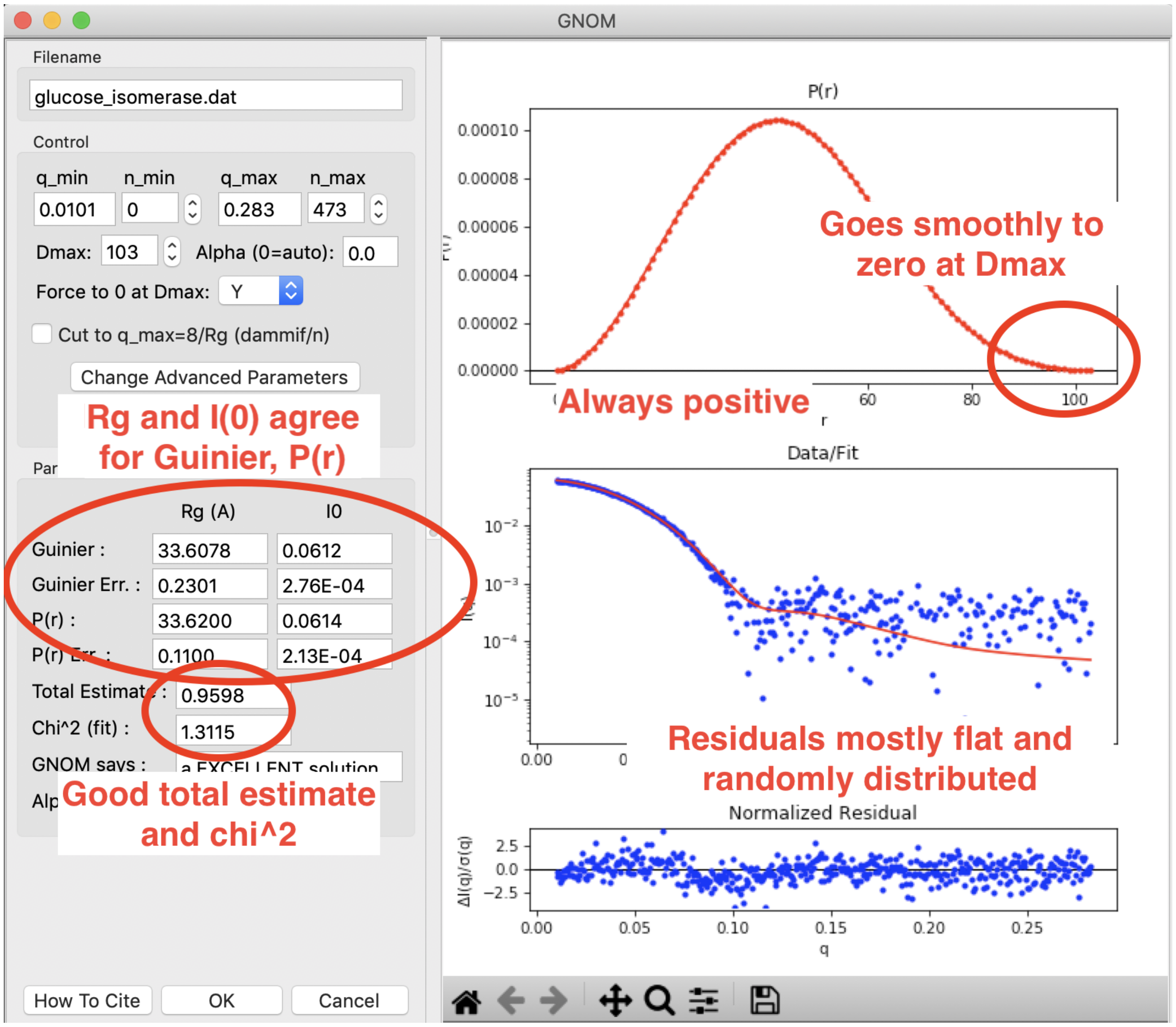
Additionally, the following criteria usually apply:

1. **The R_g and I(0) from the Guinier fit and the P(r) function agree well.**
2. **The P(r) function is always positive.**

A more thorough discussion of each criterion is given below.

The P(r) function falls gradually to zero at D_{max}

This is perhaps the most important and most subjective criterion for determining whether you have a good P(r) function. The idea is straightforward: macromolecules do not have perfectly sharp boundaries. Because they have side chains that stick out, and have some amount of flexibility in solution (even if limited to solvent exposed side chains), there is no distance at which you go from many electrons pairs to no electron pairs within the macromolecule. As such, the P(r) function should gradually approach zero at the maximum dimension, rather than being cut off.


GNOM
Filename
Control
q_min
n_min
q_max
n_max
Dmax:
Alpha (0=auto):
Force to 0 at Dmax:
Cut to q_max=8/Rg (dammif/n)
Change Advanced Parameters
Rg and I(0) agree for Guinier, P(r)
Rg (A)
I0
Good total estimate and chi^2
How To Cite
OK
Cancel
P(r)
Goes smoothly to zero at Dmax
Always positive
r
Data/Fit
Residuals mostly flat and randomly distributed
Normalized Residual
ΔI(q)/σ(q)
q

Filename: glucose_isomerase.dat

q_min: 0.0101, n_min: 0, q_max: 0.283, n_max: 473

Dmax: 103, Alpha (0=auto): 0.0

Force to 0 at Dmax: Y

	Rg (A)	I0
Guinier :	33.6078	0.0612
Guinier Err. :	0.2301	2.76E-04
P(r) :	33.6200	0.0614
P(r) Err. :	0.1100	2.13E-04

Total Estimate : 0.9598

Chi^2 (fit) : 1.3115

GNOM says : a EXCELLENT solution

Fig. 8: A P(r) function done in RAW using GNOM for glucose isomerase (available in the RAW Tutorial data). This shows what a good P(r) function looks like. The function goes smoothly to zero at D_{max}, it is always positive, there is good agreement between the Guinier and P(r) R_g and I(0) values, and the residuals are mostly flat and randomly distributed. You can see a small systematic deviation in the residuals below q~0.125. This could be smoothed out by reducing the α value, which may be slightly over weighting the regularization parameters vs. the actual fit to the data.

Essentially, if this criterion is met you have picked an appropriate D_{max} for the system. If you underestimate the D_{max}, then the P(r) function has an abrupt descent to zero, while an overestimated D_{max} usually shows an oscillation about zero. This is shown in the figure below.

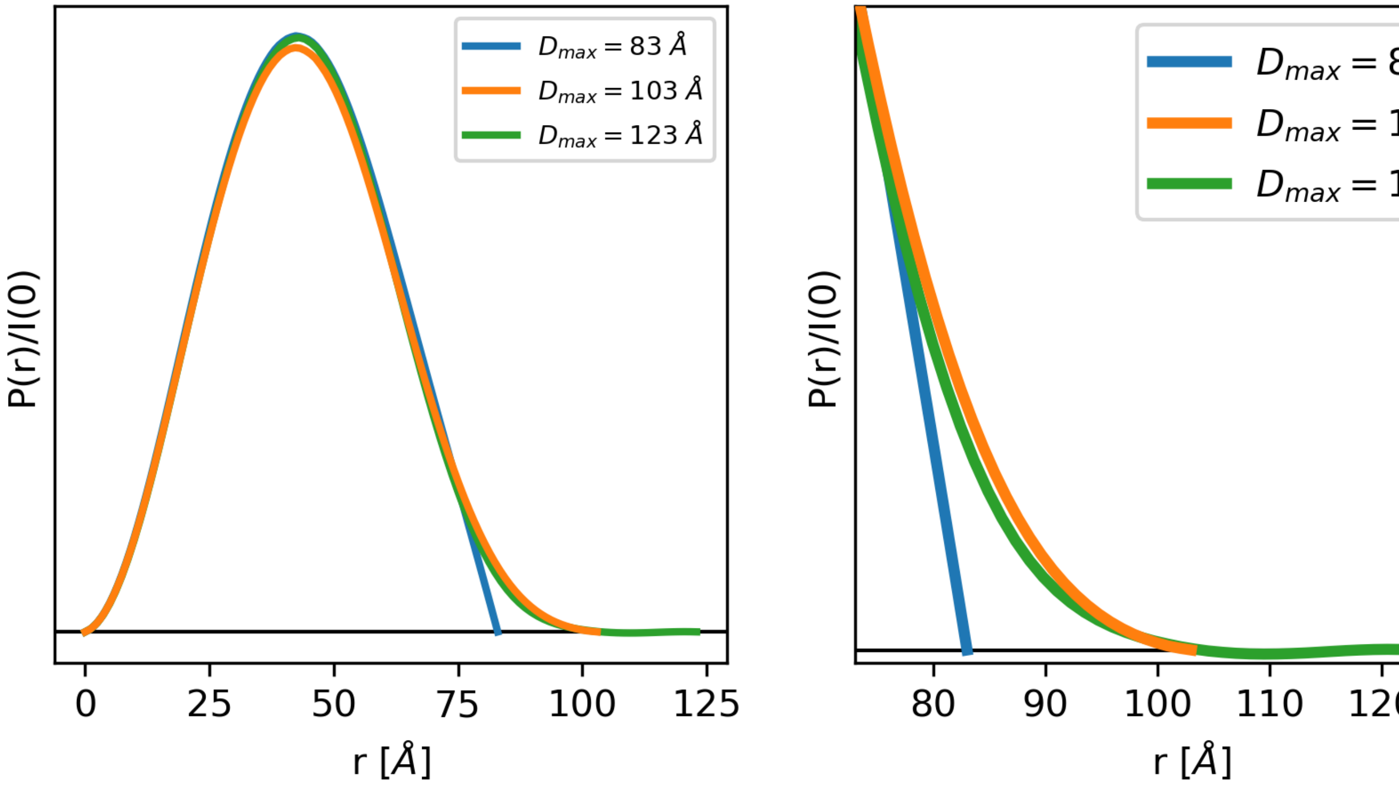


Fig. 9: The left and right plots show three different P(r) functions for the same protein (glucose isomerase, available in the RAW Tutorial Data). The difference between the three is the D_{max}, which is either 83 (blue), 103 (orange) or 123 (green) Angstrom. The left plot shows the full P(r) function. The different D_{max} values yield similar P(r) functions, so much so that they end up plotted on top of each other for most of their r values. The right plot is the same functions showing just the end, as P(r) approaches zero at D_{max}.

In the plot above, we can clearly see that for a D_{max} of 83, the P(r) function is forced abruptly down. For a D_{max} of 103, the function has a smooth approach to zero. For a D_{max} of 123 the function reaches zero and then oscillates about it. From this we can conclude that 103 is a good value for D_{max}, whereas 83 is underestimated and 123 is overestimated.

The P(r) function fits the measured scattering profile

This criterion is straightforward. The transformation of the P(r) function to I(q) should fit the measured scattering profile. This can be evaluated both through the χ^2 value of the fit, which should be close to 1, and the normalized residuals between the fit and the data, which should be flat and randomly distributed about zero.

The P(r) function goes to zero at $\mathrm{r} = 0$ and $\mathrm{r} = \mathrm{D_{max}}$.

The reason for this criterion is straightforward. The P(r) function should go to zero at r=0 because it is the r^2 weighted number of electron pairs in the macromolecule. As r goes to zero, so does r^2, and thus so must P(r). The P(r) function should go to zero at $r = D_{max}$ because D_{max} is the maximum dimension of the particle. Beyond that distance there should be no electron pairs in the particle. This criterion is usually enforced by conditions in the IFT calculation.

The R_g and I(0) from the Guinier fit and the P(r) function agree well

The R_g and I(0) values can be determined directly from the P(r) function. This provides a complementary approach to the Guinier fit. For well behaved rigid systems, R_g and I(0) should agree well between both methods. If they do not, it may suggest a problem in either the Guinier fit or the P(r) function. However, for flexible and disordered systems, it has been observed that the P(r) R_g and I(0) values are characteristically larger, and more reliable, than the Guinier R_g and I(0) values [3].

The P(r) function is always positive

This criterion usually applies, as for most macromolecules the presence of a negative number of electron pairs has no meaning. However, when dealing with membrane proteins that are encapsulated in lipids or detergents, this criterion is no longer valid. In those cases, the lipid/detergent may have a lower electron density than the buffer. As scattering is measured relative to the solvent, this lower density will appear as negative electron pairs in the P(r) function. For example, proteins embedded in lipid nanodiscs have a characteristic dip in the P(r) function that can go negative.

Determining a good P(r) function using GNOM

While it takes some practice to learn how to properly evaluate the P(r) function, there is a set of steps that I regularly follow when creating a P(r) function using GNOM via the RAW interface:

1. Open the GNOM interface. It defaults to what GNOM thinks is a reasonable D_{max} (using datgnom).
2. If necessary, set the starting q value for the P(r) function to match that of the Guinier fit (newer versions of RAW do this automatically).
3. Set the D_{max} value to 2-3 times larger than the initial value.
4. Look for where the P(r) function drops to 0 naturally. Set the D_{max} value to this point.
5. Turn off the force to zero at D_{max} condition.
6. Tweak D_{max} up and down until it naturally goes to zero (with the force to zero turned off).
7. Turn the force to zero at D_{max} condition back on.
8. If needed, truncate the P(r) function to a maximum q of $8/R_g$, or 0.25-0.3 1/Angstrom, whichever is smaller, if using for bead model reconstructions with DAMMIF/N. You may have to tweak the D_{max} a bit after truncation.

If you have good quality data, this ought to produce a good P(r) function.

Note that even for good quality data with a mostly rigid globular macromolecule like glucose isomerase (shown in the plots above), there usually isn't a single right value of D_{max}. For this data, a best case scenario, you could reasonably pick a D_{max} value from ~99-104, which is a 5% variation. For macromolecules with more flexibility, D_{max} is even more poorly defined. As a rule of thumb, D_{max} is usually never determined to better than 5%, sometimes the uncertainty is closer to 10%.

Other tips:

- If the residual has too much systematic deviation, you can manually set the α to something smaller than the automatic value. Start with half the automatically determined value, and tweak from there.
- Don't forget to check that the R_g and I(0) values agree (if you have a rigid system). Generally speaking, increasing D_{max} will increase the P(r) R_g and I(0) values, so that can help guide your choice of D_{max}.
- Don't truncate your P(r) function for electron density reconstructions with DENSS.

What is a bad P(r) function, and what does it mean?

Even if you follow all of the guidelines above, sometimes you can end up with a bad P(r) function. Typically what bad means is one of two things:

1. You can't find a good D_{max} value. You keep increasing D_{max} and the P(r) function never smoothly goes to zero.
2. If you increase D_{max} the P(r) function goes negative.

In these cases, it is most likely that your data has a problem. The figure below gives a quick summary of the most common pathologies, more detail is available in the sections below.

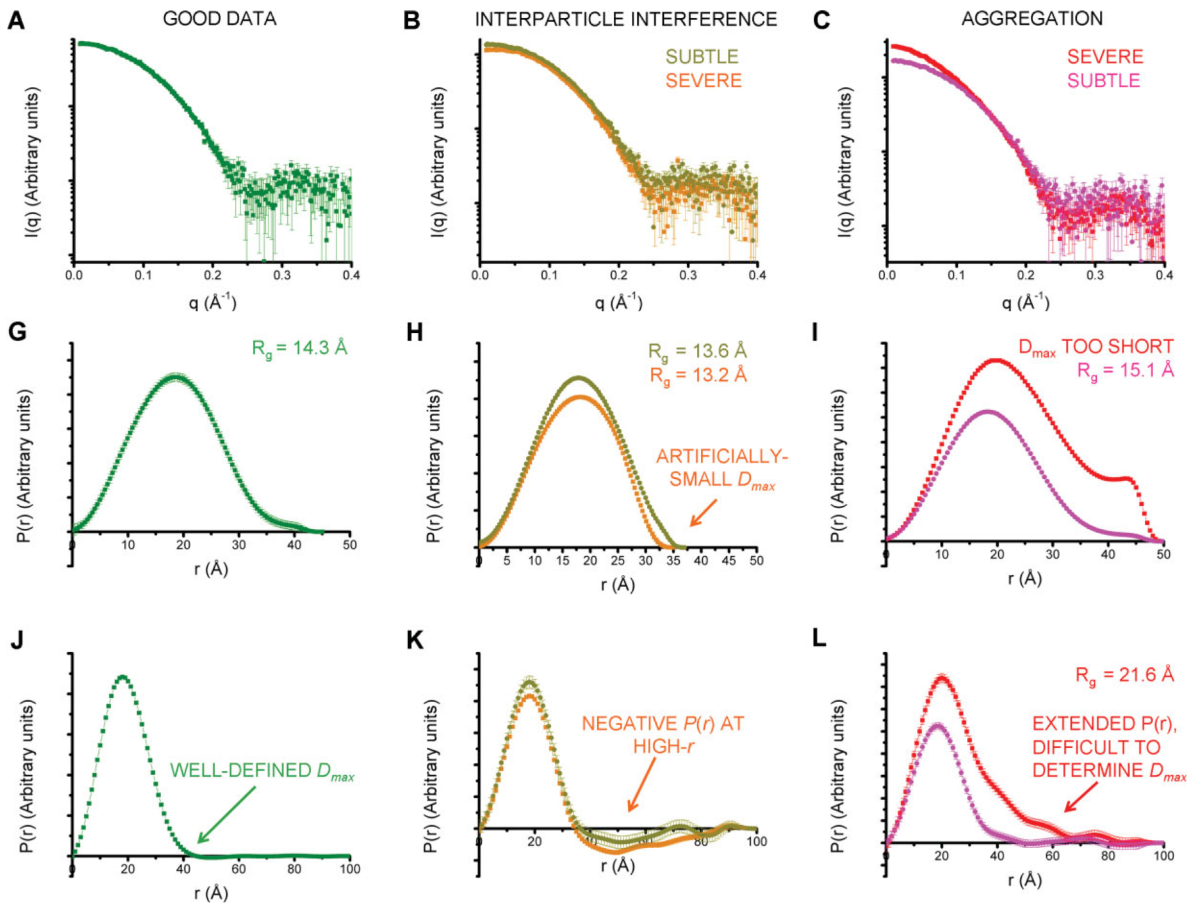


Fig. 10: Figure 3 from [4]. A, G, and J show a good (monodisperse) scattering profile and P(r) function. B, H, and K show a scattering profile and P(r) function with varying degrees of interparticle interference. C, I, and L show a scattering profile and P(r) function with varying degrees of aggregation. the middle row shows the effect on the P(r) function, while the last row shows that to really judge the effect of the change on the P(r) function you should extend the D_{max} value out significantly.

Note that both pathologies are easiest to see when you plot the P(r) function well past the point where you think the correct D_{max} value is (panels J-L in the above figure). This means that it is important to always extend the D_{max} value when generating your P(r) function to verify that the P(r) function stays flat and close to zero (usually small oscillations about zero), as in panel J, rather than dipping negative (panel K, repulsive interparticle interference) or staying slightly positive (panel L, aggregation).

Aggregation

Aggregation in solution means some amount of larger particles are present. The presence of these larger particles causes the D_{max} to be hard to determine. Typically this manifests as there being a significantly extended tail on the P(r) distribution, which does not fall to zero naturally regardless of the chosen D_{max}. The P(r) function calculated R_g and I(0) will also be larger than they would be for the monodisperse sample.

Small amounts of aggregation can look similar to the P(r) function for a flexible system, so other methods should be used to determine the true state of the system. The Guinier analysis will usually reveal the presence of aggregates, and Kratky plot will show if the system is flexible. If you are unsure whether your P(r) function is showing aggregation or flexibility, check with these other techniques.

Interparticle interference

Interparticle interference usually manifests as repulsion in solution. This repulsive effect manifests as an artificially small D_{max}. The P(r) function calculated R_g and I(0) values are reduced compared to what they would be for the non-interacting sample.

Since you usually don't know the D_{max} of your sample prior to making the measurement, it can be hard to tell if the D_{max} is artificially small. In this case, the easiest way to see this effect is to extend the D_{max} out past what you found to be a 'good' D_{max}. If the P(r) function goes and stays negative, as in K of the above figure, then you have a repulsive interaction. If it stays near zero (possibly with some small oscillation about zero), as in J of the above figure, then you are not seeing a repulsive interaction.

How to interpret features of a P(r) function

The P(r) function provides a significant amount of information on particle shape and size in solution. The easiest parameters to interpret are the D_{max}, R_g, and I(0), which all inform on particle size. The D_{max} is simply the maximum dimension of the particle. The R_g is the radius of gyration, and I(0) is the scattering at zero angle, which is proportional to the molecular weight and concentration. Beyond these parameters, the shape of the P(r) function contains significant information about the particle shape. This is seen clearly in the plot of P(r) functions for different geometric bodies shown below.

In the above figure, there are several things worth noting:

1. In the P(r) function for a long rod, the initial peak at short distance comes from electron pairs across the short dimension of the rod. The long extend tail comes from electron pairs along the length of the rod.
2. In the P(r) function for a dumbbell, the peak at lower r comes from the electron pairs within each individual domain of the dumbbell. The peak at longer distance comes from the electron pairs between the two domains of the dumbbell.
3. For the hollow sphere, the peak at long distance is coming from the electron pairs across the diameter of the sphere.

While macromolecules do not have P(r) functions that exactly match those of geometric bodies, the dominant features of the P(r) function can be used to determine overall shape characteristics of the macromolecule. Most usefully:

- Globular proteins tend to have P(r) functions similar to that of a solid sphere.
- Long rigid rod-like systems, like duplex DNA, have P(r) functions very similar to that of a long rod.
- Multi-domain proteins have two peaks, like that of a dumbbell. However, as the domains are not usually symmetric in size or widely separated, the peaks are usually more of a strong peak and an overlapping shoulder peak.

The plot below shows the P(r) function for several actual macromolecules.

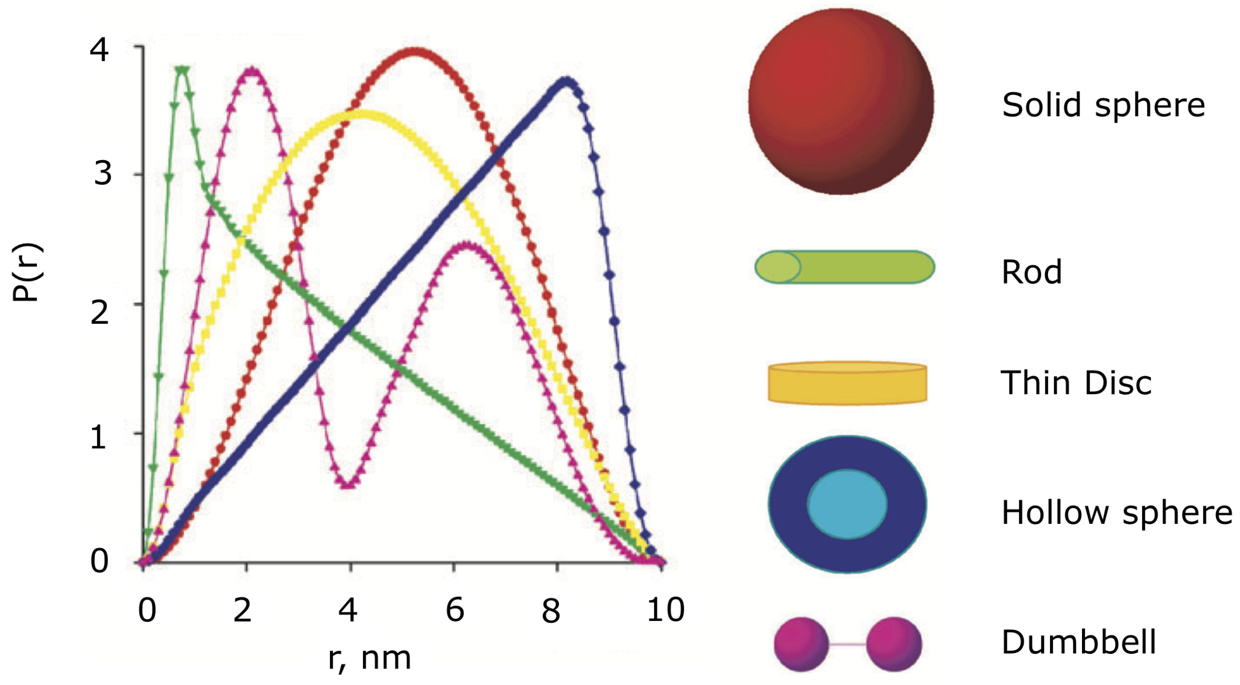


Fig. 11: P(r) functions for geometric bodies, adapted from figure 5 of [5].

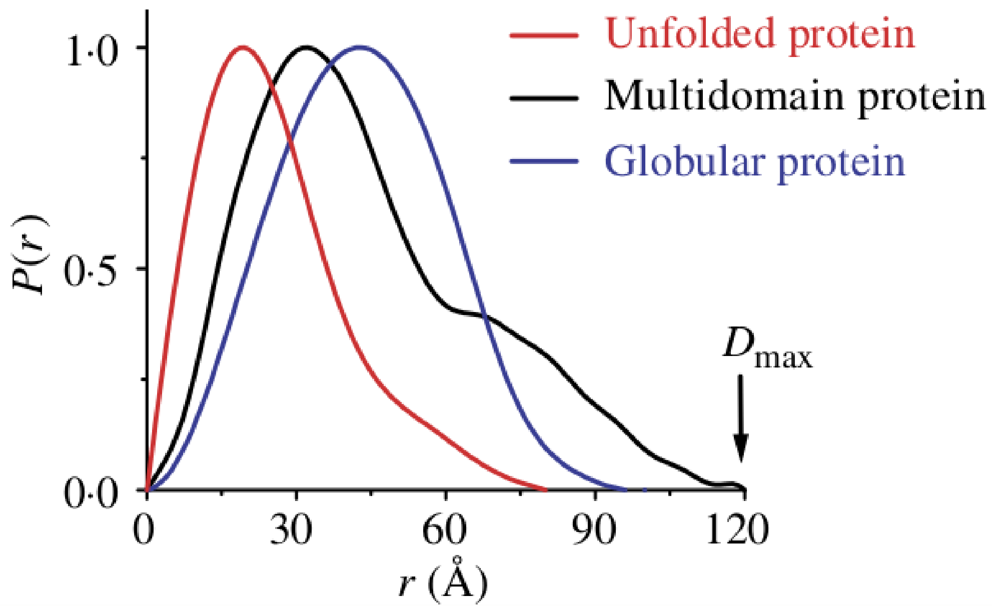


Fig. 12: P(r) functions for flexible (unfolded), multidomain, and globular proteins. Figure 24 in [6].

FAQ

Is a small change in D_{max} significant?

Small changes in D_{max} are usually not significant. Even for good quality data with a mostly rigid globular macromolecule like glucose isomerase (shown in the plots above), you could reasonably pick a D_{max} value from ~99-104, which is a 5% variation. For macromolecules with more flexibility, D_{max} is even more poorly defined. As a rule of thumb, D_{max} is usually never determined to better than 5%, sometimes the uncertainty is closer to 10%.

My P(r) function goes negative. Is that okay?

Generally speaking, no. However, if you have a system that is detergent or lipid bound, such as a protein embedded in a nanodisc, then you may see a negative dip in your P(r) function.

References

1. Hansen, S. J. Appl. Crystallogr.(2000) 33, 1415-1421. DOI: 10.1107/S0021889800012930
2. Svergun D.I. (1992). J. Appl. Crystallogr. 25, 495-503. DOI: 10.1107/S0021889892001663
3. Kikhney, A. G. & Svergun, D. I. (2015). FEBS Lett. 589, 2570–2577. DOI: 10.1016/j.febslet.2015.08.027
4. Jacques, D. A. & Trewhella, J. (2010). Protein Sci. 19, 642–657. DOI: 10.1002/pro.35
5. Svergun, D. I. & Koch, M. H. J. (2003). Reports Prog. Phys. 66, 1735–1782. DOI: 10.1088/0034-4885/66/10/R05
6. Putnam, C. D., Hammel, M., Hura, G. L. & Tainer, J. a (2007). Q. Rev. Biophys. 40, 191–285. DOI: 10.1017/S0033583507004635

5.4.5 Bead model reconstructions

This tutorial covers basic principles and best practices for creating bead (dummy atom) model reconstructions of macromolecule shape from SAXS data. This is not a tutorial on how to use RAW for this type of analysis. For that, please see the RAW tutorial for *DAMMIF/N*.

Overview

A natural desired outcome for many SAXS experiments is determining the 'solution structure' of the sample, i.e. the structure of the macromolecule as it exists in solution. Unfortunately, unlike crystallography, cryoEM, and NMR, SAXS data cannot be used to generate a high resolution 3D shape (though it can be used to constrain other methods of structure determination). What SAXS can often provide, and what is a common end point of SAXS analysis, is a low resolution shape reconstruction of the sample.

For many years, bead modeling was the state of the art approach for generating these low resolution shape reconstructions. Recently, other techniques have been developed, such as direct reconstruction of the electron density at low resolutions. Despite this, bead modeling remains the de facto standard for shape reconstruction.

Why do we do bead model reconstructions?

Bead models, despite their low resolution, can be powerful tools for understanding the system in solution. High resolution structures can be docked into the bead model, allowing visual analysis of how well the high resolution structure agrees with the solution structure. In the absence of other structure information, bead models can be used to provide important clues to the overall shape and size of the system, which can often be enough to draw conclusions about important aspects of the macromolecular function or interactions with other systems.

While bead modeling can be quite useful, it is important to always keep in mind two things. First, it is very easy to get poor reconstructions, even with good quality data, and you must carefully evaluate the results of your reconstructions before using them. Second, SAXS is much more accurate at hypothesis testing than it is at generating bead models. Put another way SAXS is very good at telling you what something isn't, but not so good at telling you what it is. For example, if you want to compare a high resolution structure to SAXS data and see if they agree, you are always better off testing the calculated scattering profile of the high resolution structure against the measured scattering profile than docking the structure into the bead model. In this case, the bead model might be useful in visualizing the differences between the solution shape and your high resolution structure.

How do we do bead model reconstructions?

There are a number of different programs available to do bead modeling, some suited for general systems and some tuned for very specific applications like reconstructions of membrane proteins with detergent halos. Regardless, all of these methods share a similar approach.

1. Generate a volume of beads (aka 'dummy atoms') and randomly assign the beads to be one of the allowed phases. In general, beads are either solvent or macromolecule, but some programs allow more than 2 phases, such as differentiating between protein and RNA or macromolecule and lipid.
2. Calculate a scattering profile from the bead model and fit that against the data.
3. Flip a randomly chosen set of beads between phases (e.g. from solvent to macromolecule or vice versa).
4. Recalculate the scattering profile from the model and the to the data.
5. If the fit is better, accept the changes to the beads. If the fit is worse, accept the changes to the beads with some probability (avoids local minima).
6. Repeat steps 3-5 until the convergence criteria is met.

Additional, programs usually impose physical constraints on the bead models to improve the model. Common constraints are requiring connectivity of the model, imposing penalties for extended models, and constraining the size of the model based on the R_g and/or D_{max}.

It turns out that a scattering profile does not generate a unique reconstruction, so to account for this a Monte Carlo like approach is taken where a number (usually 10-20) of models are generated, and then averaged to give a consensus reconstruction.

The most common program for creating bead model reconstructions (though far from the only such program) is DAMMIF (or DAMMIN) [1-2] from the ATSAS package. For the remainder of this tutorial we will exclusively discuss using DAMMIF/N, though much of the discussion should apply to other programs as well.

Using DAMMIF/N for bead model reconstructions

DAMMIF is the most commonly used program for bead model reconstructions. Here we discuss some of the practical aspects. A detailed tutorial of *how to use DAMMIF in RAW* is also available.

Input data

DAMMIF requires a P(r) function generated by GNOM (a .out file) as input. Note that, as mentioned in the *IFT tutorial*, the scattering profile for the IFT should be truncated to a maximum q value of 8/R_g or ~0.25-0.30 1/Angstrom, whichever is smaller. This is because the hydration layer and internal structure are not modeled by DAMMIF, which leads to errors at higher q. The truncation removes the potentially problematic higher q data.

Generating models

As bead modeling does not generate a unique solution. In order to generate a reasonable solution, we create 10-20 bead model reconstructions and then average them. I recommend 15 reconstructions. This means that we need to run DAMMIF 15 different times.

The most accessible settings for DAMMIF are the Mode, Symmetry, and Anisometry.

Mode: For mode, options are Fast or Slow. Fast is quick, but less detailed, Slow is the opposite. For a final reconstruction, use Slow mode.

Symmetry: Adding in symmetry constraints can improve the reconstruction. If you know the symmetry of the particle, you can specify this. However, it is always recommend that you do an additional set of reconstructions with P1 symmetry, to verify that the symmetry did not overly constrain the reconstructions.

Anisometry: Adding in anisometry constraints can improve the reconstruction. If you know the anisometry of the particle, you can specify this. However, it is always recommend that you do an additional set of reconstructions with no anisometry, to verify that the symmetry did not overly constrain the reconstructions.

Additional advanced options are available, and are described in the DAMMIF manual.

If you only want a quick look at the shape (such as when collecting data at a beamline) 3 reconstructions in Fast mode will work for that purpose.

Averaging and clustering models

After models are generated the next step is to average and cluster the models. Averaging generates a consensus shape from the individual models, and provides statistics on how stable the reconstruction is. This is done with DAMAVER [3]. The average outputs both damaver.pdb and damfilt.pdb model files. These correspond to two different consensus shapes of the model, loosely and tightly defined respectively. However, neither of these models actually fits the data, and so generally should not be used to display your reconstructions. DAMAVER will also specify the most probably individual model. If you do not refine the results of DAMAVER (below) you should use the most probable model as your final result.

Clustering is done with DAMCLUST [4] and clusters models that are more similar to each other than they are to the rest of the models. This is a way of assessing the ambiguity of the reconstruction, and we will discuss it further in the section on evaluating reconstructions below.

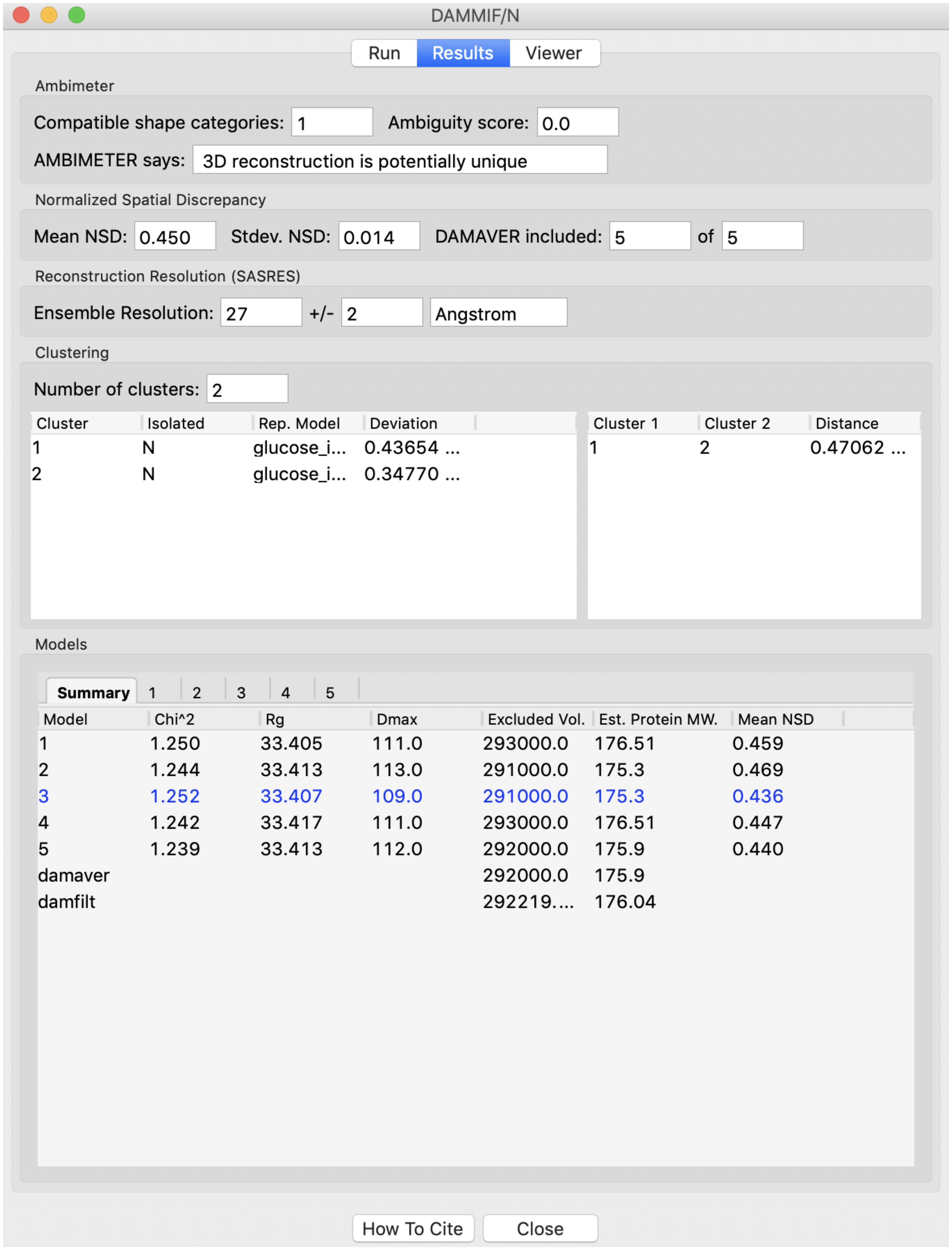
Creating a final refined model

The output of DAMAVER, specifically the damstart.pdb file, can be used as input for DAMMIN to create a final refined model. Essentially, the damstart.pdb represents a conservative core of the most probably occupied volume as determined by averaging all the reconstructions using DAMAVER. DAMMIN keeps this core fixed, and refines the outside of the model to match the scattering profile. I've seen mixed recommendations (even from the makers of the software) on whether you should do a refinement. I typically do, but it seems you can often do just as well with the most probable model determined by DAMAVER.

Evaluating DAMMIF/N reconstructions

SAXS data contains very limited information, both because it is measured at relatively low q, and because it is measured from a large number of particles in solution oriented at random angles. The SAXS scattering profile represents the scattering from a single particle, averaged over all possible orientations. The practical consequence of this is that there are often several possible shapes that could generate the same (or so similar as to be indistinguishable within experimental noise) scattering profiles. As such, it may simply not be possible to generate a bead model reconstruction from a dataset that accurately represents the solution shape, regardless of the overall data quality. If the sample is flexible or otherwise exists in multiple conformational or oligomeric states in solution the reconstruction is also challenging or impossible. **In summary, high quality SAXS data is not a guarantee of a good bead model reconstruction. This makes it very important to critically evaluate every reconstruction done, regardless of the underlying data quality.**

The information needed to evaluate the reconstructions is generated when running DAMMIF, DAMAVER, DAMCLUST, SASRES [5] (run as part of DAMAVER) and AMBIMETER [6]. While it can all be accessed through the files these programs generate, RAW gathers and presents it for you when you run DAMMIF in RAW.


DAMMIF/N

Run | Results | Viewer

Ambimeter

Compatible shape categories: 1 Ambiguity score: 0.0

AMBIMETER says: 3D reconstruction is potentially unique

Normalized Spatial Discrepancy

Mean NSD: 0.450 Stdev. NSD: 0.014 DAMAVER included: 5 of 5

Reconstruction Resolution (SASRES)

Ensemble Resolution: 27 +/- 2 Angstrom

Clustering

Number of clusters: 2

Cluster	Isolated	Rep. Model	Deviation
1	N	glucose_i...	0.43654 ...
2	N	glucose_i...	0.34770 ...

Cluster 1	Cluster 2	Distance
1	2	0.47062 ...

Models

Summary | 1 | 2 | 3 | 4 | 5

Model	Chi^2	Rg	Dmax	Excluded Vol.	Est. Protein MW.	Mean NSD
1	1.250	33.405	111.0	293000.0	176.51	0.459
2	1.244	33.413	113.0	291000.0	175.3	0.469
3	1.252	33.407	109.0	291000.0	175.3	0.436
4	1.242	33.417	111.0	293000.0	176.51	0.447
5	1.239	33.413	112.0	292000.0	175.9	0.440
damaver				292000.0	175.9	
damfilt				292219....	176.04	

How To Cite | Close

Criteria for a good DAMMIF/N reconstruction

- Ambiguity score < 2.5 (preferably < 1.5)
- NSD < 1.0
- Few (0-2) models rejected from the average
- Only one cluster of models
- Model χ^2 near 1.0 for all models
- Model R_g and D_{max} close to values from P(r) function for all models
- M.W. estimated from model volume close to expected M.W.

More about these criteria can be found below.

Ambiguity

It is possible to evaluate the potential ambiguity of your bead model reconstructions before doing the reconstructions. The AMBIMETER program in the ATSAS package can be run on P(r) functions from GNOM to assess how likely you are to get a good reconstruction. The program has a database of scattering profiles representing all possible shapes made out of up to 7 beads. Your scattering profile is compared against these shapes, and AMBIMETER reports how many match your profile. The more profiles from AMBIMETER that match yours, the more possible shapes could have generated your profile.

AMBIMETER reports both the number of shapes and the log (base 10) of the number shapes, which is the Ambiguity score. They provide the following interpretations:

- Ambiguity score < 1.5 - Reconstruction is likely unique
- Ambiguity score of 1.5-2.5 - Take care when doing the reconstruction
- Ambiguity score > 2.5 - Reconstruction is most likely ambiguous.

This provides a quick initial assessment of whether you should even attempt a shape reconstruction for your dataset. *You can run AMBIMETER from RAW*.

Normalized spatial discrepancy

DAMAVER reports a number of different results. The most useful is the normalized spatial discrepancy (NSD). This is essentially a size normalized metric for comparing how similar two different models are. When DAMAVER is run, it reports the average and standard deviation of the NSD between all the reconstructions. It also reports the average NSD for each model.

The average NSD is commonly used to evaluate the stability of the reconstruction. Roughly speaking we evaluate reconstruction stability as:

- NSD < 0.6 - Good stability of reconstructions
- NSD between 0.6 and 1.0 - Fair stability of reconstructions
- NSD > 1.0 - Poor stability of reconstructions

Generally speaking, if your average NSD is less than 1.0, the reconstruction can probably be trusted (if all of the other validation metrics also check out), while if it is greater than 1.0 you should proceed with caution, or not use the reconstructions at all.

The NSD is also used to determine which models to include in the average. If the average NSD of a given model is more than two standard deviations above the overall average NSD, that model is not included in the average. If more than ~2 models are rejected (out of 15), that may be a sign of an unstable reconstruction.

Clusters

DAMCLUST creates clusters of models that are more similar to each other than they are to the rest of the models. This is a way of assessing the ambiguity of the reconstruction. If you have more than one cluster of models in your reconstructions, you may have several distinct shapes that are being reconstructed by the DAMMIF algorithm. This typically indicates that there are several distinct shapes in solution that could generate the measured scattering profile, and so is another indication of a highly ambiguous reconstruction.

The caveat to this is that with good quality data that is very low ambiguity (ambiguity score from AMBIMETER < 0.5) and yields a set of reconstructions with a very small average NSD (<0.5, typically) and NSD standard deviation (~0.01), I have seen several (often >5) clusters identified with DAMCLUST. I believe that in this case there are not actually multiple clusters, but the extremely low deviation between the models is fooling the DAMCLUST algorithm.

Note that the different clusters should not be taken as representatives of different distinct shapes in solution. Even if there are a finite number of distinct shapes scattering in the solution (such as an open and closed state of a protein), the measured scattering profile is an average of the scattering from each component, and each individual reconstruction fits that measured scattering profile. As such, there is no way for an individual reconstruction to fit just the scattering from one of the components and so the different clusters cannot be representative of the different shapes in the solution.

Model fit and parameters

Each model has the following parameters that can be used to evaluate the success of an individual reconstruction: χ^2, R_g, D_{max}, volume, molecular weight estimated from volume, and the normalized residual of the model fit to the data. For a good fit to the data, the model χ^2 should be close to 1 and the normalized residual between the model fit and the data should be flat and randomly distributed about zero. However, in my experience the normalized residual often shows some small systematic deviations, and so this should not be too concerning. A χ^2 value significantly larger than 1 (1.5-2 or larger) indicates either a poor fit to the data or that the uncertainty for the data is underestimated. To differentiate between these two cases, look at the normalized residual. If it is flat and randomly distributed, then the uncertainty is most likely underestimated. If it shows significant systematic deviations then the fit quality is poor.

The R_g and D_{max} obtained from the model should be close to those calculated from the P(r) function. If that is not the case, you should reevaluate your P(r) function and redo the reconstruction if necessary. If the discrepancy persists, it is an indication that your reconstruction isn't a good representation of what is in solution, and shouldn't be trusted. While there's no hard and fast rule here on how closely R_g and D_{max} should agree, my experience is generally that for high quality data R_g agrees to better than ~5% and D_{max} to ~10%.

The volume is reported for each bead model, but it is usually easier to compare the molecular weight calculated from that volume with the expected molecular weight. In this case, M.W. is calculated by dividing the volume (nominally representing the sample's excluded volume) by an empirically determined constant [4] of 1.66 (used in RAW, other programs may use different values). This value is approximate, and varies between roughly 1.5 and 2.0 depending on the shape of the macromolecule. This M.W. is less well determined than *other SAXS methods*, given the variation in the coefficient. As such, it is mostly useful for indicating general agreement between the overall size of the reconstruction and the expected size. If the M.W. is different from the expected M.W. by more than 20-25% you should consider the reconstructions to be suspect.

Limitations of bead models

While bead models can be quite useful, they have a number of limitations, many of which are mentioned in previous parts of the tutorial. In summary:

- Bead models can be ambiguous, even if the data quality is very high. This is because multiple different shapes in solution can produce the same scattering profile, so there is no guaranteed unique solution to a reconstruction, and the success of the reconstruction depends not just on the input data quality but also the inherent shape of the particle and how ambiguous that shape is for SAXS. Because of this, all models should be thoroughly evaluated as described above.
- Ignoring ambiguity, bead models still only work best with particular particle shapes. An excellent discussion of how well bead models work for different types of shapes is found in [3]. The summary is that bead models tend to be less reliable for high aspect ratio objects, such as long rods or thin discs, objects with voids (such as a spherical shell), and rings. They are most reliable for things that are generally globular.
- Bead models are low resolution. Small variations of the surface of the model are likely insignificant. I rarely see estimated model resolutions less than ~20 Angstroms, often they are much larger.
- Bead models do not (typically) model the hydration layer or internal structure of the particle. This requires that you use only data out to a maximum q of 8/R_g or ~0.25-0.30 1/Angstrom, whichever is less.
- The most common bead modeling programs cannot model multiple electron densities within a sample, such a protein-nucleic acid complex or a membrane protein with a detergent halo. There are specialized programs (such as MONSA or Memprot) that can handle these cases, but these require the input of additional information to provide extra constraints.
- The bead model is only as good as the input data. In particular, bead models are quite sensitive to the presence of larger particles in solution, either oligomers or non-specific aggregate. In one simple simulation I've seen, as little as 0.7% aggregate caused a significant change in the bead model. Non-specific aggregation usually manifests as an extended protrusion from the main model.

As you can see, while bead models can certainly be useful for your research, you should proceed with caution and ensure that you have a trustworthy reconstruction before proceeding with your bead models.

Visualizing DAMMIF/N reconstructions

Visualizing DAMMIF/N bead model reconstruction is slightly different from displaying a typical macromolecular structure. There are two main ways that these are visualized, either as individual beads or, more commonly, as an envelope that defines the edges of the model. Both representations are usually made semi-transparent so that a high resolution structure docked with the bead model is simultaneously visible.

The main detail to remember is that to get a correct visualization you have to set the correct bead size for the model, which is given in the header of the DAMMIF/N .pdb file.

Below are two quick tutorials for visualizing models in Chimera (or ChimeraX) and PyMOL.

Visualizing bead models with Chimera

Note: There are some differences between Chimera and the newer ChimeraX. Differences for ChimeraX are noted in **bold**.

1. Open Chimera.
2. Load in the DAMMIF/N .pdb file of interest.
3. If necessary, open the Model Panel and the Command Line from the Tools-> General Controls menu.
4. In the Select->Chain menu choose the bead model ("no ID" or **the filename**).
5. In the Actions->Ribbons (**Actions->Cartoon**) menu, choose "Hide".
6. In the Actions->Atoms/Bonds menu, choose "Show".
7. In the Actions->Atoms/Bonds (**Actions->Atoms/Bonds->Atom Style**) menu choose "Sphere".

8. **In the Camera section of the ChimeraX "Graphics" ribbon, click "View selected".**
9. Open the PDB header by double clicking on the model in the Model Panel, then clicking on "PDB Headers…" at the bottom of the panel that pops up.
   - *Note:* **This doesn't seem to be available in ChimeraX. You'll have to open the .pdb file in a text editor.**
   - *Tip:* You can also open the .pdb file in a text editor and read the header there.
10. Find the "Atomic Radius" (DAMMIF/DAMAVER model) or "DAM packing radius" (DAMMIN model) value. That is the bead size you need to set.
11. In the command line, enter the command `vdwdefine x #y` where `x` is the bead size from the PDB header and `y` is the ID number of the bead model shown in the model panel.
    - **The command is "size atomradius x" in ChimeraX.**
12. Your beads are now the right size. If you want to make an envelope proceed with the following optional steps. If you'd rather use the individual bead display, you can stop here.
13. To make an envelope, in the command line enter the command `molmap #y z` where `y` is the ID number of the bead model shown in the model panel and `z` is 3x the bead size that you found in the previous steps.
    - *Tip:* The last number controls the smoothness of the envelope. You may need to vary it from 3*(bead size), depending on the size of your beads and how smooth you want your envelope. I recommend leaving at least a hint of the beads visible (not overly smoothing the envelope) to help whoever sees the graphic to remember that an envelope is not an electron density contour.
14. Hide the bead model using the "Hide" button in the model panel.
15. In the Volume Viewer window that appeared when you entered the molmap command, in the Features menu select the "Surface and Mesh options'.
    - *Note:* This menu doesn't exist in ChimeraX.
16. Check the box for Surface smoothing and set the iterations to 2 and the factor to 1.
    - *Note:* This option doesn't exist in ChimeraX.
17. Check the box for Subdivide surface and set it to 2 times.
    - *Note:* This option doesn't exist in ChimeraX.
18. Click on the color box to set color and opacity. I find that 0.4 (**40%**) is a good opacity for overlaying with high resolution models.
19. Load in your aligned (such as with *SUPCOMB*) high resolution structure if available.

Visualizing bead models with PyMOL

1. Open PyMOL
2. Load in the DAMMIF/N .pdb file of interest.
3. Using the model Hide menu ('H'), hide 'everything'.
4. Using the model Show menu ('S'), show 'spheres'.
5. Open the .pdb file in a text editor and find the "Atomic Radius" (DAMMIF/DAMAVER model) or "DAM packing radius" (DAMMIN model) value in the PDB header.
6. In the PyMOL command line, enter the command `alter <model_name>, vdw=x` where `x` is the size you found in the previous step. This sets the spheres to be the correct size of the beads in the model.
7. Click the 'Rebuild' button to refresh the view of the model.

8. Your model is now displayed correctly with beads. If you want to make an envelope, proceed with the following optional steps. If you'd rather use the individual bead display, you can stop here.
9. Using the model Show menu, show 'surface'.
10. Using the model Hide menu, hide 'spheres'.
11. You can smooth the surface by increasing the probe radius using the command `set solvent_radius, 3.0` (where you can vary the size from 3.0).
12. You can improve the quality of the surface using the command `set surface_quality, 1`.
    - *Note:* Values larger than 1 may take a long time to render.
13. Set your surface transparency to 50% for overlaying with high resolution models using the command `set transparency, 0.5`
14. Load in your aligned (such as with *SUPCOMB*) high resolution structure if available.

FAQ

Do I have to make a bead model?

No. It all depends on what you're trying to say about the data. However, particularly if your system shows signs of flexibility or AMBIMETER reports a high ambiguity score you probably shouldn't bother making a bead model even if you want to.

How do I fit my high resolution structure into my bead model?

If your high resolution structure is relatively complete (contains all residues in solution, and ideally post-translational modifications), you can use a program like SUPCOMB [7] to automatically *fit the structure into the bead model*. If you are missing significant amounts of the structure (such as a large flexible loop) or have only one subunit of a multi-subunit complex you may have to manually dock the structure in the envelope.

My bead model and my high resolution structure disagree. Which one is right?

Maybe both, maybe neither! It really depends on your inputs. If you've validated the bead model as above and it seems good, then it likely represents the low resolution shape in solution. You should also verify that your high resolution shape contains all of the residues in your system, often high resolution structures are missing things like flexible loops or N and C terminus regions.

If both models are good, then depending on how you obtained your high resolution shape it might also be correct, but represent the shape under different conditions. For example, it is common in crystallography to see structural artifacts induced by the packing of the macromolecule into the crystal.

Of course, the best way to compare your high resolution structure to SAXS data isn't by docking it in a bead model, but by fitting it against the data using a program like CRYSOL or FoXS. If these fits are bad, then your high resolution structure doesn't match the data, regardless of what the bead model shows. If these fits are good, and the bead model doesn't agree with the high resolution structure, then the bead model is wrong.

My bead model isn't good, what should I do instead?

There are many more approaches available than I can list here, but a couple of the more common ones are:

- If your data is flexible, you can try some kind of ensemble based approach, such as EOM, SASSIE, or BilboMD.

- If your data is more rigid and consists of several subunits you can consider rigid body modeling such as SASREF.

References

1. Franke, D. and Svergun, D.I. (2009) DAMMIF, a program for rapid ab-initio shape determination in small-angle scattering. J. Appl. Cryst., 42, 342-346.
2. D. I. Svergun (1999) Restoring low resolution structure of biological macromolecules from solution scattering using simulated annealing. Biophys J. 2879-2886.
3. V. V. Volkov and D. I. Svergun (2003). Uniqueness of ab-initio shape determination in small-angle scattering. J. Appl. Cryst. 36, 860-864.
4. Petoukhov, M.V., Franke, D., Shkumatov, A.V., Tria, G., Kikhney, A.G., Gajda, M., Gorba, C., Mertens, H.D.T., Konarev, P.V. and Svergun, D.I. (2012) New developments in the ATSAS program package for small-angle scattering data analysis. J. Appl. Cryst. 45, 342-350
5. Anne T. Tuukkanen, Gerard J. Kleywegt and Dmitri I. Svergun(2016) Resolution of ab initio shapes determined from small-angle scattering IUCrJ. 3, 440-447.
6. M.V. Petoukhov and D.I. Svergun (2015) Ambiguity assessment of small-angle scattering curves from monodisperse systems Acta Cryst. D71, 1051-1058.
7. M.Kozin & D.Svergun (2001) Automated matching of high- and low-resolution structural models J Appl Cryst. 34, 33-41.

5.5 Videos

We have made various demo and instructional videos for RAW.

5.5.1 Tutorial videos

Every section of the RAW tutorial has an associated tutorial video. These can be found in the appropriate section. A playlist of all the videos is available:

5.5.2 Older videos

The videos below were made with older versions of RAW. Some of the content may be out of date, but they might still be useful for you.

RAW demo for SBGrid

Radiation damage

For this videos, tutorial data is available from https://bit.ly/bioxtasd

Aggregation

For this videos, tutorial data is available from https://bit.ly/bioxtasd

Concentration Effects

For this videos, tutorial data is available from https://bit.ly/bioxtasd

5.6 Cite

5.6.1 RAW

If you use RAW in your research, please cite the newest RAW paper:

BioXTAS RAW: improvements to a free open-source program for small-angle X-ray scattering data reduction and analysis. J. B. Hopkins, R. E. Gillilan, and S. Skou. Journal of Applied Crystallography (2017). 50, 1545-1553.

DOI: 10.1107/S1600576717011438

You can also cite the previous RAW paper if you like:

BioXTAS RAW, a software program for high-throughput automated small-angle X-ray scattering data reduction and preliminary analysis. S. S. Nielsen, K. Noergaard Toft, D. Snakenborg, M. G. Jeppesen, J. K. Jacobsen, B. Vestergaard, J. P. Kutteraand L. Arleth. Journal of Applied Crystallography (2009). 42, 959-964.

DOI: 10.1107/S0021889809023863

5.6.2 ATSAS

If you use RAW to control any of the ATSAS programs (AMBIMETER, DAMMIF, DAMMIN, DAMAVER, DAMCLUST, DATCLASS, GNOM, SASRES, SUPCOMB), in addition to the RAW paper, please cite the appropriate paper given on their documentation pages.

5.6.3 BIFT

If you used the BIFT method to determine a P(r) function in RAW, in addition to the RAW paper please cite the following paper:

Bayesian estimation of hyperparameters for indirect Fourier transformation in small-angle scattering. Hansen, S. Journal of Applied Crystallography (2000) 33, 1415-1421.

DOI: 10.1107/S0021889800012930

5.6.4 Evolving Factor Analysis (EFA)

If you used the EFA function in RAW to deconvolve overlapping chromatography data, in addition to the RAW paper please cite the following paper:

Domain Movements upon Activation of Phenylalanine Hydroxylase Characterized by Crystallography and Chromatography-Coupled Small-Angle X-ray Scattering. S. P. Meisburger, A. B. Taylor, C. A. Khan, S. Zhang, P. F. Fitzpatrick, N. Ando. Journal of the American Chemical Society (2016). 138(20), 6506-6516.

DOI: 10.1021/jacs.6b01563

5.6.5 Electron Density (DENSS)

DENSS

If you used the electron density (DENSS) function in RAW to calculate electron density, in addition to the RAW paper please cite the following paper:

Ab initio electron density determination directly from solution scattering data. T. D. Grant. Nature Methods (2018) 15, 191–193.

DOI: 10.1038/nmeth.4581

5.6.6 REGularized Alternating Least Squares (REGALS)

If you used the REGALS function in RAW to deconvolve SAXS data, in addition to the RAW paper please cite the following paper:

REGALS: a general method to deconvolve X-ray scattering data from evolving mixtures S. P. Meisburger, D. Xu, and N. Ando. IUCrJ (2021). 8(2), 225-237.

DOI: 10.1107/S2052252521000555

REGALS source code is available here: https://github.com/ando-lab/regals

5.6.7 Molecular Weight

Volume of Correlation

If you used the volume of correlation method to determine molecular weight in RAW, in addition to the RAW paper please cite the following paper:

Accurate assessment of mass, models and resolution by small-angle scattering. Rambo, R.P. & Tainer, J.A. Nature (2013). 496, 477-481

DOI: 10.1038/nature12070

Corrected Porod Volume

If you used the corrected Porod volume method to determine molecular weight in RAW, in addition to the RAW paper please cite the following paper:

SAXSMoW 2.0: Online calculator of the molecular weight of proteins in dilute solution from experimental SAXS data measured on a relative scale. V. Piiadov, E. Ares de Araujo, M. Oliveira Neto, A. F. Craievich, and I. Polikarpov. Protein Science (2019). 28(2), 454-473.

DOI: 10.1002/pro.3528

Bayesian Inference

If you used the Bayesian inference method (datbayes) to determine molecular weight in RAW, in addition to the RAW paper please cite the following paper:

Consensus Bayesian assessment of protein molecular mass from solution X-ray scattering data. Hajizadeh, N. R., Franke, D., Jeffries, C. M. & Svergun, D. I. (2018). Sci. Rep. 8, 7204.

DOI: 10.1038/s41598-018-25355-2

Comparison to known structures

If you used the comparison to known structures (Shape&Size) method to determine molecular weight in RAW, in addition to the RAW paper please cite the following paper:

Machine Learning Methods for X-Ray Scattering Data Analysis from Biomacromolecular Solutions Franke, D., Jeffries, C. M. & Svergun, D. I. (2018). Biophys. J. 114, 2485–2492.

DOI: 10.1016/j.bpj.2018.04.018

5.6.8 Baseline Correction

If you used the integral baseline correction method in RAW, in addition to the RAW paper please cite the following paper:

US-SOMO HPLC-SAXS module: dealing with capillary fouling and extraction of pure component patterns from poorly resolved SEC-SAXS data. E. Brookes, P. Vachette, M. Rocco, and J. Pérez. Journal of Applied Crystallography (2016). 49, 1827-1841.

DOI: 10.1107/S1600576716011201

5.7 RAW API

The RAW API allows you to install BioXTAS RAW as a python package without the GUI. This lets you import RAW directly into your python scripts and use the API to call any of the functions in RAW. This is great for creating custom processing scripts, either for unusual datasets that aren't handled in the GUI, or to ensure reproducibility in your analysis. It can also be used as the basis for an open-source automated SAXS data processing pipeline at a beamline or homesource.

5.7.1 Installing the API

Installation

Users should proceed as if *installing RAW from source on their OS of choice.* If you are not going to use the RAW GUI from source, then you do not need to install the wx package. Once you've installed all required packages and downloaded and unpacked the source code, you then proceed differently:

1. Open a terminal window and cd into the top level source directory.
2. Install the RAW API using pip with the command `pip install .`

The RAW API is now installed and ready to use.

Using the RAW GUI from the installed package

If you don't want to have RAW hanging around in several places, it is possible to run the RAW GUI directly from the installed API package on most systems. When you install the RAW package, it creates a bioxtas_raw script that should be in the system path. You can simply run bioxtas_raw from the command line and the RAW GUI should start.

The only catch to this is, that on MacOS using the anaconda python distribution, that doesn't work. This has to do with the fact that using the anaconda python you have to run GUI programs as pythonw, not python (on a deeper level this is an issue with Framework builds of python), which can't be done for this script.

5.7.2 Getting Started

This will guide you through several basic tasks using the API, including how to import the API, load settings, and load and save data. Specific file names refer to the RAW Tutorial Data, which you can download. File paths are given from the top level directory of that data.

Importing the API

Once installed, the RAW API is imported just like any other python package. We recommend that you import just the RAWAPI package, as in the following example.

```
import bioxtasraw.RAWAPI as raw
```

Loading settings

Many functions in the API use RAW settings to provide certain parameters for the function (e.g. the calibration parameters and mask used to radially average images). So it is a good idea to load a settings file at the start of your program.

```
my_settings = raw.load_settings('./standards_data/SAXS.cfg')
```

While settings can be created and saved using the API, we recommend using the RAW GUI to create and save your settings, then importing them into the API.

Loading data

Usually you'll need to start by loading data into your program. Here we show how to load images, profiles, IFTs, and series.

Loading images as scattering profiles

One of the most common tasks is loading images and radially averaging them into 1D scattering profiles. This is easily accomplished with the API.

```
##Define a list of image filenames to load.
buffer_images = ['./standards_data/GIbuf2_A9_18_001_0000.tiff',
    './standards_data/GIbuf2_A9_18_001_0001.tiff']

#Load and radially average images
profiles, imgs = raw.load_and_integrate_images(buffer_images, my_settings)
```

Loading scattering profiles

Another common task is loading data that is already saved as a 1D scattering profile, usually a .dat file.

```
#Define a list of profile filenames to load
profile_names = ['./reconstruction_data/glucose_isomerase.dat']

#Load the profiles
profiles = raw.load_profiles(profile_names)
```

Loading inverse Fourier transforms (IFTs)

You can use the API to load IFT files containing P(r) functions, either GNOM .out files or .ift files from RAW's BIFT algorithm.

```
#Define a list of IFT filenames to load
ift_names = ['./reconstruction_data/gi_complete/glucose_isomerase.out',
    './reconstruction_data/gi_complete/glucose_isomerase.ift']

#Load the IFTs
ifts = raw.load_ifts(ift_names)
```

Loading series

There are two ways you can load a series. The first is loading a .hdf5 or .sec series file saved by RAW.

```
#Define a list of series filenames to load
series_names = ['./series_data/phehc_sec.hdf5', './series_data/xylanase.hdf5']

#Load the series
series = raw.load_series(series_names)
```

Alternatively, you can load in all of the individual profiles in the series, then use the API to convert those set of profiles into a series.

```
import glob

#Define a list of profile filenames to load
profile_names = sorted(glob.glob('./series_data/sec_sample_2/BSA_001_*.dat'))

#Load the profiles
profiles = raw.load_profiles(profile_names)

#Convert the profiles to a series
series = raw.profiles_to_series(profiles)
```

Note that the input profiles should be in the order they appear in the series.

Working with profiles

RAW uses a custom defined class called a SASM (SAS measurement) to contain information about scattering profiles, including the q, I, and uncertainty data as well as metadata data about analysis results.

Accessing q, I, and uncertainty data

RAW SASMs contain several different versions of the q, I, and uncertainty data. Most commonly, you'll want to access the data using the `getQ()`, `getI()` and `getErr()` functions

```
profile_names = ['./reconstruction_data/glucose_isomerase.dat']
profiles = raw.load_profiles(profile_names)

gi_profile = profiles[0]

```

(continues on next page)

(continued from previous page)

```
q = gi_profile.getQ()
intensity = gi_profile.getI()
error = gi_profile.getErr()
```

This contains data that has been truncated, scaled and offset according to the profile settings. If you want to access the scaled, offset, and un-truncated data (e.g. without zeros at the beginning skipped for loaded images) you can access `profile.q`, `profile.i` and `profile.err` attributes. If there is any truncation, you can get that using `profile.getQrange()`. So, for example

```
q_range = gi_profile.getQrange()

gi_profile.getQ() == gi_profile.q[q_range[0]:q_range[1]]
gi_profile.getI() == gi_profile.i[q_range[0]:q_range[1]]
gi_profile.getErr() == gi_profile.err[q_range[0]:q_range[1]]
```

are all true.

If you want the raw profile data, without any truncation, scaling, or offset, you can use the `getRawQ()` `getRawI()` and `getRawErr()` functions.

Analyzing the profile

Many of the RAW analysis functions act on a single scattering profile. For example, to automatically find the best range for the Guinier fit and calculate the Rg and I(0), you can do:

```
guinier_results = raw.auto_guinier(gi_profile)
```

Accessing profile metadata

The profile saves various bits of metadata to a dictionary. If the profile was created by RAW this includes information on how the profile was created and various metadata parameters from the data collection. It also includes analysis information. To get all of the metadata you can do:

```
metadata = gi_profile.getAllParameters()
```

To get a specific category of metadata,

```
analysis = gi_profile.getParameter('analysis')

guinier_rg = analysis['guinier']['Rg']
```

Working with IFTs

RAW uses a custom defined class called a IFTM (IFT measurement) to contain information about IFTs, including the P(r) function, the fit of the P(r) function to the data, and metadata about the P(r) function.

Access the P(r) function and fit

All of the P(r) data and fit is accessible as attributes of the class.

```
ift_names = ['./reconstruction_data/gi_complete/glucose_isomerase.out']
ifts = raw.load_ifts(ift_names)

gi_ift = ifts[0]

#Get the P(r) function itself
p = gi_ift.p #P(r)
r = gi_ift.r
err = gi_ift.err #Uncertainty in P(r)

#Get the original data and the P(r) fit to the original data
q = gi_ift.q_orig
i = gi_ift.i_orig
err = gi_ift.err_orig
fit = gi_ift.i_fit

#Get the fit extrapolated to q=0.
q_extrap = gi_ift.q_extrap
fit_extrap = gi_ift.i_extrap
```

Analyzing the IFT

There are several functions that take the IFTM as input for analysis, including ambimeter and the various 3D reconstruction methods. Note that analysis methods from the ATSAS package require a GNOM IFT, whereas those natively implemented in RAW (DENSS) work on either GNOM or BIFT IFTs.

```
score, categories, evaluation = raw.ambimeter(gi_ift)
```

Accessing IFT metadata

IFT metadata can be accessed in the same way as for profiles:

```
metadata = gi_ift.getAllParameters()

dmax = gi_ift.getParameter('dmax')
```

Working with series

RAW uses a custom defined class called a SECM (SEC measurement, a slightly outdated name) to contain information about series, including the individual scattering profiles, total and mean intensity as a function of frame number, and calculated parameters such as R_g as a function of frame number.

Accessing the series data

In order to visualize the series data it is common to plot total or mean intensity as a function of frame number. You can get that data as:

```
series_names = ['./series_data/baseline.hdf5']
series = raw.load_series(series_names)

```

(continues on next page)

(continued from previous page)

```
my_series = series[0]

frames = my_series.getFrames()
total_i = my_series.getIntI()
mean_i = my_series.getMeanI()
```

The calculated parameter data is similarly accessed:

```
rg = my_series.getRg()
i0 = my_series.getI0()
mw_vc = my_series.getVcMW()[0]
mw_vp = my_series.getVpMW()[0]
```

The intensity for subtracted of baseline corrected data is accessed by specifying the data type

```
subtracted_total_i = my_series.getIntI('sub')
subtracted_mean_i = my_series.getMeanI('sub')
```

Note that for data with baseline corrected profiles you would use 'baseline'

If you want to access the underlying profiles, it is done similarly.

```
#Gets all profiles in the series
profiles = my_series.getAllSASMs()
sub_profiles = my_series.getAllSASMs('sub')

#Gets a single profile in the series, zero indexed
profile_5 = my_series.getSASM(5)
sub_profile_5 = my_series.getSASM(5, 'sub')

#Get profiles from the series in a given range, zero indexed
profiles_roi = my_series.getSASMList(10, 20)
sub_profiles_roi = my_series.getSASMList(10, 20, 'sub')
```

Analyzing the series

Any analysis you can do on series in the GUI can be done with the API. For example, to automatically find a good buffer region:

```
success, region_start, region_end = raw.find_buffer_range(my_series)
```

Accessing series metadata

Series in RAW have a lot of associated metadata, such as the buffer range used for subtraction, the start and end of the baseline correction ranges, or the sample range. Most of these are accessible as attributes of the SECM.

```
buffer_range = my_series.buffer_range
sample_range = my_series.sample_range
```

Saving data

After you process your data you will want to save it. Here we show how to save profiles, IFTs, and series.

Saving scattering profiles

Suppose you have the scattering profile `my_profile`. You would save the profile as:

```
raw.save_profile(my_profile, 'my_profile.dat', './my_profile_dir')
```

Saving inverse Fourier transforms (IFTs)

Suppose you have the IFT `my_ift`. You would save the IFT as:

```
raw.save_ift(my_ift, 'my_ift.out', './my_ift_dir')
```

Note that you use the `.out` extension for GNOM IFTs, and the `.ift` extension for BIFT IFTs.

Saving series

Suppose you have the series `my_series`. You would save the series as:

```
raw.save_series(my_series, 'my_series.hdf5', './my_series_dir')
```

5.7.3 Examples

Here are some simple examples of how to use the RAW API:

Analyzing a scattering profile

One of the most common tasks is analyzing a scattering profile. Here's an example of how that might be done, from Guinier analysis through creating a 3D reconstruction.

Guinier fit and MW

```
import os
import shutil
import numpy as np
import bioxtasraw.RAWAPI as raw


#Load the settings
settings = raw.load_settings('./standards_data/SAXS.cfg')


#Load the profile of interest
profile_names = ['./reconstruction_data/glucose_isomerase.dat']
profiles = raw.load_profiles(profile_names)

gi_prof = profiles[0]


#Automatically calculate the Guinier range and fit
```

(continues on next page)

(continued from previous page)

```
(rg, i0, rg_err, i0_err, qmin, qmax, qrg_min, qrg_max, idx_min, idx_max,
    r_sq) = raw.auto_guinier(gi_prof, settings=settings)


#Calculate M.W. using 6 different methods
mw_ref = raw.mw_ref(gi_prof, 0.47, settings=settings)
mw_abs = raw.mw_abs(gi_prof, 0.47, settings=settings)
mw_vp, pvol_cor, pvol, vp_qmax = raw.mw_vp(gi_prof, settings=settings)
mw_vc, vcor, mw_err, vc_qmax = raw.mw_vc(gi_prof, settings=settings)
mw_bayes, mw_prob, ci_lower, ci_upper, ci_prob = raw.mw_bayes(gi_prof)
mw_datclass, shape, dmax_datclass = raw.mw_datclass(gi_prof)
```

Calculate IFTs

```
#Calculate the IFT using BIFT
(gi_bift, gi_bift_dmax, gi_bift_rg, gi_bift_i0, gi_bift_dmax_err,
    gi_bift_rg_err, gi_bift_i0_err, gi_bift_chi_sq, gi_bift_log_alpha,
    gi_bift_log_alpha_err, gi_bift_evidence,
    gi_bift_evidence_err) = raw.bift(gi_prof, settings=settings, single_proc=True)


#Calculate the IFT using DATGNOM
(gi_datgnom_ift, gi_datgnom_dmax, gi_datgnom_rg, gi_datgnom_i0,
    gi_datgnom_rg_err, gi_datgnom_i0_err, gi_datgnom_total_est,
    gi_datgnom_chi_sq, gi_datgnom_alpha, gi_datgnom_quality) = raw.datgnom(gi_prof)


#Use the auto_dmax function to find the best Dmax and then calculate the IFT using␣
↪GNOM
dmax = raw.auto_dmax(gi_prof)

(gi_gnom_ift, gi_gnom_dmax, gi_gnom_rg, gi_gnom_i0, gi_gnom_rg_err,
    gi_gnom_i0_err, gi_gnom_total_est, gi_gnom_chi_sq, gi_gnom_alpha,
    gi_gnom_quality) = raw.gnom(gi_prof, dmax)
```

Save the profile, IFTs, and analysis summary

```
#Save the profile, IFTs, and analysis summary
if not os.path.exists('./api_results'):
    os.mkdir('./api_results')

raw.save_profile(gi_prof, 'gi.dat', './api_results')
raw.save_ift(gi_bift, 'gi.ift', './api_results')
raw.save_ift(gi_gnom_ift, 'gi.out', './api_results')

raw.save_report('gi_report.pdf', './api_results', [gi_prof],
    [gi_gnom_ift, gi_bift])
```

Create a bead model reconstruction

```
#Calculate the ambiguity using AMBIMETER
a_score, a_cats, a_eval = raw.ambimeter(gi_gnom_ift)


#Create individual bead model reconstructions
if not os.path.exists('./api_results/gi_dammif'):
    os.mkdir('./api_results/gi_dammif')
else:
    files = os.listdir('./api_results/gi_dammif')
    for f in files:
        os.remove(os.path.join('./api_results/gi_dammif', f))

chi_sq_vals = []
rg_vals = []
dmax_vals = []
mw_vals = []
ev_vals = []

for i in range(5):
    chi_sq, rg, dmax, mw, ev = raw.dammif(gi_gnom_ift,
        'gi_{:02d}'.format(i+1), './api_results/gi_dammif', mode='Fast')

    chi_sq_vals.append(chi_sq)
    rg_vals.append(rg)
    dmax_vals.append(dmax)
    mw_vals.append(mw)
    ev_vals.append(ev)


#Average the bead model reconstructions
damaver_files = ['gi_{:02d}-1.pdb'.format(i+1) for i in range(5)]

(mean_nsd, stdev_nsd, rep_model, result_dict, res, res_err,
    res_unit) = raw.damaver(damaver_files, 'gi',
    './api_results/gi_dammif')


#Cluster the bead model reconstructions
cluster_list, distance_list = raw.damclust(damaver_files, 'gi',
    './api_results/gi_dammif')


#Refine the bead model
chi_sq, rg, dmax, mw, ev = raw.dammin(gi_gnom_ift, 'refine_gi',
    './api_results/gi_dammif', 'Refine',
    initial_dam='gi_damstart.pdb')


#Align the refined bead model with a high resolution structure
shutil.copy('./reconstruction_data/gi_complete/1XIB_4mer.pdb',
    './api_results/gi_dammif')

raw.supcomb('refine_gi-1.pdb', '1XIB_4mer.pdb', './api_results/gi_dammif')
```

Create an electron density reconstruction

```
#Create individual electron density reconstructions
if not os.path.exists('./api_results/gi_denss'):
    os.mkdir('./api_results/gi_denss')
else:
    files = os.listdir('./api_results/gi_denss')
    for f in files:
        os.remove(os.path.join('./api_results/gi_denss', f))

rhos = []
chi_vals = []
rg_vals = []
support_vol_vals = []
sides = []
fit_data = []

for i in range(5):
    (rho, chi_sq, rg, support_vol, side, q_fit, I_fit, I_extrap,
        err_extrap, all_chi_sq, all_rg, all_support_vol) = raw.denss(gi_gnom_ift,
        'gi_{:02d}'.format(i+1), './api_results/gi_denss', mode='Fast')

    rhos.append(rho)
    chi_vals.append(chi_sq)
    rg_vals.append(rg)
    support_vol_vals.append(support_vol)
    sides.append(side)
    fit_data.append([q_fit, I_fit, I_extrap, err_extrap])


#Average the electron reconstructions
(average_rho, mean_cor, std_cor, threshold, res, scores,
    fsc) = raw.denss_average(np.array(rhos), side, 'gi_average',
    './api_results/gi_denss')


#Refine the electron density
(refined_rho, refined_chi_sq, refined_rg, refined_support_vol, refined_side,
    refined_q_fit, refined_I_fit, refined_I_extrap,
    refined_err_extrap, all_chi_sq, all_rg,
    all_support_vol) = raw.denss(gi_gnom_ift, 'gi_refine',
    './api_results/gi_denss', mode='Fast',
    initial_model=average_rho)


#Align the electron density with a high resolution structure
shutil.copy('./reconstruction_data/gi_complete/1XIB_4mer.pdb',
    './api_results/gi_denss')

aligned_density, score = raw.denss_align(refined_rho, refined_side,
     '1XIB_4mer.pdb', './api_results/gi_denss',
     'gi_refined_aligned', './api_results/gi_denss')
```

Creating a subtracted scattering profile

This example shows how to create a subtracted scattering profile from batch mode SAXS data. While this particular example uses images as the starting point, the same process is easily applied to .dat file as well.

```python
import glob
import os
import bioxtasraw.RAWAPI as raw

#Load settings
settings = raw.load_settings('./standards_data/SAXS.cfg')

#Load buffer and sample profiles
buffer_files = sorted(glob.glob('./standards_data/GIbuf2_A9_18_001_*.tiff'))
buffers, imgs = raw.load_and_integrate_images(buffer_files, settings)

sample_files = sorted(glob.glob('./standards_data/GI2_A9_19_001_*.tiff'))
samples, imgs = raw.load_and_integrate_images(sample_files, settings)

#Test for similarity between buffer and sample profiles
buf_pvals, buf_corrected_pvals, buf_failed_comps = raw.cormap(buffers[1:],
    buffers[0])
sam_pvals, sam_corrected_pvals, sam_failed_comps = raw.cormap(samples[1:],
    samples[0])

#Average buffer and sample profiles
buf_avg = raw.average(buffers)
sam_avg = raw.average(samples)

#Subtract buffer and sample profiles
gi_prof = raw.subtract([sam_avg], buf_avg)[0]

#Save the averaged and subtracted profiles
if not os.path.exists('./api_results/gi_denss'):
    os.mkdir('./api_results/gi_denss')

raw.save_profile(buf_avg, datadir='./api_results')
raw.save_profile(sam_avg, datadir='./api_results')
raw.save_profile(gi_prof, datadir='./api_results')
```

Analyzing SEC-SAXS data

The following examples shows how to carry out analysis on SEC-SAXS data.

Finding and setting buffer and sample regions

```python
import glob
import os
import bioxtasraw.RAWAPI as raw

#Load the series
profile_names = sorted(glob.glob('./series_data/sec_sample_1/profile_001_*.dat'))
profiles = raw.load_profiles(profile_names)
series = raw.profiles_to_series(profiles)
```

(continues on next page)

(continued from previous page)

```python
#Find an appropriate buffer range for subtraction
success, start_idx, end_idx = raw.find_buffer_range(series)

#Set the buffer range for the series
buffer_range = [[start_idx, end_idx]]

(sub_profiles, rg, rger, i0, i0er, vcmw, vcmwer,
    vpmw) = raw.set_buffer_range(series, buffer_range)

#Find an appropriate sample range for subtraction
success, start_idx, end_idx = raw.find_sample_range(series)

#Set the sample range for the series
sample_range = [[start_idx, end_idx]]

sub_profile = raw.set_sample_range(series, sample_range)

#Save the analysis done to the series
if not os.path.exists('./api_results'):
    os.mkdir('./api_results')

raw.save_series(series, 'profile_series.hdf5', './api_results')
```

Once you have the subtracted profile generated by `set_sample_region` you can carry out analysis on the individual profile as in *the scattering profile analysis tutorial.*

Applying a linear baseline correction

```python
import os
import bioxtasraw.RAWAPI as raw

#Load series
xyl_series = raw.load_series(['./series_data/xylanase.hdf5'])[0]

#Set buffer range
success, start, end = raw.find_buffer_range(xyl_series)

(sub_profiles, rg, rger, i0, i0er, vcmw, vcmwer,
    vpmw) = raw.set_buffer_range(xyl_series, [[start, end]])

#Validate baseline range
(lin_valid, lin_valid_results, lin_similarity_results, lin_svd_results,
    lin_intI_results, lin_other_results) = raw.validate_baseline_range(
    xyl_series, [0, 10], [1132, 1142], 'Linear')

#Do baseline correction
(lin_bl_cor_profiles, lin_rg, lin_rger, lin_i0, lin_i0er, lin_vcmw, lin_vcmwer,
    lin_vpmw, lin_bl_corr, lin_fit_results) = raw.set_baseline_correction(
    xyl_series, [0, 10], [1132, 1142], 'Linear')

#Find an appropriate sample range
success, start_idx, end_idx = raw.find_sample_range(xyl_series,
    profile_type='baseline')
```

(continues on next page)

(continued from previous page)

```
sample_range = [[start_idx, end_idx]]

sub_profile = raw.set_sample_range(xyl_series, sample_range,
    profile_type='baseline')

#Save the analysis done to the series
if not os.path.exists('./api_results'):
    os.mkdir('./api_results')

raw.save_series(xyl_series, 'xyl_series.hdf5', './api_results')
```

Note that setting a buffer range is only necessary if buffer subtraction has not already been performed on the series.

Applying an integral baseline correction

```
import os
import bioxtasraw.RAWAPI as raw

#Load series
series = raw.load_series(['./series_data/baseline.hdf5'])[0]

#Find baseline range
(start_found, end_found, start_range,
    end_range) = raw.find_baseline_range(series)

#Do baseline correction
(int_bl_cor_profiles, int_rg, int_rger, int_i0, int_i0er, int_vcmw,
    int_vcmwer, int_vpmw, int_bl_corr,
    int_fit_results) = raw.set_baseline_correction(series, start_range,
    end_range, 'Integral')

#Set an appropriate sample range for subtraction
success, start_idx, end_idx = raw.find_sample_range(series,
    profile_type='baseline')

sample_range = [[start_idx, end_idx]]

sub_profile = raw.set_sample_range(series, sample_range,
    profile_type='baseline')

#Save the analysis done to the series
if not os.path.exists('./api_results'):
    os.mkdir('./api_results')

raw.save_series(series, 'profile_series_bl.hdf5', './api_results')
```

Validating buffer and sample regions

You can validate whatever buffer or sample region you want to set. Note that this validation is done as part of the `find_buffer_region` and `find_sample_region` functions, so there's no need to do it on regions found with those functions.

```
import bioxtasraw.RAWAPI as raw

#Load series
xyl_series = raw.load_series(['./series_data/xylanase.hdf5'])[0]

#Validate buffer region
buffer_range = [[180, 240], [500, 560]]

(valid, similarity_results, svd_results,
    intI_results) = raw.validate_buffer_range(xyl_series, buffer_range)

if not valid:
    success, start, end = raw.find_buffer_range(xyl_series)
    buffer_range = [[start, end]]

(sub_profiles, rg, rger, i0, i0er, vcmw, vcmwer,
        vpmw) = raw.set_buffer_range(xyl_series, [[start, end]])

#Validate sample region
sample_range = [[785, 815]]

(valid, similarity_results, param_results, svd_results,
    sn_results) = raw.validate_sample_range(xyl_series, sample_range)

if not valid:
    success, start_idx, end_idx = raw.find_sample_range(xyl_series)
    sample_range = [[start_idx, end_idx]]

sub_profile = raw.set_sample_range(xyl_series, sample_range)
```

Validating baseline regions

You can validate baseline regions. Note that this validation is done as part of the `find_baseline_region` for an integral baseline, so is not necessary in that case. Also, the linear baseline validation is not terribly useful at the moment, it almost always returns invalid.

```
import bioxtasraw.RAWAPI as raw

#Load series
series = raw.load_series(['./series_data/baseline.hdf5'])[0]

#Validate linear baseline range
(valid, valid_results, similarity_results, svd_results, intI_results,
    other_results) = raw.validate_baseline_range( series, [0, 10],
    [953, 963], 'Linear')

#Validate integral baseline range
(valid, valid_results, similarity_results, svd_results, intI_results,
    other_results) = raw.validate_baseline_range(series, [539, 568],
    [817, 846])
```

Carrying out SVD, EFA, and REGALS

You can carry out SVD, EFA, and REGALS from the API (though without the GUI you have to know what the appropriate ranges are for each EFA component as input).

```
import bioxtasraw.RAWAPI as raw

# Load data
phehc_series = raw.load_series(['./series_data/phehc_sec.hdf5'])[0]

#Do SVD
svd_s, svd_U, svd_V = raw.svd(phehc_series)

#Do EFA
efa_ranges = [[149, 197], [164, 321], [320, 364]]

(efa_profiles, efa_converged, efa_conv_data,
    efa_rotation_data) = raw.efa(phehc_series, efa_ranges)

# Do REGALS
prof1_settings = {
    'type'          : 'simple',
    'lambda'        : 0.0,
    'auto_lambda'   : True,
    'kwargs'        : {},
    }

conc1_settings = {
    'type'          : 'smooth',
    'lambda'        : 1.0,
    'auto_lambda'   : True,
    'kwargs'                : {
        'xmin'              : 145,
        'xmax'              : 195,
        'Nw'                : 50,
        'is_zero_at_xmin'   : True,
        'is_zero_at_xmax'   : True,
        }
    }

prof2_settings = {
    'type'          : 'simple',
    'lambda'        : 0.0,
    'auto_lambda'   : True,
    'kwargs'        : {},
    }

conc2_settings = {
    'type'          : 'smooth',
    'lambda'        : 3.0e3,
    'auto_lambda'   : False,
    'kwargs'                : {
        'xmin'              : 160,
        'xmax'              : 325,
        'Nw'                : 50,
        'is_zero_at_xmin'   : True,
        'is_zero_at_xmax'   : True,
```

(continues on next page)

(continued from previous page)

```
        }
    }

prof3_settings = {
    'type'          : 'simple',
    'lambda'        : 0.0,
    'auto_lambda'   : True,
    'kwargs'        : {},
    }

conc3_settings = {
    'type'          : 'smooth',
    'lambda'        : 1.0,
    'auto_lambda'   : True,
    'kwargs'                : {
        'xmin'              : 320,
        'xmax'              : 383,
        'Nw'                : 50,
        'is_zero_at_xmin'   : True,
        'is_zero_at_xmax'   : True,
        }
    }

comp_settings = [(prof1_settings, conc1_settings),
    (prof2_settings, conc2_settings), (prof3_settings, conc3_settings)]

(regals_profiles, regals_ifts, concs, reg_concs, mixture, params,
    residual) = raw.regals(phehc_series, comp_settings)
```

As the API gets more use, we hope there will be some open source projects that you can check out that use the API in some way. If you have such a project you'd like linked here, *please let us know*.

Projects of interest:

- The BioCAT processing pipeline

5.7.4 The API

Main API

`bioxtasraw.RAWAPI.`**`ambimeter`**(*ift*, *qRg_max=4*, *save_models='none'*, *save_prefix=None*, *datadir=None*, *write_ift=True*, *filename=None*, *atsas_dir=None*)

Evaluates ambiguity of a potential 3D reconstruction from a GNOM IFT (.out file) by running Ambimeter from the ATSAS package. Requires separate installation of the ATSAS package. Doesn't work on BIFT IFTs. Returns -1 and '' if it fails to run.

Parameters

- **ift** (*bioxtasraw.SASM.IFTM*) – The GNOM IFT to be evaluated. If write_ift is False, an IFT already on disk is used and this parameter can be `None`.
- **qRg_max** (*float, optional*) – The maximum qRg to be used when evaluating the ambiguity. Allowed range is 3-7, default is 4.
- **save_models** (*{'all', 'best', 'none'} str, optional*) – Whether to save all, the single best, or none of the models that ambimeter finds to be similar to the

input ift. Default is 'none'. If set to 'all' or 'best', save_prefix and datadir parameters must be provided.

- **save_prefix** (*str, optional*) – The prefix to use for the saved modes, if any are saved.
- **datadir** (*str, optional*) – The datadir to use for reading a IFT already on disk and saving models from ambimeter.
- **write_profile** (*bool, optional*) – If True, the input ift is written to file. If False, then the input ift is ignored, and the ift specified by datadir and filename is used. This is convenient if you are trying to process a lot of files that are already on disk, as it saves having to read in each file and then save them again. Defaults to True. If False, you must provide a value for the datadir and filename parameters.
- **filename** (*str, optional*) – The filename of an ift on disk. Used if write_profile is False.
- **atsas_dir** (*str, optional*) – The directory of the atsas programs (the bin directory). If not provided, the API uses the auto-detected directory.

Returns

- **score** (*float*) – The ambiguity score (A score), which is log base 10 of the number of compatible shape categories.
- **categories** (*int*) – The number of compatible shape categories.
- **evaluation** (*str*) – The Ambimeter evaluation of ift based on the ambiguity score.

Raises `SASEXceptions.NoATSASError` – If the Ambimeter program cannot be found in the ATSAS directory or running Ambimeter times out (>120 s).

`bioxtasraw.RAWAPI.`**`auto_dmax`**(*profile, dmax_thresh=0.01, dmax_low_bound=0.5, dmax_high_bound=1.5, settings=None, use_atsas=True, single_proc=True*)

Automatically calculate the maximum dimension (Dmax) value of a profile. By default uses BIFT, DATGNOM, and DATCLASS to find a starting value and then refines that starting value using GNOM. If use_atsas is False it just returns the BIFT value. It requires having an Rg from the Guinier fit.

Parameters

- **profile** (*bioxtasraw.SASM.SASM*) – The profile to find the Dmax for.
- **dmax_thresh** (*float, optional*) – The threshold for refining the Dmax value. If the value of the P(r) at Dmax is greater than this threshold times the maximum value of the P(r) function Dmax is extended until the value falls below this fractional threshold or the Dmax exceeds the initial estimated value times the dmax_high_bound value. Defaults is 0.01.
- **dmax_low_bound** (*float, optional*) – If the end of the P(r) function contains negative values, Dmax is reduced until either no negative values exist or the Dmax becomes less than this parameter times the initial estimated value of Dmax. Default is 0.5.
- **dmax_high_bound** (*float, optional*) – If the value of the P(r) at Dmax is greater than dmax_thres times the maximum value of the P(r) function Dmax is extended until the value falls below that fractional threshold or the Dmax exceeds the initial estimated value times this parameter. Default is 1.5.
- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. Passed to BIFT. Default is None, which uses the default RAW settings.

- **use_atsas** (*bool, optional*) – Whether to use ATSAS functions. If False, simply returns the Dmax found by BIFT. Default is True.
- **single_proc** (*bool, optional*) – Whether to use one or multiple processors. Defaults to True.

Returns **dmax** – The maximum dimension found by this algorithm. Returns -1 if not found.

Return type int

bioxtasraw.RAWAPI.**auto_guinier**(*profile*, *error_weight=True*, *single_fit=True*, *settings=None*)

Automatically calculates the Rg and I(0) values from the Guinier fit by determining the best range for the Guinier fit.

Parameters

- **profile** (*bioxtasraw.SASM.SASM*) – The profile to perform the Guineir fit on.
- **error_weight** (*bool, optional*) – If True (default), then the Guinier fit is calculated in an error weighted fashion. If not, the Guinier fit is calculated without error weight. This is overridden by the value in the settings if a settings object is provided.
- **single_fit** (*bool, optional*) – If True (default), then after the correct range for the Guinier fit is found a traditional Guinier fit is performed using that range. If False, currently the same is true. In the future, if False then the Rg and I(0) values may be averages over some range of best Guinier fit intervals.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – RAW settings containing relevant parameters. If provided, the error_weight parameter will be overridden with the value in the settings. Default is None.

Returns

- **rg** (*float*) – The Rg value of the fit.
- **i0** (*float*) – The I(0) value of the fit.
- **rg_err** (*float*) – The uncertainty in Rg. This is calculated as the largest of the uncertainty returned from autorg and the uncertatiny as calculated from the covariance of the Guinier fit with the autorg determined ranges.
- **i0_err** (*float*) – The uncertainty in I(0). This is calculated as the largest of the uncertainty returned from autorg and the uncertatiny as calculated from the covariance of the Guinier fit with the autorg determined ranges.
- **qmin** (*float*) – The minimum q value of the Guinier fit.
- **qmax** (*float*) – The maximum q value of the Guinier fit.
- **qRg_min** (*float*) – The q*Rg value at the minimmum q value of the Guinier fit.
- **qRg_max** (*float*) – The q*Rg value at the maximum q value of the Guinier fit.
- **idx_min** (*int*) – The minimum index of the q vector used for Guinier fit.
- **idx_max** (*int*) – The maximum index of the q vector used for the GUinier fit.
- **r_sqr** (*float*) – The r^2 value of the fit.

bioxtasraw.RAWAPI.**average**(*profiles*, *forced=False*)

Averages the input profiles into a single averaged profile. Note that unlike in the RAW GUI there is no automatic testing for similarity in this average function.

Parameters

- **profiles** (*list*) – A list of profiles (*bioxtasraw.SASM.SASM*) to average.

- **forced** (*bool, optional*) – If True, RAW will attempt to average profiles even if the q vectors do not agree. Defaults to False.

Returns **avg_profile** – The average profile.

Return type *bioxtasraw.SASM.SASM*

Raises `SASExceptions.DataNotCompatible` – If the average list contains data sets with different q vectors and not forced (or if it fails to find a solution even if forced).

`bioxtasraw.RAWAPI.`**`bift`**(*profile*, *idx_min=None*, *idx_max=None*, *pr_pts=100*, *alpha_min=150*, *alpha_max=10000000000.0*, *alpha_pts=16*, *dmax_min=10*, *dmax_max=400*, *dmax_pts=10*, *mc_runs=300*, *use_guinier_start=True*, *single_proc=True*, *nprocs=None*, *settings=None*)

Calculates the Bayesian indirect Fourier transform (BIFT) of a scattering profile to generate a P(r) function and determine the maximum dimension Dmax. Returns None and -1 values if BIFT fails.

Parameters

- **profile** (*bioxtasraw.SASM.SASM*) – The profile to calculate the BIFT for.
- **idx_min** (*int, optional*) – The index of the q vector that corresponds to the minimum q point to be used in the IFT. Default is to use the first point of the q vector, unless use_guinier_start is set.
- **idx_max** (*int, optional*) – The index of the q vector that corresponds to the maximum q point to be used in the IFT. Default is to use the last point of the q vector.
- **pr_pts** (*int, optional*) – The number of points in the calculated P(r) function. This should be less than the number of points in the scattering profile. If settings are provided, this is overridden by the value in the settings.
- **alpha_min** (*float, optional*) – The minimum value of alpha for the parameter search step. If settings are provided, this is overridden by the value in the settings. The value of alpha can go beyond this bound in the optimization step, so this is not a hard limit on alpha.
- **alpha_max** (*float, optional*) – The maximum value of alpha for the parameter search step. If settings are provided, this is overridden by the value in the settings. The value of alpha can go beyond this bound in the optimization step, so this is not a hard limit on alpha.
- **alpha_pts** (*int, optional*) – The number of points in the alpha search space, which will be logarithmically spaced between alpha_min and alpha_max. If settings are provided, this is overridden by the value in the settings.
- **dmax_min** (*float, optional*) – The minimum value of Dmax for the parameter search step. If settings are provided, this is overridden by the value in the settings. The value of Dmax can go beyond this bound in the optimization step, so this is not a hard limit on Dmax.
- **dmax_max** (*float, optional*) – The maximum value of Dmax for the parameter search step. If settings are provided, this is overridden by the value in the settings. The value of Dmax can go beyond this bound in the optimization step, so this is not a hard limit on Dmax.
- **dmax_pts** (*int, optional*) – The number of points in the Dmax search space, which will be linearly spaced between dmax_min and dmax_max. If settings are provided, this is overridden by the value in the settings.
- **mc_runs** (*int, optional*) – The number of monte carlo runs used to generate the uncertainty estimates for the P(r) function.

- **use_guiner_start** (*bool, optional*) – If set to True, and no idx_min idx_min is provided, if a Guinier fit has been done for the input profile, the start point of the Guinier fit is used as the start point for the IFT.
- **single_proc** (*bool, optional*) – Whether to use one or multiple processors. Defaults to True. In limited testing the single processor version has been found to be 2-3x faster than the multiprocessor version, but actual results may depend on the computer and the number of gird search points.
- **nprocs** (*int, optional*) – If specified, and single_proc is False, determines the number of processors to use for BIFT. Otherwise defaults to number of processors in the computer -1 (minimum 1).
- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, the pr_Pts, alpha_min, alpha_max, alpha_pts, dmax_min, dmax_max, dmax_pts, and mc_runs parameters will be overridden with the values in the settings. Default is None.

Returns

- **ift** (*bioxtasraw.SASM.IFTM*) – The IFT calculated by BIFT from the input profile.
- **dmax** (*float*) – The maximum dimension of the P(r) function found by BIFT.
- **rg** (*float*) – The real space radius of gyration (Rg) from the P(r) function.
- **i0** (*float*) – The real space scattering at zero angle (I(0)) from the P(r) function.
- **dmax_err** (*float*) – The uncertainty in the maximum dimension of the P(r) function found by BIFT.
- **rg_err** (*float*) – The uncertainty in the real space radius of gyration (Rg) from the P(r) function.
- **i0_err** (*float*) – The uncertainty in the real space scattering at zero angle (I(0)) from the P(r) function.
- **chi_sq** (*float*) – The chi squared value of the fit of the scattering profile calculated from the P(r) function to the input scattering profile.
- **log_alpha** (*float*) – Log base 10 of the alpha value for the IFT.
- **log_alpha_err** (*float*) – Log base 10 of the uncertainty in the alpha value for the IFT.
- **evidence** (*float*) – The Bayesian evidence of the IFT.
- **evidence_err** (*float*) – The uncertainty in the Bayesian evidence of the IFT.

`bioxtasraw.RAWAPI.`**`cormap`**(*profiles*, *ref_profile=None*, *correction='Bonferroni'*, *settings=None*)

Runs the cormap comparison test between the input profiles. If a reference profile is provided, then all of the profiles are compared to the reference profile. If not reference profile is provided, then all possible pairwise comparisons are run between the input profiles.

Parameters

- **profiles** (*list*) – The input profiles (*bioxtasraw.SASM.SASM*) to be compared.
- **ref_profile** (*bioxtasraw.SASM.SASM*, optional) – The reference profile to be used. If provided, all profiles are compared to this profile. If not provided (default) then all profiles are compared pairwise to each other.
- **correction** (*{'None', 'Bonferroni'} str, optional*) – What multiple testing correction to apply to the calculated pvalues. A value of 'None' applies no correction.

- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, the correction parameter will be overridden with the value in the settings. Default is None.

Returns

- **pvals** (*np.array*) – The p values from the comparison. If no reference is provided this is an NxN array, where N is the number of input profiles, and each index corresponds to the profile in profiles. So for example, pvals[0, 5] would correspond to a comparison between profiles[0] and profiles[5].

  If a reference is provided, pvals is a 1D array with the same size as profiles. There is a direct correspondence between the index of pvals and profiles. E.g. pvals[2] would correspond to the comparison of profiles[2] and the ref_profile.

- **corrected_pvals** (*np.array*) – As pvals, but with the corrected p values based on the input correction.

- **failed_comparisons** (*list*) – If any comparisons fail, the list contains the names of the two profiles (as determined by profile.getParameter('filename')) for which the comparison failed.

`bioxtasraw.RAWAPI.`**`damaver`**(*files*, *prefix*, *datadir*, *symmetry='P1'*, *atsas_dir=None*, *abort_event=<threading.Event object>*)

Runs DAMAVER from the ATSAS package on a set of files. Requires a separate installation of the ATSAS package. Function blocks until DAMAVER finishes.

Parameters

- **files** (*list*) – A list of strings of the filenames on disk that are the DAMAVER inputs. Must be just filenames, no paths, and all files must be in the same directory.
- **prefix** (*str*) – The prefix to be appended to the DAMAVER output files.
- **datadir** (*str*) – The data directory in which all of the input files are located. Also the location of the DAMAVER output.
- **symmetry** (*str, optional*) – The symmetry that DAMAVER will use during alignment. Accepts any symmetry that DAMAVEr will accept. Defaults to 'P1'.
- **atsas_dir** (*str, optional*) – The directory of the atsas programs (the bin directory). If not provided, the API uses the auto-detected directory.
- **abort_event** (`threading.Event`, optional) – A `threading.Event` or `multiprocessing.Event`. If this event is set it will abort the damaver run.

Returns

- **mean_nsd** (*float*) – The mean NSD of the models.
- **stdev_nsd** (*float*) – The standard deviation of the NSD of the models.
- **rep_model** (*str*) – The name of the representative model determined by DAMAVER.
- **result_dict** (*dict*) – A dictionary of the model specific results. The keys are the input filenames. The values are lists of the form ['Include' mean_model_nsd] where 'Include' indicates the model was in the average whereas a different value indicates the model was excluded from the average.
- **res** (*float*) – The resolution of the reconstructions. Only available if more than 3 models were averaged.
- **res_err** (*float*) – The uncertainty in the resolution.
- **res_unit** (*str*) – The unit of the resolution.

`bioxtasraw.RAWAPI.`**`damclust`**(*files*, *prefix*, *datadir*, *symmetry='P1'*, *atsas_dir=None*, *abort_event=<threading.Event object>*)

Runs DAMCLUST from the ATSAS package on a set of files. Requires a separate installation of the ATSAS package. Function blocks until DAMCLUST finishes.

Parameters

- **`files`** (*`list`*) – A list of strings of the filenames on disk that are the DAMAVER inputs. Must be just filenames, no paths, and all files must be in the same directory.
- **`prefix`** (*`str`*) – The prefix to be appended to the DAMAVER output files.
- **`datadir`** (*`str`*) – The data directory in which all of the input files are located. Also the location of the DAMAVER output.
- **`symmetry`** (*`str, optional`*) – The symmetry that DAMAVER will use during alignment. Accepts any symmetry that DAMAVEr will accept. Defaults to 'P1'.
- **`atsas_dir`** (*`str, optional`*) – The directory of the atsas programs (the bin directory). If not provided, the API uses the auto-detected directory.
- **`abort_event`** (`threading.Event`, optional) – A `threading.Event` or `multiprocessing.Event`. If this event is set it will abort the damclust run.

Returns

- **cluster_list** (*list*) – A list of `collections.namedtuple` items. Each list item has entries: 'num', 'rep_model', and 'dev', where num is the number of models in the cluster, rep_model is the representative model of the cluster, and dev is the deviation within the cluster.
- **distance_list** (*list*) – A list of `collections.namedtuple` items. Each list item has entries 'cluster1', 'cluster2', and 'cluster_dist', where cluster1 is the first cluster, cluster2 is the second cluster, and cluster_dist is the distance between cluster1 and cluster2. If there is only one cluster the distance list will be empty.

`bioxtasraw.RAWAPI.`**`dammif`**(*ift*, *prefix*, *datadir*, *mode='Slow'*, *symmetry='P1'*, *anisometry='Unknown'*, *write_ift=True*, *ift_name=None*, *atsas_dir=None*, *settings=None*, *unit='Unknown'*, *omit_solvent=True*, *chained=False*, *expected_shape='u'*, *random_seed="*, *constant="*, *max_bead_count=-1*, *dam_radius=-1*, *harmonics=-1*, *prop_to_fit=-1*, *curve_weight='e'*, *max_steps=-1*, *max_iters=-1*, *max_success=-1*, *min_success=-1*, *T_factor=-1*, *rg_penalty=-1*, *center_penalty=-1*, *loose_penalty=-1*, *abort_event=<threading.Event object>*)

Creates a bead model (dummy atom) reconstruction using DAMMIF from the ATSAS package. Requires a separate installation of the ATSAS package. Function blocks until DAMMIF finishes.

Parameters

- **`ift`** (*`bioxtasraw.SASM.IFTM`*) – The GNOM IFT to be used as DAMMIF input. If write_ift is False, an IFT already on disk is used and this parameter can be `None`.
- **`prefix`** (*`str`*) – The output prefix for the DAMMIF model.
- **`datadir`** (*`str`*) – The output directory for the DAMMIF model. If using an IFT on disk, then the IFT must be in this directory.
- **`mode`** (*`{'Fast', 'Slow' 'Custom'} str, optional`*) – The DAMMIF mode. Note that most of the advanced settings require that DAMMIF be in 'Custom' mode to use. Defaults to slow.
- **`symmetry`** (*`str, optional`*) – The symmetry applied to the reconstruction. Accepts any symmetry known to DAMMIF. Defaults to P1.

- **anisometry** (*{'Unknown', 'Prolate', 'Oblate'} str, optional*) – The anisometry applied to the reconstruction. Defaults to Unknown.
- **write_ift** (*bool, optional*) – If True, the input IFT is written to disk. If False, an IFT already on disk used, as defined by ift_name (directory must be datadir).
- **ift_name** (*str, optional*) – The IFT name on disk. Used if write_ift is False.
- **atsas_dir** (*str, optional*) – The directory of the atsas programs (the bin directory). If not provided, the API uses the auto-detected directory.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – RAW settings containing relevant parameters. If provided, every model parameter except mode, symmetry, and anisometry is overridden with the value in the settings file. Default is None.
- **unit** (*{'Unknown', 'Angstrom', 'Nanometer'} str, optional*) – The unit of the P(r) function. Defaults to 'Unknown'.
- **omit_solvent** (*bool, optional*) – Whether the solvent file (-0.pdb) should be omitted. Defaults to True.
- **chained** (*bool, optional*) – Whether the beads should be connected in pseudo-chains. Defaults to False.
- **expected_shape** (*{'u', 'c', 'e', 'f', 'r', 'h', 'hs', 'rc'} str, optional*) – Expected shape of the reconstruction: (u)nknown, (c)ompact, (e)xtended, (f)lat, (r)ing, (h) compact-hollow, (hs) hollow-sphere, (rc) random-chain. Default is unknown.
- **random_seed** (*str, optional*) – Random seed for the reconstruction. Default is to let DAMMIF generate the seed.
- **constant** (*str, optional*) – Constant offset for reconstruction. Default is to let DAMMIF determine the offset.
- **max_bead_count** (*int, optional*) – Maximum bead count for the model. Default is to let DAMMIF determine the parameter value.
- **dam_radius** (*float, optional*) – Dummy atom radius of the reconstruction, in Angstrom (>=1.0). Default is to let DAMMIF determine the parameter value.
- **harmonics** (*int, optional*) – Number of spherical harmonics to use in the reconstruction. Default is the DAMMIF default.
- **prop_to_fit** (*float, optional*) – Proportion of the curve to fit for the reconstruction. Default is the DAMMIF default.
- **curve_weight** (*{'l', 'p', 'e', 'n'} str, optional*) – Curve weighting function, [l]log, [p]orod, [e]mphasized porod, or [n]one. Default is 'e'.
- **max_steps** (*int, optional*) – Maximum number of steps in the annealing procedure. Default is the DAMMIF default.
- **max_iters** (*int, optional*) – Maximum number of iterations within a single temperature step. Default is the DAMMIF default.
- **max_success** (*int, optional*) – Maximum number of successes in a temperature step. Default is the DAMMIF default.
- **min_success** (*int, optional*) – Minimum number of successes in a temperature step. Default is the DAMMIF default.
- **T_factor** (*float, optional*) – Temperature schedule factor. Default is the DAMMIF default.

- **rg_penalty** (*float, optional*) – Rg penalty weight. Default is the DAMMIF default.
- **center_penalty** (*float, optional*) – Center penalty weight. Default is the DAMMIF default.
- **loose_penalty** (*float, optional*) – Looseness penalty weight. Default is the DAMMIF default.
- **abort_event** (threading.Event, optional) – A `threading.Event` or `multiprocessing.Event`. If this event is set it will abort the dammin run.

Returns

- **chi_sq** (*float*) – The chi squared of the model's scattering profile to the data.
- **rg** (*float*) – The Rg of the model.
- **dmax** (*float*) – The Dmax of the model.
- **mw** (*float*) – The estimated molecular weight of the model, based on the excluded volume.
- **excluded_volume** (*float*) – The excluded volume of the model.

`bioxtasraw.RAWAPI.`**`dammin`**(*ift*, *prefix*, *datadir*, *mode='Slow'*, *symmetry='P1'*, *anisometry='Unknown'*, *initial_dam=None*, *write_ift=True*, *ift_name=None*, *atsas_dir=None*, *settings=None*, *unit='Unknown'*, *constant=0*, *dam_radius=-1*, *harmonics=-1*, *prop_to_fit=-1*, *curve_weight='1'*, *max_steps=-1*, *max_iters=-1*, *max_success=-1*, *min_success=-1*, *T_factor=-1*, *loose_penalty=-1*, *knots=20*, *sphere_diam=-1*, *coord_sphere=-1*, *disconnect_penalty=-1*, *periph_penalty=1*, *abort_event=<threading.Event object>*)

Creates a bead model (dummy atom) reconstruction using DAMMIN from the ATSAS package. Requires a separate installation of the ATSAS package. Function blocks until DAMMIN finishes. Can be used to refine damstart.pdb files.

Parameters

- **ift** (*bioxtasraw.SASM.IFTM*) – The GNOM IFT to be used as DAMMIN input. If write_ift is False, an IFT already on disk is used and this parameter can be `None`.
- **prefix** (*str*) – The output prefix for the DAMMIN model.
- **datadir** (*str*) – The output directory for the DAMMIN model. If using an IFT on disk, then the IFT must be in this directory.
- **mode** (*{'Fast', 'Slow' 'Custom', 'Refine'} str, optional*) – The DAMMIN mode. Note that most of the advanced settings require that DAMMIN be in 'Custom' mode to use. Defaults to slow. If using 'Refine' mode then initial_dam must be specified.
- **symmetry** (*str, optional*) – The symmetry applied to the reconstruction. Accepts any symmetry known to DAMMIN. Defaults to P1.
- **anisometry** (*{'Unknown', 'Prolate', 'Oblate'} str, optional*) – The anisometry applied to the reconstruction. Defaults to Unknown.
- **initial_dam** (*str, optional*) – Name of the input model file for refinement. Must be in datadir.
- **write_ift** (*bool, optional*) – If True, the input IFT is written to disk. If False, an IFT already on disk used, as defined by ift_name (directory must be datadir).
- **ift_name** (*str, optional*) – The IFT name on disk. Used if write_ift is False.

- **atsas_dir** (*str, optional*) – The directory of the atsas programs (the bin directory). If not provided, the API uses the auto-detected directory.
- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, every model parameter except mode, symmetry, and anisometry is overridden with the value in the settings file. Default is None.
- **unit** (*{'Unknown', 'Angstrom', 'Nanometer'} str, optional*) – The unit of the P(r) function. Defaults to 'Unknown'.
- **constant** (*str, optional*) – Constant offset for reconstruction. Default is to let DAMMIN determine the offset.
- **dam_radius** (*float, optional*) – Packing radius of the dummy atoms. Default is to let DAMMIN determine the parameter value.
- **harmonics** (*int, optional*) – Number of spherical harmonics to use in the reconstruction. Default is the DAMMIN default.
- **prop_to_fit** (*float, optional*) – Proportion of the curve to fit for the reconstruction. Default is the DAMMIN default.
- **curve_weight** (*{'0', '1', '2'} str, optional*) – Curve weighting function. 0 - Porod weighting. 1 - Porod weighting with emphasis of initial points (default), 2 - logarithmic weighting.
- **max_steps** (*int, optional*) – Maximum number of steps in the annealing procedure. Default is the DAMMIN default.
- **max_iters** (*int, optional*) – Maximum number of iterations within a single temperature step. Default is the DAMMIN default.
- **max_success** (*int, optional*) – Maximum number of successes in a temperature step. Default is the DAMMIN default.
- **min_success** (*int, optional*) – Minimum number of successes in a temperature step. Default is the DAMMIN default.
- **T_factor** (*float, optional*) – Temperature schedule factor. Default is the DAMMIN default.
- **loose_penalty** (*float, optional*) – Looseness penalty weight. Default is the DAMMIN default.
- **knots** (*int, optional*) – Number of knots in curve to fit. Default is the DAMMIN default.
- **sphere_diam** (*float, optional*) – Sphere diameter in Angstrom. Default is the DAMMIN default.
- **coord_sphere** (*float, optional*) – Radius of the first coordination sphere. Default is the DAMMIN default.
- **disconnect_penalty** (*float, optional*) – Disconnectivity penalty weight. Default is DAMMIN default.
- **periph_penalty** (*float, optional*) – Peripheral penalty weight. Default is DAMMIN default.
- **abort_event** (`threading.Event`, optional) – A `threading.Event` or `multiprocessing.Event`. If this event is set it will abort the dammif run.

Returns

- **chi_sq** (*float*) – The chi squared of the model's scattering profile to the data.
- **rg** (*float*) – The Rg of the model.
- **dmax** (*float*) – The Dmax of the model.
- **mw** (*float*) – The estimated molecular weight of the model, based on the excluded volume.
- **excluded_volume** (*float*) – The excluded volume of the model.

`bioxtasraw.RAWAPI.`**`datgnom`**(*profile*, *rg=None*, *idx_min=None*, *idx_max=None*, *atsas_dir=None*, *use_rg_from='guinier'*, *use_guinier_start=True*, *cut_8rg=False*, *write_profile=True*, *datadir=None*, *filename=None*, *save_ift=False*, *savename=None*)

Calculates the IFT and resulting P(r) function using datgnom from the ATSAS package to automatically find the Dmax value. This requires a separate installation of the ATSAS package to use. The input profile needs to have a calculated Rg value, either from a Guinier fit or from a IFT P(r) function, so the Rg value is known. If datgnom fails, values of None, -1, or '' are returned.

Parameters

- **`profile`** (*`bioxtasraw.SASM.SASM`*) – The profile to calculate the IFT for. If using write_file false, you can pass None here. In that case you must pass a value for rg.
- **`rg`** (*`float, optional`*) – The Rg to be used in calculating the IFT If not provided, then the Rg is taken from the analysis dictionary of the profile, in conjunction with the use_rg_from setting.
- **`idx_min`** (*`int, optional`*) – The index of the q vector that corresponds to the minimum q point to be used in the IFT. Defaults to the first point in the q vector. Overrides use_guinier_start.
- **`idx_max`** (*`int, optional`*) – The index of the q vector that corresponds to the maximum q point to be used in the IFT. Defaults to the last point in the q vector. If write_profile is false and no profile is provided then this cannot be set, and datgnom will truncate to 8/Rg automatically. Overrides cut_8rg.
- **`atsas_dir`** (*`str, optional`*) – The directory of the atsas programs (the bin directory). If not provided, the API uses the auto-detected directory.
- **`use_rg_from`** (*`{'guinier', 'gnom', 'bift'} str, optional`*) – Determines whether the Rg value used for the IFT calculation is from the Guinier fit, or the GNOM or BIFT P(r) function. Ignored if the rg parameter is provided.
- **`use_guiner_start`** (*`bool, optional`*) – If set to True, and no idx_min is provided, if a Guinier fit has been done for the input profile, the start point of the Guinier fit is used as the start point for the IFT. Ignored if there is no input profile.
- **`cut_8rg`** (*`bool, optional`*) – If set to True and no idx_max is provided, then the profile is automatically truncated at q=8/Rg.
- **`write_profile`** (*`bool, optional`*) – If True, the input profile is written to file. If False, then the input profile is ignored, and the profile specified by datadir and filename is used. This is convenient if you are trying to process a lot of files that are already on disk, as it saves having to read in each file and then save them again. Defaults to True. If False, you must provide a value for the rg parameter.
- **`datadir`** (*`str, optional`*) – If write_file is False, this is used as the path to the scattering profile on disk.
- **`filename`** (*`str, optional.`*) – If write_file is False, this is used as the filename of the scattering profile on disk.

- **save_ift** (*bool, optional*) – If True, the IFT from datgnom (.out file) is saved on disk. Requires specification of datadir and savename parameters.
- **savename** (*str, optional*) – If save_ift is True, this is used as the filename of the .out file on disk. This should just be the filename, no path. The datadir parameter is used as the parth.

Returns

- **ift** (*bioxtasraw.SASM.IFTM*) – The IFT calculated by BIFT from the input profile.
- **dmax** (*float*) – The maximum dimension of the P(r) function found by BIFT.
- **rg** (*float*) – The real space radius of gyration (Rg) from the P(r) function.
- **i0** (*float*) – The real space scattering at zero angle (I(0)) from the P(r) function.
- **rg_err** (*float*) – The uncertainty in the real space radius of gyration (Rg) from the P(r) function.
- **i0_err** (*float*) – The uncertainty in the real space scattering at zero angle (I(0)) from the P(r) function.
- **total_est** (*float*) – The GNOM total estimate.
- **chi_sq** (*float*) – The chi squared value of the fit of the scattering profile calculated from the P(r) function to the input scattering profile.
- **alpha** (*float*) – The alpha value determined by datgnom.
- **quality** (*str*) – The GNOM qualitative interpretation of the total estimate.

bioxtasraw.RAWAPI.**denss**(*ift, prefix, datadir, mode='Slow', symmetry=0, sym_axis='X', sym_type='Cyclical', initial_model=None, n_electrons=None, settings=None, voxel=5, oversampling=3, steps=10000, recenter=True, recenter_step=[1001, 1501, 2001, 2501, 3001, 3501, 4001, 4501, 5001, 5501, 6001, 6501, 7001, 7501, 8001], recenter_mode='com', positivity=True, extrapolate=True, shrinkwrap=True, sw_sigma_start=3.0, sw_sigma_end=1.5, sw_sigma_decay=0.99, sw_sigma_thresh=0.2, sw_iter=20, sw_min_step=5000, connected=True, connectivity_step=[7500], chi_end_frac=0.001, cut_output=False, write_xplor=False, sym_step=[3000, 5000, 7000, 9000], seed=None, abort_event=<threading.Event object>, gpu=False*)

Generates an electron density reconstruction using DENSS. Function blocks until DENSS finishes. Can be used to refine an existing model.

Parameters

- **ift** (*bioxtasraw.SASM.IFTM*) – The IFT to be used as DENSS input.
- **prefix** (*str*) – The output prefix for the DENSS model.
- **datadir** (*str*) – The output directory for the DENSS model.
- **mode** (*{'Fast', 'Slow' 'Custom'} str, optional*) – The DENSS mode. Note that some of the advanced settings require that DENSS be in 'Custom' mode to use. Defaults to slow.
- **symmetry** (*int, optional*) – Rotational symmetry applied as n-fold symmetry about the sym_axis. Default is 0, i.e. no symmetry.
- **sym_axis** (*{'X', 'Y', 'Z'} str, optional*) – The symmetry axis used if a symmetry is specified. Correspond to the xyz principal axes.

- **sym_type** (*{'Cyclical', 'Dihedral'}*) – The symmetry type to use, either cyclical or dihedral.
- **initial_model** (class:*numpy.array*, optional) – Initial electron density model as a numpy array. If input is provided, then the model will be refined.
- **n_electrons** (*int, optional*) – Number of electrons in the molecule. If provided, the output density will be scaled so that the sum of the density across the occupied volume is equal to this value.
- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, every model parameter except mode, symmetry, and sym_axis, and n_electrons is overridden with the value in the settings file. Default is None.
- **voxel** (*float, optional*) – The voxel size for the model. Only used in Custom mode.
- **oversampling** (*int, optional*) – The sampling ratio.
- **steps** (*int, optional*) – Maximum number of iterations of the denss algorithm. Only used in Custom mode.
- **recenter** (*bool, optional*) – Whether the particle should be recentered at the origin during reconstruction. Default is True.
- **recenter_step** (*list, optional*) – A list of integers specifying the steps at which recentering should be carried out. Only used in custom mode.
- **recenter_mode** (*{'com', 'max'} str, optional*) – Recenter based on the center of mass (com) or maximum density (max). Default is com.
- **positivity** (*bool, optional*) – Enforces positive density. Only used in Custom mode. Default is True. Set to False in Membrane mode.
- **extrapolate** (*bool, optional*) – Whether to extrapolate the measured scattering profile based on the voxel size. Default is True.
- **shrinkwrap** (*bool, optional*) – Whether to apply the shrinkwrap algorithm to determine the underlying support. Default is True. Not recommended to change.
- **sw_sigma_start** (*float, optional*) – The starting value to use for blurring during the shrinkwrap algorithm. Default is 3.0.
- **sw_sigma_end** (*float, optional*) – The ending value to use for blurring during the shrinkwrap algorithm. Default is 1.5.
- **sw_sigma_decay** (*float, optional*) – How quickly the sw_sigma value transitions from start to end.
- **sw_sigma_thresh** (*float, optional*) – The minimum threshold for inclusion of a voxel in the support during the shrinkwrap algorithm. Membrane mode sets this to 0.1.
- **sw_iter** (*int, optional*) – How often the shrinkwrap algorithm is applied. Not recommended to change.
- **sw_min_step** (*int, optional*) – The first step at which the shrinkwrap algorithm is applied. Only used in Custom mode.
- **connected** (*bool, optional*) – Whether connectivity is enforced for the reconstruction. Default is True.
- **connectivity_step** (*list, optional*) – A list of integers specifying the steps at which connectivity is enforced. Only used in Custom model.

- **`chi_end_frac`** (*`float, optional`*) – The convergence criteria. Set as the minimum threshold of the chi squared standard deviation in the last 100 steps, as a fraction of the median chi squared in those steps.
- **`cut_output`** (*`bool, optional`*) – Whether to remove unused parts of the search space when writing the output electron density files. Default is False.
- **`write_xplor`** (*`bool, optional`*) – Whether to write the output density as xplor files and mrc files, or just mrc files. Default is False.
- **`sym_step`** (*`list, optional`*) – A list of integers specifying the steps at which the symmetry constraint should be applied.
- **`seed`** (*`int, optional`*) – The random seed to be used for the DENSS reconstruction. If None (default) than a new seed is generated.
- **`abort_event`** (`threading.Event`, optional) – A `threading.Event` or `multiprocessing.Event`. If this event is set it will abort the denss run.
- **`gpu`** (*`bool, optional`*) – Whether to use GPU computing for DENSS. CuPy must be installed.

Returns

- **rho** (`numpy.array`) – The calculated electron density of the model.
- **chi_sq** (*float*) – The chi squared value of the model fit to the data.
- **rg** (*float*) – The radius of gyration of the model.
- **support_vol** (*float*) – The support volume of the mode.
- **side** (*float*) – The real space box width in Angstroms of the reconstruction.
- **q_fit** (`numpy.array`) – The q values, including any extrapolation, of the model fit to the data.
- **I_fit** (`numpy.array`) – The I values, including any extrapolation, of the model fit to the data.
- **I_extrap** (`numpy.array`) – The experimental intensity, including any extrapolation, used in the reconstruction.
- **err_extrap** (`numpy.array`) – The experimental uncertainty, including any extrapolation, used in the reconstruction.
- **all_chi_sq** (`numpy.array`) – The value of chi squared at all iterations of the DENSS algorithm. Useful to check convergence.
- **all_rg** (`numpy.array`) – The value of rg at all iterations of the DENSS algorithm. Useful to check convergence.
- **all_support_vol** (`numpy.array`) – The value of support volume at all iterations of the DENSS algorithm. Useful to check convergence.

`bioxtasraw.RAWAPI.`**`denss_align`**(*density*, *side*, *ref_file*, *ref_datadir='.'*, *prefix=''*, *save_datadir='.'*, *save=True*, *center=True*, *resolution=15.0*, *enantiomer=True*, *n_proc=1*, *abort_event=None*)

Aligns each input electron density against a reference model. The reference model can either be a PDB model (.pdb) or electron density (.mrc). The aligned model can be saved to disk.

Parameters

- **`density`** (`numpy.array`) – A `numpy.array` of the electron density to be aligned to the reference model.

- **side** (*float*) – The real space box width in Angstroms of the reconstruction.
- **ref_file** (*str*) – The name (without path) of the reference model file to align the input densities to. Should be either a .pdb or .mrc file.
- **ref_datadir** (*str, optional*) – The directory where ref_file is located. Defaults to the current directory.
- **prefix** (*str*) – The name the aligned model will be saved with (minus extension) if save is True.
- **save_datadir** (*str*) – The directory to save the aligned model in if save is True.
- **save** (*bool, optional*) – Whether or not to save the aligned model to disk.
- **center** (*bool, optional*) – Whether the reference file should first be centered on the origin. Defaults to True. Only used if the reference file is a .pdb file. If used, a _centered.pdb file is written to the same folder as the ref_datadir.
- **resolution** (*float, optional*) – If a .pdb file is the reference file, this is the resolution used to generate an electron density map for comparison with the input densities.
- **enantiomer** (*bool, optional*) – Check for enantiomers during alignment.
- **n_proc** (*int, optional*) – The number of processors to use. This could be up to as many cores as your computer has.
- **abort_event** (`multiprocessing.Manager.Event`, optional) – A `multiprocessing.Manager.Event` If this event is set it will abort the denss alignment run.

Returns

- **aligned_density** (`numpy.array`) – The input electron density aligned to the reference model.
- **scores** (*float*) – The correlation score of the model to the reference model.

`bioxtasraw.RAWAPI.`**`denss_average`**(*densities*, *side*, *prefix*, *datadir*, *n_proc=1*, *abort_event=None*)

Averages multiple electron densities to produce a single average density. Uses the averaging procedure from the DENSS package. Function blocks until the average is complete. There must be at least four densities to average.

Parameters

- **densities** (`numpy.array`) – A `numpy.array` where the the first axis (e.g. density[0], density[1], etc) corresponds to each electron density to be averaged.
- **side** (*float*) – The real space box width in Angstroms of the reconstruction.
- **prefix** (*str*) – The output prefix for the averaged model.
- **datadir** (*str*) – The output directory for the averaged model.
- **n_proc** (*int*) – The number of processors to use. This could be up to as many cores as your computer has.
- **abort_event** (`multiprocessing.Manager.Event`, optional) – A `multiprocessing.ManagerEvent` If this event is set it will abort the denss average run.

Returns

- **average_rho** (`numpy.array`) – The average electron density. Of the input models, after rejecting the outliers.
- **mean_cor** (*float*) – The mean correlation score of the models.

- **std_cor** (*float*) – The standard deviation of the model correlation scores.
- **threshold** (*float*) – The threshold used to reject models from the average.
- **res** (*float*) – The estimated model resolution in Angstrom from the Fourier shell correlation.
- **scores** (`numpy.array`) – An array of the correlation scores. Each entry in the array is the score for the corresponding entry in the input densities array.
- **fsc** (`numpy.array`) – The average Fourier shell correlation between each model and a average reference model. This is used to estimate the resolution.

`bioxtasraw.RAWAPI.`**`efa`**(*series*, *ranges*, *profile_type='sub'*, *framei=None*, *framef=None*, *method='Hybrid'*, *niter=1000*, *tol=1e-12*, *norm=True*, *force_positive=None*, *previous_results=None*)

Runs evolving factor analysis (EFA) on the input series to deconvolve overlapping elution peaks in the data.

Parameters

- **`series`** (list or *`bioxtasraw.SECM.SECM`*) – The input series to be deconvolved. It should either be a list of individual scattering profiles (*`bioxtasraw.SASM.SASM`*) or a single series object (*`bioxtasraw.SECM.SECM`*).
- **`ranges`** (*`iterable`*) – A list, `numpy.array`, or other iterable of the EFA ranges, which each item is the range of a given component, and contains two integers, the first is start of that component range the second the end of the component range. Must be a type that can be cast as a numpy array. Should be relative to the full range of the series, even if framei and framef are provided.
- **`profile_type`** (*`{'unsub', 'sub', 'baseline'} str, optional`*) – Only used if a *`bioxtasraw.SECM.SECM`* is provided for the series argument. Determines which type of profile to use from the series for the EFA. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'.
- **`framei`** (*`int, optional`*) – The initial frame in the series to use for EFA. If not provided, it defaults to the first frame in the series.
- **`framef`** (*`int, optional`*) – The final frame in the series to use for EFA. If not provided, it defaults to the last frame in the series.
- **`method`** (*`{'Hybrid', 'Iterative', 'Explicit'} str, optional`*) – Sets the method used for the EFA rotation step as either 'Hybrid', 'Iterative', or 'Explicit'.
- **`niter`** (*`int, optional`*) – The maximum number of iterations to use for the rotation in either hybrid or iterative mode. Defaults to 1000. Can be increased if EFA fails to converge.
- **`tol`** (*`float, optional`*) – The tolerance used as the convergence criteria for the EFA rotation in hybrid or iterative mode. Defaults to 1e-12. Can be decreased if EFA fails to converge.
- **`norm`** (*`bool, optional`*) – Whether error normalized intensity should be used for EFA. Defaults to True. Recommended to not change this.
- **`force_positive`** (*`list, optional`*) – Whether EFA ranges should be constrained to force positive. By default all ranges are forced positive. If provided, it should be a list of bool values with a length equal to the number of EFA ranges. Each list item determines whether to force positive the range defined by the same index in the ranges parameter.
- **`previous_results`** (*`list, optional`*) – This list contains the previous results of EFA, if any. This is used to improve the starting guess. The first entry in the list is a boolean corresponding to whether the previous results converged, and the second is a dictionary of the previous results, corresponding to the rotation_data dictionary returned by this function. Defaults to None, which should be used if no previous results are available.

Returns

- **efa_profiles** (*list*) – A list of profiles (`bioxtasraw.SASM.SASM`) determined by EFA. If EFA converged, the profile at each list index was determined for the corresponding range in the ranges input parameter. If EFA failed to converge an empty list is returned.
- **converged** (*bool*) – True if EFA converged, otherwise False.
- **conv_data** (*dict*) – A dictionary containing information about the convergence. Keys are 'steps' - A list containing the value of the convergence criteria at each iteration, if available; 'iterations' - The number of iterations carried out during the rotation, if available; 'final_step' - The value of the convergence criteria in the final iteration step, if available; 'options' - A dictionary containing the input convergence options; 'failed' - A bool indicating if convergence failed.
- **rotation_data** (*dict*) – A dictionary containing the rotation results, if available. If the rotation failed to converge, the dictionary is empty. Keys are: 'C' - Concentration data. A numpy array where the first axis is concentration as a function of frame and the second axis is component number. 'chisq' - Mean error weighted chi squared data. A numpy array of chi squared vs. frame number. 'int' - Scattering intensity. A numpy array where the first axis is intensity and the second axis is component number. ' err' - Scattering intensity uncertainty. A numpy array where the first axis is intensity and the second axis is component number. 'M' - The EFA M rotation matrix.

Raises `SASExceptions.EFAError` – If initial SVD cannot be carried out.

`bioxtasraw.RAWAPI.`**`find_baseline_range`**(*series*, *baseline_type='Integral'*, *profile_type='sub'*, *window_size=5*, *int_type='total'*, *q_val=None*, *q_range=None*, *settings=None*, *sim_test='CorMap'*, *sim_cor='Bonferroni'*, *sim_thresh=0.01*)

Automatically determine an appropriate range for the baseline correction. Currently only works for integral baseline corrections.

Parameters

- **`series`** (list or `bioxtasraw.SECM.SECM`) – The input series to find the baseline range for. It should either be a list of individual scattering profiles (`bioxtasraw.SASM.SASM`) or a single series object (`bioxtasraw.SECM.SECM`).
- **`baseline_type`** (*`{'Integral', 'Linear'} str, optional`*) – Defines the baseline type for validation purposes.
- **`profile_type`** (*`{'unsub', 'sub', 'baseline'} str, optional`*) – Only used if a `bioxtasraw.SECM.SECM` is provided for the series argument. Determines which type of profile to use from the series to validate the baseline range. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'.
- **`window_size`** (*`int, optional`*) – The size of the average window used for calculating parameters from series data. Used to help set the size of the search window for the buffer region. Defaults to 5.
- **`int_type`** (*`{'total', 'mean', 'q_val', 'q_range'} str, optional`*) – The intensity type to use for the validation of the baseline range. Total integrated intensity - 'total', mean intensity - 'mean', intensity at a particular q value - 'q_val', intensity in a given q range - 'q_range'. Use of q_val or q_range requires the corresponding parameter to be provided.
- **`q_val`** (*`float, optional`*) – If int_type is 'q_val', the q value used for the intensity is set by this parameter.

- **q_range** (*list, optional*) – This should have two entries, both floats. The first is the minimum q value of the range, the second the maximum q value of the range. If int_type is 'q_range', the q range used for the intensity is set by this parameter.
- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, sim_test, sim_cor, and sim_threshold are overridden by the values in the settings.
- **sim_test** (*{'CorMap'} str, optional*) – Sets the type of similarity test to be used. Currently only CorMap is supported as an option. Is overridden if settings are provided.
- **sim_cor** (*{'Bonferroni', 'None'} str, optional*) – Sets the multiple testing correction to be used as part of the similarity test. Default is Bonferroni. Is overridden if settings are provided.
- **sim_thresh** (*float, optional*) – Sets the p value threshold for the similarity test. A higher value is a more strict test (range from 0-1). Is overridden if settings are provided.

Returns

- **start_found** (*bool*) – True if a start region was successfully found.
- **end_found** (*bool*) – True if an end region was successfully found.
- **start_range** (*tuple*) – A tuple where the first entry is the start of the start region and the second entry is the end of the start region. Returns -1 for each value if not start_found.
- **end_range** (*tuple*) – A tuple where the first entry is the start of the end region and the second entry is the end of the end region. Returns -1 for each value if not end_found.

`bioxtasraw.RAWAPI.`**`find_buffer_range`**(*series*, *profile_type='unsub'*, *int_type='total'*, *q_val=None*, *q_range=None*, *window_size=5*, *settings=None*, *sim_test='CorMap'*, *sim_cor='Bonferroni'*, *sim_thresh=0.01*)

Automatically determine the appropriate buffer range from subtraction from the input series. This is designed to work with SEC-SAXS data, but may work in other circumstances.

Parameters

- **series** (list or *bioxtasraw.SECM.SECM*) – The input series to find the buffer range for. It should either be a list of individual scattering profiles (*bioxtasraw.SASM.SASM*) or a single series object (*bioxtasraw.SECM.SECM*).
- **profile_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – Only used if a *bioxtasraw.SECM.SECM* is provided for the series argument. Determines which type of profile to use from the series to find the buffer range. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'.
- **int_type** (*{'total', 'mean', 'q_val', 'q_range'} str, optional*) – The intensity type to use for the automated determination of buffer range. Total integrated intensity - 'total', mean intensity - 'mean', intensity at a particular q value - 'q_val', intensity in a given q range - 'q_range'. Use of q_val or q_range requires the corresponding parameter to be provided.
- **q_val** (*float, optional*) – If int_type is 'q_val', the q value used for the intensity is set by this parameter.
- **q_range** (*list, optional*) – This should have two entries, both floats. The first is the minimum q value of the range, the second the maximum q value of the range. If int_type is 'q_range', the q range used for the intensity is set by this parameter.

- **window_size** (*int, optional*) – The size of the average window used for calculating parameters from series data. Used to help set the size of the search window for the buffer region. Defaults to 5.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – RAW settings containing relevant parameters. If provided, sim_test, sim_cor, and sim_threshold are overridden by the values in the settings.
- **sim_test** (*{'CorMap'} str, optional*) – Sets the type of similarity test to be used. Currently only CorMap is supported as an option. Is overridden if settings are provided.
- **sim_cor** (*{'Bonferroni', 'None'} str, optional*) – Sets the multiple testing correction to be used as part of the similarity test. Default is Bonferroni. Is overridden if settings are provided.
- **sim_thresh** (*float, optional*) – Sets the p value threshold for the similarity test. A higher value is a more strict test (range from 0-1). Is overridden if settings are provided.

Returns

- **success** (*bool*) – If a buffer range was successfully found.
- **region_start** (*int*) – The starting index of the buffer region found.
- **region_end** (*int*) – The ending index of the buffer region found.

bioxtasraw.RAWAPI.**find_sample_range**(*series, profile_type='sub', window_size=5, int_type='total', q_val=None, q_range=None, rg=None, vcmw=None, vpmw=None, settings=None, sim_test='CorMap', sim_cor='Bonferroni', sim_thresh=0.01*)

Automatically determine the appropriate sample range to average from the input series. This is designed to work with SEC-SAXS data, but may work in other circumstances.

Parameters

- **series** (list or *bioxtasraw.SECM.SECM*) – The input series to find the sample range for. It should either be a list of individual scattering profiles (*bioxtasraw.SASM.SASM*) or a single series object (*bioxtasraw.SECM.SECM*).
- **profile_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – Only used if a *bioxtasraw.SECM.SECM* is provided for the series argument. Determines which type of profile to use from the series to find the sample range. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'.
- **window_size** (*int*) – The size of the average window used when calculating Rg and MW.
- **int_type** (*{'total', 'mean', 'q_val', 'q_range'} str, optional*) – The intensity type to used when finding the sample range. Total integrated intensity - 'total', mean intensity - 'mean', intensity at a particular q value - 'q_val', intensity in a given q range - 'q_range'. Use of q_val or q_range requires the corresponding parameter to be provided.
- **q_val** (*float, optional*) – If int_type is 'q_val', the q value used for the intensity is set by this parameter.
- **q_range** (*list, optional*) – This should have two entries, both floats. The first is the minimum q value of the range, the second the maximum q value of the range. If int_type is 'q_range', the q range used for the intensity is set by this parameter.

- **rg** (*list*) – A list of the Rg values corresponding to the input series data. Only required if inputing a list of profiles rather than a series object.
- **vcmw** (*list*) – A list of the volume of correlation M.W. values corresponding to the input series data. Only required if inputing a list of profiles rather than a series object.
- **vpmw** (*list*) – A list of the Porod volume M.W. values corresponding to the input series data. Only required if inputing a list of profiles rather than a series object.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – RAW settings containing relevant parameters. If provided, sim_test, sim_cor, and sim_threshold are overridden by the values in the settings.
- **sim_test** (*{'CorMap'} str, optional*) – Sets the type of similarity test to be used. Currently only CorMap is supported as an option. Is overridden if settings are provided.
- **sim_cor** (*{'Bonferroni', 'None'} str, optional*) – Sets the multiple testing correction to be used as part of the similarity test. Default is Bonferroni. Is overridden if settings are provided.
- **sim_thresh** (*float, optional*) – Sets the p value threshold for the similarity test. A higher value is a more strict test (range from 0-1). Is overridden if settings are provided.

Returns

- **success** (*bool*) – If a buffer range was successfully found.
- **region_start** (*int*) – The starting index of the sample region found.
- **region_end** (*int*) – The ending index of the sample region found.

bioxtasraw.RAWAPI.**gnom**(*profile*, *dmax*, *rg=None*, *idx_min=None*, *idx_max=None*, *dmax_zero=True*, *alpha=0*, *atsas_dir=None*, *use_rg_from='guinier'*, *use_guinier_start=True*, *cut_dam=False*, *write_profile=True*, *datadir=None*, *filename=None*, *save_ift=False*, *savename=None*, *settings=None*, *dmin_zero=True*, *npts=0*, *system=0*, *radius56=-1*, *rmin=-1*)

Calculates the IFT and resulting P(r) function using gnom from the ATSAS package. This requires a separate installation of the ATSAS package to use. If gnom fails, values of None, -1, or '' are returned.

Parameters

- **profile** (*bioxtasraw.SASM.SASM*) – The profile to calculate the IFT for. If using write_file false, you can pass None here.
- **dmax** (*float*) – The Dmax to be used in calculating the IFT.
- **rg** (*float, optional*) – The Rg to be used in calculating the 8/rg cutoff, if cut_8/rg is True. If not provided, then the Rg is taken from the analysis dictionary of the profile, in conjunction with the use_rg_from setting.
- **idx_min** (*int, optional*) – The index of the q vector that corresponds to the minimum q point to be used in the IFT. Defaults to the first point in the q vector. Overrides use_guinier_start.
- **idx_max** (*int, optional*) – The index of the q vector that corresponds to the maximum q point to be used in the IFT. Defaults to the last point in the q vector. If write_profile is false and no profile is provided then this cannot be set, and datgnom will truncate to 8/Rg automatically. Overrides cut_dam.
- **dmax_zero** (*bool, optional*) – If True, force P(r) function to zero at Dmax.
- **alpha** (*bool, optional*) – If not zero, force alpha value to the input value. If zero (default), then alpha is automatically determined by GNOM.

- **atsas_dir** (*str, optional*) – The directory of the atsas programs (the bin directory). If not provided, the API uses the auto-detected directory.
- **use_rg_from** (*{'guinier', 'gnom', 'bift'} str, optional*) – Determines whether the Rg value used for the 8/rg cutoff calculation is from the Guinier fit, or the GNOM or BIFT P(r) function. Ignored if the rg parameter is provided. Only used if cut_8/rg is True.
- **use_guiner_start** (*bool, optional*) – If set to True, and no idx_min is provided, if a Guinier fit has been done for the input profile, the start point of the Guinier fit is used as the start point for the IFT. Ignored if there is no input profile.
- **cut_dam** (*bool, optional*) – If set to True and no idx_max is provided, then the profile is automatically truncated at q=8/Rg or 0.3 1/A, whichever is smaller. This is useful for bead models
- **write_profile** (*bool, optional*) – If True, the input profile is written to file. If False, then the input profile is ignored, and the profile specified by datadir and filename is used. This is convenient if you are trying to process a lot of files that are already on disk, as it saves having to read in each file and then save them again. Defaults to True. If False, you must provide a value for the rg parameter.
- **datadir** (*str, optional*) – If write_file is False, this is used as the path to the scattering profile on disk.
- **filename** (*str, optional.*) – If write_file is False, this is used as the filename of the scattering profile on disk.
- **save_ift** (*bool, optional*) – If True, the IFT from datgnom (.out file) is saved on disk. Requires specification of datadir and savename parameters.
- **savename** (*str, optional*) – If save_ift is True, this is used as the filename of the .out file on disk. This should just be the filename, no path. The datadir parameter is used as the parth.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – RAW settings containing relevant parameters. If provided, the dmin_zero, npts, angular_scale, system, form_factor, radius56, rmin, fwhm, ah, lh, aw, lw, and spot parameters will be overridden with the values in the settings. Default is None.
- **dmin_zero** (*bool, optional*) – If True, force P(r) function to zero at Dmin.
- **npts** (*int, optional*) – If provided, fixes the number of points in the P(r) function. If 0 (default), number of points in th P(r) function is automatically determined.
- **system** (*int, optional*) – Defines the job type as in the GNOM manual. Default is 0, a monodisperse system.
- **radius56** (*float, optional*) – The radius/thickness for system type 5/6. Default is not used.
- **rmin** (*float, optional*) – Minimum size for system types 1-6. Default is not used.

Returns

- **ift** (*bioxtasraw.SASM.IFTM*) – The IFT calculated by GNOM from the input profile.
- **dmax** (*float*) – The maximum dimension of the P(r) function.
- **rg** (*float*) – The real space radius of gyration (Rg) from the P(r) function.
- **i0** (*float*) – The real space scattering at zero angle (I(0)) from the P(r) function.

- **rg_err** (*float*) – The uncertainty in the real space radius of gyration (Rg) from the P(r) function.
- **i0_err** (*float*) – The uncertainty in the real space scattering at zero angle (I(0)) from the P(r) function.
- **total_est** (*float*) – The GNOM total estimate.
- **chi_sq** (*float*) – The chi squared value of the fit of the scattering profile calculated from the P(r) function to the input scattering profile.
- **alpha** (*float*) – The alpha value determined by datgnom.
- **quality** (*str*) – The GNOM qualitative interpretation of the total estimate.

`bioxtasraw.RAWAPI.`**`guinier_fit`**(*profile*, *idx_min*, *idx_max*, *error_weight=True*, *settings=None*)

Calculates the Rg and I(0) values from the Guinier fit defined by the input idx_min and idx_max parameters.

Parameters

- **`profile`** (*`bioxtasraw.SASM.SASM`*) – The profile to perform the Guineir fit on.
- **`idx_min`** (*`int`*) – The index of the q vector that corresponds to the minimum q point to be used in the Guinier fit.
- **`idx_max`** (*`int`*) – The index of the q vector that corresponds to the maximum q point to be used in the Guinier fit.
- **`error_weight`** (*`bool, optional`*) – If True (default), then the Guinier fit is calculated in an error weighted fashion. If not, the Guinier fit is calculated without error weight. This is overridden by the value in the settings if a settings object is provided.
- **`settings`** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, the error_weight parameter will be overridden with the value in the settings. Default is None.

Returns

- **rg** (*float*) – The Rg value of the fit.
- **i0** (*float*) – The I(0) value of the fit.
- **rg_err** (*float*) – The uncertainty in Rg. This is calculated as the largest of the uncertainty returned from autorg and the uncertainty as calculated from the covariance of the Guinier fit with the autorg determined ranges.
- **i0_err** (*float*) – The uncertainty in I(0). This is calculated as the largest of the uncertainty returned from autorg and the uncertainty as calculated from the covariance of the Guinier fit with the autorg determined ranges.
- **qmin** (*float*) – The minimum q value of the Guinier fit.
- **qmax** (*float*) – The maximum q value of the Guinier fit.
- **qRg_min** (*float*) – The q*Rg value at the minimmum q value of the Guinier fit.
- **qRg_max** (*float*) – The q*Rg value at the maximum q value of the Guinier fit.
- **r_sqr** (*float*) – The r^2 value of the fit.

`bioxtasraw.RAWAPI.`**`integrate_image`**(*img*, *settings*, *name*, *img_hdr={}*, *counters={}*, *load_path=''*)

Processes a loaded image into a 1D scattering profile.

Parameters

- **`img`** (`numpy.array`) – The image as a numpy array.

- **settings** (bioxtasraw.RAWSettings.RAWSettings) – The RAW settings to be used when integrating the image.
- **name** (*str*) – The name to be used for the scattering profile.
- **img_hdr** (*dict, optional*) – The image header associated with the given image. May be required for normalization.
- **counters** (*dict, optional*) – The counters associated with a given image. May be required for normalization.
- **load_path** (*str, optional*) – The load path of the image. Only used for metadata purposes.

Returns **profile** – The integrated 1D scattering profile.

Return type *bioxtasraw.SASM.SASM*

bioxtasraw.RAWAPI.**interpolate**(*profiles*, *ref_profile*)

Interpolates the input profiles onto the reference profile's q vector.

Parameters

- **profiles** (*list*) – A list of profiles (*bioxtasraw.SASM.SASM*) to be interpolated.
- **ref_profile** (*bioxtasraw.SASM.SASM*) – The reference profile the profiles are interpolated onto.

Returns **interpolated_profiles** – A list of interpolated profiles. Each entry in the list corresponds to the same entry in the input profiles list interpolated to the reference profile.

Return type list

bioxtasraw.RAWAPI.**load_and_integrate_images**(*filename_list*, *settings*, *return_all_images=False*)

Loads in image files and radially averages them into 1D scattering profiles. This is a convenience wrapper for *load_files()* that only returns profiles and images.

Parameters

- **filename_list** (*list*) – A list of strings containing the full path to each file to be loaded in.
- **settings** (bioxtasraw.RAWSettings.RAWSettings) – The RAW settings to be used when loading in the files, such as the calibration values used when radially averaging images.
- **return_all_images** (*bool*) – If True, all loaded images are returned. If false, only the first loaded image of the last file is returned. Useful for minimizing memory use if loading and processing a large number of images. False by default.

Returns

- **profile_list** (*list*) – A list of individual scattering profile (*bioxtasraw.SASM.SASM*) items loaded in, including those obtained from radially averaging any images.
- **img_list** (*list*) – A list of individual images (numpy.array) loaded in.

bioxtasraw.RAWAPI.**load_counter_values**(*filename_list*, *settings*, *new_filename_list=[]*)

Loads in the counter values from a separate header file associated with a given image.

Parameters

- **filename_list** (*list*) – A list of strings containing the full path to each image filename to load the counter values for.

- **settings** (bioxtasraw.RAWSettings.RAWSettings) – The RAW settings to be used when loading in the header, which defines what type of header to look for.
- **new_filename_list** (*list, optional*) – A list of strings containing optional filenames for the images. If an image file contains multiple images such as some hdf5 files), this filename is used to determine what image inside the image file the header should be loaded for. RAW expects this to be in the form <image_name>_00001.<ext>, where 00001 would be first image in a file, 00002 the second image, and so on.

Returns **counter_list** – A list of dictionaries where each key is a counter name (such as I0) and each value is the value of that counter for the input image name.

Return type list

bioxtasraw.RAWAPI.**load_files**(*filename_list*, *settings*, *return_all_images=False*)

Loads all types of files that RAW knows how to load. If images are included in the list, then the images are radially averaged as part of being loaded in.

Parameters

- **filename_list** (*list*) – A list of strings containing the full path to each file to be loaded in.
- **settings** (bioxtasraw.RAWSettings.RAWSettings) – The RAW settings to be used when loading in the files, such as the calibration values used when radially averaging images.
- **return_all_images** (*bool*) – If True, all loaded images are returned. If false, only the first loaded image of the last file is returned. Useful for minimizing memory use if loading and processing a large number of images. False by default.

Returns

- **profile_list** (*list*) – A list of individual scattering profile (*bioxtasraw.SASM.SASM*) items loaded in, including those obtained from radially averaging any images.
- **ift_list** (*list*) – A list of individual IFT (*bioxtasraw.SASM.IFTM*) items loaded in.
- **series_list** (*list*) – A list of individual series (*bioxtasraw.SECM.SECM*) items loaded in.
- **img_list** (*list*) – A list of individual images (numpy.array) loaded in.

bioxtasraw.RAWAPI.**load_ifts**(*filename_list*)

Loads IFT files: .out GNOM files and .ift BIFT files. This is a convenience wrapper for *load_files()* that only returns IFTs.

Parameters **filename_list** (*list*) – A list of strings containing the full path to each IFT file to be loaded in.

Returns **ift_list** – A list of individual IFT (*bioxtasraw.SASM.IFTM*) items loaded in.

Return type list

bioxtasraw.RAWAPI.**load_images**(*filename_list*, *settings*, *frame_num=None*)

Loads in image files.

Parameters

- **filename_list** (*list*) – A list of strings containing the full path to each file to be loaded in.

- **settings** (bioxtasraw.RAWSettings.RAWSettings) – The RAW settings to be used when loading in the files, such as the calibration values used when radially averaging images.
- **frame_num** (*int*) – If a frame number is passed, only that specific frame is returned from a multi-image file. If no frame number is passed, all frames are returned from a multi-image file. Should be either None (the default) or 0 for single image files.

Returns

- **img_list** (*list*) – A list of individual images (numpy.array) loaded in.
- **imghdr_list** (*list*) – A list of the image header values associated with each image as dictionaries.

Raises SASExceptions.WrongImageFromat – If you load in an image that RAW can't read. This could be an error with your settings, or it could fundamentally be an unreadable image type, either due to it being an unknown format or the image being corrupted.

bioxtasraw.RAWAPI.**load_mrc**(*filename_list*)

Loads DENSS .mrc files.

Parameters **filename_list** (*list*) – A list of strings containing the full path to each mrc file to be loaded in.

Returns

- **rho** (*list*) – A list of numpy.array of the calculated electron density of the models.
- **side** (*list*) – A list of floats of the real space box width in Angstroms of the models.

bioxtasraw.RAWAPI.**load_profiles**(*filename_list*, *settings=None*)

Loads individual scattering profiles from text files. This could be .dat files, but other file types such as .fit, .fir, .int, or .csv can also be loaded. This is a convenience wrapper for *load_files()* that only returns profiles. It should not be used for images, instead use *load_and_integrate_images()*.

Parameters

- **filename_list** (*list*) – A list of strings containing the full path to each profile to be loaded in.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – The RAW settings to be used when loading in the files, such as the calibration values used when radially averaging images. Default is none, this is commonly not used.

Returns **profile_list** – A list of individual scattering profile (*bioxtasraw.SASM.SASM*) items loaded in, including those obtained from radially averaging any images.

Return type list

bioxtasraw.RAWAPI.**load_series**(*filename_list*, *settings=None*)

Loads in series data. If all filenames provided at individual scattering profiles (e.g. .dat files or images that can be radially averaged into a scattering profile) they are loaded in as a single series. If all files provided are series files (e.g. .sec or .hdf5 files), each file is loaded in as a separate series. If a mixture of profiles and series are provided then an error is raised.

Parameters

- **filename_list** (*list*) – A list of strings containing the full path to each file to be loaded in.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – The RAW settings to be used when loading in the files, such as the calibration values used when radially averaging images. Default is None. This is required if you are loading images into a series or

if you wish to set the header style of the series loaded in, which is necessary for calculating the time point of each frame in the series.

Returns **series_list** – A list of individual series (*bioxtasraw.SECM.SECM*) items loaded in.

Return type list

Raises `SASExceptions.UnrecognizedDataFormat` – If you attempt to load in both scattering profiles and series files (.sec or .hdf5) at the same time.

`bioxtasraw.RAWAPI.`**`load_settings`**(*file*, *settings=None*)

Loads RAW settings from a file.

Parameters

- **`file`** (*`str`*) – The full path to a RAW settings (.cfg) file.
- **`settings`** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – A settings object containing already existing settings. Any settings duplicated between the settings loaded in and the settings provided will be overwritten by the settings from file. This parameter is generally not used.

Returns **settings** – The RAW settings stored in the .cfg file.

Return type `bioxtasraw.RAWSettings.RAWSettings`

`bioxtasraw.RAWAPI.`**`make_profile`**(*q*, *i*, *err*, *name*, *q_err=None*)

Makes a profile (*bioxtasraw.SASM.SASM*) from q, I, and uncertainty vectors. All three input vectors must be the same length.

Parameters

- **`q`** (*`iterable`*) – The q vector for the scattering profile. Should be an iterable that can be cast to a `numpy.array`, such as a list or `numpy.array`.
- **`i`** (*`iterable`*) – The intensity vector for the scattering profile. Should be an iterable that can be cast to a `numpy.array`, such as a list or `numpy.array`.
- **`err`** (*`iterable`*) – The uncertainty vector for the scattering profile. Should be an iterable that can be cast to a `numpy.array`, such as a list or `numpy.array`.
- **`name`** (*`str`*) – The name of the profile. When loading a profile from a file, this is by default set as the filename (without the full path).
- **`q_err`** (*`iterable, optional`*) – The uncertainty vector in q for the scattering profile. Should be either None or an iterable that can be cast to a `numpy.array', such as a list or :class:`numpy.array`. Default is None. Typically only used for SANS data.

Returns **profile** – A scattering profile object that can be used with the other functions in the API.

Return type *bioxtasraw.SASM.SASM*

`bioxtasraw.RAWAPI.`**`merge`**(*profiles*)

Merges the input profiles onto a single profile. Overlapping regions are averaged. Merging is done by sorting profiles by q range, then merging adjacent profiles. Typically this might be two profiles, for example a SAXS and a WAXS detector, but the function can merge an arbitrary number of profiles. You must input at least two profiles to merge.

Parameters **`profiles`** (*`list`*) – A list of profiles (*bioxtasraw.SASM.SASM*) to be merged.

Returns **merged_profile** – A single merged profile.

Return type *bioxtasraw.SASM.SASM*

`bioxtasraw.RAWAPI.`**`mw_abs`**(*profile*, *conc=0*, *i0=None*, *rho_Mprot=3.22e+23*, *rho_solv=3.34e+23*, *psv=0.7425*, *settings=None*, *use_i0_from='guinier'*, *r0=2.8179e-13*)

Calculates the M.W. of the input profile using the reference to known standard method. The input profile needs to have a calculated I(0) value, either from a Guinier fit or from a IFT P(r) function, so the I(0) value is known. You must supply either ref_i0, ref_conc, and ref_mw, or settings.

Parameters

- **`profile`** (*`bioxtasraw.SASM.SASM`*) – The profile to calculate the M.W. for.
- **`conc`** (*`float, optional`*) – The concentration of the measured profile. If not provided, then the value from profile.getParameter('Conc') is used.
- **`i0`** (*`float, optional`*) – The I(0) to be used in calculating the M.W. If not provided, then the I(0) is taken from the analysis dictionary of the profile, in conjunction with the use_i0_from setting.
- **`rho_Mprot`** (*`float, optional`*) – Number of electrons per dry mass of protein, in e-/g
- **`rho_solv`** (*`float, optional`*) – Number of electrons per volume of aqueous solvent, in e-/cm^-3
- **`psv`** (*`float, optional`*) – The protein partial specific volume.
- **`settings`** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, the ref_i0, ref_conc, and ref_mw parameters will be overridden with the values in the settings. Default is None.
- **`use_i0_from`** (*`{'guinier', 'gnom', 'bift'} str, optional`*) – Determines whether the I(0) value used for the M.W. calculation is from the Guinier fit, or the GNOM or BIFT P(r) function. Ignored if the i0 parameter is provided.
- **`r0`** (*`float, optional`*) – The scattering length of an electron, in cm. Not recommended to change.

Returns **mw** – The M.W.

Return type float

`bioxtasraw.RAWAPI.`**`mw_bayes`**(*profile*, *rg=None*, *i0=None*, *first=None*, *atsas_dir=None*, *use_i0_from='guinier'*, *write_profile=True*, *datadir=None*, *filename=None*)

Calculates the M.W. of the input profile using the Bayesian inference method implemented in datmw in the ATSAS package. This requires a separate installation of the ATSAS package to use. The input profile needs to have calculated Rg and I(0) values, either from a Guinier fit or from a IFT P(r) function, so the Rg and I(0) values are known. All return values are -1 if datmw fails.

Parameters

- **`profile`** (*`bioxtasraw.SASM.SASM`*) – The profile to calculate the M.W. for. If using write_profile false, you can pass None here. In that case you must pass values for rg, i0, and first.
- **`rg`** (*`float, optional`*) – The Rg to be used in calculating the M.W. If not provided, then the Rg is taken from the analysis dictionary of the profile, in conjunction with the use_i0_from setting.
- **`i0`** (*`float, optional`*) – The I(0) to be used in calculating the M.W. If not provided, then the I(0) is taken from the analysis dictionary of the profile, in conjunction with the use_i0_from setting.

- **first** (*int, optional*) – The first point in the q vector to be used in calculating the M.W. If not provided, then the first point is taken from the guinier analysis of the profile if available. If not available, then the first point of the profile is used.
- **atsas_dir** (*str, optional*) – The directory of the atsas programs (the bin directory). If not provided, the API uses the auto-detected directory.
- **use_i0_from** (*{'guinier', 'gnom', 'bift'} str, optional*) – Determines whether the Rg and I(0) value used for the M.W. calculation is from the Guinier fit, or the GNOM or BIFT P(r) function. Ignored if both rg and i0 parameters are provided.
- **write_profile** (*bool, optional*) – If True, the input profile is written to file. If False, then the input profile is ignored, and the profile specified by datadir and filename is used. This is convenient if you are trying to process a lot of files that are already on disk, as it saves having to read in each file and then save them again. Defaults to True. In this case, Rg and I(0) must be specified, as they are not read from the file on disk.
- **datadir** (*str, optional*) – If write_profile is False, this is used as the path to the scattering profile on disk.
- **filename** (*str, optional*) – If write_profile is False, this is used as the filename of the scattering profile on disk.

Returns

- **mw** (*float*) – The M.W.
- **mw_prob** (*float*) – The Bayesian estimated probability that the given M.W. is correct.
- **ci_lower** (*float*) – The lower bound of the Bayesian confidence interval for the M.W. value.
- **ci_upper** (*float*) – The upper bound for the Bayesian confidence interval for the M.W. value.
- **ci_prob** (*float*) – The Bayesian estimated probability that the M.W. for the scattering profile is within the confidence interval.

bioxtasraw.RAWAPI.**mw_datclass**(*profile, rg=None, i0=None, first=None, atsas_dir=None, use_i0_from='guinier', write_profile=True, datadir=None, filename=None*)

Calculates the M.W. of the input profile using the Bayesian inference method implemented in datmw in the ATSAS package. This requires a separate installation of the ATSAS package to use. The input profile needs to have calculated Rg and I(0) values, either from a Guinier fit or from a IFT P(r) function, so the Rg and I(0) values are known. All return values are -1 or None if datclass fails.

Parameters

- **profile** (*bioxtasraw.SASM.SASM*) – The profile to calculate the M.W. for. If using write_profile false, you can pass None here. In that case you must pass values for rg, i0, and first.
- **rg** (*float, optional*) – The Rg to be used in calculating the M.W. If not provided, then the Rg is taken from the analysis dictionary of the profile, in conjunction with the use_i0_from setting.
- **i0** (*float, optional*) – The I(0) to be used in calculating the M.W. If not provided, then the I(0) is taken from the analysis dictionary of the profile, in conjunction with the use_i0_from setting.
- **first** (*int, optional*) – The first point in the q vector to be used in calculating the M.W. If not provided, then the first point is taken from the guinier analysis of the profile if available. If not available, then the first point of the profile is used.

- **atsas_dir** (*str, optional*) – The directory of the atsas programs (the bin directory). If not provided, the API uses the auto-detected directory.
- **use_i0_from** (*{'guinier', 'gnom', 'bift'} str, optional*) – Determines whether the Rg and I(0) value used for the M.W. calculation is from the Guinier fit, or the GNOM or BIFT P(r) function. Ignored if both rg and i0 parameters are provided.
- **write_profile** (*bool, optional*) – If True, the input profile is written to file. If False, then the input profile is ignored, and the profile specified by datadir and filename is used. This is convenient if you are trying to process a lot of files that are already on disk, as it saves having to read in each file and then save them again. Defaults to True.
- **datadir** (*str, optional*) – If write_profile is False, this is used as the path to the scattering profile on disk.
- **filename** (*str, optional.*) – If write_profile is False, this is used as the filename of the scattering profile on disk.

Returns

- **mw** (*float*) – The M.W.
- **shape** (*str*) – The datclass shape category of the profile.
- **dmax** (*float*) – The datclass estimated Dmax of the profile.

`bioxtasraw.RAWAPI.`**`mw_ref`**(*profile, conc=0, i0=None, ref_i0=0, ref_conc=0, ref_mw=0, settings=None, use_i0_from='guinier'*)

Calculates the M.W. of the input profile using the reference to known standard method. The input profile needs to have a calculated I(0) value, either from a Guinier fit or from a IFT P(r) function, so the I(0) value is known. You must supply either ref_i0, ref_conc, and ref_mw, or settings.

Parameters

- **profile** (*bioxtasraw.SASM.SASM*) – The profile to calculate the M.W. for.
- **conc** (*float, optional*) – The concentration of the measured profile. If not provided, then the value from profile.getParameter('Conc') is used.
- **i0** (*float, optional*) – The I(0) to be used in calculating the M.W. If not provided, then the I(0) is taken from the analysis dictionary of the profile, in conjunction with the use_i0_from setting.
- **ref_i0** (*float, optional*) – The I(0) value for the reference standard. If settings are provided, this is overridden by the value in the settings.
- **ref_conc** (*float, optional*) – The concentration of the reference standard. If settings are provided, this is overridden by the value in the settings.
- **ref_mw** (*float, optional*) – The M.W. of the reference standard. If settings are provided, this is overridden by the value in the settings.
- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, the ref_i0, ref_conc, and ref_mw parameters will be overridden with the values in the settings. Default is None.
- **use_i0_from** (*{'guinier', 'gnom', 'bift'} str, optional*) – Determines whether the I(0) value used for the M.W. calculation is from the Guinier fit, or the GNOM or BIFT P(r) function. Ignored if the i0 parameter is provided.

Returns **mw** – The M.W.

Return type float

bioxtasraw.RAWAPI.**mw_vc**(*profile*, *rg=None*, *i0=None*, *protein=True*, *cutoff='Manual'*, *qmax=0.3*, *settings=None*, *use_i0_from='guinier'*, *A_prot=1.0*, *B_prot=0.1231*, *A_rna=0.808*, *B_rna=0.00934*)

Calculates the M.W. of the input profile using the volume of correlation method. The input profile needs to have calculated Rg and I(0) values, either from a Guinier fit or from a IFT P(r) function, so the Rg and I(0) values are known. You must supply either protein, cutoff, and qmax, or settings.

Parameters

- **profile** (*bioxtasraw.SASM.SASM*) – The profile to calculate the M.W. for.
- **rg** (*float, optional*) – The Rg to be used in calculating the M.W. If not provided, then the Rg is taken from the analysis dictionary of the profile, in conjunction with the use_i0_from setting.
- **i0** (*float, optional*) – The I(0) to be used in calculating the M.W. If not provided, then the I(0) is taken from the analysis dictionary of the profile, in conjunction with the use_i0_from setting.
- **protein** (*bool*) – True if the sample is protein, False if the sample is RNA. Determines which set of coefficients to use for calculating M.W.
- **cutoff** (*{''Default', '8/Rg', 'log(I0/I(q))', 'Manual''} str, optional*) – The method to use to calculate the maximum q value used for the M.W. calculation. Defaults to 'Manual'
- **qmax** (*float, optional*) – The maximum q value to be used if the 'Manual' cutoff method is selected. Defaults to 0.3.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – RAW settings containing relevant parameters. If provided, the density, cutoff, and qmax parameters will be overridden with the values in the settings. parameter. Default is None.
- **use_i0_from** (*{'guinier', 'gnom', 'bift'} str, optional*) – Determines whether the Rg and I(0) value used for the M.W. calculation is from the Guinier fit, or the GNOM or BIFT P(r) function. Ignored if both rg and i0 parameters are provided.
- **A_prot** (*float*) – The A coefficient for protein. Not recommended to be changed.
- **B_prot** (*float*) – The B coefficient for protein. Not recommended to be changed. Note that here B is defined as 1/B from the original paper.
- **A_rna** (*float*) – The A coefficient for RNA. Not recommended to be changed.
- **B_rna** (*float*) – The B coefficient for RNA. Not recommended to be changed. Note that here B is defined as 1/B from the original paper.

Returns

- **mw** (*float*) – The M.W.
- **vcor** (*float*) – The volume of correlation.
- **mw_err** (*float*) – The estimated uncertainty in the M.W.
- **qmax** (*float*) – The maximum q used to calculate the Porod volume.

bioxtasraw.RAWAPI.**mw_vp**(*profile*, *rg=None*, *i0=None*, *density=0.00083*, *cutoff='Default'*, *qmin=None*, *qmax=0.5*, *settings=None*, *use_i0_from='guinier'*)

Calculates the M.W. of the input profile using the corrected Porod volume method. The input profile needs to have calculated Rg and I(0) values, either from a Guinier fit or from a IFT P(r) function, so the Rg and I(0) values are known. You must supply either density, cutoff, and qmax, settings.

Parameters

- **profile** (*bioxtasraw.SASM.SASM*) – The profile to calculate the M.W. for.
- **rg** (*float, optional*) – The Rg to be used in calculating the M.W. If not provided, then the Rg is taken from the analysis dictionary of the profile, in conjunction with the use_i0_from setting.
- **i0** (*float, optional*) – The I(0) to be used in calculating the M.W. If not provided, then the I(0) is taken from the analysis dictionary of the profile, in conjunction with the use_i0_from setting.
- **density** (*float, optional*) – The density used to the calculate the M.W. in kDa/A^3. Defaults to 0.83*10**(-3).
- **cutoff** (*{''Default', '8/Rg', 'log(I0/I(q))', 'Manual''} str, optional*) – The method to use to calculate the maximum q value used for the M.W. calculation. Defaults to 'Default'
- **qmin** (*float, optional*) – The minimum q value to be used if rg and I(0) are supplied. Ignored if rg and i0 parameters are not provided.
- **qmax** (*float, optional*) – The maximum q value to be used if the 'Manual' cutoff method is selected. Defaults to 0.5.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – RAW settings containing relevant parameters. If provided, the density, cutoff, and qmax parameters will be overridden with the values in the settings. parameter. Default is None.
- **use_i0_from** (*{'guinier', 'gnom', 'bift'} str, optional*) – Determines whether the Rg and I(0) value used for the M.W. calculation is from the Guinier fit, or the GNOM or BIFT P(r) function. Ignored if both rg and i0 parameters are provided.

Returns

- **mw** (*float*) – The M.W.
- **pvol_cor** (*float*) – The corrected Porod volume.
- **pvol** (*float*) – The uncorrected Porod volume.
- **qmax** (*float*) – The maximum q used to calculate the Porod volume.

bioxtasraw.RAWAPI.**profiles_to_series**(*profiles*, *settings=None*)

Converts a set of individual scattering profiles (*bioxtasraw.SASM.SASM*) into a single series object (*bioxtasraw.SECM.SECM*).

Parameters

- **profiles** (*list*) – A list of profiles (*bioxtasraw.SASM.SASM*) to be converted into a series.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – The RAW settings to be used when converting profiles to a series. Default is None. This is required if you wish to set the header style of the series loaded in, which is necessary for calculating the time point of each frame in the series.

Returns **series** – A series made from the individual input profiles.

Return type *bioxtasraw.SECM.SECM*

bioxtasraw.RAWAPI.**rebin**(*profiles*, *npts=100*, *rebin_factor=1*, *log_rebin=False*)

Rebins the input profiles to either the given number of points or by the specified factor.

Parameters

- **profiles** (*list*) – A list of profiles (*bioxtasraw.SASM.SASM*) to be rebinned.

- **npts** (*int, optional*) – The number of points in each rebinned profile. Only used if rebin_factor is left to the default value of 1. Default is 100.
- **rebin_factor** (*int, optional*) – The factor by which to rebin each profile, e.g. a rebin_factor of 2 will result in half as many q points in the rebinned profile. If set to a value other than the default of 1, it overrides the npts parameter.
- **log_rebin** (*bool, option.*) – Specifies whether the rebinning should be done in linear (False) or logarithmic (True) space. Defaults to linear (False).

Returns rebinned_profiles – A list of rebinned profiles. Each entry in the list corresponds to the same entry in the input profiles list rebinned.

Return type list

`bioxtasraw.RAWAPI.`**`regals`**(*series, comp_settings, profile_type='sub', framei=None, framef=None, x_vals=None, min_iter=25, max_iter=1000, tol=0.0001, conv_type='Chi^2', use_previous_results=False, previous_results=None*)

Runs regularized alternating least squares (REGALS) on the input series to deconvolve overlapping components in the data.

Parameters

- **series** (list or *bioxtasraw.SECM.SECM*) – The input series to be deconvolved. It should either be a list of individual scattering profiles (*bioxtasraw.SASM.SASM*) or a single series object (*bioxtasraw.SECM.SECM*).
- **comp_settings** (*list*) – A list where each entry is the settings for a REGALS component. REGALS component settings are themselves lists with two entries, one for the profile settings and one for the concentration settings. Each of these is a dictionary. Profile and settings are:

```
prof_settings = {
    'type'           : 'simple', #simple, smooth, or realspace
    'lambda'         : 1.0, #float of the regularizer
    'auto_lambda'    : True, #Whether to automatically determine␣
↪lambda
    'kwargs'         : {
        'Nw'     : 50, #Number of control points (smooth/realspace)
        'dmax'   : 100, #Maximum dimension (realspace)
        'is_zero_at_r0' : True, #realspace
        'is_zero_at_damx': True, #realspace
    },
}

conc_settings = {
    'type'   : 'simple', #simple or smooth
    'lambda'         : 1.0, #float of the regularizer
    'auto_lambda'    : True, #Whether to automatically determine␣
↪lambda
    'kwargs'         : {
        'xmin'   : 0, #Minimum x value for the component
        'xmax'   : 1, #Maximum x value for the component
        'Nw'     : 50, #Number of control points (smooth)
        'is_zero_at_xmin'    : True, #Smooth
        'is_zero_at_xmax'    : True, #Smooth

    },
}
```

- **profile_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – Only used if a *bioxtasraw.SECM.SECM* is provided for the series argument. Determines which type of profile to use from the series for the REGALS. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'.
- **framei** (*int, optional*) – The initial frame in the series to use for REGALS. If not provided, it defaults to the first frame in the series.
- **framef** (*int, optional*) – The final frame in the series to use for REGALS. If not provided, it defaults to the last frame in the series.
- **min_iter** (*int, optional*) – The minimum number of iterations of the REGALS algorithm to run. Defaults to 25.
- **max_iter** (*int, optional*) – The maximum number of iterations of the REGALS algorithm to run. Defaults to 1000. If convergence method is set to 'Iterations' then this value is used as the number of iterations to run.
- **tol** (*float, optional*) – The relative tolerance to use for the 'Chi^2' convergence criteria. Defaults to 0.001.
- **conv_type** (*str, optional*) – The convergence type to use. Can be either 'Iterations' or 'Chi^2'. Iterations runs a number of iterations equal to max_iter. The chi^2 criteria runs iterations until the average of the past min_iter iterations is stable within the tolerance defined by tol, up to the max_iter number of iterations.
- **use_previous_results** (*bool, optional*) – Whether to use previous results as the initial profile and concentration vectors. Requires previous_results input. Defaults to False.
- **previous_results** (bioxtasraw.REGALS.mixture, optional) – The mixture output from a previous REGALS run, which will be used as the initial profile and concentration vectors for this REGALS run. Only used of use_previous_results is True.

Returns

- **regals_profiles** (*list*) – A list of profiles (*bioxtasraw.SASM.SASM*) determined by REGALS.
- **regals_ifts** (*list*) – A list of IFTS (*bioxtasraw.SASM.IFTM*) determined by REGALS. Only contains results for components using the 'realspace' constraint.
- **concs** (*list*) – The concentrations sampled at the input experimental points. Returns a list where each list item is a tuple of (x, conc, conc_sigma).
- **reg_concs** (*list*) – The regularized concentrations sampled at the grid/control points. Returns a list where each list item is a list of (x, conc).
- **mixture** (*REGALS.mixture*) – The final mixture result from REGALS. Can be used as input to REGALS to start with the previously determined values for each component.
- **params** (*dict*) – Contains the final convergence parameters for REGALS. Each parameter is for the final iteration of the algorithm. 'x2': chi^2, 'delta_concentration': the difference between the final and final-1 iterations concentrations. 'delta_profile': the difference between the final and final-1 iterations profiles. 'delta_u_concentration': the difference between the final and final-1 iterations u concentration matrix. 'delta_u_profile': the difference between the final and final-1 iterations U profiles matrix. 'total_iter': the total number of iterations.
- **residual** (numpy.array) – The residual between the initial input intensities and the deconvolved intensities. This is a matrix where each column corresponds to an input intensity.

`bioxtasraw.RAWAPI.`**`save_ift`**(*ift*, *fname=None*, *datadir='.'*)

Saves an individual ift as a .out (GNOM) or .ift (BIFT) file.

Parameters

- **`ift`** (*`bioxtasraw.SASM.IFTM`*) – The ift to be saved.
- **`fname`** (*`str, optional`*) – The output filename, without the directory path. If no filename is provided, the filename associated with the profile (e.g. obtained by using `profile.getParameter('filename')`) is used.
- **`datadir`** (*`str, optional`*) – The directory to save the profile in. If no directory is provided, the current directory is used.

`bioxtasraw.RAWAPI.`**`save_profile`**(*profile*, *fname=None*, *datadir='.'*, *settings=None*)

Saves an individual profile as a .dat file.

Parameters

- **`profile`** (*`bioxtasraw.SASM.SASM`*) – The profile to be saved.
- **`fname`** (*`str, optional`*) – The output filename, without the directory path. If no filename is provided, the filename associated with the profile (e.g. obtained by using `ift.getParameter('filename')`) is used.
- **`datadir`** (*`str, optional`*) – The directory to save the profile in. If no directory is provided, the current directory is used.
- **`settings`** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – The RAW settings to be used when saving, which contain settings for how to write the output .dat file.

`bioxtasraw.RAWAPI.`**`save_report`**(*fname*, *datadir='.'*, *profiles=[]*, *ifts=[]*, *series=[]*, *dammif_data=[]*)

Saves a .pdf report/summary of the input data.

Parameters

- **`fname`** (*`str`*) – The output filename without the directory path.
- **`datadir`** (*`str, optional`*) – The directory to save the report in. If no directory is provided, the current directory is used.
- **`profiles`** (*`list`*) – A list of *`bioxtasraw.SASM.SASM`* profiles to add to the report.
- **`ifts`** (*`list`*) – A list of *`bioxtasraw.SASM.IFTM`* IFTs to add to the report.
- **`series`** (*`list`*) – A list of *`bioxtasraw.SECM.SECM`* series to add to the report.
- **`dammif_data`** (*`list`*) – A list of paths to the summary files of dammif runs from RAW (.csv files) to add to the report.

`bioxtasraw.RAWAPI.`**`save_series`**(*series*, *fname=None*, *datadir='.'*)

Saves an individual series as a .hdf5 file.

Parameters

- **`series`** (*`bioxtasraw.SASM.IFTM`*) – The series to be saved.
- **`fname`** (*`str, optional`*) – The output filename, without the directory path. If no filename is provided, the filename associated with the profile (e.g. obtained by using `series.getParameter('filename')`) is used.
- **`datadir`** (*`str, optional`*) – The directory to save the profile in. If no directory is provided, the current directory is used.

`bioxtasraw.RAWAPI.`**`save_settings`**(*settings*, *fname*, *datadir='.'*)

Saves the settings to a file.

Parameters

- **settings** (`bioxtasraw.RAWSettings.RAWSettings`) – The settings to be saved.
- **fname** (*str*) – The save filename (without path). Should be a .cfg file.
- **datadir** (*str, optional*) – The directory to save the settings. Defaults to the current directory.

Returns **success** – Whether the save was successful.

Return type bool

`bioxtasraw.RAWAPI.`**`series_calc`**(*sub_profiles*, *window_size=5*, *settings=None*, *error_weight=True*, *vp_density=0.00083*, *vp_cutoff='Default'*, *vp_qmax=0.5*, *vc_protein=True*, *vc_cutoff='Manual'*, *vc_qmax=0.3*, *vc_a_prot=1.0*, *vc_b_prot=0.1231*, *vc_a_rna=0.808*, *vc_b_rna=0.00934*)

Calculates Rg and MW for the input subtracted profiles. If you are working with a `SECM.SECM` series object then use *set_buffer_range()* instead of this function.

Parameters

- **sub_profiles** (*list*) – A list of subtracted profiles (`SASM.SASM`) to calculate the Rg and M.W. values for using the series calculation function.
- **window_size** (*int, optional*) – The size of the average window used when calculating Rg and MW. So if the window is 5, 5 a window is size 5 is slid along the series, and profiles in that window are averaged before being used to calculate Rg and MW. For example, frames 1-5, 2-6, 3-7, etc would be averaged and then have Rg and MW calculated from that average.
- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, err_weight, vp_density, vp_cutoff, vp_qmax, vc_protein, vc_cutoff, and vc_qmax are overridden by the values in the settings.
- **error_weight** (*bool, optional*) – Whether to use error weighting when calculating the Rg.
- **vp_density** (*float, optional*) – The density used for the Porod volume M.W. calculation in kDa/A^3. Defaults to 0.83*10**(-3).
- **vp_cutoff** (*{''Default', '8/Rg', 'log(I0/I(q))', 'Manual''} str, optional*) – The method to use to calculate the maximum q value used for the Porod volume M.W. calculation. Defaults to 'Default'
- **vp_qmax** (*float, optional*) – The maximum q value to be used if the 'Manual' cutoff method is selected for the Porod volume M.W. calculation. Defaults to 0.5.
- **vc_protein** (*bool*) – True if the sample is protein, False if the sample is RNA. Determines which set of coefficients to use for calculating M.W.
- **vc_cutoff** (*{''Default', '8/Rg', 'log(I0/I(q))', 'Manual''} str, optional*) – The method to use to calculate the maximum q value used for the M.W. calculation. Defaults to 'Manual'
- **vc_qmax** (*float, optional*) – The maximum q value to be used if the 'Manual' cutoff method is selected. Defaults to 0.3.

- **vc_a_prot** (*float*) – The volume of correlation A coefficient for protein. Not recommended to be changed.
- **vc_b_prot** (*float*) – The volume of correlation B coefficient for protein. Not recommended to be changed. Note that here B is defined as 1/B from the original paper.
- **vc_a_rna** (*float*) – The volume of correlation A coefficient for RNA. Not recommended to be changed.
- **vc_b_rna** (*float*) – The volume of correlation B coefficient for RNA. Not recommended to be changed. Note that here B is defined as 1/B from the original paper.

Returns

- **rg** (`numpy.array`) – An array of the Rg values calculated for each subtracted profile. If no Rg value could be calculated then the value is -1. Each array index is the Rg corresponding to the subtracted profile at that index in the sub_profiles list.
- **rger** (`numpy.array`) – An array of the uncertainty in the Rg values calculated for each subtracted profile. If no Rg value could be calculated then the value is -1. Each array index is the uncertainty corresponding to the subtracted profile at that index in the sub_profiles list.
- **i0** (`numpy.array`) – An array of the I(0) values calculated for each subtracted profile. If no I(0) value could be calculated then the value is -1. Each array index is the I(0) corresponding to the subtracted profile at that index in the sub_profiles list.
- **i0er** (`numpy.array`) – An array of the uncertainty in the I(0) values calculated for each subtracted profile. If no I(0) value could be calculated then the value is -1. Each array index is the uncertainty corresponding to the subtracted profile at that index in the sub_profiles list.
- **vcmw** (`numpy.array`) – An array of the volume of correlation M.W. values calculated for each subtracted profile. If no M.W. value could be calculated then the value is -1. Each array index is the M.W. corresponding to the subtracted profile at that index in the sub_profiles list.
- **vcmwer** (`numpy.array`) – An array of the uncertainty in the volume of correlation M.W. values calculated for each subtracted profile. If no M.W. value could be calculated then the value is -1. Each array index is the uncertainty corresponding to the subtracted profile at that index in the sub_profiles list.
- **vpmw** (`numpy.array`) – An array of the Porod volume M.W. values calculated for each subtracted profile. If no M.W. value could be calculated then the value is -1. Each array index is the M.W. corresponding to the subtracted profile at that index in the sub_profiles list.

`bioxtasraw.RAWAPI.`**`set_baseline_correction`**(*series*, *start_range*, *end_range*, *baseline_type*, *bl_extrap=True*, *int_type='total'*, *q_val=None*, *q_range=None*, *window_size=5*, *settings=None*, *min_iter=100*, *max_iter=2000*, *calc_thresh=1.02*, *error_weight=True*, *vp_density=0.00083*, *vp_cutoff='Default'*, *vp_qmax=0.5*, *vc_protein=True*, *vc_cutoff='Manual'*, *vc_qmax=0.3*, *vc_a_prot=1.0*, *vc_b_prot=0.1231*, *vc_a_rna=0.808*, *vc_b_rna=0.00934*)

Calculates and sets the baseline correction for the input series. Then recalculates the series Rg and M.W. values based on the baseline corrected profiles. The input profile must have both unsubtracted and subtracted profiles.

Parameters

- **series** (*bioxtasraw.SECM.SECM*) – The input series to calculate the baseline for.
- **start_range** (*list*) – A list defining the baseline start range to be validated. The list is two integers, the start of the range and the end of the range.
- **end_range** (*list*) – A list defining the baseline end range to be validated. The list is two integers, the start of the range and the end of the range.
- **baseline_type** (*{'Integral', 'Linear'} str*) – Defines the baseline type as either Integral or Linear.
- **bl_extrap** (*bool, optional*) – Used for a linear baseline. If True, the linear baseline correction is extrapolated to the entire series, if not it is only applied between the start_range and end_range.
- **int_type** (*{'total', 'mean', 'q_val', 'q_range'} str, optional*) – The intensity type to use the calculation of the baseline range. Total integrated intensity - 'total', mean intensity - 'mean', intensity at a particular q value - 'q_val', intensity in a given q range - 'q_range'. Use of q_val or q_range requires the corresponding parameter to be provided.
- **q_val** (*float, optional*) – If int_type is 'q_val', the q value used for the intensity is set by this parameter.
- **q_range** (*list, optional*) – This should have two entries, both floats. The first is the minimum q value of the range, the second the maximum q value of the range. If int_type is 'q_range', the q range used for the intensity is set by this parameter.
- **window_size** (*int, optional*) – The size of the average window used when calculating Rg and MW. So if the window is 5, 5 a window is size 5 is slid along the series, and profiles in that window are averaged before being used to calculate Rg and MW. For example, frames 1-5, 2-6, 3-7, etc would be averaged and then have Rg and MW calculated from that average.
- **settings** (bioxtasraw.RAWSettings.RAWSettings, optional) – RAW settings containing relevant parameters. If provided, sim_test, sim_cor, and sim_threshold are overridden by the values in the settings.
- **min_iter** (*int, optional*) – The minimum number of iterations for calculating the integral baseline correction. Overridden if settings are provided.
- **max_iter** (*int, optional*) – The maximum number of iterations for calculating the integral baseline correction. Overridden if settings are provided.
- **calc_thresh** (*float, optional*) – If the ratio of the scattering profile intensity to the average buffer intensity is greater than this threshold, the Rg and MW for the profile is calculated. Defaults to 1.02.
- **error_weight** (*bool, optional*) – Whether to use error weighting when calculating the Rg.
- **vp_density** (*float, optional*) – The density used for the Porod volume M.W. calculation in kDa/A^3. Defaults to 0.83*10**(-3).
- **vp_cutoff** (*{''Default', '8/Rg', 'log(I0/I(q))', 'Manual''} str, optional*) – The method to use to calculate the maximum q value used for the Porod volume M.W. calculation. Defaults to 'Default'
- **vp_qmax** (*float, optional*) – The maximum q value to be used if the 'Manual' cutoff method is selected for the Porod volume M.W. calculation. Defaults to 0.5.

- **vc_protein** (*bool*) – True if the sample is protein, False if the sample is RNA. Determines which set of coefficients to use for calculating M.W.
- **vc_cutoff** (*{''Default', '8/Rg', 'log(I0/I(q))', 'Manual''} str, optional*) – The method to use to calculate the maximum q value used for the M.W. calculation. Defaults to 'Manual'
- **vc_qmax** (*float, optional*) – The maximum q value to be used if the 'Manual' cutoff method is selected. Defaults to 0.3.
- **vc_a_prot** (*float*) – The volume of correlation A coefficient for protein. Not recommended to be changed.
- **vc_b_prot** (*float*) – The volume of correlation B coefficient for protein. Not recommended to be changed. Note that here B is defined as 1/B from the original paper.
- **vc_a_rna** (*float*) – The volume of correlation A coefficient for RNA. Not recommended to be changed.
- **vc_b_rna** (*float*) – The volume of correlation B coefficient for RNA. Not recommended to be changed. Note that here B is defined as 1/B from the original paper.

Returns

- **bl_cor_profiles** (*list*) – A list of the baseline corrected profiles.
- **rg** (`numpy.array`) – An array of the Rg values calculated for each subtracted profile. If no Rg value could be calculated then the value is -1. Each array index is the Rg corresponding to the subtracted profile at that index in the sub_profiles list.
- **rger** (`numpy.array`) – An array of the uncertainty in the Rg values calculated for each subtracted profile. If no Rg value could be calculated then the value is -1. Each array index is the uncertainty corresponding to the subtracted profile at that index in the sub_profiles list.
- **i0** (`numpy.array`) – An array of the I(0) values calculated for each subtracted profile. If no I(0) value could be calculated then the value is -1. Each array index is the I(0) corresponding to the subtracted profile at that index in the sub_profiles list.
- **i0er** (`numpy.array`) – An array of the uncertainty in the I(0) values calculated for each subtracted profile. If no I(0) value could be calculated then the value is -1. Each array index is the uncertainty corresponding to the subtracted profile at that index in the sub_profiles list.
- **vcmw** (`numpy.array`) – An array of the volume of correlation M.W. values calculated for each subtracted profile. If no M.W. value could be calculated then the value is -1. Each array index is the M.W. corresponding to the subtracted profile at that index in the sub_profiles list.
- **vcmwer** (`numpy.array`) – An array of the uncertainty in the volume of correlation M.W. values calculated for each subtracted profile. If no M.W. value could be calculated then the value is -1. Each array index is the uncertainty corresponding to the subtracted profile at that index in the sub_profiles list.
- **vpmw** (`numpy.array`) – An array of the Porod volume M.W. values calculated for each subtracted profile. If no M.W. value could be calculated then the value is -1. Each array index is the M.W. corresponding to the subtracted profile at that index in the sub_profiles list.
- **bl_corr** (*list*) – A list of the baseline correction applied. Each item is a *bioxtasraw.SASM.SASM*, and there is one for every baseline corrected profile. The intensity is the value

subtracted from the starting intensity of the corresponding profile to achieve the baseline corrected intensity.

- **fit_results** (*list*) – Only contains items if a linear correction is done. In that case, each item is the linear fit results a, b, and corresponding covariances for a given q value. There is one item per q value of the input profiles.

`bioxtasraw.RAWAPI.`**`set_buffer_range`**(*series*, *buffer_range*, *int_type='total'*, *q_val=None*, *q_range=None*, *already_subtracted=False*, *window_size=5*, *settings=None*, *calc_thresh=1.02*, *sim_test='CorMap'*, *sim_cor='Bonferroni'*, *sim_thresh=0.01*, *error_weight=True*, *vp_density=0.00083*, *vp_cutoff='Default'*, *vp_qmax=0.5*, *vc_protein=True*, *vc_cutoff='Manual'*, *vc_qmax=0.3*, *vc_a_prot=1.0*, *vc_b_prot=0.1231*, *vc_a_rna=0.808*, *vc_b_rna=0.00934*)

Sets the buffer range for a series, carries out the subtraction, and calculates Rg and MW vs. frame number.

Parameters

- **`series`** (*`bioxtasraw.SECM.SECM`*) – The input series to set the buffer range for.
- **`buffer_range`** (*`list`*) – A list defining the input buffer range to be set. The list is made up of a set of sub-ranges, each defined by an entry in the list. Each sub-range item should be a list or tuple where the first entry is the starting index of the range and the second entry is the ending index of the range. So a list like `[[0, 10], [100, 110]]` would define a buffer range consisting of two sub-ranges, the first from profiles 0-10 in the series and the second from profiles 100-110 in the series.
- **`int_type`** (*`{'total', 'mean', 'q_val', 'q_range'} str, optional`*) – The intensity type to use when setting the buffer range. Total integrated intensity - 'total', mean intensity - 'mean', intensity at a particular q value - 'q_val', intensity in a given q range - 'q_range'. Use of q_val or q_range requires the corresponding parameter to be provided.
- **`q_val`** (*`float, optional`*) – If int_type is 'q_val', the q value used for the intensity is set by this parameter.
- **`q_range`** (*`list, optional`*) – This should have two entries, both floats. The first is the minimum q value of the range, the second the maximum q value of the range. If int_type is 'q_range', the q range used for the intensity is set by this parameter.
- **`already_subtracted`** (*`bool, optional`*) – Whether the series is already subtracted. If True, any buffer_range input is ignored and the series unsubtracted profiles are set as the subtracted profiles. Defaults to False
- **`window_size`** (*`int, optional`*) – The size of the average window used when calculating Rg and MW. So if the window is 5, 5 a window is size 5 is slid along the series, and profiles in that window are averaged before being used to calculate Rg and MW. For example, frames 1-5, 2-6, 3-7, etc would be averaged and then have Rg and MW calculated from that average.
- **`settings`** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, calc_thresh, sim_test, sim_cor, sim_thresh, err_weight, vp_density, vp_cutoff, vp_qmax, vc_protein, vc_cutoff, and vc_qmax are overridden by the values in the settings.
- **`calc_thresh`** (*`float, optional`*) – If the ratio of the scattering profile intensity to the average buffer intensity is greater than this threshold, the Rg and MW for the profile is calculated. Defaults to 1.02.

- **sim_test** (*{'CorMap'} str, optional*) – Sets the type of similarity test to be used. Currently only CorMap is supported as an option. Is overridden if settings are provided.
- **sim_cor** (*{'Bonferroni', 'None'} str, optional*) – Sets the multiple testing correction to be used as part of the similarity test. Default is Bonferroni. Is overridden if settings are provided.
- **sim_thresh** (*float, optional*) – Sets the p value threshold for the similarity test. A higher value is a more strict test (range from 0-1). Is overridden if settings are provided.
- **error_weight** (*bool, optional*) – Whether to use error weighting when calculating the Rg.
- **vp_density** (*float, optional*) – The density used for the Porod volume M.W. calculation in kDa/A^3. Defaults to 0.83*10**(-3).
- **vp_cutoff** (*{''Default', '8/Rg', 'log(I0/I(q))', 'Manual''} str, optional*) – The method to use to calculate the maximum q value used for the Porod volume M.W. calculation. Defaults to 'Default'
- **vp_qmax** (*float, optional*) – The maximum q value to be used if the 'Manual' cutoff method is selected for the Porod volume M.W. calculation. Defaults to 0.5.
- **vc_protein** (*bool*) – True if the sample is protein, False if the sample is RNA. Determines which set of coefficients to use for calculating M.W.
- **vc_cutoff** (*{''Default', '8/Rg', 'log(I0/I(q))', 'Manual''} str, optional*) – The method to use to calculate the maximum q value used for the M.W. calculation. Defaults to 'Manual'
- **vc_qmax** (*float, optional*) – The maximum q value to be used if the 'Manual' cutoff method is selected. Defaults to 0.3.
- **vc_a_prot** (*float*) – The volume of correlation A coefficient for protein. Not recommended to be changed.
- **vc_b_prot** (*float*) – The volume of correlation B coefficient for protein. Not recommended to be changed. Note that here B is defined as 1/B from the original paper.
- **vc_a_rna** (*float*) – The volume of correlation A coefficient for RNA. Not recommended to be changed.
- **vc_b_rna** (*float*) – The volume of correlation B coefficient for RNA. Not recommended to be changed. Note that here B is defined as 1/B from the original paper.

Returns

- **sub_profiles** (*list*) – A list of `SASM.SASM` subtracted profiles. Each profile is created by creating an average buffer from the range defined by buffer_range and subtracting that from every unsubtracted profile in the series.
- **rg** (`numpy.array`) – An array of the Rg values calculated for each subtracted profile. If no Rg value could be calculated then the value is -1. Each array index is the Rg corresponding to the subtracted profile at that index in the sub_profiles list.
- **rger** (`numpy.array`) – An array of the uncertainty in the Rg values calculated for each subtracted profile. If no Rg value could be calculated then the value is -1. Each array index is the uncertainty corresponding to the subtracted profile at that index in the sub_profiles list.

- **i0** (`numpy.array`) – An array of the I(0) values calculated for each subtracted profile. If no I(0) value could be calculated then the value is -1. Each array index is the I(0) corresponding to the subtracted profile at that index in the sub_profiles list.
- **i0er** (`numpy.array`) – An array of the uncertainty in the I(0) values calculated for each subtracted profile. If no I(0) value could be calculated then the value is -1. Each array index is the uncertainty corresponding to the subtracted profile at that index in the sub_profiles list.
- **vcmw** (`numpy.array`) – An array of the volume of correlation M.W. values calculated for each subtracted profile. If no M.W. value could be calculated then the value is -1. Each array index is the M.W. corresponding to the subtracted profile at that index in the sub_profiles list.
- **vcmwer** (`numpy.array`) – An array of the uncertainty in the volume of correlation M.W. values calculated for each subtracted profile. If no M.W. value could be calculated then the value is -1. Each array index is the uncertainty corresponding to the subtracted profile at that index in the sub_profiles list.
- **vpmw** (`numpy.array`) – An array of the Porod volume M.W. values calculated for each subtracted profile. If no M.W. value could be calculated then the value is -1. Each array index is the M.W. corresponding to the subtracted profile at that index in the sub_profiles list.

`bioxtasraw.RAWAPI.`**`set_sample_range`**(*series*, *sample_range*, *profile_type='sub'*)

Sets the sample range for the series and returns the subtracted scattering profile corresponding to the specified sample range.

Parameters

- **`series`** (*`bioxtasraw.SECM.SECM`*) – The input series to set the sample range for.
- **`sample_range`** (*`list`*) – A list defining the input sample range to be set. The list is made up of a set of sub-ranges, each defined by an entry in the list. Each sub-range item should be a list or tuple where the first entry is the starting index of the range and the second entry is the ending index of the range. So a list like `[[0, 10], [100, 110]]` would define a sample range consisting of two sub-ranges, the first from profiles 0-10 in the series and the second from profiles 100-110 in the series.
- **`profile_type`** (*`{'unsub', 'sub', 'baseline'} str, optional`*) – Determines which type of profile to use from the series to set the sample range. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'.

Returns **sub_profile** – The subtracted scattering profile calculated from the specified sample_range.

Return type *`bioxtasraw.SASM.SASM`*

`bioxtasraw.RAWAPI.`**`subtract`**(*profiles*, *bkg_profile*, *forced=False*, *full=False*)

Subtracts a background profile from the other input profiles.

Parameters

- **`profiles`** (*`list`*) – A list of profiles (*`bioxtasraw.SASM.SASM`*) to be subtracted.
- **`bkg_profile`** (*`bioxtasraw.SASM.SASM`*) – The background profile to subtract from the profiles.
- **`forced`** (*`bool, optional`*) – If True, RAW will attempt to subtract profiles even if the q vectors do not agree. Defaults to False.

- **full** (*bool, optional*) – If False, RAW will only use the portion of the profile between the defined q start and q end indices. If True, RAW will use the full q range of the profile, regardless of the defined q start and end indices. Defaults to False.

Returns **sub_profiles** – A list of subtracted profiles. Each entry in the list corresponds to the same entry in the input profiles list with the bkg_profile subtracted from it.

Return type list

bioxtasraw.RAWAPI.**supcomb**(*target*, *ref_file*, *datadir*, *mode='fast'*, *superposition='ALL'*, *enantiomorphs='YES'*, *proximity='NSD'*, *symmetry='P1'*, *fraction='1.0'*, *atsas_dir=None*, *abort_event=<threading.Event object>*)

Aligns the target to the reference file using SUPCOMB from the ATSAS package. Require a separate installation of ATSAS. Both files must be in the same folder, and the aligned file is output in the folder. The aligned file will have the same name as the target file with _aligned appended to the end of the name. This function blocks until SUPCOMB is done.

Parameters

- **target** (*str*) – The target file name, without path. This file is aligned to the reference file, and must be in the same folder as the reference file.
- **ref_file** (*str*) – The reference file name, without path.
- **datadir** (*str*) – The directory containing both the target and ref_file. It is also the directory for the output file.
- **mode** (*{'fast', 'slow'} str, optional*) – The alignment mode. Must be 'slow' if symmetry is not P1. Default is fast.
- **superposition** (*{'ALL', 'BACKBONE'} str, optional*) – Whether to align all atoms or just the backbone. Default is ALL.
- **enantiomorphs** (*{'YES', 'NO'} str, optional*) – Whether to generate enantiomorphs during alignment. Default is YES.
- **proximity** (*{'NSD, 'VOL'} str, optional*) – What method to use to determine distance between two models. Default is NSD.
- **symmetry** (*str, optional*) – The symmetry applied to the alignment. If the symmetry is not P1, then the mode must be slow. Any symmetry allowed in the SUPCOMB manual is an acceptable input.
- **fraction** (*str, optional*) – The amount of the structure closer to the surface to use for NSD calculations.
- **atsas_dir** (*str, optional*) – The directory of the atsas programs (the bin directory). If not provided, the API uses the auto-detected directory.
- **abort_event** (threading.Event, optional) – A threading.Event or multiprocessing.Event. If this event is set it will abort the supcomb run.

bioxtasraw.RAWAPI.**superimpose**(*profiles*, *ref_profile*, *scale=True*, *offset=False*)

Superimposes the profiles onto the reference profile using either a scale, offset, or both.

Parameters

- **profiles** (*list*) – A list of profiles (*bioxtasraw.SASM.SASM*) to be superimposed.
- **ref_profile** (*bioxtasraw.SASM.SASM*) – The reference profile the profiles are superimposed onto.
- **scale** (*bool, optional*) – Whether a scale is used when superimposing. Default is True.

- **offset** (*bool, optional*) – Whether an offset is used when superimposing. Default is False.

Returns **sup_profiles** – A list of superimposed profiles. Each entry in the list corresponds to the same entry in the input profiles list superimposed on the reference profile.

Return type list

`bioxtasraw.RAWAPI.`**`svd`**(*series*, *profile_type='sub'*, *framei=None*, *framef=None*, *norm=True*)

Runs singular value decomposition (SVD) on the input series.

Parameters

- **series** (list or *bioxtasraw.SECM.SECM*) – The input series to be deconvolved. It should either be a list of individual scattering profiles (*bioxtasraw.SASM.SASM*) or a single series object (*bioxtasraw.SECM.SECM*).
- **profile_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – Only used if a *bioxtasraw.SECM.SECM* is provided for the series argument. Determines which type of profile to use from the series for the EFA. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'.
- **framei** (*int, optional*) – The initial frame in the series to use for EFA. If not provided, it defaults to the first frame in the series.
- **framef** (*int, optional*) – The final frame in the series to use for EFA. If not provided, it defaults to the last frame in the series.
- **norm** (*bool, optional*) – Whether error normalized intensity should be used for EFA. Defaults to True. Recommended to not change this.

Returns

- **svd_s** (class:*numpy.array*) – The singular values.
- **svd_U** (class:*numpy.array*) – The left singular vectors
- **svd_V** (class:*numpy.array*) – The right singular vectors.

Raises `SASExceptions.EFAError` – If SVD cannot be carried out.

`bioxtasraw.RAWAPI.`**`validate_baseline_range`**(*series*, *start_range*, *end_range*, *baseline_type='Integral'*, *profile_type='sub'*, *int_type='total'*, *q_val=None*, *q_range=None*, *fast=False*, *settings=None*, *sim_test='CorMap'*, *sim_cor='Bonferroni'*, *sim_thresh=0.01*)

Validates whether the input start and end ranges are a trustworthy baseline range or not. This is designed to work with SEC-SAXS data, but may work in other circumstances. Note that currently the validation for linear baselines almost always returns false, and is of little use.

Parameters

- **series** (list or *bioxtasraw.SECM.SECM*) – The input series to validate the baseline range for. It should either be a list of individual scattering profiles (*bioxtasraw.SASM.SASM*) or a single series object (*bioxtasraw.SECM.SECM*).
- **start_range** (*list*) – A list defining the baseline start range to be validated. The list is two integers, the start of the range and the end of the range.
- **end_range** (*list*) – A list defining the baseline end range to be validated. The list is two integers, the start of the range and the end of the range.
- **baseline_type** (*{'Integral', 'Linear'} str, optional*) – Defines the baseline type for validation purposes.

- **profile_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – Only used if a *bioxtasraw.SECM.SECM* is provided for the series argument. Determines which type of profile to use from the series to validate the baseline range. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'.
- **int_type** (*{'total', 'mean', 'q_val', 'q_range'} str, optional*) – The intensity type to use for the validation of the baseline range. Total integrated intensity - 'total', mean intensity - 'mean', intensity at a particular q value - 'q_val', intensity in a given q range - 'q_range'. Use of q_val or q_range requires the corresponding parameter to be provided.
- **q_val** (*float, optional*) – If int_type is 'q_val', the q value used for the intensity is set by this parameter.
- **q_range** (*list, optional*) – This should have two entries, both floats. The first is the minimum q value of the range, the second the maximum q value of the range. If int_type is 'q_range', the q range used for the intensity is set by this parameter.
- **fast** (*bool, optional*) – Whether the test should be done in fast mode or not. A fast test stops at the first failed check. In a normal test (not fast), all metrics are checked. Using a fast test is best when trying to automatically determine a baseline range, something which can take many separate validation checks. A normal test is best when trying to determine what, if anything, about your selected range might be problematic.
- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, sim_test, sim_cor, and sim_threshold are overridden by the values in the settings.
- **sim_test** (*{'CorMap'} str, optional*) – Sets the type of similarity test to be used. Currently only CorMap is supported as an option. Is overridden if settings are provided.
- **sim_cor** (*{'Bonferroni', 'None'} str, optional*) – Sets the multiple testing correction to be used as part of the similarity test. Default is Bonferroni. Is overridden if settings are provided.
- **sim_thresh** (*float, optional*) – Sets the p value threshold for the similarity test. A higher value is a more strict test (range from 0-1). Is overridden if settings are provided.

Returns

- **valid** (*bool*) – If the baseline start and end ranges are a valid baseline range.
- **valid_reuslts** (*tuple*) – A tuple of bools. The first entry is whether the start range is valid, the second is whether the end range is valid. The start range is always valid for a linear baseline correction.
- **similarity_results** (*tuple*) – A tuple of dicts. The first entry is the start range similarity results, the second entry is the end range similarity results. A similarity test is only done for integral baselines. For linear baselines an empty dictionary is returned. For an integral baseline, each dictionary has the following keys are: 'all_similar' - whether all profiles in the selected range are similar over the entire profile. 'low_q_similar' - whether all profiles in the selected range are similar over the low q region of the profile. 'high_q_similar' - whether all profiles in the selected range are similar over the high q region. 'max_idx' - The index of the profile used as the reference for the similarity test, corresponding to the profile with the highest intensity in the region. 'all_outliers' - Indices of the outlier profiles of the similarity test at all q. 'low_q_outliers' - Indices of the outlier profiles of the similarity test at low q. 'high_q_outliers' - Indices of the outlier profiles of the similarity test at high q.

- **svd_results** (*tuple*) – A tuple of dicts. The first entry is the start range svd results, the second is the end range svd results. A SVD test is only done for integral baselines. For linear baselines an empty dictionary is returned. For an integral baseline the dictionary keys are: 'svals' - the number of significant singular vectors in the region. 'U' - the left singular vectors. 'V' - the right singular vectors. 'u_autocor' - The autocorrelation of the left singular vectors. 'v_autocor' - The autocorrelation of the right singular vectors.
- **intI_results** (*tuple*) – A tuple of dicts. The first entry is the start range intensity results, the second entry is the end range intensity results. The intensity test is only done for integral baselines. For linear baselines an empty dictionary is returned. For an integral baseline, each dictionary has the following keys: 'intI_r' - the Spearman correlation coefficient of the intensity in the region. 'inti_pval' - the p-value from the Spearman correlation test on the intensity of the region. 'intI_valid' - Whether the range is a valid buffer range based on the intensity correlation. The same keys are provided but with smoothed in front, indicating the test results on the smoothed intensity.
- **other_results** (*tuple*) – A tuple of dicts. The first entry is the start range other results, the second entry is the end range other results. For either baseline type, only the end range has other results, as this is an evaluation based on comparing the end range to the start range. For an integral baseline, This contains the following keys: 'zero_valid' - whether all q points in the end region are higher than the same q points in the start region. 'zero_outliers' - An array of bools in the shape of the input profile q vector. True where q values are more than 4 sigma lower in the end region than the start region. 'zero_q' - the q vector. For a linear baseline the following keys are available: 'fit_valid' - whether the fit is valid. Note that fit_valid almost always returns False, and is currently of little use.

`bioxtasraw.RAWAPI.`**`validate_buffer_range`**(*series*, *buf_range*, *profile_type='unsub'*, *int_type='total'*, *q_val=None*, *q_range=None*, *fast=False*, *settings=None*, *sim_test='CorMap'*, *sim_cor='Bonferroni'*, *sim_thresh=0.01*)

Validates whether the input data range is a trustworthy buffer range or not. This is designed to work with SEC-SAXS data, but may work in other circumstances.

Parameters

- **series** (list or *bioxtasraw.SECM.SECM*) – The input series to validate the buffer range for. It should either be a list of individual scattering profiles (*bioxtasraw.SASM.SASM*) or a single series object (*bioxtasraw.SECM.SECM*).
- **buf_range** (*list*) – A list defining the input buffer range to be validated. The list is made up of a set of sub-ranges, each defined by an entry in the list. Each sub-range item should be a list or tuple where the first entry is the starting index of the range and the second entry is the ending index of the range. So a list like `[[0, 10], [100, 110]]` would define a buffer range consisting of two sub-ranges, the first from profiles 0-10 in the series and the second from profiles 100-110 in the series.
- **profile_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – Only used if a *bioxtasraw.SECM.SECM* is provided for the series argument. Determines which type of profile to use from the series to validate the buffer range. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'.
- **int_type** (*{'total', 'mean', 'q_val', 'q_range'} str, optional*) – The intensity type to use for the validation of the buffer range. Total integrated intensity - 'total', mean intensity - 'mean', intensity at a particular q value - 'q_val', intensity in a given q range - 'q_range'. Use of q_val or q_range requires the corresponding parameter to be provided.
- **q_val** (*float, optional*) – If int_type is 'q_val', the q value used for the intensity is set by this parameter.

- **q_range** (*list, optional*) – This should have two entries, both floats. The first is the minimum q value of the range, the second the maximum q value of the range. If int_type is 'q_range', the q range used for the intensity is set by this parameter.
- **fast** (*bool, optional*) – Whether the test should be done in fast mode or not. A fast test stops at the first failed check. In a normal test (not fast), all metrics are checked. Using a fast test is best when trying to automatically determine a buffer range, something which can take many separate validation checks. A normal test is best when trying to determine what, if anything, about your selected region might be problematic.
- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, sim_test, sim_cor, and sim_threshold are overridden by the values in the settings.
- **sim_test** (*{'CorMap'} str, optional*) – Sets the type of similarity test to be used. Currently only CorMap is supported as an option. Is overridden if settings are provided.
- **sim_cor** (*{'Bonferroni', 'None'} str, optional*) – Sets the multiple testing correction to be used as part of the similarity test. Default is Bonferroni. Is overridden if settings are provided.
- **sim_thresh** (*float, optional*) – Sets the p value threshold for the similarity test. A higher value is a more strict test (range from 0-1). Is overridden if settings are provided.

Returns

- **valid** (*bool*) – If the input buffer range is a valid buffer range.
- **similarity_results** (*dict*) – A dictionary with the results of the similarity test. In particular, keys are: 'all_similar' - whether all profiles in the selected range are similar over the entire profile. 'low_q_similar' - whether all profiles in the selected range are similar over the low q region of the profile. 'high_q_similar' - whether all profiles in the selected range are similar over the high q region. 'max_idx' - The index of the profile used as the reference for the similarity test, corresponding to the profile with the highest intensity in the region. 'all_outliers' - Indices of the outlier profiles of the similarity test at all q. 'low_q_outliers' - Indices of the outlier profiles of the similarity test at low q. 'high_q_outliers' - Indices of the outlier profiles of the similarity test at high q.
- **svd_results** (*dict*) – A dictionary with the results of the SVD test. In particular, keys are: 'svals' - the number of significant singular vectors in the region. 'U' - the left singular vectors. 'V' - the right singular vectors. 'u_autocor' - The autocorrelation of the left singular vectors. 'v_autocor' - The autocorrelation of the right singular vectors.
- **intI_results** (*dict*) – A dictionary with the results of the intensity test. In particular, keys are: 'intI_r' - the Spearman correlation coefficient of the intensity in the region. 'inti_pval' - the p-value from the Spearman correlation test on the intensity of the region. 'intI_valid' - Whether the range is a valid buffer range based on the intensity correlation. The same keys are provided but with smoothed in front, indicating the test results on the smoothed intensity.

`bioxtasraw.RAWAPI.`**`validate_sample_range`**(*series, sample_range, profile_type='sub', int_type='total', q_val=None, q_range=None, rg=None, vcmw=None, vpmw=None, fast=False, settings=None, sim_test='CorMap', sim_cor='Bonferroni', sim_thresh=0.01*)

Validates whether the input data range is a trustworthy sample range or not. This is designed to work with SEC-SAXS data, but may work in other circumstances.

Parameters

- **series** (list or *bioxtasraw.SECM.SECM*) – The input series to validate the sample range for. It should either be a list of individual scattering profiles (*bioxtasraw.SASM.SASM*) or a single series object (*bioxtasraw.SECM.SECM*).
- **sample_range** (*list*) – A list defining the input sample range to be validated. The list is made up of a set of sub-ranges, each defined by an entry in the list. Each sub-range item should be a list or tuple where the first entry is the starting index of the range and the second entry is the ending index of the range. So a list like `[[0, 10], [100, 110]]` would define a sample range consisting of two sub-ranges, the first from profiles 0-10 in the series and the second from profiles 100-110 in the series.
- **profile_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – Only used if a *bioxtasraw.SECM.SECM* is provided for the series argument. Determines which type of profile to use from the series to validate the sample range. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'.
- **int_type** (*{'total', 'mean', 'q_val', 'q_range'} str, optional*) – The intensity type to use for the validation of the sample range. Total integrated intensity - 'total', mean intensity - 'mean', intensity at a particular q value - 'q_val', intensity in a given q range - 'q_range'. Use of q_val or q_range requires the corresponding parameter to be provided.
- **q_val** (*float, optional*) – If int_type is 'q_val', the q value used for the intensity is set by this parameter.
- **q_range** (*list, optional*) – This should have two entries, both floats. The first is the minimum q value of the range, the second the maximum q value of the range. If int_type is 'q_range', the q range used for the intensity is set by this parameter.
- **rg** (*list*) – A list of the Rg values corresponding to the input series data. Only required if inputing a list of profiles rather than a series object.
- **vcmw** (*list*) – A list of the volume of correlation M.W. values corresponding to the input series data. Only required if inputing a list of profiles rather than a series object.
- **vpmw** (*list*) – A list of the Porod volume M.W. values corresponding to the input series data. Only required if inputing a list of profiles rather than a series object.
- **fast** (*bool, optional*) – Whether the test should be done in fast mode or not. A fast test stops at the first failed check. In a normal test (not fast), all metrics are checked. Using a fast test is best when trying to automatically determine a sample range, something which can take many separate validation checks. A normal test is best when trying to determine what, if anything, about your selected region might be problematic.
- **settings** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, sim_test, sim_cor, and sim_threshold are overridden by the values in the settings.
- **sim_test** (*{'CorMap'} str, optional*) – Sets the type of similarity test to be used. Currently only CorMap is supported as an option. Is overridden if settings are provided.
- **sim_cor** (*{'Bonferroni', 'None'} str, optional*) – Sets the multiple testing correction to be used as part of the similarity test. Default is Bonferroni. Is overridden if settings are provided.
- **sim_thresh** (*float, optional*) – Sets the p value threshold for the similarity test. A higher value is a more strict test (range from 0-1). Is overridden if settings are provided.

Returns

- **valid** (*bool*) – If the input sample range is a valid sample range.
- **similarity_results** (*dict*) – A dictionary with the results of the similarity test. In particular, keys are: 'all_similar' - whether all profiles in the selected range are similar over the entire profile. 'low_q_similar' - whether all profiles in the selected range are similar over the low q region of the profile. 'high_q_similar' - whether all profiles in the selected range are similar over the high q region. 'max_idx' - The index of the profile used as the reference for the similarity test, corresponding to the profile with the highest intensity in the region. 'all_outliers' - Indices of the outlier profiles of the similarity test at all q. 'low_q_outliers' - Indices of the outlier profiles of the similarity test at low q. 'high_q_outliers' - Indices of the outlier profiles of the similarity test at high q.
- **param_results** (*dict*) – A dictionary with the results of the Rg and M.W. tests. In particular, keys are: 'rg_r' - the Spearman correlation coefficient of the Rg values in the region. 'rg_pval' - the p-value from the Spearman correlation test on the Rg values in the region. 'rg_valid' - Whether the range is a valid sample range based on the Rg correlation. The same keys are provided for vcmw and vpmw. Additional keys are 'param_range_valid' - Whether Rg is not calculated anywhere in the defined range. 'param_bad_frames' - The frames at which the Rg is undefined, if any. 'param_valid' - Whether all evaluation metrics (rg_valid, vcmw_valid, vpmw_valid, param_range_valid) are True.
- **svd_results** (*dict*) – A dictionary with the results of the SVD test. In particular, keys are: 'svals' - the number of significant singular vectors in the region. 'U' - the left singular vectors. 'V' - the right singular vectors. 'u_autocor' - The autocorrelation of the left singular vectors. 'v_autocor' - The autocorrelation of the right singular vectors.
- **sn_results** (*dict:*) – A dictionary with the results of the signal to noise test. In particular, keys are 'low_sn' - an array of the indices of profiles that, when averaged, lower the signal to noise of the resulting averaged profile. 'sn_valid' - Whether all profiles averaged improve signal to noise.

`bioxtasraw.RAWAPI.`**`weighted_average`**(*profiles*, *weight_by_error=True*, *weight_counter=''*, *forced=False*, *settings=None*)

Averages the input profiles into a single averaged profile, using a weighted average. Note that unlike in the RAW GUI there is no automatic testing for similarity in this average function.

Parameters

- **`profiles`** (`list`) – A list of profiles (`bioxtasraw.SASM.SASM`) to average.
- **`weight_by_error`** (`bool, optional`) – If true, weight in the average is determined by the profiles' uncertainties. If False, then the weighting is done by a counter value (such as incident intensity) specified by the weight_counter parameter. Defaults to True.
- **`weight_counter`** (`str, optional`) – If weight_by_error is False, this is the counter used to do the weighting, for example this might be incident intensity. This counter must be present in the header of all of the profiles (i.e. either in the 'counters' or 'imageHeader' dictionaries of the profiles).
- **`forced`** (`bool, optional`) – If True, RAW will attempt to average profiles even if the q vectors do not agree. Defaults to False.
- **`settings`** (`bioxtasraw.RAWSettings.RAWSettings`, optional) – RAW settings containing relevant parameters. If provided, the weight_by_error and weight_counter parameters will be overridden with the values in the settings. Default is None.

Returns **avg_profile** – The average profile.

Return type `bioxtasraw.SASM.SASM`

Raises `SASExceptions.DataNotCompatible` – If the average list contains data sets with different q vectors and not forced (or if it fails to find a solution even if forced).

Profile and IFT objects

class `bioxtasraw.SASM.`**`SASM`**(*i*, *q*, *err*, *parameters*, *q_err=None*)

Bases: `object`

Small Angle Scattering Measurement (SASM) Object. Essentially a scattering profile with q, i, and uncertainty, plus a lot of metadata.

Variables

- **q** (*numpy.array*) – The scaled q vector, without the trimming specified by *setQrange()*.
- **i** (*numpy.array*) – The scaled intensity vector, without the trimming specified by *setQrange()*.
- **err** (*numpy.array*) – The scaled error vector, without the trimming specified by *setQrange()*.
- **q_err** (*numpy.array*) – The scaled q error vector, without the trimming specified by *setQrange()*. Typically only used with SANS data.

`__init__`(*i*, *q*, *err*, *parameters*, *q_err=None*)

Constructor

Parameters

- **i** (*numpy.array*) – The intensity vector.
- **q** (*numpy.array*) – The q vector.
- **err** (*numpy.array*) – The error vector.
- **parameters** (*dict*) – A dictionary of metadata for the object. This should contain at least {'filename': filename_with_no_path}. Other reserved keys are: 'counters' : [(countername, value),...] Info from counter files 'fileHeader' : [(label, value),...] Info from the header in the loaded file

static closest(*qlist*, *q*)

A convenience function which returns the index of the nearest q point in qlist to the input q value.

Parameters

- **qlist** (*np.array*) – The q list to search in.
- **q** (*float*) – The q value to search for in the qlist.

Returns **index** – The index of the q value in qlist closest to the input q.

Return type int

copy()

Creates a copy of the SASM, without scale information, using, for example, the scaled, trimmed intensity as the raw intensity for the new SASM.

The preferred method of copying a profile is to use copy.deepcopy on the profile.

Returns **profile** – The copied profile

Return type *bioxtasraw.SASM.SASM*

extractAll()

Extracts the raw and scaled q, intensity, and error, the scale and offset values, the selected q range, and the parameters in a single dictionary.

Returns **all_data** – A dictionary with keys q_raw, i_raw, err_raw, q, i, err, scale_factor, offset_value, q_scale_factor, selected_qrange, and parameters, which correspond to those values from the SASM.

Return type dict

getAllParameters()

Returns all of the metadata parameters associated with the profile as a dictionary.

Returns **parameters** – The metadata associated with the profile.

Return type dict

getErr()

Gets the scaled, offset, trimmed error vector. Usually this is what you want to use to the get the error vector.

getI()

Gets the scaled, offset, trimmed intensity vector. Usually this is what you want to use to the get the intensity vector.

getIofQ(*qref*)

Gets the intensity at a specific q value (or the closest such value in the q vector).

Parameters **qref** (*float*) – The reference q to get the intensity at.

Returns **intensity** – The intensity at the q point nearest the provided qref.

Return type float

getIofQRange(*q1*, *q2*)

Gets the total integrated intensity in the q range from q1 to q2 (or the closest such values in the q vector).

Parameters

- **q1** (*float*) – The starting q value in the q range
- **q2** (*float*) – The ending q value in the q range.

Returns **total_intensity** – The total intensity.

Return type float

getLine()

Returns the plotted line for the profile. Only used in the RAW GUI.

Returns **line** – The plotted line.

Return type matplotlib.lines.Line2D

getMeanI()

Gets the mean intensity of the intensity vector.

Returns **mean_intensity** – The mean intensity.

Return type float

getOffset()

Returns the offset for the profile.

Returns **offset** – The offset.

Return type float

getParameter(*key*)
Gets a particular metadata parameter based on the provided key.

Parameters **key** (`str`) – A string that is a key in the parameters metadata dictionary.

Returns The parameter associated with the specified key. If the key is not in the parameter dictionary, None is returned.

Return type parameter

getQ()
Gets the scaled, offset, trimmed q vector. Usually this is what you want to use to the get the q vector.

getQErr()
Gets the scaled, offset, trimmed q error vector. Usually this is what you want to use to the get the q error vector. Q errors are usually only available with SANS data. Returns None if no q error has been defined.

getQrange()
Returns the currently selected q range as described in `setQrange()`.

Returns **q_range** – A tuple with 2 indices, the start and end of the selected q range, such that q[start:end] returns the desired q range.

Return type tuple

getRawErr()
Gets the raw error vector, without scaling or offset from `scale()` and `offset()` and without trimming based on `setQrange()`.

Returns **err_raw** – The raw error vector.

Return type numpy.array

getRawI()
Gets the raw intensity vector, without scaling or offset from `scale()` and `offset()` and without trimming based on `setQrange()`.

Returns **i_raw** – The raw intensity vector.

Return type numpy.array

getRawQ()
Gets the raw q vector, without scaling based on the `scaleQ()` and without trimming based on `setQrange()`.

Returns **q_raw** – The raw q vector.

Return type numpy.array

getRawQErr()
Gets the raw q error vector, without scaling or offset from `scale()` and `offset()` and without trimming based on `setQrange()`.

Returns **q_err_raw** – The raw error vector.

Return type numpy.array

getScale()
Returns the scale factor for the profile.

Returns **scale** – The scale factor.

Return type float

getTotalI()
Gets the total integrated intensity of the intensity vector.

Returns **total_intensity** – The total intensity.

Return type float

`offset`(*offset_value*)
Applies an absolute offset to the profile intensity. For example, if the offset is 1, then all the the intensities in the profile are increased by 1. Offset supersedes the existing offset, so if the current offset is 1, and an offset_value of 2 is provided, the resulting offset is 2 (not 3).

Parameters **`offset_value`** (*`float`*) – The offset to be applied to the profile intensity.

`offsetRawIntensity`(*offset*)
Offsets the raw intensity and error values. These are the intensity and error values without any offset applied. The raw offset factor is not tracked, so this cannot easily be undone, unlike the :func:scaleQ function.

Parameters **`offset`** (*`float`*) – The offset to be applied to the raw profile intensity and error values.

`removeParameter`(*key*)
Removes a particular metadata parameter based on the provided key.

Parameters **`key`** (*`str`*) – A string that is a key in the parameters metadata dictionary.

`removeZingers`(*start_idx=0*, *window_length=10*, *stds=4.0*)
Removes spikes (zingers) from radially averaged data based on smoothing of the intensity profile.

Parameters

- **`start_idx`** (*`int`*) – The initial index in the intensity to start the dezingering process.
- **`window_length`** (*`int`*) – The size of the window used to search for an replace spikes, as the number of intensity points.
- **`stds`** (*`The standard deviation threshold used to detect spikes.`*) –

`reset`()
Removes scale and offset values from the intensity, uncertainty, and q.

`scale`(*scale_factor*)
Applies an absolute scale to the profile intensity. The scale factor supersedes the existing scale factor. For example, suppose the scale factor is currently 2. If a scale of 4 is provided, the resulting scale factor is 4 (not 8). Scale factors are always positive.

Parameters **`scale_factor`** (*`float`*) – The scale factor to be applied to the the profile intensity and uncertainty.

`scaleQ`(*q_scale_factor*)
Scales the profile q values by a factor. The scale factor supersedes the existing scale factor. For example, suppose the scale factor is currently 2. If a scale of 4 is provided, the resulting scale factor if 4 (not 8). Useful for converting between 1/A and 1/nm, for example.

Parameters **`q_scale_factor`** (*`float`*) – The scale factor to be applied to the profile q values.

`scaleRawIntensity`(*scale*)
Scales the raw intensity and error values. These are the intensity and error values without any scale applied. The raw scale factor is not tracked, so this cannot easily be undone, unlike the :func:scaleQ function.

Parameters **`scale`** (*`float`*) – The scale factor to be applied to the raw profile intensity and error values.

scaleRawQ(*scale_factor*)
Scales the raw q values. These are the q values without any scale applied. The raw q scale factor is not tracked, so this cannot easily be undone, unlike the :func:scaleQ function.

Parameters **scale_factor** (`float`) – The scale factor to be applied to the raw profile q values.

scaleRelative(*relscale*)
Applies a relative scale to the profile intensity. If the scale factor is currently 1, then this is the same as `scale()`. Otherwise, this scales relative to the current scale factor. For example, suppose the scale factor is currently 2. If a relative scale of 2 is provided, the resulting scale factor if 4. Scale factors are always positive.

Parameters **relscale** (`float`) – The relative scale factor to be applied to the the profile intensity and uncertainty.

scaleRelativeQ(*relscale*)
Applies a relative scale to the profile q. If the scale factor is currently 1, then this is the same as `scale()`. Otherwise, this scales relative to the current scale factor. For example, suppose the scale factor is currently 2. If a relative scale of 2 is provided, the resulting scale factor if 4.

Parameters **relscale** (`float`) – The relative scale factor to be applied to the the profile intensity and uncertainty.

setAllParameters(*new_parameters*)
Sets the parameters dictionary, which contains the profile metadata, to the new input value.

Parameters **new_parameters** (`dict`) – A dictionary containing the new parameters.

setParameter(*key*, *value*)
Sets a particular metadata parameter based on the provided key and value.

Parameters

- **key** (`str`) – The name of the new bit of metadata.
- **value** (`object`) – The value of the new bit of metadata. Could be anything that is an acceptable value for a dictionary.

setQrange(*qrange*)
Sets the q range used for the profile. Useful for trimming leading or trailing values of the q profile that are not useful data.

Parameters **qrange** (`tuple or list`) – A tuple or list with two items. The first item is the starting index of the q vector to be used, the second item is the ending index of the q vector to be used, such that q[start:end] returns the desired q range.

setRawErr(*new_raw_err*)
Sets the raw error vector. Will overwrite whatever error vector is already in the object! Typically only used during calibration.

Parameters **new_raw_err** (`numpy.array`) – The new error vector.

setRawI(*new_raw_i*)
Sets the raw q vector. Will overwrite whatever q vector is already in the object! Typically only used during calibration.

Parameters **new_raw_i** (`numpy.array`) – The new intensity vector.

setRawQ(*new_raw_q*)
Sets the raw intensity vector. Will overwrite whatever intensity vector is already in the object! Typically only used during calibration.

Parameters **`new_raw_q`** (*`numpy.array`*) – The new q vector.

`setRawQErr`(*new_raw_q_err*)

Sets the raw q error vector. Will overwrite whatever error vector is already in the object! Typically only used during calibration. Q errors are typically only found in SANS data, and currently are only carried and written out with the data set, they are not used in any processing.

Parameters **`new_raw_err`** (*`numpy.array`*) – The new error vector.

`setScaleValues`(*scale_factor*, *offset_value*, *q_scale_factor*)

A convenience method that lets you set the scale offset, and q scale values all at once.

Parameters

- **`scale_factor`** (*`float`*) – The scale factor to be applied to the the profile intensity and uncertainty.
- **`offset_value`** (*`float`*) – The offset to be applied to the profile intensity.
- **`q_scale_factor`** (*`float`*) – The scale factor to be applied to the profile q values.

class `bioxtasraw.SASM.`**`IFTM`**(*p*, *r*, *err*, *i_orig*, *q_orig*, *err_orig*, *i_fit*, *parameters*, *i_extrap=[]*, *q_extrap=[]*)

Bases: `object`

Inverse Fourier transform measurement (IFTM) object. Contains the P(r), r and error vectors, as well as the original data, the fit of the P(r) to the data, and all associated metadata.

Variables

- **`r`** (*`numpy.array`*) – The r vector of the P(r) function.
- **`p`** (*`numpy.array`*) – The values of the P(r) function.
- **`err`** (*`numpy.array`*) – The errors of the P(r) function.
- **`q_orig`** (*`numpy.array`*) – The q vector of the input data.
- **`i_orig`** (*`numpy.array`*) – The intensity vector of the input data.
- **`err_orig`** (*`numpy.array`*) – The error vector of the input data.
- **`i_fit`** (*`numpy.array`*) – The intensity vector of the fit to the input data.
- **`q_extrap`** (*`numpy.array`*) – The q vector of the input data extrapolated to q=0.
- **`i_extrap`** (*`numpy.array`*) – The intensity vector of the fit to the input data extrapolated to q=0.

`__init__`(*p*, *r*, *err*, *i_orig*, *q_orig*, *err_orig*, *i_fit*, *parameters*, *i_extrap=[]*, *q_extrap=[]*)

Constructor

Parameters

- **`p`** (*`numpy.array`*) – The input P(r) values.
- **`r`** (*`numpy.array`*) – The input r values for the P(r) function.
- **`err`** (*`numpy.array`*) – The input error values for the P(r) function.
- **`i_orig`** (*`numpy.array`*) – The intensity values of the data used to do the IFT.
- **`q_orig`** (*`numpy.array`*) – The q values of the data used to do the IFT.
- **`err_orig`** (*`numpy.array`*) – The error values of the data used to do the IFT.
- **`i_fit`** (*`numpy.array`*) – The intensity values of the fit of the P(r) function to the data.

- **parameters** (*dict*) – A dictionary of the metadata. Should ontain at least {'filename': filename_with_no_path}
- **i_extrap** (*numpy.array, optional*) – The intensity values of the fit of the P(r) function to the data extrapolated to q=0. If not provided, an empty array is used.
- **q_extrap** (*numpy.array, optional*) – The q values of the input data extrapolated to q=0. If not provided, an empty array is used.

copy()

Creates a copy of the IFT.

Returns **ift** – The copied IFTM

Return type *bioxtasraw.SASM.IFTM*

extractAll()

Extracts the raw and scaled q, intensity, and error, the scale and offset values, the selected q range, and the parameters in a single dictionary.

Returns **all_data** – A dictionary with keys r_raw, p_raw, err_raw, i_orig_raw, q_orig_raw, err_orig_raw, i_fit_raw, i_extrap_raw, q_extrap_raw, and parameters, which correspond to those values from the IFTM.

Return type dict

getAllParameters()

Returns all of the metadata parameters associated with the IFT as a dictionary.

Returns **parameters** – The metadata associated with the IFT.

Return type dict

getLine()

Returns the plotted line for the P(r) function. Only used in the RAW GUI.

Returns **line** – The plotted line.

Return type matplotlib.lines.Line2D

getOffset()

Returns the offset for the P(r) function.

Returns **offset** – The offset.

Return type float

getParameter(*key*)

Gets a particular metadata parameter based on the provided key.

Parameters **key** (*str*) – A string that is a key in the parameters metadata dictionary.

Returns The parameter associated with the specified key. If the key is not in the parameter dictionary, None is returned.

Return type parameter

getScale()

Returns the scale factor for the P(r) function.

Returns **scale** – The scale factor.

Return type float

setAllParameters(*new_parameters*)

Sets the parameters dictionary, which contains the IFT metadata, to the new input value.

Parameters **new_parameters** (*dict*) – A dictionary containing the new parameters.

setParameter (*key*, *value*)

Sets a particular metadata parameter based on the provided key and value.

Parameters

- **key** (*str*) – The name of the new bit of metadata.
- **value** (*object*) – The value of the new bit of metadata. Could be anything that is an acceptable value for a dictionary.

Series objects

class bioxtasraw.SECM.**SECM** (*file_list*, *sasm_list*, *frame_list*, *parameters*, *settings*)

Bases: `object`

Series measurement object. Was originally a SEC-SAXS measurement (SECM) object. Contains all the information about a series, including the individual scattering profiles, subtracted scattering profiles, baseline corrected scattering profiles, calculated parameter values such as Rg, total and mean intensity for each profile, information about selected buffer ranges and baseline correction ranges, etc.

Variables

- **qref** (*float*) – The reference q value specified by `I()`.
- **qrange** (*tuple*) – The q range specified by `calc_qrange_I()`.
- **buffer_range** (*list*) – A list defining the set buffer range. The list is made up of a set of sub-ranges, each defined by an entry in the list. Each sub-range item should be a list or tuple where the first entry is the starting index of the range and the second entry is the ending index of the range. So a list like `[[0, 10], [100, 110]]` would define a buffer range consisting of two sub-ranges, the first from profiles 0-10 in the series and the second from profiles 100-110 in the series.
- **window_size** (*int*) – The size of the average window used when calculating parameters such as Rg.
- **mol_type** (*str*) – The macromolecule type used when calculating the Vc M.W.
- **mol_density** (*float*) – The macromolecular density used when calculating the Vp M.W.
- **already_subtracted** (*bool*) – Whether the initial input profiles represent an already subtracted series or not.
- **average_buffer_sasm** (`bioxtasraw.SASM.SASM`) – The average buffer profile from the buffer_range.
- **baseline_start_range** (*tuple*) – A tuple where the first item is the start of the baseline start range and the second item is the end of the baseline start range.
- **baseline_end_range** (*tuple*) – A tuple where the first item is the start of the baseline end range and the second item is the end of the baseline end range.
- **baseline_corr** (*list*) – A list of the baseline correction applied. Each item is a `bioxtasraw.SASM.SASM`, and there is one for every baseline corrected profile. The intensity is the value subtracted from the starting intensity of the corresponding profile to achieve the baseline corrected intensity.
- **baseline_type** (*str*) – The baseline type.
- **baseline_extrap** (*bool*) – Whether the baseline was extrapolated to all profiles.

- **baseline_fit_results** (*list*) – Only contains items if a linear baseline correction is done. In that case, each item is the linear fit results a, b, and corresponding covariances for a given q value. There is one item per q value of the input profiles.
- **sample_range** (*list*) – A list defining the set sample range. The list is made up of a set of sub-ranges, each defined by an entry in the list. Each sub-range item should be a list or tuple where the first entry is the starting index of the range and the second entry is the ending index of the range. So a list like `[[0, 10], [100, 110]]` would define a sample range consisting of two sub-ranges, the first from profiles 0-10 in the series and the second from profiles 100-110 in the series.
- **use_subtracted_sasm** (*list*) – A list of bools defining which frames should be used when calculating Rg, MW. This is based on how the total intensity in the frame compares to the average intensity of the buffer region. If the intensity ratio of a profile divided by the average buffer profile is greater than the threshold defined in the RAW settings, that frame is used to calculate Rg and MW, otherwise it is not. This is simply to increase speed of calculation by not trying to calculate these parameters for buffer data.
- **use_baseline_subtracted_sasm** (*list*) – As use_subtracted_sasm, but for the baseline corrected profiles (if available).

I(*qref*)

Sets the reference q value and returns the intensity of each profile at the specified q value (or the closest such value in each profile).

Parameters **qref** (*float*) – The reference q to get the intensity at.

Returns **intensity** – The intensity of each profile at the given q value.

Return type numpy.array

__init__(*file_list*, *sasm_list*, *frame_list*, *parameters*, *settings*)

Constructor

Parameters

- **file_list** (*list*) – A list of strings corresponding to the filenames of each input sasm.
- **sasm_list** (*list*) – A list of bioxtasraw.SASM.SASM objects, which are the individual scattering profiles that make up the series.
- **frame_list** (*list*) – A list of the frame numbers of each item in the sasm_list. Usually just range(len(sasm_list))
- **parameters** (*dict*) – A dictionary of metadata for the object. This should contain at least {'filename': filename_with_no_path}.
- **settings** (`bioxtasraw.RAWSettings.RawGuiSettings`) – RAW settings. Used to try compute the time associated with each input profile.

acquireSemaphore()

Acquires a processing semaphore. Useful for multi-threading operations on the series.

append(*filename_list*, *sasm_list*, *frame_list*)

Appends new data to the series. Used when operating in an 'online' mode during active data collection.

Parameters

- **filename_list** (*list*) – A list of strings corresponding to the filenames of each input sasm.
- **sasm_list** (*list*) – A list of bioxtasraw.SASM.SASM objects, which are the individual scattering profiles that make up the series.

- **frame_list** (*list*) – A list of the frame numbers of each item in the sasm_list. Usually just range(len(sasm_list))

appendBCSubtractedSASMs (*sub_sasm_list*, *use_sasm_list*, *window_size*)
Appends new baseline corrected subtracted data to the series. Used when operating in an 'online' mode during active data collection.

Parameters

- **sub_sasm_list** (*list*) – A list of the baseline corrected subtracted profiles.
- **use_sasm_list** (*list*) – A list of bools indicating whether or not the baseline corrected subtracted profiles should be used when calculating parameters such as Rg.
- **window_size** (*int*) – The averaging window size used to calculate the parameters.

appendCalcValues (*rg*, *rger*, *i0*, *i0er*, *vcmw*, *vcmwer*, *vpmw*, *first_frame*, *window_size*)
Appends new calculated parameter data to the series. Used when operating in an 'online' mode during active data collection.

Parameters

- **rg** (*np.array*) – An array of the new Rg values.
- **rger** (*np.array*) – An array of the new uncertainty in the Rg values.
- **i0** (*np.array*) – An array of the new I(0) values.
- **i0er** (*np.array*) – An array of the new uncertainty in the I(0) values.
- **vcmw** (*np.array*) – An array of the new volume of correlation M.W. values.
- **vcmwer** (*np.array*) – An array of the new uncertainty in the Vc M.W. values.
- **vpmw** (*np.array*) – An array of the new corrected Porod volume M.W. values.
- **first_frame** (*int*) – The first frame index of the new data.
- **window_size** (*int*) – The averaging window size used to calculate the parameters.

appendSubtractedSASMs (*sub_sasm_list*, *use_sasm_list*, *window_size*)
Appends new subtracted data to the series. Used when operating in an 'online' mode during active data collection.

Parameters

- **sub_sasm_list** (*list*) – A list of the subtracted profiles.
- **use_sasm_list** (*list*) – A list of bools indicating whether or not the subtracted profiles should be used when calculating parameters such as Rg.
- **window_size** (*int*) – The averaging window size used to calculate the parameters.

averageFrames (*range_list*, *series_type*, *sim_test*, *sim_thresh*, *sim_cor*, *forced=False*)
Creates an average profile from the frame ranges defined in the range_list.

Parameters

- **range_list** (*list*) – A list defining the input range to be averaged. The list is made up of a set of sub-ranges, each defined by an entry in the list. Each sub-range item should be a list or tuple where the first entry is the starting index of the range and the second entry is the ending index of the range. So a list like `[[0, 10], [100, 110]]` would define a buffer range consisting of two sub-ranges, the first from profiles 0-10 in the series and the second from profiles 100-110 in the series.

- **`series_type`** (*`{'unsub', 'sub', 'baseline'} str, optional`*) – Determines which type of profile to average. Unsubtracted profiles - 'unsub', subtracted profiles - 'sub', baseline corrected profiles - 'baseline'
- **`sim_test`** (*`{'CorMap'} str, optional`*) – Sets the type of similarity test to be used. Currently only CorMap is supported as an option.
- **`sim_thresh`** (*`float, optional`*) – Sets the p value threshold for the similarity test. A higher value is a more strict test (range from 0-1).
- **`sim_cor`** (*`{'Bonferroni', 'None'} str, optional`*) – Sets the multiple testing correction to be used as part of the similarity test. Default is Bonferroni.
- **`forced`** (*`bool, optional`*) – If True, RAW will attempt to average profiles even if the q vectors do not agree or the profiles are not similar. Defaults to False.

Returns

- **average_profile** (*bioxtasraw.SASM.SASM*) – The average profile. If averaging fails, returns None.
- **success** (*bool*) – Whether the average succeeded.
- **error** (*tuple*) – A tuple of strings. Both are empty if success. If the average failed, it contains either 'sim' or 'q_vector' to indicate the issue is either the profiles are not all similar or the q vectors do not all match. The second string is the names of the profiles where the check failed.

`calc_qrange_I`(*qrange*)

Sets the reference q range and returns the intensity of each profile in the specified q range (or the closest such q values in each profile).

Parameters **`qrange`** (*`tuple or list`*) – A tuple or list with two items. The first item is the starting index of the q vector to be used, the second item is the ending index of the q vector to be used, such that q[start:end] returns the desired q range.

Returns **intensity** – The total intensity of each profile in the given q range.

Return type numpy.array

`extractAll`()

Extracts all of the series data and returns it as a dict. Useful for pickling the series.

Returns **all_data** – A dictionary with all the series data.

Return type dict

`getAllParameters`()

Returns all of the metadata parameters associated with the series as a dictionary.

Returns **parameters** – The metadata associated with the series.

Return type dict

`getAllSASMs`(*int_type='unsub'*)

Gets the all profiles in the series.

Parameters **`int_type`** (*`{'unsub', 'sub', 'baseline'} str, optional`*) – The type of profile to get. Either 'unsub' - unsubtracted, 'sub' - subtracted, or 'baseline' - baseline corrected.

Returns **profiles** – A list of bioxtasraw.SASM.SASM profiles corresponding to the selected type.

Return type list

getBCSubQrange()

Returns the currently selected q range for each baseline corrected profile in the series as described in *setBCSubQrange()*.

Returns **q_range** – A tuple with 2 indices, the start and end of the selected q range, such that q[start:end] returns the desired q range.

Return type tuple

getCalcLine()

Returns the plotted line for the calculated values of the series, such as Rg. Only used in the RAW GUI.

Returns **line** – The plotted line.

Return type matplotlib.lines.Line2D

getFrames()

Returns the list of frames suitable for plotting the series intensity. Note that this may be different from the input frame list.

Returns **frame_list** – An array that starts from 0 and runs to the length of the series.

Return type numpy.array

getI0()

Returns the I(0) and uncertainty in I(0) values for each profile in the series. If the I(0) value is not available for a given profile, then a -1 is returned.

Returns

- **i0** (*numpy.array*) – The I(0) values for each profile in the series.
- **i0er** (*numpy.array*) – The uncertainty in the I(0) values for each profile in the series.

getIntI(*int_type='unsub'*)

Returns the total integrated intensity of each profile in the series.

Parameters **int_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – The type of profile to get. Either 'unsub' - unsubtracted, 'sub' - subtracted, or 'baseline' - baseline corrected.

Returns **intensity** – The total integrated intensity of each profile for the selected profile type.

Return type numpy.array

getIofQ(*int_type='unsub'*)

Returns the intensity of each profile at the specified q value (or the closest such value in each profile). Use this instead of *I()* if you don't want to recalculate the intensity at the given reference q value.

Parameters **int_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – The type of profile to get. Either 'unsub' - unsubtracted, 'sub' - subtracted, or 'baseline' - baseline corrected.

Returns **intensity** – The intensity of each profile at the given q value for the selected profile type.

Return type numpy.array

getIofQRange(*int_type='unsub'*)

Returns the intensity of each profile in the specified q range (or the closest such q values in each profile). Use this instead of *calc_qrange_I()* if you don't want to recalculate the intensity for the given reference q range.

Parameters **`int_type`** (*{'unsub', 'sub', 'baseline'} str, optional*) – The type of profile to get. Either 'unsub' - unsubtracted, 'sub' - subtracted, or 'baseline' - baseline corrected.

Returns **intensity** – The total intensity of each profile in the given q range for the selected profile type.

Return type numpy.array

`getLine`()

Returns the plotted line for the series. Only used in the RAW GUI.

Returns **line** – The plotted line.

Return type matplotlib.lines.Line2D

`getMeanI`(*int_type='unsub'*)

Returns the mean intensity of each profile in the series.

Parameters **`int_type`** (*{'unsub', 'sub', 'baseline'} str, optional*) – The type of profile to get. Either 'unsub' - unsubtracted, 'sub' - subtracted, or 'baseline' - baseline corrected.

Returns **intensity** – The mean intensity of each profile for the selected profile type.

Return type numpy.array

`getOffset`()

Returns the offset for the series.

Returns **offset** – The offset.

Return type float

`getParameter`(*key*)

Gets a particular metadata parameter based on the provided key.

Parameters **`key`** (*str*) – A string that is a key in the parameters metadata dictionary.

Returns The parameter associated with the specified key. If the key is not in the parameter dictionary, None is returned.

Return type parameter

`getQrange`()

Returns the currently selected q range for each profile in the series as described in `setQrange()`.

Returns **q_range** – A tuple with 2 indices, the start and end of the selected q range, such that q[start:end] returns the desired q range.

Return type tuple

`getRg`()

Returns the Rg and uncertainty in Rg values for each profile in the series. If the Rg value is not available for a given profile, then a -1 is returned.

Returns

- **rg** (*numpy.array*) – The Rg values for each profile in the series.
- **rger** (*numpy.array*) – The uncertainty in the Rg values for each profile in the series.

`getSASM`(*index=0*, *int_type='unsub'*)

Gets the profile at a given frame number.

Parameters

- **index** (*int, optional.*) – The index of the profile to return. Defaults to the first profile.
- **int_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – The type of profile to get. Either 'unsub' - unsubtracted, 'sub' - subtracted, or 'baseline' - baseline corrected.

Returns **profile** – A scattering profile.

Return type *bioxtasraw.SASM.SASM*

getSASMList(*initial_frame*, *final_frame*, *int_type='unsub'*)
Gets the specified profiles of the series in a given frame range.

Parameters

- **initial_frame** (*int*) – The starting frame of profiles in the series to get.
- **final_frame** (*int*) – The final frame of profiles in the series to get.
- **int_type** (*{'unsub', 'sub', 'baseline'} str, optional*) – The type of profile to get. Either 'unsub' - unsubtracted, 'sub' - subtracted, or 'baseline' - baseline corrected.

Returns **profiles** – A list of bioxtasraw.SASM.SASM profiles corresponding to the selected type and frame range.

Return type list

Raises `SASExceptions.DataNotCompatible` – If the frame range is invalid or requesting a profile type not available in the series (such as asking for baseline corrected profiles when no baseline correction has been done).

getScale()
Returns the scale factor for the series.

Returns **scale** – The scale factor.

Return type float

getSubQrange()
Returns the currently selected q range for each subtracted profile in the series as described in `setSubQrange()`.

Returns **q_range** – A tuple with 2 indices, the start and end of the selected q range, such that q[start:end] returns the desired q range.

Return type tuple

getTime()
Gets the time associated with each scattering profile, if available. Returns an arary of -1 if not available.

Returns **time** – A array of the collection time of each profile, relative to the first.

Return type numpy.array

getVcMW()
Returns the volume of correlation M.W. and uncertainty values for each profile in the series. If the M.W. value is not available for a given profile, then a -1 is returned.

Returns

- **vcmw** (*numpy.array*) – The Vc M.W. values for each profile in the series.
- **vcmwer** (*numpy.array*) – The uncertainty in the Vc M.W. values for each profile in the series.

getVpMW()
Returns the corrected Porod volume M.W. and uncertainty values for each profile in the series. If the M.W. value is not available for a given profile, then a -1 is returned. Currently uncertainty is not available, so an array of all -1 is returned.

Returns

- **vcmw** (*numpy.array*) – The Vp M.W. values for each profile in the series.
- **vcmwer** (*numpy.array*) – The uncertainty in the Vp M.W. values for each profile in the series. Currently all -1, as this value is not calculated.

offset(*offset_value*)
Applies an absolute offset to the profile intensity. For example, if the offset is 1, then all the the intensities in the profile are increased by 1. Offset supersedes the existing offset, so if the current offset is 1, and an offset_value of 2 is provided, the resulting offset is 2 (not 3).

Scale factors are applied to the individual profiles with the series, not to the overall intensity, so if the profiles are then retrieved from the series they will have the same scale factor as was applied to the series.

Parameters **offset_value** (`float`) – The offset to be applied to the profile intensity.

releaseSemaphore()
Releases a processing semaphore. Useful for multi-threading operations on the series.

reset()
Removes scale and offset values from the series.

scale(*scale_factor*)
Applies an absolute scale to the series intensity. The scale factor supersedes the existing scale factor. For example, suppose the scale factor is currently 2. If a scale of 4 is provided, the resulting scale factor is 4 (not 8). Scale factors are always positive.

Scale factors are applied to the individual profiles with the series, not to the overall intensity, so if the profiles are then retrieved from the series they will have the same scale factor as was applied to the series.

Parameters **scale_factor** (`float`) – The scale factor to be applied to the the profile intensity and uncertainty.

scaleRelative(*relscale*)
Applies a relative scale to the series intensity. If the scale factor is currently 1, then this is the same as `scale()`. Otherwise, this scales relative to the current scale factor. For example, suppose the scale factor is currently 2. If a relative scale of 2 is provided, the resulting scale factor if 4. Scale factors are always positive.

Scale factors are applied to the individual profiles with the series, not to the overall intensity, so if the profiles are then retrieved from the series they will have the same scale factor as was applied to the series.

Parameters **relscale** (`float`) – The relative scale factor to be applied to the the profile intensity and uncertainty.

setAllParameters(*new_parameters*)
Sets the parameters dictionary, which contains the series metadata, to the new input value.

Parameters **new_parameters** (`dict`) – A dictionary containing the new parameters.

setBCSubQrange(*n_min*, *n_max*)
Sets the q range used for each baseline corrected profile in the series. Useful for trimming leading or trailing values of the q profile that are not useful data.

Parameters

- **n_min** (`int`) – The starting index of the q vector to be used.

- **n_max** (*int*) – The ending index of the q vector to be used, such that q[start:end] returns the desired q range.

setBCSubtractedSASMs (*sub_sasm_list*, *use_sub_sasm*)

Sets the baseline corrected subtracted profiles for the series.

Parameters

- **sub_sasm_list** (*list*) – A list of the baseline corrected subtracted profiles.
- **use_sub_sasm** (*list*) – A list of bools indicating whether or not the profiles should be used when calculating parameters such as Rg.

setCalcValues (*rg*, *rger*, *i0*, *i0er*, *vcmw*, *vcmwer*, *vpmw*)

Sets the value of the calculated values for the series. If a value is not available for a given profile than a -1 value should be provided.

Parameters

- **rg** (*np.array*) – An array of the Rg values for each profile in the series.
- **rger** (*np.array*) – An array of the uncertainty in the Rg values for each profile in the series.
- **i0** (*np.array*) – An array of the I(0) values for each profile in the series.
- **i0er** (*np.array*) – An array of the uncertainty in the I(0) values for each profile in the series.
- **vcmw** (*np.array*) – An array of the volume of correlation M.W. values for each profile in the series.
- **vcmwer** (*np.array*) – An array of the uncertainty in the volume of correlation M.W. values for each profile in the series.
- **vpmw** (*np.array*) – An array of the corrected Porod volume M.W. values for each profile in the series.

setParameter (*key*, *value*)

Sets a particular metadata parameter based on the provided key and value.

Parameters

- **key** (*str*) – The name of the new bit of metadata.
- **value** (*object*) – The value of the new bit of metadata. Could be anything that is an acceptable value for a dictionary.

setQrange (*n_min*, *n_max*)

Sets the q range used for each profile in the series. Useful for trimming leading or trailing values of the q profile that are not useful data.

Parameters

- **n_min** (*int*) – The starting index of the q vector to be used.
- **n_max** (*int*) – The ending index of the q vector to be used, such that q[start:end] returns the desired q range.

setScaleValues (*scale_factor*, *offset_value*, *frame_scale_factor*)

A convenience method that lets you set the scale offset, and frame scale values all at once.

Note: Frame scale factor is currently not used.

Parameters

- **scale_factor** (*float*) – The scale factor to be applied to the series intensity.
- **offset_value** (*float*) – The offset to be applied to the series intensity.
- **frame_scale_factor** (*float*) – The scale factor to be applied to the series frame values.

setSubQrange (*n_min*, *n_max*)
Sets the q range used for each subtracted profile in the series. Useful for trimming leading or trailing values of the q profile that are not useful data.

Parameters

- **n_min** (*int*) – The starting index of the q vector to be used.
- **n_max** (*int*) – The ending index of the q vector to be used, such that q[start:end] returns the desired q range.

setSubtractedSASMs (*sub_sasm_list*, *use_sub_sasm*)
Sets the subtracted profiles for the series.

Parameters

- **sub_sasm_list** (*list*) – A list of the subtracted profiles.
- **use_sub_sasm** (*list*) – A list of bools indicating whether or not the subtracted profiles should be used when calculating parameters such as Rg.

subtractAllSASMs (*buffer_sasm*, *int_type*, *threshold*, *qref=None*, *qrange=None*)
Subtracts the input buffer profile from all of the profiles in the series to generate subtracted profiles. Does not save the subtracted profiles to the series.

Parameters

- **buffer_sasm** (bioxtasraw.SASM.SASM) – The buffer profile to be subtracted.
- **int_type** (*{'total', 'mean', 'q_val', 'q_range'} str, optional*) – The intensity type to use when setting the buffer range. Total integrated intensity - 'total', mean intensity - 'mean', intensity at a particular q value - 'q_val', intensity in a given q range - 'q_range'. Use of q_val or q_range requires the corresponding parameter to be provided.
- **threshold** (*float*) – If the ratio of the scattering profile intensity to the average buffer intensity is greater than this threshold, the use_subtracted_sasm flag for that profile will be set to true. This flag can later be used for calculating Rg, M.W., etc, to determine which profiles to attempt the calculation for.
- **q_val** (*float, optional*) – If int_type is 'q_val', the q value used for the intensity is set by this parameter.
- **q_range** (*list, optional*) – This should have two entries, both floats. The first is the minimum q value of the range, the second the maximum q value of the range. If int_type is 'q_range', the q range used for the intensity is set by this parameter.

Returns

- **subtracted_sasms** (*list*) – A list of the subtracted profiles (bioxtasraw.SASM.SASM).
- **use_subtracted_sasms** (*list*) – A list of bool values corresponding to the subtracted profile. If True, the corresponding profile has a intensity ratio above the buffer greater than the threshold, and so should be used when calculating Rg, M.W., and other parameters.

static subtractSASMs (*buffer_sasm*, *sasms*, *int_type*, *threshold*, *qref=None*, *qrange=None*)
Subtracts the input buffer profile from the input profiles to generate subtracted profiles.

Parameters

- **buffer_sasm** (bioxtasraw.SASM.SASM) – The buffer profile to be subtracted.
- **sasms** (*list*) – A list of profiles to subtract the buffer from.
- **int_type** (*{'total', 'mean', 'q_val', 'q_range'} str, optional*) – The intensity type to use when setting the buffer range. Total integrated intensity - 'total', mean intensity - 'mean', intensity at a particular q value - 'q_val', intensity in a given q range - 'q_range'. Use of q_val or q_range requires the corresponding parameter to be provided.
- **threshold** (*float*) – If the ratio of the scattering profile intensity to the average buffer intensity is greater than this threshold, the use_subtracted_sasm flag for that profile will be set to true. This flag can later be used for calculating Rg, M.W., etc, to determine which profiles to attempt the calculation for.
- **q_val** (*float, optional*) – If int_type is 'q_val', the q value used for the intensity is set by this parameter.
- **q_range** (*list, optional*) – This should have two entries, both floats. The first is the minimum q value of the range, the second the maximum q value of the range. If int_type is 'q_range', the q range used for the intensity is set by this parameter.

Returns

- **subtracted_sasms** (*list*) – A list of the subtracted profiles (bioxtasraw.SASM.SASM).
- **use_subtracted_sasms** (*list*) – A list of bool values corresponding to the subtracted profile. If True, the corresponding profile has a intensity ratio above the buffer greater than the threshold, and so should be used when calculating Rg, M.W., and other parameters.

Settings

class bioxtasraw.RAWSettings.**RawGuiSettings**(*settings=None*)

Bases: object

Essentially just a fancy wrapper for a big dictionary. It contains pretty much all of RAW's settings, both for the GUI and for computation.

__init__(*settings=None*)

Constructor.

Parameters settings (*dict, optional*) – A dictionary with RAW settings. If not provided, then the default values of the settings are used.

findParamById(*param_id*)

Given a particular setting id, finds the name (key) associated with that id. Note that this will only work in a GUI setting.

Parameters param_id (*int*) – The setting's id.

Returns key – The setting name.

Return type str

get(*key*)

Gets the setting value for the input key.

Parameters key (*str*) – The setting name to get the value of.

Returns setting – The setting value, which can be anything that can be included in a dictionary.

Return type object

`getAllParams`()

Gets the entire settings dictionary.

Returns **settings** – All of the settings.

Return type dict

`getId`(*key*)

Gets the Id associated with the setting. In the RAW GUI, this is a unique wx ID that can be used in windows that contain the setting value.

Parameters **`key`** (*`str`*) – The setting name to get the value of.

Returns **id** – The setting id.

Return type int

`getIdAndType`(*key*)

Gets the id and type of the setting.

Parameters **`key`** (*`str`*) – The setting name to get the value of.

Returns

- **id** (*int*) – The setting id.
- **type** (*str*) – The setting type. Can be: text, bool, choice, float, int.

`getType`(*key*)

Gets the type of the setting, useful when auto-creating a GUI based on the settings.

Parameters **`key`** (*`str`*) – The setting name to get the value of.

Returns **type** – The setting type. Can be: text, bool, choice, float, int.

Return type str

`set`(*key*, *value*)

Sets the setting value for the input key to the input value.

Parameters

- **`key`** (*`str`*) – The setting name to get the value of.
- **`value`** (*`object`*) – The new value of the setting.

5.8 Manual

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

5.8.1 Introduction to RAW and this documentation

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

RAW

BioXTAS RAW is a program for analysis of Small-Angle X-ray Scattering (SAXS) data. The software enables: creation of 1D scattering profiles from 2D detector images, standard data operations such as averaging and subtraction, analysis of radius of gyration (Rg) and molecular weight, and advanced processing using GNOM, DAMMIF, and AMBIMETER (requires ATSAS installation). It also allows easy processing of inline SEC-SAXS data.

RAW is written in python (mostly) and C++ (a few small bits for speed). It is open source and free for anyone to use. If you do use RAW, we ask that you cite the following paper if you publish or present your results:

BioXTAS RAW, a software program for high-throughput automated small-angle X-ray scattering data reduction and preliminary analysis, J. Appl. Cryst. (2009). 42, 959-964.

Some of the features of RAW include:

- Calibrate, mask, radially integrate, and normalize 2D images to make 1D scattering profiles
- Average, subtract, merge, rebin, and interpolate scattering profiles
- Easily process in-line SEC-SAXS data
- Calculate radius of gyration (Rg) and I(0) via Guinier fit
- Calculate the molecular weight via I(0) comparison to standards, absolute calibration, correlation volume, and corrected Porod volume
- Run GNOM, DAMMIF, and AMBIMETER (requires ATSAS installation)
- Run singular value decomposition (SVD) and evolving factor analysis (EFA) on datasets
- Calculate P(r) functions using a Bayesian indirect Fourier transform (BIFT)
- Can read the following image formats:
  - Pilatus Tiff
  - CBF
  - SAXSLab300
  - ADSC Quantum
  - Bruker
  - Gatan Digital Micrograph
  - EDNA-XML
  - Eiger
  - ESRF EDF
  - FReLoN
  - Nonius KappaCCD
  - Fit2D spreadsheet
  - FLICAM
  - General Electric
  - Hamamatsu CCD
  - HDF5
  - ILL SANS D11
  - MarCCD 165

  - Mar345
  - Medoptics
  - MPA (multiwire)
  - Numpy 2D Array
  - Oxford Diffraction
  - Pixi
  - Portable aNy Map
  - Rigaku SAXS format
  - 16 bit TIF
  - 32 bit TIF

RAW is currently being developed by Jesse Hopkins and Soren Nielsen.

More information on RAW is available from the RAW website:

https://sourceforge.net/projects/bioxtasraw/

RAW needs your help! If you find bugs in the program, or errors in the documentation or tutorial, please let us know. Your input is the way we get better!

This documentation

The purpose of this documentation is to clearly and completely document the functions of RAW. This document is not intended to act as a tutorial to RAW, for a tutorial please refer to the tutorial document and videos available from the RAW website. Instead, the goal is to document what features are available in each section of RAW, how to use them, and what they do.

This document is laid out in chapters discussing each general area of RAW (for example, the Files tab in the Control Panel is a single chapter). When appropriate, a chapter will cover two related areas, such as the Manipulation tab of the Control Panel and the Main Plot Panel. Some areas will be physical panels or windows of RAW, as the previous examples, while some will be features, such as the Online Mode chapter. The exceptions to this are the second chapter, which covers how to install RAW, and the third chapter, which provides a brief introduction to the different windows of RAW and how they relate.

Typically, the algorithms used will not be fully documented here, as they are available in the source code or appropriate citations (and are subject to change as we improve and expand RAW).

5.8.2 Installing RAW

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

This chapter provides an overview of how to install RAW on Windows, Linux and Macintosh. Detailed installation guides for each system are available separately. It covers:

- Licensing
- System requirements
- Installing from source code.
- Installing from pre-compiled binaries.
- Obtaining the newest version of RAW

Licensing

RAW is open source software, which means the source code is freely available and can be modified/extended as the experienced users sees fit. RAW is available under the GPL V3 license which means that if a modified version of RAW, or any software that uses any of the code available in RAW, is made available to users, then the full source code must also be made available. For a full description of the GPL licensing terms, see the file gpl-3.0.txt included with the RAW source code, or visit https://www.gnu.org/licenses/gpl-3.0.html

System requirements

The faster the machine the better the experience.

Minimum Requirements

1.6 GHz 32 bit (x86) Computer

1 GB of memory

1024x768 resolution display

Windows 7 or newer, Mac OSX 10.9 or newer, Linux

Note: The newest version of RAW may run on Windows XP or Vista and OS X 10.4+, but it has not been tested by the developers on these older systems.

Optimal Requirements

2.5+ GHz 32/64 bit (x86) Dual/Quad Core Computer

2+ GB of memory

1280x1024+ resolution display

Windows 7 or newer, Mac OSX 10.9 or newer, Linux

Installing RAW from pre-compiled binaries

A prebuilt installer is available for RAW for Windows and Mac. We recommend that most users install the prebuilt versions. The windows installer has been tested on Windows 7, 8.1, and 10. It may also work for older versions. The Mac installer has been tested on macOS/OS X 10.9 - 10.12. It may also work for older versions.

Instructions for prebuilt installers are available here:

- *Windows*
- *Mac*

Installing RAW from source code

Installing RAW from source code takes more work but has the advantage that the newest version can always be downloaded and quickly installed once the necessary tools have been installed. Installing a pre-compiled version of RAW is easier, but compiled versions are not updated as often as the source code due to the time consuming process of making a compiled executable. Compiled versions may also only be available for certain platforms.

Instructions for install RAW from source are available for

- *Windows*
- *Mac*
- *Linux*

NOTE: As of writing, the RAW is **only** compatible with Python version 2.7 and not the newer 3.x versions.

Obtaining the newest version of RAW

The newest version of RAW can always be found on Sourceforge. The released source code and compiled binaries can be downloaded at:

https://sourceforge.net/projects/bioxtasraw/files/

To obtain the very latest unreleased version (that might or might not be stable) go to: https://sourceforge.net/p/bioxtasraw/code/HEAD/tree/trunk/src/

And click "Download Snapshot" near the top of the screen. Unpack the content into a folder and run RAW.py as usual.

NOTE: When installing new source code on top of old it is important to delete the old source code, including the compiled C libraries with the extensions .so, .pyc, and .pyd, otherwise RAW could be loading old code and become unstable.

Integrating ATSAS with RAW

RAW allows you to do analysis with some of the programs from the ATSAS package directly from RAW. Currently, you can use GNOM, DAMMIF, and AMBIMETER in RAW. This requires a separate ATSAS installation, as the RAW developers are not allowed to distribute the ATSAS package with RAW.

Installing the ATSAS package

The ATSAS package is available from EMBL, and can be downloaded here:

https://www.embl-hamburg.de/biosaxs/download.html

Installation instructions are available here:

https://www.embl-hamburg.de/biosaxs/manuals/install.html

We recommend installing the packages in the default installation location.

To use all of the programs through RAW, you need ATSAS version 2.7.1 or greater. GNOM and DAMMIF may work for earlier versions of the ATSAS package, but the RAW developers have not tested this.

Locating the ATSAS package for RAW

RAW will attempt to automatically locate the ATSAS package when you start up RAW (and when you load a configuration file). It may fail to do this, in which case you will need to set the location of the ATSAS programs manually. To do this:

- Open the "ATSAS" section of the Options window.
- Uncheck the "Automatically find the ATSAS bin location"
- Either by typing the path or using the "Select Directory" button, select the "bin" folder inside the main ATSAS folder. This folder should have a dammif executable inside of it.

Running without compiled extensions

RAW compiles certain extensions that are written in C++ in order to maximize the speed of the program. These extensions are involved in the following tasks: Making polygon masks, integrating images into scattering profiles, and carrying out the BIFT analysis. All of these extensions are also available in native python code, but run much more slowly. If RAW is unable to compile these extensions, a warning message will display when the program is started.

While RAW is able to run without the extensions compiled, it will significantly impact performance of the listed tasks. We recommend troubleshooting the RAW installation, or reinstalling RAW to get these to compile. The RAW installation guides contain detailed install instructions and some solutions to common problems with the installation. Please refer to those for more details.

5.8.3 Overview of RAW

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

The main screen of RAW shows three distinct panels: The Information Panel (top left), the Control Panel (bottom left), and the Plot Panel (right). Additional windows that can be shown are the Options window, and various Analysis windows. In addition, there is a menu bar at the top of RAW (either in the RAW window or in the system menu bar, depending on your operating system).

Any of the three panels docked in the main RAW window can be moved relative to the other panels. To do this, click on the title bar of the panel, and drag the panel to the desired location (you should see a blue rectangle indicating where the panel will be put). Additionally, panels can be undocked from the main window by clicking on the pin icon on the right side of the title bar of the panel. Undocked panels can be re-docked with the main window using the same method as rearranging the panels, drag the title bar until you see a blue rectangle appear indicating where the panel will go.

The portion of the total area available for the left (information and control) and right (plot) sides of RAW is controlled by the separator bar. You can click and drag on this bar to change this ratio.

The main RAW window can be resized by clicking and dragging on a corner or edge.

The Control panel

The control panel is where you manipulate files and items loaded into RAW. It has four tabs: Files, Manipulation, IFT, and SEC. The Files tab is for viewing files on the system disk, and loading the files into RAW. The Manipulation tab is for working with individual scattering profiles, the IFT tab is for working with inverse Fourier transforms, and the SEC tab is for working with SEC-SAXS data. You change tabs just by clicking on them. The order of the tabs in the control panel can be rearranged by clicking and dragging the tabs.

The Plot panel

The plot panel contains four tabs: the Main Plot, IFT Plot, Image plot, and SEC plot tabs. The Main Plot tab is for viewing individual scattering profiles. The IFT Plot tab is for viewing inverse Fourier transforms, the Image plot tab is for viewing detector images, and the SEC plot tab is for viewing SEC-SAXS data. The order of the tabs can be changed by clicking and dragging.

Multiple different plot tabs can be displayed at the same time. Click and drag the tab to any edge of the Plot Panel (up, down, left, or right) and you should see a blue box appear indicating where the tab will go. Additional splits after the first can be accomplished by dragging other tabs, and tabs can be moved between different tab bars (in this case, you should see the bar light up blue, rather than a full blue box for the panel). The amount of screen occupied by the different plots can be set by clicking and dragging the separator bars. To recombine, drag all of the plot tabs into a single bar.

Navigation/control bar

Each Plot tab has a navigation/control bar on the bottom. Most navigation bars have some unique buttons, but they all share the following buttons:

Home. Zoom the to fit all the displayed data.

Move back and fourth in the zoom history.

Pan/Zoom. Click and drag the left mouse button to pan and the right mouse button to zoom.

Area Zoom. Click and drag to zoom to the selected area.

White-space settings. Ability to adjust the white-space around the plots.

Image save. Ability to save the displayed plot(s) in formats such as .png, .svg, .eps, .pdf and more.

Cursor readout

Every plot has a cursor readout. If the mouse is hovering over a position on the plot, the coordinates (x-axis and y-axis values) will read out in the bottom bar of RAW. For the image plot, the intensity at those coordinates is also displayed. For the SEC plot, the value of the secondary y-axis is also displayed.

Plot settings

For each of the Main Plot, IFT Plot, and SEC plot, one of the options in the right-click menu is “Plot Options”. Selecting this will open a window that allows you to set:

- Plot title, x-axis label, and y-axis label text, font size, and bold/italicization
- Legend title, font size, and bold/italication
- Whether the legend is visible (default: no)
- Legend font size, transparency, border, and shadow
- Turn auto limits on/off for the Axes and manually set the axes limits if desired

- Turn off the left/right/top/bottom plot borders
- Turn on a line at y = 0 (the Zero Line)
- Adjust the axes tick label font sizes

The axis labels and title will accept some LaTeX commands if they are placed between $ symbols, for example: $\int I(q)$ will display as $\int I(q)$. The bold and italic options will not change LaTeX text.

The settings will be changed for the plot/subplot that was clicked on.

The Information panel

The information panel displays information about a selected data item in the Manipulation panel. It will show, if available, the Rg, I(0), and MW. It provides a place to enter the sample concentration, description/notes about the sample, and quickly look at the different header values of the data item.

If the current Control panel is changed from the Manipulation panel to the IFT or SEC panel, the Information panel is cleared. If the panel is changed back to the Manipulation panel, the data in the Information panel are restored.

The Options window

The options panel is opened by clicking on the Options->Advanced Options menu item. This opens a separate window where many of the RAW settings can be viewed/changed. The panel has two pieces. On the left is the set/category of options, and on the right the actual options panel. For example, if you click on the General Settings section in the left panel, the general settings will be displayed on the right.

Note that in the options tree on the left, the triangles can be used to expand/collapse more options for many of the categories.

Once you have changed the settings, you simply need to click "OK" and they will be saved in memory. In some cases you may want to set the options without exciting the panel. To do this, click "Apply". To exit without saving any changes click "Cancel".

Analysis windows

Various analysis windows can be opened in RAW. These will be discussed in detail *later*.

Menus

The top menu bar of RAW contains the File, Options, View, Tools, and Help menu. The View and Tools menu are simply another way to access options found elsewhere, while the File, Options, and Help menu have items that cannot otherwise be accessed. These will be discussed *later*.

5.8.4 RAW Settings and the Options window

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

This section covers the RAW settings, in particular focusing on the Options window and what all of the settings there do. It also covers how to save and load settings in RAW.

Saving and loading settings

To save the current RAW settings to disk (a .cfg file):

1. Go to the “File” menu bar and select “Save Settings”
2. Select a filename and a location for the configuration (config) file and click “Save”

To load RAW settings from disk (a .cfg file):

1. Go to the “File” menu bar and select “Load Settings”
2. Select the appropriate file and click “Open”

Note: Saving a configuration file saves all of the settings in RAW that are set in the Options window. It also saves any masks that have been created.

Changing settings

Settings in RAW are generally changed in the Options window. Below we describe how to open the options window, how to change all of the settings, and what the settings do.

Opening the Options window

To open the options window:

1. Go to the “Options” menu
2. Select “Advanced Options”

There are two parts to the Options window. There is an options tree on the left that determines which options panel is displayed on the right. The part on the right displays the options associated with the selected option in the option tree.

Closing the Options window

To save the changes made to the settings in the Options window, close the window by clicking the “OK” button. To exit without saving any of the changes in the Options window, use the “Cancel” button or the system close button (an “x” in the upper left or right corner of the panel).

General settings panel

The general settings are all check boxes and can be set by checking and unchecking the boxes. They control the following items:

Hide controls on manipulation items for new plots

If this option is selected, new Manipulation data items start out *collapsed*, if not selected they start out expanded. Defaults to off.

Write header on top of dat files

The *.dat* and *.ift* file “header” information can be saved at the top or bottom of the file. If this is selected, it is written at the top, if not it is written at the bottom. Defaults to off.

Use header for mask creation (SAXSLAB instruments)

If this option is set, if a SAXSLab instrument is being used, the image header will be used to make the beamstop mask. Defaults to off.

Detector is rotated 90 degrees (SAXSLAB instruments)

If this option is set, it indicates to RAW that the SAXSLAB detector is rotated 90 degrees. This affects mask creation from the image header. Defaults to off.

Start online mode on startup

If this option is selected and an *Online mode startup directory* has been picked, then when RAW starts, it will automatically turn on online mode, with the selected directory as the target online directory.

2D Reduction

In the 2D reduction panel, there are reduction options.

Correct of the change in the solid angle of the pixels

If this option is selected, the scattering intensity is corrected for the change in solid angle of pixels as you move along the detector. This is implemented as the standard $\cos^3(2\theta)$ where 2θ is the scattering angle.

Image/Header Format panel

This panel in the Options window allows you to set the image format, the header file format (if any), load the image header into RAW to set up normalizations, and set up bindings to use information in the image header or header file as calibration and reduction parameters.

Image format

The image format can be set using the drop down menu. A full description of supported image types is available in the *file types section*.

Header file format

The header file is a file that accompanies the detector image on some beamlines. This file often contains additional information such as, diode values, ion chamber readings, exposure time and date. Currently RAW supports the following header files: I711, MAXLab; I911-4, MAXLab; F2, CHESS; G1, CHESS; G1 WAXS, CHESS; G1 Eiger, CHESS; BL19U2, SSRF; and BioCAT, APS.

Note: If you wish to have a new header file format added to RAW, please contact the developers.

Loading/Viewing header information

If you wish to view header information from either the header file or the image header, click the “Load Image” button and select the image of interest.

If you wish to use the header information to normalize the image, load the image using the “Load Image” button and then click the “Apply” button at the bottom of the screen. This will save the counter values in such a way that RAW can set up the normalization appropriately.

Using image header information for calibration and reduction (turning on and setting bindings)

RAW has the ability to use header information for calibration and reduction settings. The method for doing this is to set a “binding” between the counter value and the calibration value. The calibration values that can be obtained from the image header or header file are: Beam X Center, Beam Y Center, Detector Pixel Size, Sample Detector Distance, and Wavelength.

To create a binding:

1. Check the “Use image-header/header file for calibration and reduction parameters” box.
2. Load the image and header file values into the list as described *above*.
3. In the list of the image header and header file names and values, click on the name of the header parameter that you want to use as one of the calibration values. This will fill in the Name and Value in the appropriate fields in the lower left hand portion of the panel.
4. Using the “Binding” menu, select what calibration parameter should use this header value. In the binding column of the header list, you will see this calibration parameter displayed.

Note: These values overwrite the same values set elsewhere in the settings. So if you bind the Beam X Center to use a value from the header, no matter what you set it to in the Calibration panel of the Options window it will use the value from the header.

Note 2: Make sure that your header file values match the expected units for the calibration parameter. The beam center values should be in pixels on the detector, the detector pixel size should be in microns, the sample detector distance in mm, and the wavelength in angstroms.

Adding a modifier to a binding

Once a binding is set, it is possible to add a modifier to the binding, which affects the value obtained from the header. This might be used in a case where the header value is not in the appropriate units.

To set a modifier:

1. In the list of the image header and header file names and values, click on the name of the bound header parameter that you want to set a modifier for.
2. In the Modifier field at the bottom of the panel, type in a mathematical expression. This expression may contain any of the header values (including but not limited to the header value selected for the binding). It may contain “+” “-“ “*” and “/” for addition, subtraction, multiplication, and division. The following strings are restricted, and apply specific mathematical functions: *acos, asin, atan, atan2, ceil, cos, cosh, degrees, exp, fabs, floor, fmod, frexp, hypot, ldexp, log, log10, modf, pow, radians, sin, sinh, sqrt, tan, tanh*, all of which correspond to the functions of the same name in the python math library (https://docs.python.org/2/library/math.html#module-math).
3. Click the “Add” button. You should get a popup window that evaluates the expression for the current loaded header values. Once you close that window, the modifier should be listed in the Modifer column of the header list.

Changing or removing a modifier to a binding

To change a modifier to a binding, do the steps to add a modifier, *above*. When you click on the header item in step 1, the modifier will be shown in the Modifier field at the bottom of the panel, and you can make changes as appropriate in step 2.

To remove a modifier to a binding:

1. In the list of the image header and header file names and values, click on the name of the bound header parameter that you want to remove a modifier to.
2. Click the "Remove" button (next to the Modifier field at the bottom of the panel).

Removing bindings

To remove a single binding:

1. In the list of the image header and header file names and values, click on the name of the bound header parameter that you want to remove a binding to.
2. In the "Binding" menu at the bottom of the panel, select "No binding".

To remove all bindings, click the "Clear Bindings" button.

Disabling bindings for calibration and reduction

To disable the use of bindings for calibration and reduction, either *remove* all bindings or uncheck the "Use image-header/header file for calibration and reduction parameters" checkbox.

Clear All

The "Clear All" button clears all bindings, and removes the current loaded header/header file values from the panel.

Calibration panel

The calibration panel allows you to set the beam center, binning size, number of points skipped at the start and end of a scattering profile, the sample to detector distance, wavelength, detector pixel size, and whether or not the Q range is calibrated.

Setting calibration parameters

The calibration paramters are: Beam center (x and y), sample-detector distance, wavelength, and detector pixel size. These can all be set by entering a value in the appropriate field on this panel or using the spin controls. However, it is more natural to set these values from the *Calibration/Centering panel*.

Note: Changing the settings in the calibration/centering panel will change the values in this panel, and vice versa.

Note 2: All of these calibration values are overridden by the bindings described *above*, if a binding for the particular calibration parameter is set.

Start and end points

The "Start plots at q point number" value sets the first q point shown on the plot when a scattering profile is integrated. It is zero indexed (first point is zero). So if it is set to 5, the plot will start with the 6th q point in the q-vector. This is typically used to get rid of the beamstop shadow from the integrated profiles.

The "Skip n points at the end of the curve" value sets the last point shown on the plot when a scattering profile is integrated. If it is set to zero, all points are shown. So if it is set to 5, the last point shown will be the 5th to last point in the q-vector. This is typically used to remove end points if something, for example the downstream flight tube window, is shadowing a high q region of the detector.

Binning

The default binning for integrated scattering profiles can be set using the “Binning Size” option. It accepts integer values. A binning size of one corresponds to q bins that are one pixel wide. A binning size of 2 corresponds to q bins that are 2 pixels wide, and so on.

Note: The q size of a bin of a given pixel size will depend on the calibration parameters.

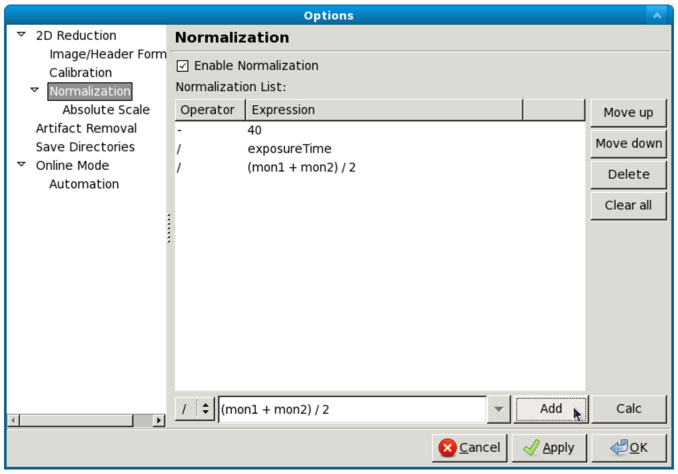
Calibrating the q-range

If you do not wish to calibrate the q-range of integrated scattering profiles, uncheck the “Calibrate Q-range” box. The scattering profile will then be displayed as intensity vs. bin number. This option is checked by default.

Normalization panel

The normalization panel allows you to normalize integrated scattering profiles by some value. Typically a counter value is used that is proportional to the beam intensity transmitted through the sample (such as a beamstop counter from an active beamstop).

Enabling and disabling normalization



Normalization

☑ Enable Normalization

Normalization List:

Operator	Expression
-	40
/	exposureTime
/	(mon1 + mon2) / 2

/ (mon1 + mon2) / 2

To enable normalization for integrated scattering profiles, check the “Enable Normalization” checkbox (checked by default). To disable, uncheck the “Enable Normalization” checkbox.

Setting up normalization operations

To add a new operation to the normalization procedure:

1. Select the operator to be used (/, *, -, + corresponding to division, multiplication, subtraction, and addition respectively)
2. Enter the desired expression in the expression box.
3. Click the “Calc” button to view the result of the entered expression.
4. Click the “Add” button.

Note: This expression may contain any of the header values (including but not limited to the header value selected for the binding). It may contain "+" "-" "*" and "/" for addition, subtraction, multiplication, and division. The following strings are restricted, and apply specific mathematical functions: *acos, asin, atan, atan2, ceil, cos, cosh, degrees, exp, fabs, floor, fmod, frexp, hypot, ldexp, log, log10, modf, pow, radians, sin, sinh, sqrt, tan, tanh*, all of which correspond to the functions of the same name in the python math library (https://docs.python.org/2/library/math.html#module-math).

Reordering and removing normalization operations

The order in which the operations are carried out can be changed by selecting the operation in the normalization list and using the Move Up and Move Down buttons. Operations can be removed by selecting the operation in the list and clicking the Delete button.

Normalizing by a header value

It is often desired to normalize the data by exposure time or incoming / transmitted beam intensity, and/or remove offsets on the detector.

To do so:

1. Load a header file into RAW as *described*. Be sure to hit the "Apply" button after loading!
2. Return to the Normalization panel.
3. Add a normalization value as in steps, in the expression box enter the name of the header value you wish to normalize by along with any other mathematical operations.

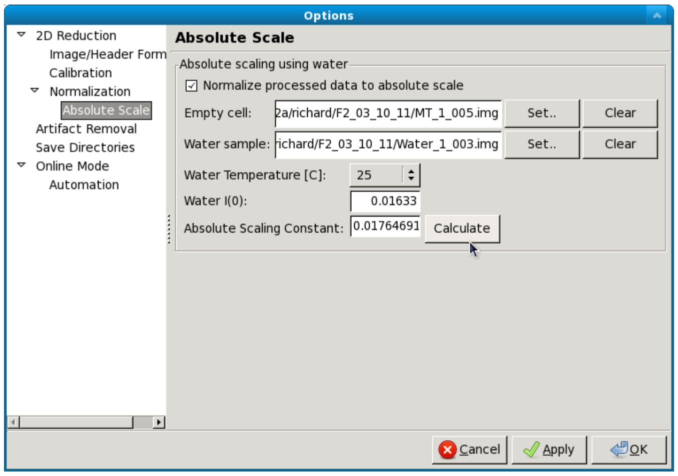
Normalizing by a region of interest (ROI)

RAW has the ability to normalize by a region of interest on the image. Every pixel in the region of interest is summed, and that can be used to normalize in the same way as a header value.

To normalize by an ROI:

1. Set an *ROI mask*.
2. Add an operation to the *normalization list*, but use "roi_counter" (without quotes) as the header value. For example, to divide by the roi value, select the "/" operator and enter roi_counter in the expression box, then add that to the list.

Absolute scale panel



Absolute Scale

Absolute scaling using water

☑ Normalize processed data to absolute scale

Empty cell: 2a/richard/F2_03_10_11/MT_1_005.img

Water sample: richard/F2_03_10_11/Water_1_003.img

Water Temperature [C]: 25

Water I(0): 0.01633

Absolute Scaling Constant: 0.01764691

RAW is able to scale loaded image data to absolute scale using water as a standard. Water has a known, temperature dependent absolute scale value at the forward scattering I(0). Water has a relatively flat scattering profile, which makes it possible to estimate the forward scattering, I(0), from an average of the intensity. To obtain the pure water signal, the water sample obtained in a sample cell must have the empty cell subtracted from it.

To set up Absolute scale:

1. Click the “Set” button for the empty cell. Select either an image or text (such as .dat) file of the empty cell scattering.
2. Click the “Set” button for the water sample. Select either an image or text (such as .dat) file of the water scattering.
3. Select the water temperature in degrees centigrade.
4. Click the “Calculate” button. An absolute scaling constant should appear.
5. Enable absolute scale normalization by checking “Normalize processed data to absolute scale” check box.

The algorithm uses the middle third part of the water scattering curve to estimate I(0) by the average intensity.

Note: The selected files must have been normalized in exactly the same way as the rest of the data that is to go on absolute scale. If loading an image, that means not changing the normalization parameters after calculating the absolute scale. If normalization parameters are changed the absolute scale constant will have to be re-calculated. It is particularly important that the images or profiles used to calculate absolute scale not have been saved with absolute scale already on (for example, from a previous calibration).

Turning off absolute scale

To turn off absolute scale, uncheck the “Normalize processed data to absolute scale” checkbox in the Absolute scale panel.

Flatfield correction panel

If a flatfield file is available, RAW can do a flatfield correction of the data. To do so, click the “Set” button, and select the flatfield image. Then check the “Enable flatfield correction” box.

When RAW applies a flatfield correction, it divides every image it processes by the flatfield image, on a per-pixel basis. The assumption is that every pixel in the flatfield image should have gotten the same intensity, so any variation comes from variation in the detector.

Turning off flatfield correction

To turn off flatfield correction, uncheck the "Enable flatfield correction" checkbox in the Flatfield correction panel.

Molecular weight panel

The molecular weight panel of the Options windows allows control of the parameters used to calculate molecular weight in the *molecular weight window* and the *SEC calculated parameters*. All four methods are described in more detail *elsewhere*.

Molecular Weight Estimation Using a Standard

This subpanel corresponds to parameters for the MW estimation by comparison to a known standard. While all of the parameters of the standard can be set/changed in this box, the standard MW (in kDa), the standard I(0), the standard concentration (in mg/ml), and the standard filename (only for reference), it is more natural to change these settings by loading the standard scattering profile into RAW and using the *Use as MW Standard* option.

Molecular Weight Estimation From Volume of Correlation

This subpanel corresponds to the parameters used for the volume of correlation method of estimating molecular weight. This method is the method used for calculating MW in the *SEC panel*). The protein and RNA coefficients correspond to the *A and B coefficients*. The default type selection selects if the MW calculation defaults to Protein or RNA. Changing this option will change whether the MW calculated in the SEC panel is for protein or RNA.

Molecular Weight Estimation From Corrected Porod Volume

This subpanel corresponds to the parameters for the MW calculation by corrected Porod volume. For this method, the only parmater that can be changed is the protein density in kDa/Å:sup:*3*.

Molecular Weight Estimation From Absolute Intensity Calibration

These parameters correspond to the parameters necessary for calculating the molecular weight when a scattering profile is on an absolute scale.

Reset MW Parameters To Defaults

If you have customized the MW parameters for a particular sample, you can restore the parameters to the RAW defaults (which are the defaults from the relevant papers for each method). There are no default settings for the estimation using a standard.

Artifact removal panel

Zingers are pixel values on the detector that are unusually high due to either cosmic radiation or readout errors. RAW includes three methods that can be used for zinger removal.

Zinger removal by smoothing

A window of "Window Length" data points can be run across the data and discard values that are more than "Std" standard deviations away from the average of the points in the window. A starting index is given to specify where on the data curve the window should start.

Zinger removal when averaging

If three or more exposures of the same sample are available, then these can be used to eliminate zingers by comparing the intensity values of each data set to the others. An intensity value in a data-set that is larger than x standard deviations (Sensitivity) from third quintile of all related data points in the rest of the data sets is removed and replaced by the average of the third quintile.

Zinger removal after radial averaging

This method is the most effective method for removing zingers. Pixel intensities in the image for the same q value are compared and should be fairly constant. Values that are more than "Sensitivity" standard deviations away from the median are discarded.

IFT panel

RAW currently supports one built-in method for determining the inverse Fourier transform (IFT) of a scattering profile, the Bayesian IFT (BIFT) method. In the future we anticipate supporting a python based implementation of the GNOM algorithm called pyGNOM, but currently that is not available.

BIFT

The BIFT panel allows you to set the BIFT Grid-Search parameters. These define the large grid that the BIFT algorithm searches over before doing the fine search near the best value on the grid.

Dmax Upper Bound

Sets the largest maximum dimension (Dmax) value that will be used in the coarse grid search, in Å.

Dmax Lower Bound

Sets the smallest Dmax value that will be used in the coarse grid search, in Å.

Dmax Search Points

Total number of Dmax values in the coarse grid search that. These are evenly distributed between the lower and upper bounds.

Alpha Upper Bound

Sets the largest alpha value that will be used in the coarse grid search.

Alpha Lower Bound

Sets the smallest alpha value that will be used in the coarse grid search.

Alpha search points

Sets the total number of alpha values in the coarse grid search. These are distributed logarithmically between the lower and upper bound.

P(r) Points

Sets the number of points in the calculated P(r) curve.

Save Directories panel

This panel controls the settings for *automated saving of data*.

Auto Save

In this subpanel, the checkboxes control whether or not RAW automatically saves Processed image files, Averaged data files, and Subtracted data files. When the boxes are checked, that file type will be automatically saved, when they are unchecked, it will not.

Save Directories

This panel allows you to selected the directories into which files will be saved for each of the automated saving file types (Processed, Averaged, Subtracted). To pick a directory, click the "Set" button and use the window that opens to select a folder. Click "Open" once the appropriate folder is selected. To clear a directory click the " Clear" button.

Note: A save directory must be selected before an Auto Save checkbox can be enabled.

Online Mode panel

This panel includes settings for the *online mode*. This lets you enable online filtering and set up the filter list.

Enabling/disabling online filtering

Online filtering filters files by filename, so that you can control which files are loaded into RAW automatically. To enable this mode, check the "Enable Online Filtering" checkbox. To disable this mode, uncheck that checkbox.

Adding a filter item to the online filter list

A filter item consists of three parts. First, there is the Ignore/Open operator. This allows you specify whether you want RAW to ignore files with the given filter string in their name, or to only open files that have the given filter string in their name. To set this option, use the dropdown selector box at the bottom of the panel and select either "Ignore" or "Open only with".

The next part of the filter is the filter string. This is the string that RAW looks for in the filename. To set this, enter a string into the filter string box at the bottom of the panel.

The final part of the filter is the location of the filter string. This sets where in the filename RAW looks for the given filter string. This can be set to: "At start", which means RAW only applies the filter Ignore/Open action to files with the filter string at the start of the file name; "Anywhere", which means RAW applies the filter Ignore/Open action to files with the filter string anywhere in the file name; and "At end", which means RAW only applies the filter Ignore/Open action to files with the filter string at the end of the file name. To set this, use the dropdown selector box at the bottom of the panel and select one of those three options.

Once you have set the Ignore/Open option, entered a filter string, and selected the location of the filter string, click the "Add" button to add the filter item to the Online Filter List.

Reordering and removing filter items

The order in which the filtering is carried out can be changed by selecting the item in the filter list and using the Move Up and Move Down buttons. Items can be removed by selecting the operation in the list and clicking the Delete button. All filter items can be removed using the "Clear all" button.

SEC-SAXS panel

The SEC-SAXS panel controls settings related to the SEC-SAXS data processing.

Intensity ratio (to background) threshold for calculationg Rg, MW, I0

In order to speed up the *calculation of Rg, MW, and I0 as a function of frame* for SEC-SAXS data, a ratio of the frame intensity to the background intensity is taken. If that value is less than the threshold set here, the frame is skipped. To attempt to calculate structural parameters for all frames, set this threshold to -1.

ATSAS panel

The top level ATSAS panel allows you to control where the ATSAS bin location is (the folder with all of the ATSAS programs in it). By default, RAW will attempt to automatically find the ATSAS installation. If you wish to set the location yourself, uncheck the "Automatically find the ATSAS bin location" checkbox, and either type the location into the ATSAS bin location field or use the "Select Directory" button to select the appropriate directory.

Note: If you uncheck and then check the Automatically find checkbox, RAW will attempt to find the ATSAS directory again. This can be useful if, for example, you install ATSAS and want RAW to find the new installation without restarting RAW.

GNOM panel

The top level GNOM panel allows you to set the commonly used advanced settings for the ATSAS software GNOM, which is run from the *GNOM window*. All of these options correspond in name and allowable values to those of GNOM as described in the GNOM manual: https://www.embl-hamburg.de/biosaxs/manuals/gnom.html

Settings can be rest to their defaults (which correspond to the GNOM defaults) by clicking the "Reset to default" button. This resets the settings in this panel and in the GNOM Advanced panel.

GNOM Advanced panel

This GNOM panel allows setting GNOM settings which are not as commonly used in GNOM. Again, all of the options correspond in name and allowable values to those of GNOM as described in the GNOM manual: https://www.embl-hamburg.de/biosaxs/manuals/gnom.html

Settings can be rest to their defaults (which correspond to the GNOM defaults) by clicking the "Reset to default" button in the GNOM panel. This resets the settings in this panel and in the GNOM panel.

DAMMIF panel

The top level DAMMIF panel allows setting two things: First, the default settings for DAMMIF that are set when the *DAMMIF window* is opened. Second, standard settings that are available in Fast and Slow mode can be set in the "Standard Settings" subpanel. All of the settings correspond in name and allowable values to those in the DAMMIF manual: https://www.embl-hamburg.de/biosaxs/manuals/dammif.html

Settings can be rest to their defaults (which generally correspond to the DAMMIF defaults) by clicking the "Reset to default" button. This resets the settings in this panel and in the DAMMIF Advanced panel.

DAMMIF Advanced panel

The settings in the DAMMIF advanced panel are only used when the "Custom" mode is selected in the DAMMIF panel. This is equivalent to the interactive DAMMIF mode at the command line. Unless otherwise noted, a value of -1 for a field indicates that it will use the default setting. The settings correspond in name and allowable values to those in the DAMMIF manual: https://www.embl-hamburg.de/biosaxs/manuals/dammif.html

Settings can be rest to their defaults (which generally correspond to the DAMMIF defaults) by clicking the "Reset to default" button in the DAMMIF panel. This resets the settings in this panel and in the DAMMIF panel.

5.8.5 The Centering/Calibration panel

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

The Centering/Calibration panel allows you to set the beam center, x-ray wavelength or energy, sample to detector distance, and detector pixel size. This allows RAW to calibrate the q position for each pixel, and integrate the image into a one dimensional scattering profile.

Opening the Centering/Calibration panel

Before opening the Centering/Calibration panel, you should make sure that an appropriate image for calibration (such as the scattering from silver behenate) is *open* in the *Image panel*. To open the Centering/Calibration panel, go to the Tools menu and click on "Centering/Calibration". The Centering/Calibration panel will be placed (temporarily) over the control panel.

Setting calibration parameters

The Wavelength, Energy, Sample-Detector Distance, and Detector Pixel Size can all be set in the Manual Centering/Calibration Adjustments panel of the Centering/Calibration panel. For each, they can be set in one of two ways. First, a value can be directly typed into the appropriate field. Second, the spin controls can be used to adjust the value in the field. The spin controls only change the last digit of the value, so if the sample detector distance is set at 1000, the up spin control will change it to 1001, whereas if it is set at 1000.0, the up spin control will change it to 1000.1.

If a centering pattern is displayed on the image, changing these calibration will change the pattern. In order for the pattern to update when a value is typed into a field, you must either hit the enter key, or click out of the field.

Note: Setting the wavelength will automatically set the appropriate energy, and vice versa. You only need to set one of these values.

Manually setting the beam center

The beam center is displayed in x and y coordinates in the Manual Center/Calibration Adjustments panel. These coordinates correspond with the x and y coordinates shown on the image. If a centering pattern is selected in the Pattern drop down menu, the eam center is also displayed on the image as a red dot 6 pixels in diameter.

The location of the beam center can be changed in three ways. First, values can be directly entered into the X center and Y center fields. As with the calibration parameters, in order to update the pattern display after entering a value in these fields, you must either click out of the field or hit the enter key.

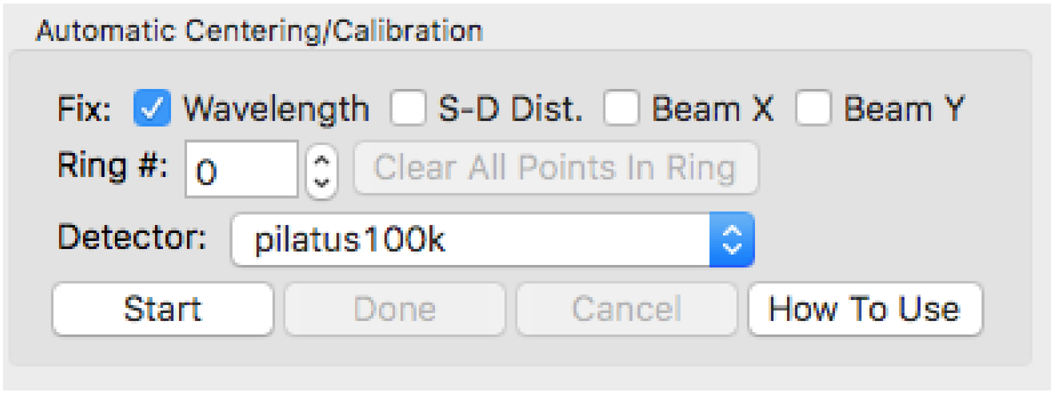
Second, the step size of the beam center in pixels can be set using the drop-down "Steps" menu. The Steps menu is also a field you can type into, so you can set it to any numerical value if the options available in the list are insufficient. Once you have set this, the large red centering arrows on the right side of the Manual Center/Calibration Adjustments box can be used to move the beam center <steps> pixels in the direction indicated by the arrow, where <steps> is the value of the "Steps" field/menu. The arrows can be held down for repeated motion in the same direction with the same step size.

Third, the target button in the center of the centering arrows can be clicked. You then click on the image, and the beam center will be set to that position on the image, to the nearest pixel.

Automatic centering and distance calibration

There are two different automated centering methods available in RAW. Which one is available in your version depends on whether or not you have the *pyFAI* library installed. With either of these methods, you will have the best chance to obtain accurate results if you manually calibrate the center and distance first, so that the automated centering can refine on that starting position.

With pyFAI (recommended)


Automatic Centering/Calibration

Fix: [✓] Wavelength [ ] S-D Dist. [ ] Beam X [ ] Beam Y

Ring #: 0

Detector: pilatus100k

In the Manual Centering/Calibration Adjustments panel:

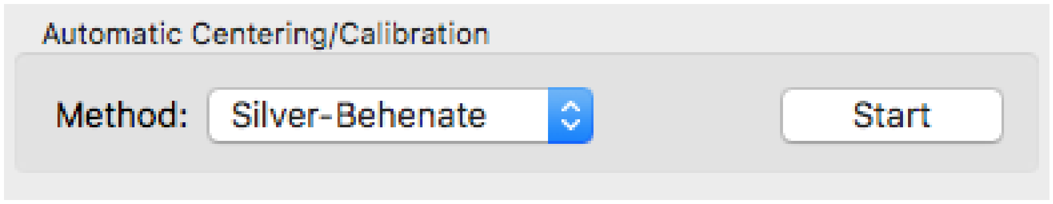
1. Select the appropriate Standard.

In the Automatic Centering/Calibration panel:

1. Select the parameters to hold constant by checking boxes in the Fix section.
2. Select the detector type to be used. If your detector is not in the list, select Other.
3. Set the Ring # to the index of the first ring visible on the detector. The ring index starts at zero for the largest d-spacing ring (nearest the beam) and increments by one for each ring thereafter. *IMPORTANT:* The first ring visible on your detector image may not be ring 0!
4. Click the Start button. Then click on the first ring in the image. Points in that ring will be automatically selected. Click on other parts of the ring as necessary to fill in points.
5. Increment the ring number as appropriate for the next ring visible on the image (usually increment by 1, for example from 0 to 1), and click on the next ring on the image to select points there. Repeat for all visible rings.
6. (If necessary) To remove points in a ring, set the Ring # to that ring, and click the Clear All Points In Ring button.
7. Click the Done button once you have selected points in all of the visible standard rings. At this point, automatic centering and calibration will be carried out.

Note: Ring points cannot be selected if the Pan or Zoom tool are selected.

Without pyFAI (not recommended)



This automated centering function attempts to position the center and find the sample-detector distance based on the innermost ring of a Silver Behenate sample:

1. Click the start button in the "Centering/Calibration Panel"
2. Select at least 3 points just outside the innermost ring in the Silver Behenate image.
3. Hit the "Done" button.

This function attempts to find the center of the circle that all of the points defined by your clicks make. If that is found, it then uses the known wavelength, pixel size, and q value of the first silver behenate ring to calculate the sample-detector distance.

Centering patterns

With pyFAI (recommended)

If the pyFAI library is installed, more than twenty different calibrants are available, all of those supported by the py-FAI package (https://github.com/silx-kit/pyFAI/tree/master/pyFAI/resources/calibration), including Silver-Behenate (AgBh). Rings are displayed as red dashed lines at all appropriate q values.

Without pyFAI (not recommended)

Only one centering pattern is available in RAW. This is the Silver-Behenate pattern, which displays red dashed rings on the detector corresponding to the expected rings from Silver-Behenate. RAW displays the first 5 rings, which at q = 0.1076, 0.2152, 0.3229, 0.4305, and 0.538 Å:sup:*-1*. The q values are labeled in RAW, but only show at the bottom of the rings, and so you typically have to zoom out of the image to show the label.

5.8.6 The Masking panel

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

The masking panel allows you to create various types of masks that will be used when the image is integrated into a one dimensional scattering profiles.

Types of masks

Beamstop mask

Unwanted regions such as the beamstop, bad pixels, or gaps between detector panels need to be excluded from the detector image before processing the image. Creating a beamstop mask allows you to do so. For a beamstop mask, when a normal mask is drawn (shown as red on the image), the area inside the mask is excluded from integration. When an inverted mask is used (green on the image), only the area within the mask is included during integration. These types of masks can be combined: normal masked regions will be excluded from integration, even if some or all of the same region is included in the inverse mask region. This allows you to create an inverse mask, and mask out undesired features within that inverse mask.

ROI counter mask

RAW allows you to define a region of interest (ROI) mask. If you have an ROI mask defined, every time an image is integrated to a scattering profile, all of the pixel intensities in the ROI mask will be summed, and the value will be available as a counter. In this case, the normal mask (red color) is an include only mask, while the inverted mask (green color) is an exclusion mask. Thus, any areas with a normal mask will be counted in the ROI, while any areas with an inverted mask will be ignored by the ROI.

Typically this is used for normalization purposes, for example to normalize by the transmitted beam through a semi-transparent beamstop. The counter name that this value goes to is *roi_counter*.

SAXSLAB Beamstop mask

The SAXSLAB home source instruments embed the beamstop mask within the image header. Selecting this beamstop mask option will allow you to show this mask on the image. This mask cannot be changed within RAW.

Note: Both a beamstop mask and a SAXSLAB beamstop mask can be simultaneously defined in RAW. In that case, they are both used to mask the image.

Readout-dark mask

The area that is selected by a readout-dark mask is averaged for each image, and that average value is subtracted off of every pixel in the image. The standard deviation in the pixel counts is added in quadrature with the noise in the scattering profile calculated during integration (the standard deviation of the counts in each q bin). In this case, the normal mask (red color) is an include only mask, while the inverted mask (green color) is an exclusion mask. Thus, any areas with a normal mask will be averaged to get the readout-dark counts, while any areas with an inverted mask will be ignored by the average.

Setting a readout-dark mask is intended to allow RAW to compensate for dark counts on the detector (most common with CCD detectors). Typically it is used when a fixed area of the detector is permanently masked, such as by lead tape applied to the detector face.

Creating a mask

In order to creating a mask, an appropriate image must first be l:ref:*open <showimage>* in the *Image panel*. Mask creation is then done as follows:

1. Select the appropriate mask type from the drop down masking menu in the Mask Creation panel.
2. Use the mask drawing tools to draw a mask on the image.
3. For the circle and rectangle tools, left click once to start drawing, move the mouse until you have the desired shape, and click again to stop drawing.
4. For a polygon, left click once to start drawing, and continue left clicking to add vertices. To finish drawing and connect the final point to the initial point, right click.
5. To invert a mask section, right click on the section and select "Inverted mask".
6. To remove a mask section, left click on the section to select it and click the delete key.
7. You may draw as many mask sections as you want.
8. Once done, click the "Set" button to save the mask.

Note: Clicking "Set" overwrites the existing mask in memory. To add new regions to an existing mask, see below.

Viewing and modifying a mask

At times you may wish to view an existing mask, and possibly modify it. To do so:

1. Select the mask type in the Mask Creation panel drop down menu.
2. Click “Show” to show the existing mask.
3. Draw new masked regions and/or remove old masked regions as above.
4. Once you have finished modifying the mask, click “Set” to save your changes.

Removing a mask

To completely remove a mask, select the appropriate type of mask in the drop down menu in the Mask Creation panel, and click the “Remove” button.

Saving/Loading a mask to/from disk

A mask may be saved to disk by itself (saving the settings saves the mask as well), using the “Save” button in the mask drawing panel, and loaded into RAW using the “Load” button. Once a mask is loaded, select the appropriate mask type from the drop down menu in the Mask Creation panel and click “Set” to set the loaded mask as that mask type.

Note: The “Save” button does not save the mask in program memory! You must use the “Set” button to set a mask as the current mask for use. Save and load are exclusively for saving/loading to/from disk.

Clearing a mask

The “Clear” button clears any mask from the image. However, to save these changes, you must click “Set” for the mask type of interest. To remove a mask from use in RAW, see *above*.

Mask drawing options

The check box “Show Beam Center” puts a red circle of diameter 6 pixels on the image where the beam center is (as set in the Centering/Calibration panel).

5.8.7 The Files tab in the Control Panel

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

Changing file directories

There are three ways to change the directory whose contents are displayed in the Files tab:

- Click on the folder icon next to the bar to open a directory window. This allows you to select the directory you want to display.
- Type a path into the bar at the top of the Files tab and hit enter to go to a directory.
- Double click the up arrow in the File list (blue arrow with a filename of ‘..’) to navigate up a directory level. Double click on any displayed directory to open it and show the contents in the Files tab.

Note: The Files tab simply displays the files in your system. None of these files are loaded into RAW until you take some action on them (for example, *plotting them*)

Updating the file list

The file list in the Files tab does not automatically update when new files are placed in the folder. In order to update the file list to show new files, you must click the Refresh button (two green arrows in a circle) to the right of the Folder button near the top right of the Files tab.

Plotting files

Plotting is done from the Files tab in the Control Panel. It can be done in two ways:

1. Simply double-click an image, compatible ASCII file, or RAW sec file to plot it.
2. Select one or more files to be plotted at the same time and click the "Plot" button

The Calibration, Normalization, Masking, and other Advanced Options will be used to convert images into one dimensional scattering profiles when an image is selected and plotted. The Normalization settings, including absolute scale factor, counter norms, and solid angle correction status, the configuration file used, and the image load path are saved as part of the manipulation history for a scattering profile in RAW created from an image.

Note 1: Selecting files is as simple as left clicking on them. Multiple files can be selected by shift or ctrl (command on OS X) left clicking on the files.

Note 2: Some image files can only be loaded if the image type in the Advanced Options is set appropriately.

Filtering and searching the file list

Below the file list is a filter/search bar. You can either select filters from a list or create your own using wildcard characters. The filter bar accepts '*' as a wildcard matching any number of characters and '?' as a wildcard matching a single character.

Examples:

- If you were looking for all files with 'BSA' anywhere in the filename you would write: *BSA* in the filter/search bar. The two * wildcard matches any characters before and after the 'BSA'.
- If you wanted all files with 'BSA' at the start of the filename you would write BSA* in the filter/search bar. The single * wildcard will match any characters after the 'BSA'.
- If you wanted all of the files with 'BSA' at the start of the filename and '.tiff' at the end of the filename you would write **BSA*.tiff**. The single * wildcard will match any characters between BSA and .tiff in the filename.
- If you wanted all of the files with 'BSA_000x.tiff' where x is a single character, you would write **BSA_000?.tiff**. The ? wildcard will match any single alphanumeric character in that position.

Sorting the file list

The columns at the top of the file list, Name, Ext, Modified and Size, show the file name, the file extension, the file modified time, and the file size respectively. By clicking on the column heading, you can sort the file list by those attributes. By default, the file list is sorted by name, ascending (low to high, A to Z). This is indicated by the upward pointing green arrow in the Name column heading. Clicking on the Name column heading will switch the order of the sort to descending (high to low, Z to A) sort. Clicking again will return it to the ascending sort. Clicking on another column heading will sort the file list by that column, and again can be switched between ascending and descending.

Files tab buttons

Quick reduce

Processes selected images and save them to ASCII files without plotting the scattering profiles using the integration parameters (calibration, masking, etc) current set in RAW.

System Viewer

Attempts to open/run the selected file using whatever software is set up in the operating system to handle the file format. You can use the 'E' keyboard key as a shortcut.

Plot

This plots the selected files in RAW, assuming the formats can be read by RAW. If the files are images, they will be integrated into one dimensional scattering profiles using the integration parameters (calibration, masking, etc) set in RAW. Files are shown in the following locations:

- Images (which RAW integrates) and 1D scattering profiles (typically ".dat" files) opened by RAW display in the Main Plot tab of the Plot panel. The Manipulation panel shows the items in the Main Plot and lets change and process the curves individually.
- IFT files (.out and .ift extensions) are loaded into the IFT Plot Panel of RAW. The IFT panel in the Control Panel shows the items in the IFT Plot and lets you change and process the curves individually.
- SEC files (.sec extensions) are loaded into the SEC Plot Panel of RAW. The SEC panel in the Control Panel shows the items in the SEC plot and lets you change and process the curves individually.

Note: If a file has previously been saved from RAW, all of the saved information (see later sections) will be loaded when the file is reloaded. For example, if a subtracted scattering profile had a Guinier analysis performed on it and was saved as a ".dat" file, when that saved file is loaded back into RAW, the Rg and I(0) values will be loaded as well.

Show image

Shows the image selected in the file list in RAW's image plot. You can use the 'S' keyboard key as a shortcut.

Note: If multiple files are selected, the image shown is from the first file in the list.

Clear all

Resets and clears all plots and also clears the item lists in the Manipulation, IFT, and SEC tab.

Plot SEC

The "Plot SEC" button plots the selected images and/or 1D scattering profiles as a *SEC curve*. The images are integrated into 1D scattering profiles.

Note: This button can also be used to plot files with the .sec extension that have previously been saved by RAW.

Manipulating files and folders

Files and folders can be manipulated in the Files tab by right clicking on a filename in the file list. The pop-up menu has the options to create a new folder, rename a file, copy/cut/paste file(s) or delete file(s).

Note: These options change the files on disk, not just in RAW! So if you delete a file here, it will be deleted from your disk. If you want to work with the files you have loaded into RAW, see the sections on the *Manipulation*, *IFT*, and *SEC Control Panels*.

5.8.8 The Manipulation Panel and Main Plot Window

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

When individual scattering profiles are loaded, they are placed in the Manipulation tab of the Control Panel, and displayed in the Main Plot window. This section will cover the features of the Manipulation panel and the Main Plot Panel.

The Manipulation Panel

The Manipulation Panel refers to the panel that is shown in the Control Panel when the Manipulation Tab is selected.

The panel consists of two parts. The top part is where individual scattering profiles that are loaded into RAW are shown. The bottom part consists of buttons that allow you to manipulation these scattering profiles. Further manipulation of the scattering profiles can be done from the right click (context) menu and from the Tools menu.

An individual item in the Manipulation Panel list is called a manipulation data item.

All items loaded into the manipulation panel are initially displayed in the top plot of the Main Plot.

Manipulation Data Items

When a new item is plotted, a data item is added to the manipulation list in the manipulation tab. This allows you to control the properties of the data item, and perform manipulations and analysis on it.

The name of the data item is displayed at the top of each item. If an item is given a different name for the plot legend, this legend name is displayed in [square brackets] next to the item name. On the same line as the item name, on the right side of the data item, there are a *series of buttons* that can be used for further manipulation of the data item.

Note: If there is a * to the left of the item name (between the checkbox and the item name), it indicates there are unsaved changes to the item. This can occur if you change the scale, offset, or q range of the item, if the item is a newly created scattering profile (such as integrated from an image, or an averaged or subtracted item), if there is new analysis information associated with the item (such as Rg and I0), and from binning and interpolating the scattering profile.

Offsetting, scaling and adjusting the q-range of a data item

Each data item lets the user interactively adjust the scaling, offset and q-range by using the spin-buttons or entering values in the associated text field and hitting enter. These changes will be updated on the scattering profile displayed on the main plot.

Offset and Scale

The spin-buttons in the scaling and offset boxes will adjust the last digit of the displayed number by 1. So if the scale is 1.0, using the up spin button will adjust it 1.1, while the down spin button will adjust it to 0.9. If the scale was set to 1.15, the up button would adjust it to 1.16, while the down button would adjust it to 1.14.

The scale can never be set to zero by the spin buttons, so if it is adjusted down in a way that would do so, it starts adjusting the next significant figure. For example, if you use the down button on a scale of 0.1, the new scale will be 0.09. A negative scale factor is treated like a positive scale factor (the absolute value of the scale factor is applied), thought it can cause unexpected behavior from the spin buttons.

Q-range

Adjusting q Min and q Max can be done in two ways. Each q limit has two boxes associated with it. The left box for each displays the actual q value, and a q value can be typed into those boxes. As there are a finite set of points in q space, RAW will automatically determine the nearest q point, and set the q value to this nearest point. The box on the right is the (0 indexed) point in the q vector. That is, if for q Min the value reads 15, then the scattering profile is being displayed starting at point number 15 in the q vector (note that this is the 16th point of the q vector, because it is zero indexed). The spin buttons adjust these point numbers, so if you use the up spinner on q Min, it will start at the next point in the q vector.

Showing/Hiding data items on the Main plot

To show/hide the scattering profile associated with a data item on the plot, click the checkbox next to the sample filename. If the checkbox is checked, the item is currently shown on the Main plot, if the checkbox is unchecked, the item is currently hidden on the main plot.

Data item buttons

On the right of the top line of the data item (the same line as the item name and show/hide checkbox) are a series of buttons for controlling the data item. In order of left to right, these buttons have the following properties.

Collapse/Expand item controls

In order to see more data items in the list, or to prevent accidental changes to the q Min, q Max, Scale, or Offset values, the controls for a data item can be hidden using the triangle button in the upper right of the data item. Note that the direction of the triangle switches when it is toggled. If the triangle is pointing up, the controls are expanded, if the triangle is pointing down the controls are collapsed.

Extended Info

By hovering the mouse over the i in the blue circle button, a tooltip will appear which shows more information about the item, including Rg and MW.

Locate Line

The target button is used to highlight the scattering profile on the graph that is associated with the data item. When the target is pressed, it ‘bolds’ the line of the scattering profile (increases the line width by several points). When the target is pressed again, the line width is set back to normal. You can tell if a line is currently bolded, as the target will be orange instead of grey.

Line Properties

The colored line button has two purposes. First, the color matches the current color of the scattering profile in the Main Plot. Second, when pressed it opens a *line properties dialog* which allows you to set the legend label; the line style, width, and color; the data point marker style, size, line color, and fill color; and the error bar line style, width, and color.

Mark

The star button marks an item. This is used when doing operations such as subtraction, syncing, superimposing or merging. In those cases, the marked (also referred to as starred) item has a special significance.

Selecting data items

A single data item can be selected by clicking on the item name in the Manipulation list (similar to how you would select files in your system file browser). When an item is selected, the color of the item background changes from white to gray. If the item is currently selected, clicking on it will cause it to be unselected. Note that for a regular click, all other selected items will be unselected when a new item is selected.

Multiple items may be selected in two ways. If the Control key (Command key on Macs) is held down while clicking on items, each item that is clicked on will be added to the set of selected items. If a single item is first selected and then the Shift key is held down and another item is selected, all of the items in the list between the two items will be selected (including the second item that is clicked on).

All of the items in the list can be selected in two ways. The first is using the *select all button*, the second is pressing Ctrl-A (Cmd-A), the Control (Command) key and the A key at the same time when you are in the Manipulation panel. All items can be unselected by clicking in a empty spot of the Manipulation list (but not above or below the list), or by clicking on an already selected item.

Note: If you have a set of selected items and wish to remove some, holding down the Control (Command) key and clicking on selected items will deselect them without affecting the other selected items.

The top buttons of the Manipulation Panel

The Manipulation Panel has a set of five buttons at the top of the panel. These buttons have the following effects, listed from left to right.

Show All

Clicking on the button that looks like an eye will show all scattering profiles. This is the same as if you manually set all of the show/hide checkboxes in the data items to on.

Hide All

Clicking on the button that looks like an eye with a red x through it will hide all scattering profiles. This is the same as if you manually set all of the show/hide checkboxes in the data items to off.

Select All

Clicking on the button that looks like a spreadsheet with selected cells will select all of the Manipulation items.

Collapse All

Clicking on the button that looks like an upward pointing arrow with a box under it will collapse all of the data item controls. This is the same as if you manually toggled the Expand/Collapse button for each data item to the collapse position.

Expand All

Clicking on the button that looks like a downward pointing arrow with a box over it will expand all of the data item controls. This is the same as if you manually toggled the Expand/Collapse button for each data item to the expand position.

Synchronizing settings for several data items

Settings for several data items can be easily synchronized.

1. Click the star icon on the data item you want the other items synchronized to.
2. *Select* one or more data items, you wish to synchronize with the starred item.
3. Click the “Sync” button
4. Select what parameters you want synchronized
5. Hit the “OK” button

Parameters that can be synchronized in this way are: q min, q max, n min, n max, scale, offset, line style, line width, and line marker.

Renaming a data item

Data items can be renamed by selecting the data item of interest and selecting “Rename” in the right click popup menu.

Note: While no characters are expressly forbidden in the filename, RAW does not sanitize file names before saving, and thus special characters such as ‘/’ and ‘\’ are likely to cause problems when the file is saved.

Saving data items

All scattering profiles can be saved to a standard 3 column ASCII format with a header. To save:

1. Select the item(s) to be saved.
2. Click the “Save” button or select “Save selected file(s)” from the right click menu.
3. In the window that pops up, navigate to the directory in which you want to save the files.
4. If you are saving a single item, the window will give you an opportunity to rename your file if desired. Click “Save” when ready.
5. If you are saving multiple items, you simply need to select the folder for the items to be saved in, and click “Open”. The items will be saved with the same names displayed in the Manipulation Panel, in the folder that you chose.

Items are saved in a standard format with three columns of data. These columns have listed headers, and are Q, I, and Error in I respectively. Depending on your settings, at the bottom (default) or top of the file there will be a “Header”. This header contains any information loaded as ‘Header’ information from the image or separate header file, the normalization factor, any notes you made in the Notes section of the information box, any analysis results (such as Rg, MW), and the manipulation history saved by the averaging, subtracting, merging, rebinning, and interpolating functions (see following sections).

The files are saved with a “.dat” extension, which is the standard for scattering profiles and can be read in by most standard SAXS processing software (including Scatter, Primus, and programs in the ATSAS package). The files are simply text files, and can be opened and viewed in any standard text editor. The header is saved in the JSON format.

Removing data items from the manipulation list

To remove one or more data items, select them and do one of the following:

1. Press the “Delete” key on the keyboard
2. Click the “Remove” button
3. Select “Remove” from the right click menu

Averaging data

Averaging data is an important part of processing SAXS data. At the moment in RAW, data can only be averaged if the q vectors match exactly. To average data:

1. *Select* two or more data items that should be averaged.
2. Either click the "Average" button or right-click on a selected item and choose "Average" from the popup menu.

After averaging, a new data item will be displayed in the manipulation panel with the filename in green and an 'A_' prefix. All averaged items are initially displayed on the top plot of the Main plot, regardless of where the items being averaged are displayed.

A record of which profiles were averaged is saved in the metadata for the averaged profile. This can be viewed within RAW by right clicking on the item and selecting "Show history", or externally if the scattering profile is saved as a .dat file and opened by an external text editor. This history contains all of the history of the individual files, so if you average a set of averaged (or subtracted) files, you can dig down through the metadata to see what files were used at each step.

Subtracting data

Subtracting data is an important part of processing SAXS data. RAW can handle subtracting data that has different q vectors, as long as there is a region of overlap between the q vectors. Data subtraction is done in the following way:

1. Click the star icon on the data item to mark the scattering profile that should be used for subtraction (typically the background or buffer scattering profile).
2. *Select* another data item, or multiple items.
3. Either click the "Subtract" button or right-click and select "Subtract" from the popup menu. Note: this subtracts the marked item (step 1) from the selected items (step 2).

After subtraction, a new data item will be displayed in the Manipulation panel with the filename in red and a 'S_' prefix. All subtracted items are initially displayed on the bottom plot of the main plot, regardless of where the items being subtracted are displayed.

A record of which profiles were involved in the subtraction is saved in the metadata for the subtracted profile This can be viewed within RAW by right clicking on the item and selecting "Show history", or externally if the scattering profile is saved as a .dat file and opened by an external text editor. This history contains all of the history of the individual files, so if you subtract a set of averaged (or subtracted) files, you can dig down through the metadata to see what files were used at each processing step.

Data with different q vectors is handled as follows. First, the range of overlap is established. Second, RAW checks whether these overlapping regions have matching q vectors, if so, RAW carries out the subtraction. If not, RAW bins both curves so that the q vectors in the overlapping region match, and then carries out the subtraction.

Superimposing data

Scattering profiles can be superimposed on each other automatically. This function tries to calculate a scale and an offset that minimizes difference between two or more curves using a least-squares method. To superimpose:

1. Click the star icon on the data item you want the other items superimposed to.
2. *Select* one or more data items, that you wish to superimpose onto the starred item.
3. Click the "Superimpose" button

Merging data

Merging scattering profiles is typically done if you have measured multiple q ranges using different detectors. RAW can merge these profiles into one single profile. To merge profiles in RAW:

1. Click the star icon on one of the curves. This items name will be used for the name of the merged item, otherwise it does not matter.
2. *Select* one or more data items, that you wish to merge together with the starred item
3. Click the "Merge" button, select "Merge" from the right click menu, or select "Merge" from the Tools->Operations menu.

After merging, a new data item will be displayed in the Manipulation panel with a filename with a 'M_' prefix. All merged items are initially displayed on the top plot of the Main plot, regardless of where the items being merged are displayed.

Merging will average together the overlapping regions, if any exist. It can be done on an arbitrary number of data items, and is done in serial. So first the two items with the two lowest q vectors are merged, then that item is merged with the data item with the next lowest q vector, and so on. For example, if you had selected items with a minimum q vector of 0.01, 0.25, and 0.75, it would first merge the 0.01 and 0.25 items, and then merge that resulting item with the 0.75 item.

A record of which profiles were merged is saved in the metadata for the averaged profile. This can be viewed within RAW by right clicking on the item and selecting "Show history", or externally if the scattering profile is saved as a .dat file and opened by an external text editor.

Note: If items are merged without an overlap region, there will simply be no data in those intervening points.

Rebinning data

Rebinning scattering profiles is typically done to improve signal to noise of individual points. As most scattering profiles are significantly oversampled (compared to the Shannon sampling frequency), there is no loss of information from doing this. To rebin curves in RAW:

1. *Select* one or more data items, that you wish to rebin with the same binning factor.
2. Select "Rebin" from the right click menu, or select "Rebin" from the Tools->Operations menu.
3. Select the bin reduction factor, and if you want logarithmic binning check the logarithmic box. Note: the log binning is log base 10.
4. Click "OK".

After rebinning, a new data item will be displayed in the Manipulation panel with a filename with an 'R_' prefix. All rebinned items are initially displayed on the top plot of the Main plot, regardless of where the items being rebinned are displayed.

A record of which profiles were averaged is saved in the metadata for the rebinned profile. This can be viewed within RAW by right clicking on the item and selecting "Show history", or externally if the scattering profile is saved as a .dat file and opened by an external text editor.

Note: The bin reduction factor can be thought of as how many q points will go into a single bin in the new profile. So a bin reduction factor of 2 will result in half as many q points in the new profile, with each q point being the average of two q points from the previous profile (note that this may not quite be true for q points at the start and end of the profile). Rebinning will slightly reduce the q range of the scattering profile, as the new q value will be less than the extreme values in each bin.

Interpolating data

RAW can interpolate a scattering profile to find values at all points in a reference q vector. This can be useful for creating scattering profiles with perfectly matched q-vectors. To interpolate curves in RAW:

1. Click the star icon on the scattering profile you wish to interpolate to. The q vector for this profile will be used as the interpolation points for the other profile(s).
2. *Select* one or more data items, that you wish to interpolate to match the q vector of the starred item
3. Select "Interpolate" from the right click menu, or select "Interpolate" from the Tools->Operations menu.

After interpolating, a new data item will be displayed in the Manipulation panel with a filename with an 'I_' prefix. All interpolated items are initially displayed on the top plot of the Main plot, regardless of where the items being merged are displayed.

Interpolation works as follows: The overlapping q region is found for the two curves. Within this region, for every point in the q vector of the reference (starred) scattering profile an intensity value is found from the item being interpolated via linear interpolation (using the scipy.interpolate.interp1d function). A new scattering profile is then created from these interpolated values and the reference q vector.

A record of which profiles were averaged is saved in the metadata for the averaged profile. This can be viewed within RAW by right clicking on the item and selecting "Show history", or externally if the scattering profile is saved as a .dat file and opened by an external text editor.

Converting q-scale

Q-values may sometimes be in inverse angstroms (1/Å), while others use inverse nanometers (1/nm). The q vector can be multiplied or divided by 10 to accommodate for this difference when comparing two curves on different q-scales.

1. Select the data item(s) to be scaled and right-click on them.
2. Select the "Convert q-scale" option and choose whether to adjust the q-scale upwards or downwards by a factor of 10.

Note: All of the processing in RAW assumes that the scattering profiles are in inverse angstroms.

Normalizing by concentration

The overall intensity of the scattering profile should be proportional to the concentration, so it can be useful to normalize scattering profiles by the measured concentration value. To do so:

1. Set the concentration of the item in the *Information panel* or the *MW window*.
2. Select the item to be normalized.
3. Right click on the item and select "Normalize by conc"

Note: This function simply calculates 1/conc and sets that value as the scale factor. This is equivalent to doing the same calculation and manually setting the scale factor of the scattering profile.

Saving analysis results

After analyzing the data using the Guinier, MW, BIFT, or GNOM panels, the resulting parameters such as Rg, I(0), MW, Dmax, and others can be saved to a comma separated format that easily loads into a spreadsheet program such as Excel. To do so:

1. Select the data items of interest.
2. Right-click on a selected data item and select “Save all analysis info”
3. Choose a name and a location for the .csv file
4. Load the .csv file into a spreadsheet program.

This saves a spreadsheet where each row is an individual scattering profile, and each column an analysis parameter.

Saving select data item information

Information about data items including any analysis parameters, header file information, image header information, scale factor, offset, concentration, and notes can be selectively saved to a comma separated format that easily loads into a spreadsheet program such as Excel. To do so:

1. Select the data items of interest
2. Right-click on a selected data item and select “Save item info”
3. A window will pop up with two panels. The panel on the left is all of the information available about the selected items, sorted into appropriate categories (such as Image Header and Guinier Analysis).
4. In the window, select the information of interest in the left hand panel by clicking (including Ctrl and Shift clicking) on it.
5. Once all of the variables of interest are selected (blue), click the “->” button to put them in the list of data that will be saved. They should appear in the right hand panel.
6. If you accidentally selected an item, select the item and click the “<-“ button to remove it from the list of data to be saved.
7. You can add additional items to the list by repeating steps 4 and 5 until everything you want to save is selected.
8. Click the “OK” button.
9. Choose a name and a location for the .csv file
10. Load the .csv file into a spreadsheet program.

This saves a spreadsheet where each row is an individual scattering profile, and each column one of the selected variables.

Note: This differs from the save function described *above* in two ways. First, it allows you select which information you want to save. Second, it can save header information and other parameters about the scattering profile beyond the analysis information.

Data point browsing

The q, I, and I error values for each individual point in a scattering profile on the data curve can be inspected using the data browser. To do so:

1. Right-click on the data item of interest.
2. Select “Show data” in the popup menu.

Displaying the detector image

If the plotted data was loaded from an image, the image can be shown in the Image tab of RAW. To do so, right-click on the data item of interest and click “Show image” in the popup menu.

Displaying the header information

If the plotted data has header information, either from an image header or separate header file, this can be displayed in RAW. To do so, right-click on the data item of interest and click “Show header” in the popup menu.

Displaying the history information

If the plotted data has history information, either from integration of an image or from manipulation inside RAW such averaging or subtraction, this can be displayed in RAW. To do so, right-click on the data item of interest and click “Show history” in the popup menu.

The Manipulation data item right click menu options

When you right click on a data item, a popup menu is shown. This section describes what each item on the menu does.

Subtract

Carries out *subtraction*.

Average

Carries out *averaging*.

Merge

Carries out *merging*.

Rebin

Carries out *rebinning*.

Interpolate

Carries out *interpolation*.

Use as MW Standard

Sets the selected item to be the MW standard used in the *I(0) reference MW* calculation method. To do this the concentration must be set, for example in the Information Panel (it can also be set in the MW panel). A Guinier fit must also have been done, as this requires a known I(0) value for the reference scattering profile. In the popup window that opens, enter the known protein MW in units of kDa.

Normalize by conc

Normalizes the profile by the concentration.

Remove

Removes the item.

Guinier fit

Opens the Guinier fit analysis panel for the selected scattering profile.

Molecular Weight

Opens the molecular weight analysis panel for the selected scattering profile.

BIFT

Opens the Bayesian indirect Fourier transform (BIFT) analysis panel for the selected scattering profile.

GNOM (ATSAS)

Opens the GNOM analysis panel for the selected scattering profile. Note: this option is only available if RAW has detected a valid ATSAS package installation.

SVD

Opens the singular value decomposition (SVD) analysis panel for the selected scattering profiles (must have at least 2 profiles selected).

EFA

Opens the evolving factor analysis (EFA) panel for the selected scattering profiles (must have at least 2 profiles selected).

Convert q-scale

Has a submenu with two options: **q * 10** which multiples the scattering profile q vector by a factor of 10, and **q / 10** which divides the scattering profile q vector by a factor of 10.

Show image

If the scattering profile was integrated from an image, and the image is still in the same location as when the profile was first integrated, it displays the image in the Image Plot Panel.

Show data

Shows the q, I(q) and I_error(q) data.

Show header

Shows the header information.

Show history

This shows the history of the scattering profile.

Move to top plot

Moves the scattering profile to the top subplot of the Main plot (no effect if the scattering profile is already plotted on the top plot).

Move to bottom plot

Moves the scattering profile to the bottom subplot of the Main plot (no effect if the scattering profile is already plotted on the top plot).

Save all analysis info

Saves all of the analysis info of the selected data item(s).

Save item info

Saves select item info.

Save selected file(s)

Saves the selected data item(s).

The Manipulation panel bottom buttons

There are six buttons at the bottom of the Manipulation panel. They are:

Save

This button saves the selected data item(s).

Sync

This button syncs the settings of the selected data item(s) to the starred data item.

Remove

This button removes the selected data item(s) from the Manipulation panel.

Superimpose

This button superimposes the selected data item(s) onto the starred data item.

Average

This button averages the selected data item(s).

Subtract

This button subtracts the selected data item(s).

The Main Plot window

The Main Plot window displays individual scattering profiles associated with items in the Manipulation panel. It has two subplots, which by default are titled "Main Plot" (top) and "Subtracted" (bottom). Any item that is initially loaded is plotted on the top plot.

The features that are general between all of the plots are described *elsewhere*. This section will describe features unique to this plot.

Changing axes and plot types

Right-click in the plot window to view a pop-up menu with different axis settings.

The available plot modes are:

- Lin-Lin
- Log-Lin
- Log-Log
- Lin-Log
- Guinier plot (ln(I(q)) vs. q^2)
- Porod plot (q^4I(q) vs. q)
- Kratky plot (q^2I(q) vs. q)

The Main plot toolbar

In addition to the plot toolbar buttons *shared by all of the plots*, the Main plot has the following buttons:

Toggle errorbars. Shows the errorbars on the plotted curves.

Top/Bottom plot. Shows both the top and the bottom plot.

1

Top plot. Shows only the top plot.

2

Bottom plot. Shows only the bottom plot.

Moving profiles between plots

Scattering profiles can be moved between the top and bottom plots. To do so:

1. Select the data item(s) of interest in the Manipulation panel.
2. Right-click on a selected data item and select “Move to top plot” or “Move to bottom plot” from the popup menu.

Setting the legend label

Instead of the filename showing in the legend, a special label can be created for that data item.

1. Click on the *Line Properties* button of the data item.
2. In the Line Properties window that pops up, enter the desired legend label in the Legend Label box.
3. Click “OK”.

The new legend label will be shown in [square brackets] next to the data item name. To revert to t he standard label (the filename), erase the custom label.

Locating data items

The data item associated with a plotted scattering profile can be quickly found by clicking plotted curve. When a curve is clicked on, it will flash ‘bold’ (increased line weight), and the associated data item will be selected in the Manipulation panel.

Note: If there are other selected items in the Manipulation panel, they will be unselected.

5.8.9 The IFT Control Panel and IFT Plot Panel

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

When inverse Fourier transforms (P(r) functions) are loaded into RAW or generated by BIFT or GNOM in RAW, they are placed in the IFT tab of the Control Panel, and displayed in the IFT Plot window. This section will cover the features of the IFT control panel and IFT plot panel.

The IFT control panel

The IFT control panel refers to the panel that is shown in the Control panel when the IFT tab is selected.

The panel consists of two parts. The top part is where individual IFT items loaded into RAW are shown. The bottom part consists of buttons that allow you to save or remove those IFT items. Further manipulation of the IFT items can be done from the right click (context) menu and from the Tools menu.

An individual item in the IFT list is called an IFT data item. It is associated with a P(r) function, the scattering profile used to generate that P(r) function, and the scattering profile generated from the P(r) function.

All items loaded into the IFT panel have a P(r) function displayed in the top plot of the IFT Plot window, and an experimental scattering profile and a scattering profile from the P(r) function displayed in the bottom plot of the IFT Plot window.

IFT Data Items

When a new item is plotted, a data item is added to the IFT list in the IFT tab. This allows you to control the properties of the data item, and perform analysis on it.

The name of the data item is displayed for each item. If an item is given a different name for the plot legend, this legend name is displayed in [square brackets] next to the item name. On the same line as the item name, on the right side of the data item, there are several buttons that can be used for *further manipulation* of the data item.

Note: If there is a * to the left of the item name (between the checkbox and the item name), it indicates there are unsaved changes to the item. This can occur if the item is newly created IFT data (such as from the BIFT or GNOM panels), or if item properties such as the name have been changed.

Showing/Hiding data items on the IFT plot

To show/hide all of the curves (P(r), experimental data, scattering profile from P(r) curve) associated with a data item on the plot, click the checkbox next to the filename. If the checkbox is checked, the item is currently shown on the IFT plot, if the checkbox is unchecked, the item is currently hidden on the IFT plot.

Data item buttons

On the right of the data are a series of buttons for controlling the data item. In order of left to right, these buttons have the following properties.

Locate Line

The target button is used to highlight the scattering profile on the graph that is associated with the data item. When the target is pressed, it ‘bolds’ the line each plotted curve associated with the data item (increases the line width by several points). When the target is pressed again, the line width is set back to normal. You can tell if a line is currently bolded, as the target will be orange instead of grey.

Line Properties

The colored line button has two purposes. First, the color matches the current color of the P(r) function in the IFT Plot. Second, when pressed it opens a *line properties dialog* which allows you to set the legend label; the line style, width, and color; the data point marker style, size, line color, and fill color; and the error bar line style, width, and color for each line associated with the IFT data item.

Selecting data items

A single data item can be selected by clicking on the item name in the IFT list (similar to how you would select files in your system file browser). When an item is selected, the color of the item background changes from white to gray. If the item is currently selected, clicking on it will cause it to be unselected. Note that for a regular click, all other selected items will be unselected when a new item is selected.

Multiple items may be selected in two ways. If the Control key (Command key on Macs) is held down while clicking on items, each item that is clicked on will be added to the set of selected items. If a single item is first selected and then the Shift key is held down and another item is selected, all of the items in the list between the two items will be selected (including the second item that is clicked on).

All of the items in the list can be selected in two ways. The first is using the *select all* button, the second is pressing Ctrl-A (Cmd-A), the Control (Command) key and the A key at the same time when you are in the IFT panel. All items can be unselected by clicking in an empty spot of the IFT list (but not above or below the list), or by clicking on an already selected item.

Note: If you have a set of selected items and wish to remove some, holding down the Control (Command) key and clicking on selected items will deselect them without affecting the other selected items.

The top buttons of the IFT Panel

The IFT Panel has a set of three buttons at the top of the panel. These buttons have the following effects, listed from left to right.

Show All

Clicking on the button that looks like an eye will show all IFT items. This is the same as if you manually set all of the show/hide checkboxes in the data items to on.

Hide All

Clicking on the button that looks like an eye with a red x through it will hide all IFT items. This is the same as if you manually set all of the show/hide checkboxes in the data items to off.

Select All

Clicking on the button that looks like a spreadsheet with selected cells will select all of the IFT data items.

Renaming a data item

Data items can be renamed by selecting the data item of interest and selecting “Rename” in the right click popup menu.

Note: While no characters are expressly forbidden in the filename, RAW does not sanitize file names before saving, and thus special characters such as ‘/’ and ‘\’ are likely to cause problems when the file is saved.

Saving data items

IFT items are saved in two different formats, depending on whether the item was generated by BIFT (“.ift”) or GNOM (“.out”). The procedure to save either is the same. To save:

1. Select the item(s) to be saved.
2. Click the “Save” button or select “Save selected file(s)” from the right click menu.
3. In the window that pops up, navigate to the directory in which you want to save the files.

4. If you are saving a single item, the window will give you an opportunity to rename your file if desired. Click "Save" when ready.
5. If you are saving multiple items, you simply need to select the folder for the items to be saved in, and click "Open". The items will be saved with the same names displayed in the IFT Panel, in the folder that you chose.

BIFT items are saved as ".ift" files, which is a text file with RAW specific formatting of the text. The first two lines are "BIFT" and the "Filename: <filename>". After that, the P(r) function is saved as 3 column data. The first column is "R", the second column is "P(R)" and the third column is "Error", these headers are included as the third line of the file. After the P(R) function there are two blank lines, followed by the scattering data. This is saved in four columns, the first three are "Q", "I(q)" and "Error" which correspond to the experimental data, while the fourth column is "Fit" which is the scattering profile from the P(r) function. After this data is written there is a "header" written, which consists of the Chisquared, algorithm used, I(0) value, log base 10 of the alpha value, Dmax, Rg, and the filename, all saved in JSON format. The files are simply text files, and can be opened and viewed in any standard text editor.

GNOM items are saved in the standard ".out" format. This is described in the ATSAS manual for GNOM. These files can be directly input into any ATSAS (or other) program that requires a GNOM .out file as input.

Removing data items from the IFT list

To remove one or more data items, select them and do one of the following:

1. Press the "Delete" key on the keyboard
2. Click the "Remove" button
3. Select "Remove" from the right click menu

Sending data to the main plot

If you wish to examine the experimental scattering profile and fit to this profile from the P(r) function more closely, the data can be sent to the main plot. To do this, right click on the IFT data item and select "To Main Plot". This will plot items on the Main plot and add them to the Manipulation list.

For a ".ift" file generated by BIFT, the following items will be added to the Manipulation list and Main plot. In all cases, <filename> corresponds to the filename of the IFT data item without the extension (so "my_ift".ift would have a filename of "my_ift").

<filename>_data – This is the experimental data that was used to generate the P(r) curve.

<filename>_fit – This is the scattering profile generated from the P(r) curve.

For a ".out" file generated by GNOM, the same two curves as for a BIFT item (above) are sent to the main plot, there is also one additional file.

<filename>_extrap – This is the scattering profile generated from the P(r) curve, extrapolated to q=0.

Running DAMMIF on the P(r) function

RAW allows you to run DAMMIF on a P(r) function from within RAW. Currently, this can only be done on P(r) items generated by GNOM (these can be generated in RAW, or loaded in after being generated outside of RAW). To run DAMMIF, select an appropriate IFT data item (a ".out" item), and either right click and select the "Run DAMMIF" option or from the Tools->ATSAS menu select "DAMMIF". This opens the *DAMMIF window*.

Running AMBIMETER on the P(r) function

RAW allows you to run AMBIMETER on a P(r) function from within RAW. Currently, this can only be done on P(r) items generated by GNOM (these can be generated in RAW, or loaded in after being generated outside of RAW). To run AMBIMETER, select an appropriate IFT data item (a ".out" item), and either right click and select the "Run AMBIMETER" option or from the Tools->ATSAS menu select "AMBIMETER". This opens the *AMBIMETER window*.

Data point browsing

Each individual point of the r; P(r); error in P(r); experimental q, I(q), and error; I(q) from the P(r) function; and, for GNOM generated IFT data items, the q and I(q) values extrapolated to q=0 vectors; can be inspected using the data browser. To do so:

1. Right-click on the data item of interest.
2. Select "Show data" in the popup menu.

The IFT data item right click menu options

When you right click on a data item, a popup menu is shown. This section describes what each item on the menu does.

Remove

Removes the item.

To Main Plot

This *sends the scattering profile data to the main plot*.

Run DAMMIF

This item is only available for ".out" files generated by GNOM. It opens the *DAMMIF window*.

Run AMBIMETER

This item is only available for ".out" files generated by GNOM. It opens the *AMBIMETER window*.

SVD

Opens the singular value decomposition (SVD) analysis panel for the selected scattering profiles (must have at least 2 items selected).

EFA

Opens the evolving factor analysis (EFA) panel for the selected scattering profiles (must have at least 2 items selected).

Show data

Shows the *individual data points*.

Rename

Renames the data item,

Save selected file(s)

Saves the selected data item(s).

The IFT panel bottom buttons

There are three buttons at the bottom of the IFT control panel. They are:

Save

This button *saves the selected data item(s)*.

Remove

This button *removes the selected data item(s)* from the IFT panel.

Clear IFT Data

This button clears all loaded IFT data. It works the same as if you had selected all of the IFT data items and then removed them.

The IFT Plot window

The IFT Plot window displays P(r) data (top plot), the scattering profiles generated from the P(r) data (bottom plot), and the experimental scattering profile used to generate the P(r) data (bottom plot). Each set of three curves is associated with a single IFT data item in the IFT control panel. The two subplots are the Pair Distance Distribution Function (top) and Data/Fit (bottom) plots.

The features that are general between all of the plots are described *elsewhere*. This section will describe features unique to this plot.

Changing axes and plot types

Right-click in the Data/Fit (top) plot to view a pop-up menu with different axis settings.

The available plot modes are:

- Lin-Lin
- Log-Lin
- Log-Log
- Lin-Log
- Guinier plot (ln(I(q)) vs. q^2)
- Porod plot (q^4I(q) vs. q)
- Kratky plot (q^2I(q) vs. q)

The axes cannot be changed for the P(r) (top) plot.

The IFT plot toolbar

In addition to the plot toolbar buttons *shared by all of the plots*, the IFT plot has the following buttons:

Toggle errorbars. Shows the error bars on the plotted curves.

Top/Bottom plot. Shows both the top and the bottom plot.

Top plot. Shows only the top plot.

Bottom plot. Shows only the bottom plot.

5.8.10 The SEC Control Panel and SEC Plot Panel

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

When SEC-SAXS data is loaded into RAW, it is placed in the SEC tab of the Control Panel, and displayed in the SEC Plot window. This section will cover the features of the SEC control panel and the SEC plot panel.

SEC-SAXS data is assumed to come from continuous collection of images while sample is being eluted from an FPLC column. Depending on the frame rate of the detector and the total elution time, this can generate thousands of images. The basic approach RAW uses to deal with this data is to load every scattering profile (or integrate every image into a scattering profile), and associate it with a single SEC data item. For each SEC item, the total (or average) intensity in each scattering profile is plotted vs. frame number, where frame number is simply what position the item was loaded in (so if 10 items are loaded, the frame number for the first item loaded would be 0, for the next item loaded 1, etc). For standard SEC-SAXS data, this intensity vs. frame number should look similar to the UV chromatograph of the sample after it passes through the column. Some analysis can be carried out directly on this SEC data item, and the scattering profiles of interest can be extracted for further analysis.

The SEC control panel

The SEC control panel refers to the panel that is shown in the Control panel when the SEC tab is selected.

The panel consists of three parts. The top part is a set of controls for loading SEC items, controlling the SEC online mode, and analyzing SEC items. The middle part is where individual SEC items loaded into RAW are shown. The bottom part consists of buttons that allow you to save or remove the SEC items. Further manipulation of the SEC items can be done from the right click (context) menu.

An individual item in the SEC list is the called a SEC data item. It is associated with two curves on the SEC plot, one plotted on the right y axis and one on the left y axis.

SEC Data Items

When a new item is plotted, a data item is added to the SEC list in the SEC control panel. This allows you to control the properties of the data item, and perform analysis on it.

The name of the data item is displayed for each item. If an item is given a different name for the plot legend, this legend name is displayed in [square brackets] next to the item name. On the same line as the item name, on the right side of the data item, there are several buttons that can be used for further manipulation of the data item.

SEC data items are different from Manipulation and IFT data items as they have many (often hundreds or thousands) of scattering profiles associated with a single data item, and they can have new profiles appended to them after they are loaded.

Note: If there is a * to the left of the item name (between the checkbox and the item name), it indicates there are unsaved changes to the item. This can occur if the item is newly created IFT data (such as from the BIFT or GNOM panels), or if item properties such as the name have been changed.

Showing/Hiding data items on the SEC plot

To show/hide the curves on the plot associated with a data item, click the checkbox next to the filename. If the checkbox is checked, the item is currently shown on the SEC plot, if the checkbox is unchecked, the item is currently hidden on the SEC plot.

Data item buttons

On the right of the data item are several buttons for controlling the data item. In order of left to right, these buttons have the following properties.

Extended Info

By hovering the mouse over the i in the blue circle button, a tooltip will appear which shows more information about the item. It shows the buffer range and window size used (if any) to calculate the structural parameters of the SEC data item as a function of frame number.

Locate Line

The target button is used to highlight the scattering profile on the graph that is associated with the data item. When the target is pressed, it 'bolds' the line each plotted curve associated with the data item (increases the line width by several points). When the target is pressed again, the line width is set back to normal. You can tell if a line is currently bolded, as the target will be orange instead of grey.

Line Properties

The colored line button has two purposes. First, the color matches the current color of the Intensity curve in the SEC Plot. Second, when pressed it opens a *line properties dialog* which allows you to set the legend label; the line style, width, and color; the data point marker style, size, line color, and fill color; and the error bar line style, width, and color for each line associated with the SEC data item.

Mark

The star button marks an item. This is used when doing operations such as sending data frames to plot or calculating structural parameters. In those cases, the marked (also referred to as starred) item has a special significance.

Loading data items

SEC data items can be loaded from the *files tab*. There is an additional way to load files available in the SEC control panel, this method is intended to be used in conjunction with the SEC online mode. This method is described in the rest of this section.

At the top of the SEC control panel, there is a section for loading files. It has a button "Select file in SEC run", and then descriptive information about the file: the image prefix, the run number, and the frame numbers of the file. Note that none of these descriptive fields can be edited. Below the descriptive fields is an "Update" button and an "Automatically Update" checkbox.

To load a SEC curve using this method, click the "Select file in SEC run" button. In the file browser that pops up, select any file in the SEC data set. The data set will automatically be loaded into RAW as a SEC item, and the descriptive information will be displayed in the appropriate boxes. This loaded file can then be used with the SEC automatic update mode described *below*.

Note: Due to how files are loaded from the SEC control panel, it only works for certain header file types. RAW must be able to automatically determine what files are associated with the SEC run, from any file in the run. At the moment, this can only be done with the G1 and G1 WAXS header file formats. If you want to have a particular beamline's file format added to this loading (and thus able to use the online mode) please contact the RAW developers.

Updating a SEC data item

If a SEC data item is loaded via the SEC control panel, it can be updated if additional files are added to the folder that are part of the data collection. This can only be done for the most recent file loaded via the SEC control panel, for which the descriptive information is shown at the top of the SEC control panel.

To do this update, hit the "Update" button on the SEC control panel. RAW will automatically determine all of the files associated with the SEC run, based on the file you selected with the "Select file in SEC run" button, determine if any of those files have not yet been loaded, and if so, load the files and add them to the SEC item as appropriate.

This is useful when working at the beamline while data is being collected, as you may want to start analysis of the SEC curve before all of the images are taken.

SEC automatic update (online mode)

The SEC control panel can be used in an online mode, where a SEC curve is automatically updated as data comes in. All of the conditions for manually updating a curve, described above, must be met. Instead of using the "Update" button as in that section, check the "Automatically Update" box in the SEC control panel.

Note: The automatic update will stay on even if you load a new file into the SEC panel using the "Select file in SEC run" button. It will switch to updating this newly loaded file. As with the "Update" button described above, the automatic update only applies to the SEC data item most recently loaded in the SEC panel.

Selecting data items

A single data item can be selected by clicking on the item name in the SEC list (similar to how you would select files in your system file browser). When an item is selected, the color of the item background changes from white to gray. If the item is currently selected, clicking on it will cause it to be unselected. Note that for a regular click, all other selected items will be unselected when a new item is selected.

Multiple items may be selected in two ways. If the Control key (Command key on Macs) is held down while clicking on items, each item that is clicked on will be added to the set of selected items. If a single item is first selected and then the Shift key is held down and another item is selected, all of the items in the list between the two items will be selected (including the second item that is clicked on).

All of the items in the list can be selected in two ways. The first is using the *select all* button, the second is pressing Ctrl-A (Cmd-A), the Control (Command) key and the A key at the same time when you are in the SEC panel. All items can be unselected by clicking in an empty spot of the SEC list (but not above or below the list), or by clicking on an already selected item.

Note: If you have a set of selected items and wish to remove some, holding down the Control (Command) key and clicking on selected items will deselect them without affecting the other selected items.

The top buttons of the SEC item list

The SEC item list has a set of three buttons at the top of the panel. These buttons have the following effects, listed from left to right.

Show All

Clicking on the button that looks like an eye will show all SEC items. This is the same as if you manually set all of the show/hide checkboxes in the data items to on.

Hide All

Clicking on the button that looks like an eye with a red x through it will hide all SEC items. This is the same as if you manually set all of the show/hide checkboxes in the data items to off.

Select All

Clicking on the button that looks like a spreadsheet with selected cells will select all of the SEC data items.

Renaming a data item

Data items can be renamed by selecting the data item of interest and selecting "Rename" in the right click popup menu.

Note: While no characters are expressly forbidden in the filename, RAW does not sanitize file names before saving, and thus special characters such as '/' and '\' are likely to cause problems when the file is saved.

Saving data items

SEC items are saved as ".sec" files, and is the only data that RAW does not save in a human readable format. To save:

1. Select the item(s) to be saved.
2. Click the "Save" button or select "Save selected file(s)" from the right click menu.
3. In the window that pops up, navigate to the directory in which you want to save the files.
4. If you are saving a single item, the window will give you an opportunity to rename your file if desired. Click "Save" when ready.
5. If you are saving multiple items, you simply need to select the folder for the items to be saved in, and click "Open". The items will be saved with the same names displayed in the SEC Panel, in the folder that you chose.

SEC items often contain hundreds or thousands of scattering profiles, so they are not saved in a human readable format. The ".sec" files that RAW saves can only be read by RAW.

Removing data items from the SEC list

To remove one or more data items, select them and do one of the following:

1. Press the "Delete" key on the keyboard
2. Click the "Remove" button
3. Select "Remove" from the right click menu

Exporting SEC data

The following data can be exported in a spreadsheet ready format: frame number, integrated intensity, mean intensity, Rg, Rg error, I(0), I(0) error, MW, filename, and, if available intensity a q=<#> where <#> is a user selected value, for each individual point.

To do so:

1. Select the item(s) to be saved.
2. Select "Export data" from the right click menu.
3. In the window that pops up, navigate to the directory in which you want to save the file(s).
4. If you are saving a single item, the window will give you an opportunity to rename your file if desired. Click "Save" when ready.
5. If you are saving multiple items, you simply need to select the folder for the items to be saved in, and click "Open". The items will be saved with the same names displayed in the SEC Panel, in the folder that you chose.

The data is saved as a comma separated value (".csv") file. This can be opened directly into most spreadsheet programs, such as Excel.

Saving all SEC scattering profiles

If you want to save every individual scattering profile loaded into a SEC data item, you can do so by:

1. Select the item(s) to be saved.
2. Select "Save all profiles as .dats" from the right click menu.
3. In the window that pops up, navigate to the directory in which you want to save the file(s).
4. Select the folder for the items to be saved in, and click "Open". The items will be saved with the same filenames displayed in the *data browser*.

Sending data to the main plot

Individual scattering profiles can be sent to the main plot for further analysis. This utilizes the middle section of the controls at the top of the SEC Control panel.

To do so:

1. Star the SEC data item containing the scattering profiles of interest (note: if only one SEC data item is loaded, it does not have to be starred).
2. Enter the data frames of interest in the "Select Data Frames:" boxes. The box on the left is the first frame of interest, the box on the right is the last frame of interest. All of the frames between those two endpoints (inclusively) are selected.
3. Either click the "Frames To Main Plot" button, which will send each individual frame selected in part 2 to the Main plot, or click "Average To Main Plot" which will send the average of the selected frames to the Main plot.
4. Click on the Manipulation panel and Main plot panel to view the scattering profiles.

Calculate structural parameters

You can calculate the Rg, MW, and I(0) as a function of frame number for a SEC profile. This is done using the "Calculate/Plot Structural Parameters" section of the SEC control panel (the bottom section of the controls at the top of the panel).

It is important to have a big picture for what is happening when this is done. First you set a buffer range. All of the scattering profiles in that range will be averaged, and then subtracted from every loaded scattering profile. Next you set a window size. This window will be slid across the curve and all of the frames within it averaged (note: this average is of the buffer subtracted scattering profiles). This Rg, MW, and I(0) for this averaged profile is then calculated. Those values are assigned to the center frame of the window. The window is then slid down the curve one frame, and the process is repeated until the window reaches the end of the SEC data. For example, if the window size is 5, the first 5 frames, frames 0, 1, 2, 3, and 4, are averaged, and have the Rg, MW, and I(0) calculated. Then window is moved over by one, and frames 1, 2, 3, 4, and 5 are averaged and have the Rg, MW, and I(0) calculated, and so on.

To do this:

1. Star the SEC data item containing the scattering profiles of interest (note: if only one SEC data item is loaded, it does not have to be starred).
2. Enter the buffer range in the "Buffer Range" boxes. The box on the left is the starting buffer frame and the box on the right is the final buffer frame.
3. Enter the window size.
4. Select the appropriate molecule type (Protein or RNA) for the molecular weight calculation.
5. Click the "Set/Update Parameters" button.
6. The structural parameters will be plotted as a function of frame number in the SEC plot. If RAW was unable to determine the parameters for a particular window, then all parameters in that window are set to -1.

The Rg and I(0) calculations are done using RAW's *automatic rg function*. The molecular weight calculation is done using the *volume of correlation method*.

Note: See *below* for how to show different structural parameters on the SEC plot.

Data point browsing

The Frame number, integrated intensity, mean intensity, Rg, Rg error, I(0), I(0) error, MW, filename, and, if available intensity a q=<#> where <#> is a user selected value, for each individual point can be inspected using the data browser. To do so:

1. Right-click on the data item of interest.
2. Select "Show data" in the popup menu.

The SEC data item right click menu options

When you right click on a data item, a popup menu is shown. This section describes what each item on the menu does.

Remove

Removes the item.

Export data

Exports the SEC data item to a spreadsheet.

Save all profiles as .dats

Saves all of the scattering profiles loaded into the SEC item as individual .dat files.

Save

Saves the selected data item(s).

SVD

Opens the singular value decomposition (SVD) analysis panel for the selected SEC item.

EFA

Opens the evolving factor analysis (EFA) panel for the selected SEC item.

Show data

Shows the individual data points.

Rename

Renames the data item.

The SEC panel bottom buttons

There are three buttons at the bottom of the SEC control panel. They are:

Save

This button saves the selected data item(s).

Remove

This button removes the selected data item(s) from the SEC panel.

Clear SEC Data

This button clears all loaded SEC data. It works the same as if you had selected all of the SEC data items and then removed them.

The SEC Plot window

The SEC plot window has only one plot, however data can be plotted on both the left and right y axes. The left axis plots intensity from scattering profiles, while the right axis plots structural parameters (Rg, MW, I(0)). The items associated with the plotted curves are shown in the SEC control panel list.

The features that are general between all of the plots are described *elsewhere*. This section will describe features unique to this plot.

Changing axes and plot types

Right-click in the plot to view a pop-up menu with different axis settings. In this plot, you can change what data is plotted on each axis.

Y Data (Left Axis)

This allows you to change intensity plotted on the left y axis. The methods are:

- Integrated intensity (default), which is

$$I_{tot} = \int_{q_{min}}^{q_{max}} I(q)dq$$

- Mean intensity, which is the average intensity across the whole scattering profile
- Intensity at q=. . . , which allows the user to specify a q value, and displays the intensity at the nearest q point to that specified value.

Y Data (Right Axis)

This allows you to change which calculated structural parameter is plotted on the right y axis.

1. RG (default) which plots the Rg
2. MW which plots the molecular weight
3. I(0) which plots the I(0)
4. None, which turns off the right y axis.

X Data

This allows you to change what is plotted on the x axis.

- Frame Number, which plots the intensity and structural parameter as a function of frame number.
- Time, which puts the x axis in terms of experimental time.

Note: The time display on the x axis is only available for certain types of header files. Currently only G1 and G1 WAXS header files will have associated time values.

5.8.11 The Image plot panel

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

The Image plot panel displays any detector image that RAW is able to read. The image name is shown at the top of the plot, while the x and y axes are in dimensions of pixels. There is no right click menu for this plot, but scrolling with a mouse scroll wheel will zoom in and out. It is possible to display regions where there is no image data, these will show up as a white background (for example, if you zoom out and show negative pixel ranges.

In addition to the plot toolbar buttons *shared by all of the plots*, the Image plot has the following buttons:

Displays the image header. This does not display any information from separate header files, only what is in the image itself.

Opens the image display settings dialog described in below.

Shows the previous/next image in the same directory as the current image (as sorted by filename, A-Z).

The Image Display Settings dialog

The Image Display Settings dialog lets you change how the image is displayed on the screen. It has three sections, Image parameters, Image scaling, and Colormaps.

Image parameters

There are three slider bars here. The Upper and Lower limit sliders let you set what pixel values in the image correspond to the most extreme values of the color scale chosen. The Brightness color bar does not work at the moment. By default, the upper and lower limit values are automatically set each time an image is loaded into the Image plot panel. However, checking the Lock values checkbox will prevent the current set values from changing. This allows you to scroll through images maintaining constant upper and lower limits.

Image scaling

Image scaling allows you to use a linear or logarithmic (base 10) color scale. The logarithmic scale uses the matplotlib.colors.SymLogNorm function (more available here: https://matplotlib.org/api/colors_api.html), with a linear threshold set at 1. So the image is logarithmic in color display above 1, and linear in color display below 1 (this is so 0 and negative value pixels can be handled).

Colormaps

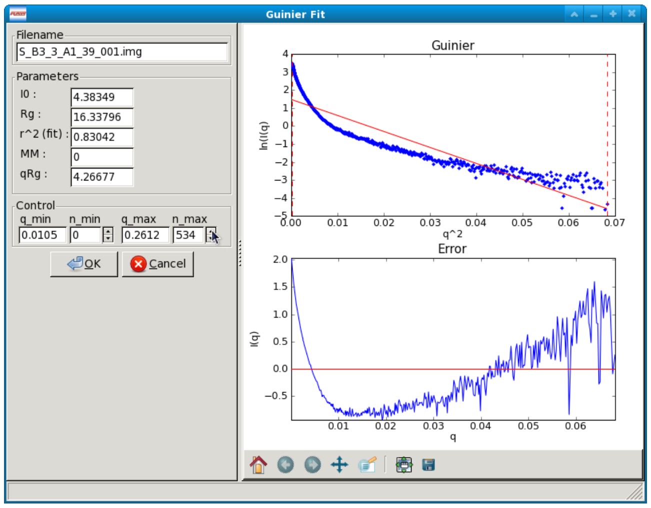
The colormaps option allows you to change the color scale used to display intensity values in the Image plot. The color maps are those available through matplotlib (https://matplotlib.org/users/colormaps.html), a limited selection of which are available in RAW: Gray, Heat, Rainbow, Jet (default), Spectral.

Note: Currently there is also a brightness setting, but it does not work properly.

5.8.12 Analysis windows

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

There are a variety of analysis windows in RAW: the Guinier Fit window, the Molecular Weight window, the BIFT window, the GNOM window, the AMBIMETER window, the DAMMIF window, the SVD window, and the EFA window. In this chapter we will cover how to open these windows and carry out the analysis.



Filename
S_B3_3_A1_39_001.img

Parameters

I0 :	4.38349
Rg :	16.33796
r^2 (fit) :	0.83042
MM :	0
qRg :	4.26677

Control

q_min	n_min	q_max	n_max
0.0105	0	0.2612	534

Guinier Fit window

Data that has been properly background subtracted can be analyzed in the standard fashion via a Guinier Fit to yield the radius of gyration and the scattering intensity at zero angle. The Guinier Fit window allows you to interactively do the fit.

Opening the Guinier fit window

The Guinier fit window can be opened by selecting a data item in the Manipulation Panel and either right clicking and selecting "Guinier fit" or selecting "Guinier fit" from the "Tools" menu. Whichever data item is selected will be the one that is being analyzed. If multiple items are selected, the first (top item in the list) will be sent to the Guinier Fit panel.

Once the window opens you will see two panels. The left panel has the filename of the data loaded into the Guinier Fit panel, the fit parameters, and the fit controls. The right panel has two plots that show the fit and residual. If no previous Guinier fit analysis has been saved for the data item, the fit range defaults to the whole scattering profile. If a previous Guinier fit analysis was saved for the data item, the fit range will be the range of that analysis.

Once you have completed a satisfactory Guinier fit, clicking the "OK" button will save the fit for the data item selected. That will allow you to save the Guinier analysis parameters when *saving all results*, view the results in the information window, and save it with the scattering profile if that is saved as a ".dat" file. Clicking the "Cancel" button (or the close window button from the operating system, typically in the upper left or right corner) will exit the window without saving the analysis results.

Note 1: Only one Guinier Fit window can be open at once. If a window is already open and you try to open a new one, it will close the current window and open a new one.

Note 2: RAW will automatically attempt to automatically determine the Rg when the Guinier window is opened, if no previous Guinier fit has been done. If a previous Guinier fit has been done, RAW will set the q min and q max to the previous values.

Guinier Fit Controls

The Guinier Fit Controls allow you to manually set the q_min and q_max values over which the Guinier fit is done. You can do this in two ways. First, if you type a value into the q_min or q_max box and hit enter, RAW will find the nearest q point, and set the q value to that nearest point. Second, you can adjust the n_min and n_max values either by typing an integer value in the appropriate box or using the spin controls. The n_min and n_max controls change the minimum and maximum points of the q vector used (zero indexed), and the q_min and q_max are updated to match. For example if n_min is set to 5 and n_max is set to 130, the 5th indexed point through the 130th indexed point in the q vector (so points 6-131, because of the zero indexing) will be used in the Guinier fit.

The AutoRG button calls a function that attempts to automatically determine the best range of data to use for the Guinier fit. It considers the following criteria: q_min*Rg as small as possible, q_max*Rg as close to 1.3 as possible, smallest fit error in the Rg, smallest fit error in the I(0) possible, r^2 of the fit as close to one as possible, and the largest range of q values used as possible.

Guinier Fit Plots

There are two plots in the Guinier fit window. The top plot is the standard Guinier plot, ln(I(q)) vs. q^2, shown in the range of n_min -20 to n_max +3 (if possible) as set in the Guinier Control panel. The scattering profile is marked by the blue points, the fit is shown as the solid straight red line, the fit extrapolated to q=0 is shown by the green dashed line, and the range of the fit, q_min and q_max, are shown by vertical red dashed lines.

The bottom plot has the residual between each point in the scattering profile and the fit, plotted as $\Delta ln(I(q))$ vs. q^2. The red line is a line at zero. Note that the x range of this plot may be slightly reduced compared to the top plot, as it only displays the fit, while the top plot can display points beyond the limits of the fit.

Note: There are no controls available on right click for these plots, but there is a standard *plot control bar*.

Guinier Fit Parameters

The fit parameters available in the Guinier Fit window are:

- I(0), the extrapolated scattering intensity at zero angle
- Rg, the radius of gyration
- r^2 (fit), the r^2 value of the fit
- qRg, which is q_min*Rg and q_max*Rg for the left and right boxes respectively.

Molecular weight

Data that has been background subtracted and had a Guinier fit carried out can be analyzed to find the molecular weight of the macromolecule. RAW provides four different ways of calculating the molecular weight, which are described below.

Opening the Molecular Weight Window

The Molecular Weight window can be opened by selecting a data item in the Manipulation Panel and either right clicking and selecting "Molecular Weight" or selecting "Molecular Weight" from the "Tools" menu. Whichever data item is selected will be the one that is being analyzed. If multiple items are selected, the first (top item in the list) will be the one analyzed.

The window has two parts. The top part is a description of the MW methods and a panel with the Guinier fit parameters. The bottom part is four panels providing the calculated MW from each method, and the ability to see more details and learn more about each method. All of the MW methods require a Guinier fit to have been done, as they rely on the I(0) value. Two of the methods require knowing the sample concentration, and the same two methods also depend on (different) calibration of the scattering profile.

Once you are satisfied with the molecular weight analysis, clicking the "OK" button will save the analysis for the data item selected. That will allow you to save the molecular weight analysis parameters when *saving all results*, view the results in the information window, and save the results with the scattering profile if that is saved as a ".dat" file. Clicking the "Cancel" button (or the close window button from the operating system, typically in the upper left or right corner) will exit the window without saving the analysis results.

Note: Only one molecular weight window can be open at once. If a window is already open and you try to open a new one, it will close the current window and open a new one.

The molecular weight controls

There are several controls available in the molecular weight panel. In the top panel, the Guinier Fit button can be used to open the Guinier panel and (re)analyze the scattering profile using that method, so that an I(0) value is available for the molecular weight panel.

Each of the four lower panels has a "Show Details" and "More Info" button. The "More Info" button simply provides an extended description of the method used, and, when relevant, a citation. The "Show Details" button provides extended information on the parameters used to calculate the MW for a given method. None of these parameters are editable in the molecular weight panel. When the "Show Details" button is clicked, it becomes a "Hide Details" button, which hides the extra parameters.

The concentration box in both the first and fourth panels can have the sample concentration entered, if it has not already been (for example, in the Information panel). These are linked, so that any change to one also changes the other. Sample concentration should be in mg/ml.

The volume of correlation method uses different parameters for Proteins and RNA, and the drop down menu can be used to toggle between those two sets of parameters. The default setting can be changed in the Options window.

If RAW is set to normalize scattering profiles to an absolute scale, the “Intensity on Absolute Scale” box will be checked in the fourth panel. If RAW is not normalizing to an absolute scale, it will be unchecked. If that is set incorrectly for a particular scattering profile for some reason, it can be manually toggled to the correct position.

At the bottom of the window there are three buttons. The “OK” and “Cancel” buttons have been described above. The “Change Advanced Parameters” button opens the Options window, with the Molecular Weight section shown. This allows you to change the parameters used to calculate the molecular weight for each method of calculation.

The molecular weight parameters

There are a number of parameters listed in the molecular weight panel. In the top panel there are:

Filename

Gives the filename of the data item being analyzed.

Guinier parameters

Gives the I(0) and Rg from the Guinier fit.

In the I(0) Ref. MW panel there are:

Concentration

The sample concentration in mg/ml, this field accepts input.

MW

The sample MW calculated by this method, in kDa.

Ref. I(0)

The I(0) of the molecular weight standard set for RAW. This field corresponds to the I(0) field in the “Molecular Weight Estimation Using a Standard” box in the Molecular Weight section of the Options window, and can be changed there or by *setting the MW standard*.

Ref. MW

The molecular weight of the molecular weight standard set for RAW. This field corresponds to the MW field in the “Molecular Weight Estimation Using a Standard” box in the Molecular Weight section of the Options window, and can be changed there or by *setting the MW standard*.

Ref. Concentration

The concentration of the molecular weight standard set for RAW. This field corresponds to the Conc. field in the “Molecular Weight Estimation Using a Standard” box in the Molecular Weight section of the Options window, and can be changed there or by *setting the MW standard*.

File – The data item name of the molecular weight standard set for RAW. This field corresponds to the Filename field in the “Molecular Weight Estimation Using a Standard” box in the Molecular Weight section of the Options window, and can be changed there or by *setting the MW standard*.

In the Vc MW panel there are:

MW

The sample molecular weight calculated by this method, in kDa.

Vc

The volume of correlation calculated by this method, in $Å^2$.

Qr

The Qr parameter calculated by this method, in Å^3.

a

The macromolecular type (protein/RNA) dependent "a" parameter used for the calculation. This field corresponds to the "Protein (RNA) Coef. A" field in the "Molecular Weight Estimation From Volume of Correlation" box in the Molecular Weight section of the Options window, and can be changed there. The value depends on whether Protein or RNA is selected in the drop down menu at the top of this panel.

b

The macromolecular type (protein/RNA) dependent "b" parameter used for the calculation. This field corresponds to the "Protein (RNA) Coef. B" field in the "Molecular Weight Estimation From Volume of Correlation" box in the Molecular Weight section of the Options window, and can be changed there. The value depends on whether Protein or RNA is selected in the drop down menu at the top of this panel.

This panel also has a plot which shows $\int qI(q)dq$ vs. q, over the q-range of the scattering profile. For this method to be accurate, the integral value needs to have converged at high q (the graph needs to be flat at high q).

In the Vp MW panel there are:

MW

The sample molecular weight calculated by this method, in kDa.

Vp

The Porod volume calculated by direct integration of the scattering profile, in Å^3.

Corrected Vp

The corrected Porod volume based on the method described *below*, in Å^3.

Macromolecule Density

The density of the macromolecule, used to calculate the molecular weight. This field corresponds to the "Density" field in the "Molecular Weight Estimation from Corrected Porod Volume" box in the Molecular Weight section of the Options window, and can be changed there.

In the Abs. MW panel there are:

Concentration

The sample concentration in mg/ml, this field accepts input.

MW

The sample MW calculated by this method, in kDa.

# electrons per mass dry macromolecule

The dry mass number density of electrons for the macromolecule, in e-/g. This field corresponds to the "Electrons per dry mass of macromolecule" field in the "Molecular Weight Estimation from Absolute Intensity Calibration" box in the Molecular Weight section of the Options window, and can be changed there.

# electrons per volume of buffer

The number density of electrons for the protein buffer/solvent, in e-/cm^3. This field corresponds to the "Electrons per volume of aqueous solvent" field in the "Molecular Weight Estimation from Absolute Intensity Calibration" box in the Molecular Weight section of the Options window, and can be changed there.

Protein partial specific volume

The partial specific volume of the macromolecule, in cm^3/g. This field corresponds to the “Partial specific volume of the macromolecule” field in the “Molecular Weight Estimation from Absolute Intensity Calibration” box in the Molecular Weight section of the Options window, and can be changed there.

Scattering length of an electron

The scattering length of an electron in cm. This field corresponds to the “Scattering length of an electron” field in the “Molecular Weight Estimation from Absolute Intensity Calibration” box in the Molecular Weight section of the Options window, and can be changed there.

Scattering contrast per mass

The calculated scattering contrast per mass. This is calculated from the other parameters as $r_0(\rho_{Mmac} - \rho_{solv}\bar{\nu})$ where r_0 is the scattering length of an electron, ρ_{Mmac} is the electrons per dry mass of macromolecule, ρ_{solv} is the electrons per volume of aqueous solvent, and $\bar{\nu}$ is the partial specific volume of the protein.

The molecular weight methods

Four different methods are used to calculate the molecular weight of the macromolecule from the background subtracted scattering profile.

I(0) Referenced molecular weight calculation (I(0) Ref. MW panel)

The scattering at zero angle, I(0) is proportional to the molecular weight of the macromolecule, and the concentration and contrast of the macromolecule in solution. If a reference sample of known molecular weight and concentration is measured, it can be used to calibrate the molecular weight of any other scattering profile with known concentration (assuming constant contrast between reference and sample, and a monodisperse sample). Molecular weight is calculated as:

$$MW_m = \left(\frac{I(0)_m}{c_m}\right)\left(\frac{MM_{st}}{I(0)_{st}/c_{st}}\right)$$

where MW is the molecular weight, c is the concentration, and the m and st subscripts denote quantities from the macromolecule of interest and the standard respectively. For a reference see, among many, Mylonas, E. & Svergun, D. I. (2007). J. Appl. Crystallogr. 40, s245-s249.

This method can yield inaccurate results if the reference is not properly calibrated, I(0) is not well estimated from the Guinier fit, or the contrast between the macromolecule and buffer is significantly different between the reference and sample.

Volume of correlation based molecular weight calculation (Vc MW panel)

This method uses the approach described in: Rambo, R. P. & Tainer, J. A. (2013). Nature. 496, 477-481. First, the volume of correlation, V_c, is calculated as

$$V_c = \frac{I(0)}{\int qI(q)dq}$$

Unlike the Porod volume, V_c is expected to converge for both compact and flexible macromolecules. Physically, V_c can be interpreted as the particle volume per self-correlation length, and has units of Å^2. V_c and the radius of gyration, Rg, are then used to calculate a parameter $Q_r = V_c^2/R_g$. The molecular weight is then calculated as:

$$MW = \left(\frac{Q_r}{b}\right)^a$$

where *a* and *b* are empirically determined constants that depend upon the type of macromolecule. More details on the calculation are in the reference. The authors claim the error in MW determination is ~5-10%.

This method can yield inaccurate results if the integral $\int qI(q)dq$ doesn’t converge, which may indicate the scattering profile is not measured to high enough q or that there is a bad buffer match. It also requires accurate determination of I(0) and Rg. It doesn’t work for protein-nucleic acid complexes.

Corrected Porod Volume method (Vp MW panel)

This method uses the approach described in: Fischer, H., de Oliveira Neto, M., Napolitano, H. B., Polikarpov, I., & Craievich, A. F. (2009). J. Appl. Crystallogr. 43, 101-109. First, the Porod volume, V_p, is determined. True determination of the Porod volume requires the scattering profile measured to infinite q. A correction is applied to V_p to account for the limited range of the measurement. The authors report a maximum of 10% uncertainty for calculated molecular weight from globular proteins.

This method can yield inaccurate results if the molecule is not globular. It requires accurate determination of I(0). It also requires an accurate protein density. It only works for proteins.

Note: To do the integration, RAW extrapolates the scattering profile to I(0) using the Guinier fit. The authors of the original paper used smoothed and extrapolated scattering profiles generated by GNOM. This may cause discrepancy. To use this method on GNOM profiles, use the online SAXS MoW calculator located at: http://saxs.ifsc.usp.br/

Absolute calibrated intensity method (Abs. MW panel)

This uses the absolute calibration of the scattering profile to determine the molecular weight, as described in Orthaber, D., Bergmann, A., & Glatter, O. (2000). J. Appl. Crystallogr. 33, 218-225. By determining the absolute scattering at I(0), if the sample concentration is also known, the molecular weight is calculated as:

$$MW = \frac{N_A I(0)/c}{\Delta\rho_M^2}$$

where N_A is the Avagadro number, c is the concentration, and $\Delta\rho_M^2$ is the scattering contrast per mass described *above*. The accuracy of this method was assessed in Mylonas, E. & Svergun, D. I. (2007). J. Appl. Crystallogr. 40, s245-s249, and for most proteins is <~10%.

This method can yield inaccurate results if the absolute calibration is off, or if the partial specific volume of the macromolecule in solution is incorrect. I(0) and the concentration in solution must be well determined. Unless the scattering contrast is adjusted, this method will only work for proteins.

BIFT

The BIFT window allows you to run a Bayesian Indirect Fourier Transform (BIFT) method on background subtracted scattering profiles to find the P(r) function. The advantage to this method over the method implemented by GNOM is that once the search parameters are set, there is no subjective input required from the user, a single "best" solution is provided by the algorithm. The BIFT algorithm being used to find the P(r) is that of: Hansen, S. (2000). J. Appl. Crystallogr. 33, 1415-1421.

Opening the BIFT Window

The BIFT window can be opened by selecting a data item in the Manipulation Panel and either right clicking and selecting "BIFT" or selecting "BIFT" from the "Tools" menu. Whichever data item is selected will be the one that is being analyzed. If multiple items are selected, the first (top item in the list) will be the one analyzed.

The window has two parts. The left part shows the file being analyzed, and the controls, parameter outputs, and status for the BIFT. The right is two plots, the top showing the P(r) function found by the BIFT and the bottom showing the experimental data and the scattering profile generated from the P(r) function. The space allotted to each side can be adjusted by clicking and dragging the separator bar. The whole window can be resized by clicking and dragging an edge or corner.

When the window is first opened, it runs a BIFT analysis to find the P(r) function of the data, based on the current settings. These settings can be altered from the *BIFT panel* in the Options window.

Once you are satisfied with the BIFT results, clicking the "OK" button will save the Dmax, real space Rg, real space I(0), the χ^2 for the fit, and $\log_{10}\alpha$ for the data item selected. That will allow you to save the BIFT analysis parameters when *saving all results* and save the results with the scattering profile if that is saved as a ".dat" file. Additionally, a new IFT data item will be created, which will be shown in the *IFT Control and Plot panels*. Clicking the "Cancel" button (or the close window button from the operating system, typically in the upper left or right corner) will exit the window without saving the analysis results or new IFT item. IFT items created by BIFT will have an extension ".ift".

Note: Only one BIFT window can be open at once. If a window is already open and you try to open a new one, it will close the current window and open the new one.

The BIFT Controls

A BIFT analysis of the scattering profile is automatically run when the window is opened. BIFT has very few controls available to the user, though settings can be customized. The controls consist of three buttons:

Run

Reruns the BIFT analysis. Needs to be done if the settings are changed after the BIFT panel is opened.

Abort

Aborts the BIFT analysis if it is currently running.

Settings

Opens the Options window and shows the settings for BIFT. If settings are changed, the Run button must be used to generate a new P(r) function with the changed settings.

The BIFT parameters

The BIFT panel displays the following parameters:

Dmax

The maximum dimension of the P(r) function found by the BIFT algorithm. This is in units of 1/q, which RAW assumes to be Å.

Log(Alpha)

The log base 10 of the alpha value found as optimal by the BIFT search.

Rg (A)

The radius of gyration in Å (assumed, actual units of 1/q). This is shown from the Guinier fit (if available) and the P(r) function. The value from the P(r) function is the value calculated in real space by

$$R_g = \frac{\int_0^{D_{max}} r^2 P(r)dr}{2\int_0^{D_{max}} P(r)dr}$$

I(0)

The scattering at zero angle. This is shown from the Guinier fit (if available) and the P(r) function. The value from the P(r) function is the value calculated in real space by

$$I(0) = 4\pi \int_0^{D_{max}} P(r)dr$$

chi^2 (fit)

The χ^2 value of the scattering profile from the P(r) function to the experimental data.

The BIFT status

The status box for the BIFT search shows parameters that update as the BIFT search is performed. Once the search is over, they show the parameters of the final solution. The status items displayed are:

Status

An overall status, which can be: Performing search grid, Performing Fine Search, BIFT done, or BIFT canceled.

Evidence

The evidence value for a given search point.

Chi

The χ^2 value of a given search point.

Alpha

The log base 10 of the alpha value of a given search point.

Dmax

The maximum dimension of the current search point.

Current Search Point

The current search point (numbered along the search grid, essentially arbitrary).

Total Search Points

The total number of search points, equal to the number of Dmax search points multiplied by the number of alpha search points.

Note: The status window does not update the evidence, chi, alpha, or dmax values during the fine search, only at the end of the fine search.

The BIFT plots

There are two plots in the BIFT window. The top plot shows the P(r) function in red. The units for the r (bottom) axis of this plot are 1/q, which RAW assumes to be Å. A black line is displayed at zero on the plot for reference. The bottom plot shows the measured scattering profile data as blue points, and the scattering profile generated from the P(r) function in red.

Note: There are no controls available on right click for these plots, but there is a standard *plot control bar*.

The BIFT algorithm

The algorithm used is described in Hansen, S. (2000). J. Appl. Crystallogr. 33, 1415-1421. In RAW, a coarse grid is used for an initial search, and then a fine optimization is performed from the best point in that search space. The limits of the coarse grid and the number of points in the coarse grid can be set in the Options window.

GNOM (ATSAS)

RAW allows you to run certain analyses using the ATSAS software package from within RAW. One of the programs that can be run from RAW is GNOM, which performs an IFT to find the P(r) function. Using the ATSAS package programs requires a *separate installation* and (possibly) some additional configuration of RAW.

Opening the GNOM Window

The GNOM window can be opened by selecting a data item in the Manipulation Panel and either right clicking and selecting “GNOM (ATSAS)” or selecting “GNOM” from the “Tools”->“ATSAS” menu. Whichever data item is selected will be the one that is being analyzed. If multiple items are selected, the first (top item in the list) will be the one analyzed.

The window has two parts. The left part shows the filename being analyzed, and the controls and parameter outputs for GNOM. The right part has two plots, the top showing the P(r) function found by the GNOM, and the bottom showing the experimental data and the scattering profile generated from the P(r) function. The space allotted to each side can be adjusted by clicking and dragging the separator bar. The whole window can be resized by clicking and dragging an edge or corner.

When the window is first opened, if no previous GNOM analysis is available for the data item, RAW runs DATGNOM from the ATSAS package analysis to find a P(r) function of the data. Generally, better results are obtained from DATGNOM when an Rg value is available from the Guinier fit. If GNOM analysis has previously be done on the data item, RAW will display the P(r) function corresponding to the Dmax value found by that analysis.

Once you are satisfied with the GNOM results, clicking the “OK” button will save the Dmax, Total Estimate, real space Rg, real space I(0), and the starting and ending q values for the data item selected. That will allow you to save the GNOM analysis parameters when *saving all results* and save the results with the scattering profile if that is saved as a “.dat” file. Additionally, a new IFT data item will be created, which will be shown in the *IFT Control and Plot panels*. Clicking the “Cancel” button (or the close window button from the operating system, typically in the upper left or right corner) will exit the window without saving the analysis results or new IFT item. IFT items created by GNOM will have an extension “.out”.

Note 1: Only one GNOM window can be open at once. If a window is already open and you try to open a new one, it will close the current window and open the new one.

Note 2: If the GNOM option is unavailable in the right click menu for a data item, it indicates that RAW is unable to find the *ATSAS programs* on your computer.

The GNOM Controls

The GNOM Controls allow you to manually set the q_min and q_max values GNOM uses. You can do this in two ways. First, if you type a value into the q_min or q_max box and hit enter, RAW will find the nearest q point, and set the q value to that nearest point. Second, you can adjust the n_min and n_max values either by typing an integer value in the appropriate box or using the spin controls. The n_min and n_max controls change the minimum and maximum points of the q vector used (zero indexed), and the q_min and q_max are updated to match. For example if n_min is set to 5 and n_max is set to 130, the 5th indexed point through the 130th indexed point in the q vector (so points 6-131, because of the zero indexing) will be used by GNOM. The Dmax value can be either by typing an integer value in the appropriate box or using the spin controls. Changing any of these values (q_min, q_max, and Dmax) automatically updates the P(r) function.

The “Change Advanced Parameters” button opens the Options panel and shows the options for GNOM. This allows you to change advanced parameters for your GNOM analysis.

The “DATGNOM” button runs the DATGNOM program from the ATSAS software package. The Dmax it finds is rounded to the nearest integer, and GNOM is run with that Dmax value to generate the P(r) function.

The GNOM parameters

The GNOM panel displays the following parameters from the GNOM fit:

Rg (A)

The radius of gyration in Å (assumed, actual units of 1/q). This is shown from the Guinier fit (if available) and the P(r) function. The value from the P(r) function is the value calculated in real space by

$$R_g = \frac{\int_0^{D_{max}} r^2 P(r) dr}{2\int_0^{D_{max}} P(r) dr}$$

I(0)

The scattering at zero angle. This is shown from the Guinier fit (if available) and the P(r) function. The value from the P(r) function is the value calculated in real space by

$$I(0) = 4\pi \int_0^{D_{max}} P(r) dr$$

Total Estimate

The "Total Estimate" produced by GNOM. A value close to 1 is good.

chi^2 (fit)

The χ^2 value of the scattering profile from the P(r) function to the experimental data.

GNOM says

The subject interpretation of the quality of the P(r) function provided by GNOM.

The GNOM plots

There are two plots in the GNOM window. The top plot shows the P(r) function in red. The units for the r (bottom) axis of this plot are 1/q, which RAW assumes to be Å. A black line is displayed at zero on the plot for reference. The bottom plot shows the measured scattering profile data as blue points, and the scattering profile generated from the P(r) function in red.

Note: There are no controls available on right click for these plots, but there is a standard *plot control bar*.

AMBIMETER (ATSAS)

RAW allows you to run certain analyses using the ATSAS software package from within RAW. One of the programs that can be run from RAW is AMBIMETER, which provides an estimate of the ambiguity a 3D shape reconstruction will have, based on the scattering profile generated from the P(r) function. Using the ATSAS package programs requires equires a *separate installation* and (possibly) some additional configuration of RAW.

Opening the AMBIMETER Window

The AMBIMETER window can be opened by selecting a data item in the IFT Panel and either right clicking and selecting "Run AMBIMETER" or selecting "AMBIMETER" from the "Tools" -> "ATSAS" menu. Whichever data item is selected will be the one that is being analyzed. If multiple items are selected, the first (top item in the list) will be the one analyzed. Currently, AMBIMETER only works on IFT items generated by GNOM (".out" files in the IFT panel). The AMBIMETER window shows the file it is being run on, the Rg (real space form the P(r) function), and controls and results.

When the window is first opened, AMBIMETER is run on the data. Once you are satisfied with the GNOM results, clicking the "OK" or "Cancel" will close the window. Because of the strict save format required for ".out" files to be used by the ATSAS package, the AMBIMETER results are not saved anywhere, and must be manually saved by the user (such as writing it down).

Note 1: Only one AMBIMETER window can be open at once. If a window is already open and you try to open a new one, it will close the current window and open the new one.

Note 2: If the AMBIMETER option is unavailable in the right click menu for a data item, it indicates that RAW is unable to find the *ATSAS programs* on your computer or that you do not have a recent enough version of the ATSAS package installed (version 2.7.1 or greater required for AMBIMETER).

The AMBIMETER Controls

The AMBIMETER controls allow you to adjust the maximum q value used by AMBIMETER, by adjusting the upper q*Rg limit between 3 and 7. Note that if the maximum q value of the scattering profile times the Rg is less than the limit set, the whole curve is used.

The AMBIMETER program can also save output shapes. For more information about this, see the AMBIMETER manual available with the ATSAS software. In the window, you can select which shapes to save, None (default), Best (one shape, the best fit), or All (all of the shapes that fit). If you are saving shapes, you should then select the output directory to save them in, and provide an output prefix. The shapes will with the prefix value provided in the Output prefix box, as described in the AMBIMETER manual. Clicking the Run button is necessary to rerun AMBIMETER after any settings have been changed.

The AMBIMETER Results

The results section shows the output from AMBIMETER. It reports:

Number of compatible shape categories

The number of compatible shape categories, as described in the AMBIMETER manual.

Ambiguity score

Log base 10 of the number of compatible shape categories.

AMBIMETER says

The subjective interpretation of the ambiguity score provided by AMBIMETER.

DAMMIF (ATSAS)

RAW allows you to run certain analyses using the ATSAS software package from within RAW. One of the programs that can be run from RAW is DAMMIF and the accompanying programs DAMAVER and DAMCLUST, which carry out 3D shape reconstructions based on the P(r) function and scattering profile. Using the ATSAS programs requires a *separate installation* and (possibly) some additional configuration of RAW.

Opening the DAMMIF Window

The DAMMIF window can be opened by selecting a data item in the IFT Panel and either right clicking and selecting “Run DAMMIF” or selecting “DAMMIF” from the “Tools”->“ATSAS” menu. Whichever data item is selected will be the one that is being analyzed. If multiple items are selected, the first (top item in the list) will be the one analyzed. Currently, DAMMIF only works on IFT items generated by GNOM (“.out” files in the IFT panel).

The DAMMIF window shows current DAMMIF settings, controls, the log output for each separate DAMMIF and DAMAVER run, and the overall status of the processing. The “Close” button closes the window. If this is done before the DAMMIF processing is finished, it will abort the processing.

Note 1: Only one DAMMIF window can be open at once. If a window is already open and you try to open a new one, it will close the current window and open the new one.

Note 2: If the DAMMIF option is unavailable in the right click menu for a data item, it indicates that RAW is unable to find the *ATSAS programs* on your computer.

Note 3: The DAMMIF processing can be run in the background while further data processing is done in RAW.

The DAMMIF settings

The DAMMIF window settings section allows you to change the most commonly used DAMMIF settings. The "Change Advanced Settings" button allows you to change all of the advanced settings of DAMMIF.

Output directory

This sets the output directory for DAMMIF and DAMAVER results, and can be set either by typing a directory into the box and hitting enter, or using the Select/Change Directory button. The directory defaults to the directory showing the Files tab.

Output prefix

The output prefix will be appended to each DAMMIF and DAMAVER file. It defaults to the filename being analyzed. This should contain no spaces.

Number of reconstructions

The total number of DAMMIF models to construct.

Number of simultaneous runs

The number of DAMMIF models to reconstruct simultaneously. DAMMIF runs on a single core, and typically takes 100% of the resources of that core. On multicore machines, the number of simultaneous runs can be set between 1 and the total number of cores. The default is the number of cores minus one.

Mode

Sets the Mode used by DAMMIF for the reconstructions. The "Fast" and "Slow" modes are as described in the DAMMIF manual. The "Custom" mode is equivalent to the "Interactive" mode for DAMMIF, and allows the use of the more advanced settings, as set in the Options panel in the "DAMMIF Advanced" section.

Symmetry

Allows the symmetry to be set, if known, as described in the DAMMIF manual.

Anisometry

Allows the anisometry to be set, if known, as described in the DAMMIF manual.

Change Advanced Settings

This opens the Options panel and shows the options for DAMMIF. This allows you to change advanced parameters for your DAMMIF analysis.

Align and average envelopes (damaver)

If this is selected and two or more reconstructions are generated using DAMMIF, then once all reconstructions are finished DAMAVER is automatically run on the DAMMIF reconstructions. This runs in a mode equivalent to "damaver –a" at the command line.

Align and cluster envelopes (damclust)

If this is selected and two or more reconstructions are generated using DAMMIF, then once all reconstructions are finished DAMCLUST is automatically run on the DAMMIF reconstructions.

Note: The damaver and damclust options are mutually exclusive, you can select DAMAVER or DAMCLUST but not both.

The DAMMIF controls

There are only two control buttons, “Start”, which starts the DAMMIF reconstructions, and “Abort”, which aborts the DAMMIF reconstructions. Start is only available if DAMMIF reconstructions are not currently running. Abort is only available if DAMMIF reconstructions are running.

Note: DAMMIF requires that the P(r) function be written to disk as a “.out” file. RAW will check whether there is an existing “.out” file with the same name that will be overwritten before running. It will also check whether the DAMMIF files generated will overwrite any existing files. In either case, it will provide a warning to let you know that is happening.

Note 2: DAMMIF gets the starting random seed value from the computer clock time in seconds. In order to produce different reconstructions, the start of each reconstruction must occur at least 1s after the previous one. This introduces a noticeable delay when starting up a large number simultaneous reconstructions.

The DAMMIF status

The status panel provides an overview of the current status of the DAMMIF runs. It updates with the following status:

Starting Processing

Indicates that the initial processing has started.

Starting DAMMIF run <#>

Here the <#> corresponds to the run number (1 up to the total number of reconstructions), and this corresponds to the numbers in the Log panel. This indicates that the given DAMMIF run has started.

Finished DAMMIF run <#>

Here the <#> corresponds to the run number (1 up to the total number of reconstructions), and this corresponds to the numbers in the Log panel. This indicates that the given DAMMIF run has finished.

Starting DAMVER

Indicates that the reconstructions are now being aligned and averaged by DAMAVER.

Finished DAMAVER

Indicates that the reconstructions have finished being aligned and averaged by DAMAVER.

Finished Processing

Indicates that all processing has finished.

Processing Aborted!

Indicates that the processing was aborted before everything finished.

If necessary, the status window is scrollable.

The DAMMIF log

The log area provides details of the current and finished DAMMIF and DAMAVER processing. This is the output that would be displayed in the console window if DAMMIF or DAMAVER were run from the command line.

When the DAMMIF window is first opened, the log window will be empty. Once DAMMIF processing is started, a set of different panels accessible via the tabs at the top will be opened. Any tab with a number corresponds with a DAMMIF run (output from DAMMIF run 1 will display in tab "1", and so on). The runs are simply numbered sequentially starting with 1 and ending with the total number of reconstructions. If DAMAVER will also be run, the last tab is "Damaver", which shows the output of that processing.

Before a given run is started, the window associated with the tab will be empty. Once it starts, it will be updated with the output from DAMMIF or DAMAVER that would normally display in the console. Once it finishes, it will no longer update, but the log can still be viewed. The long windows are scrollable. You can change which output you are viewing by clicking on the tabs.

Note: The output in the log window is not saved, however by default DAMMIF writes this output to a log file, and the DAMAVER output is available in a set of output files. See the ATSAS manuals for each program for more details.

SVD

Singular value decomposition (SVD) is a mathematical technique that is a model independent approach that provides information on the number of unique elements in a data set. Formally, singular value decomposition of a m x n matrix M is a factorization of into three matrices such that

$$M = U\Sigma V^*$$

where U is an m x m unitary matrix, called the left singular values; Σ is a diagonal m x n matrix, where the diagonal values are the singular values, and V^* is the conjugate transpose of an n x n unitary matrix V, the right singular vectors. A typical interpretation of singular value decompositions is that the number of singular values significantly above the baseline level represents the number of significant distinct components in the data set.

RAW allows the user to use either scattering profiles or P(r) functions as the data set. This is typically applied to scattering profiles in a SEC-SAXS data set, and the number of significant singular values corresponds to the number of distinct scatterers in the data set. For SVD done across a single well-separated peak from the chromatograph, there would be two significant components: one from the buffer and one from the protein. For SVD done on a poorly separated monomer-dimer peak, there would be three significant components: buffer, monomer, and dimer. RAW allows users to select a range of scattering profiles for SVD, and displays the singular values σ_i and the autocorrelation of the left and right singular vectors R_i for each ith singular value defined as

$$R_i = \Sigma_n X_{i,n} X_{i,n+1} \tag{5.1}$$

where X is the U or V singular vector matrix.

Opening the SVD Window

The SVD window can be opened by selecting a single data item in the SEC Panel or two or more data items in the Manipulation or IFT Panels and either right clicking and selecting "SVD" or selecting "SVD" from the "Tools" menu.

The SVD window shows current SVD settings, controls, and results. The "Okay" button closes the window and saves the parameters used, the "Cancel" button closes the window and saves nothing.

Note: Only one SVD window can be open at once. If a window is already open and you try to open a new one, it will close the current window and open the new one.

The SVD Controls

The SVD controls give the user (up to) three items to control.

Use

This option allows the user to select whether they use Unsubtracted or Subtracted data. Note that this choice is only available for SEC curves, where both unsubtracted and subtracted data can exist within the same data item. Subtracted data can only be selected if the SEC curve has had *structural parameters* calculated for it.

Use Frames

This option changes the range of frames used for the SVD. If a SEC data item is selected, the frame number corresponds to the frame number in the SEC Plot. If Manipulation or IFT data items are selected in, the frame number is the same as how it would be displayed if the data were plotted on the *SEC plot*. The plot in the controls window shows the SEC curve of the selected. The red points correspond to all of the data in the data set, the blue points correspond to the data being used for SVD.

Normalize by uncertainty

If this option is selected, the intensity at a given q (SEC and Manipulation data) or r (P(r) data) value is divided by the average uncertainty across all frames at that q or r value. When doing EFA analysis, the data is normalized in this way, so this option allows corresponding SVD analysis.

The SVD Results

The SVD results are plotted in the two plots on the right side of the SVD panel. The top plot shows the singular values as a function of index (by default singular values are ordered from largest to smallest). The bottom plot shows the autocorrelation, defined in equation (5.1), of the left (red) and right (blue) singular vectors as a function of index.

The results control panel allows you to control which singular value indices are plotted. It has two further controls:

Save Plotted Values

This saves the singular values and autocorrelation values as a function of index that are plotted in the plots on the right. The data is saved as a comma separated value (.csv) file.

Save All

This saves all of the singular value information, which is to say, the full U, Σ, and V^* matrices. It also saves the autocorrelation values for all indices. The data is saved as a comma separated value (.csv) file.

EFA

Evolving factor analysis (EFA) is a model independent approach that extends SVD to allow separation of scattering profiles from mixed solutions, particularly overlapping chromatographic peaks from different species. This method was recently applied to SEC-SAXS data (see: Meisburger, S. P., Taylor, A. B., Khan, C. A., Zhang, S., Fitzpatrick, P. F., & Ando, N. (2016). J. Am. Chem. Soc. jacs.6b01563). An improved version of the method described by Meisburger et al. has been implemented in RAW. EFA in RAW starts with SVD, then proceeds by finding the component start and end points in the EFA plot and then rotating the significant singular value vectors into real scattering profiles. RAW implements two new methods for rotation of the singular vectors besides the iterative approached described by Meisburger et al. The first new method is the explicit calculation method described in: Maeder, M. (1987). Anal. Chem. 59, 530–533, while the second is a hybrid method that uses the explicit calculation as a seed for the iterative approach, allowing much faster convergence of the rotation.

Because EFA analysis is relative complex, it consists of three distinct screens in RAW, which are reached by clicking the "Next" and "Back" buttons in the EFA window. These screens step you through the EFA analysis. The window can be closed at any time with the "Cancel" button. If the EFA analysis succeeds, the window can be closed with the "Done" button, which sends the extracted scattering profiles to the Main plot, and saves information about the parameters used in the SEC item (if a SEC item was selected).

Opening the EFA Window

The EFA window can be opened by selecting a single data item in the SEC Panel or two or more data items in the Manipulation or IFT Panels and either right clicking and selecting “EFA” or selecting “EFA” from the “Tools” menu.

The EFA window shows current EFA settings, controls, and results. The “Okay” button closes the window and saves the parameters used, the “Cancel” button closes the window and saves nothing.

Note: Only one EFA window can be open at once. If a window is already open and you try to open a new one, it will close the current window and open the new one.

EFA Page 1 – SVD

The first EFA page essentially reproduces the SVD window described *above*. The difference is that there is a panel for User Input, “# Significant SVs” where the user inputs the number of significant singular values/vectors in the data set. RAW will automatically attempt to determine that, but the user can adjust it. If the user changes any of the SVD controls, RAW will not refine the guess for the number of significant singular values.

Once the user is happy with the data range and has determined number of significant singular vectors in the data, they click the “Next” button to move to Page 2.

EFA Page 2 – Evolving Factors

The second EFA page presents the results of the Evolving Factor Analysis. The details are described in the above referenced papers. In short, for Forward EFA, SVD is done on pieces of the data set, starting with just the first two frames, then the first three frames, and so on, until the entire data set is used. The singular values are then plotted as a function of final frame index used in that particular SVD, the top plot in this second page. This lets the user determine where certain components start in the data set, when there is a strong increase in a singular value is when a component starts in the data set. For Backward EFA, the same thing is done, except using the last two frames, then the last three frames, etc. The Backward EFA plot, the bottom plot on this page, shows the user where components exit the data set. Both the Forward and Backward EFA plots show one more value than the user marked as significant in Page 1, so that the user can judge where the value diverges from the baseline.

The User Input panel on the left side of the page consists of “Forward” and “Backward” sections each with a number of values equal to the user input of significant singular values from Page 1. These controls allow the user to set where each value diverges from the baseline.

Once the starting points for the values are set for both Forward and Backward EFA, clicking the “Next” button will take the user to Page 3.

Clicking the “Back” button allows the user to go back to Page 1 and adjust the results there.

EFA Page 3 – Rotation

The third EFA page allows the user to control the rotation of the singular vectors into scattering profiles and view the result of that rotation. There are several different control boxes:

Component Range Controls

These allow the user to adjust the ranges of the components in the scattering profile. There are a number of controls equal to the number of significant singular vectors selected on Page 1. Each control allows you to set the start and end point of the range, and each control has a “C>=0” checkbox. When checked, that forces the concentration profile for that component to be never negative (a physical constraint that can help with rotation by may hide mathematical errors). When the ranges are adjusted, the plot above this control is updated. The arrows and dashed lines of this plot shown the component ranges, with colors that correspond to the colors in the legend of the Scattering Profiles plot in the top right.

Rotation Controls

Method

This controls the rotation method. There are three choices. Iterative applies the approach of Meisburger et al. (2016). Explicit applies the approach of Maeder (1987). Hybrid uses the explicit approach to seed the iterative approach.

Number of iterations

This can only be set for the Hybrid and Iterative approaches, and sets the maximum number of iterations the algorithm will attempt. If convergence is not reached by the end of these iterations, the method will fail.

Convergence Threshold

This sets the threshold at which the iterative and hybrid solutions are said to be converged. This threshold is defined in Meisburger et al. (2016), and for the nth rotation is the absolute difference between the concentration profiles of the n-1 and n solutions.

Status

The status box displays the status of the rotation. If the rotation has succeeded it will say "Rotation Successful." If the rotation is in progress it will say "Rotation in progress." If the rotation has failed it will provide a message with some on the failure, that starts with "Rotation failed."

Results

Save EFA Data (not profiles)

This button saves the EFA data, including the SVD data, the number of significant values, the ranges Forward and Backward EFA data, the ranges and concentration constraints for each component, the rotation method and other parameters used, and the mean error weighted χ^2 and concentration data. This is all saved as a .csv file.

In addition to the control boxes, there are also three plots. The top plot shows the scattering profiles obtained via the rotation (if the rotation is successful). The bottom plots show: Left – the mean error weighted χ^2 as a function of frame, which is a measure difference between the scattering profiles in a given frame as measured and as produced from the rotated scattering profiles and concentration profiles; Right – the concentration profiles for each component, which are color as the scattering profile (top) plot. The concentration profiles are normalized to an area of 1, and so are on an arbitrary scale.

5.8.13 Menus

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

RAW has 5 program wide menus: File, Options, View, Tools, and Help.

The File menu

The File menu has five choices:

Load Settings

This loads RAW settings from a .cfg file. This can also be done by double clicking on the .cfg file in the Files tab of the Control Panel.

Save Settings

This saves all of the RAW settings, including image calibration and the image mask, to a .cfg file so they can be loaded again later.

Load Workspace

This loads a RAW *workspace*.

Save Workspace

This saves a RAW *workspace*.

Quit

This quits the RAW program. If you have unsaved data or are running DAMMIF reconstructions you will be asked to confirm that you wish to quite.

The Options menu

The options menu has one choice and one submenu. The "Advanced Options" choice opens the *Options window*. The submenu is labeled "Online mode" and the three options available there are described *here*.

The View menu

The view menu has three top level options and five submenus. The three top level options are: "Show image", "Show data", and "Show header". All of these require that a single manipulation item be selected, and otherwise behave as the options of the same name in the Manipulation data item *right click menu*.

The view menus named "Top Main Plot Axis" and "Bottom Main Plot Axis" change the plot axes/scale of the top and bottom main plots respective. Each option in the two menus corresponds to the options in the main plot *right click menu*.

The view menus named "SEC Plot Left Y Axis", "SEC Plot Right Y Axis", and "SEC Plot X Axis" change the data plotted on the SEC Plot axes as the options in the SEC plot *right click menu*.

The Tools menu

The Tools menu is split into three parts. In all of the items in the top part of the menu apply to Manipulation data items and all of them require one or more Manipulation data items to be selected. The first submenu is "Operations". This contains a set of operations that can be run on Manipulation data items, and have identical effects to items of the same name in the manipulation *right click menu*. The second submenu is "Convert q-scale" and the options in the menu have identical effects to items of the same name in the Manipulation data items, and have identical effects to items of the same name in the manipulation *right click menu*. The third item is "Use as MW Standard" and has an identical effect as the item of the same name in the manipulation *right click menu*.

The middle part of the Tools menu consists of analysis tools. The "Guinier Fit", "Molecular Weight" and "BIFT" options, and the "ATSAS" submenu option "GNOM" all require a single Manipulation data item to be selected. These all open the corresponding *analysis windows*. The "DAMMIF" and "AMBIMETER" options require a single IFT data item to be selected, and open the corresponding *analysis windows*. The "SVD" and "EFA" require either that multiple Manipulation or IFT data items be selected for that a single SEC data item be selected, and open the corresponding *analysis windows*.

The lower part of the Tools menu consists of two options. The "Centering/Calibration" option requires that an appropriate image for centering/calibration be loaded in the Image plot panel. It then opens the *Centering/Calibration panel*. The "Masking" option requires that an appropriate image for masking be loaded in the Image plot panel. It then opens the *Masking panel*.

The Help menu

The Help menu has two options on it. The "Help!" option shows a window describing how to find help for RAW (including a reference to this document). The "About" provides a very brief description of RAW, includes the RAW citation, provides the license agreement, and lists the developers.

5.8.14 Line properties dialog

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

The Manipulation, IFT, and SEC data items can all open a line properties dialog. This dialog will let you control the following properties for each line associated with the object:

Line: Style (solid, dashed, dotted), width, color

Error bars: Style (solid, dashed, dotted), width, color

Data point marker: Shape (the marker shapes are those used by matplotlib, which are defined here: https://matplotlib.org/api/markers_api.html), size, edge line color, fill color, and whether or not the markers are hollow.

The line properties dialog also lets you set the legend label for the object, if any is needed.

If the dialog is exited with "OK", all changes take effect. If the dialog is exited with "Cancel", the old settings are restored.

5.8.15 The Information panel

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

The information panel displays information about a selected data item in the Manipulation panel. It shows the name of the item at the top. It also shows the following analysis, if available, the Rg in Å, I(0), and MW in kDa (from reference to a set MW standard). There is a box into which sample concentration can be entered in mg/mL. Notes or a description of the sample can also be entered, and will be saved as part of the "header" when a .dat file is saved. Finally, the drop down list in the header browser can be used to display and quickly view any header value available for the data item.

If the current Control panel is changed from the Manipulation panel to the IFT or SEC panel, the Information panel is cleared. If the panel is changed back to the Manipulation panel, the data in the Information panel are restored.

5.8.16 Workspaces

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

RAW allows you to save and load sets of files that you are analyzing, called "workspaces". To save a workspace, go the File menu and select "Save Workspace". To load a workspace, go to the File menu and select "Load Workspace".

When you save a workspace, all of the items in the Manipulation, IFT, and SEC panel are saved. When you load a workspace, all of those saved items are loaded back up, with all of the analysis information and line settings they had previously. Any files already open in RAW remain open when a workspace is loaded.

Note: The settings are not saved when a workspace is saved.

5.8.17 Online mode

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

RAW has an online mode that will watch a folder, and automatically load any files placed in the folder that RAW can read. This is typically used to automatically load in data as it is being collected at the beamline. This is distinct from the *automatic update feature* of the SEC Control panel, and both features can be active at once.

Turning on/off online mode

In order to turn on online mode, go to the Options->Online Mode menu and click the "Online" option. When you go online, a file browser will open and you will pick a folder. This folder (and only this folder, online mode does not recurse into subfolders) will be monitored by RAW. Every time a file is added (or updated) in the folder, RAW checks to see whether it can load the file. If it can, it will be loaded into RAW (for example, detector images will be integrated and loaded into the Manipulation panel and Main plot).

To disable online mode, go to the Options->Online Mode menu and click the "Offline" option.

Note: The online status of RAW is indicated in the status bar at the bottom of the main RAW window. On the right side, it says "Mode: <status>" where <status> is OFFLINE or ONLINE. This is only toggled by this online mode, not the automatic update feature of the SEC panel.

Changing directories in online mode

If you wish to change the directory that RAW is monitoring in online mode, go to the Options->Online Mode menu and click the "Change Directory" option. A file browser will appear, and you will use that to select the new folder for the Online mode.

Online mode filter

In some cases, you may not want every file that enters the directory (that can be read by RAW) to be loaded into RAW. In this case, it is possible to set a filter so that only certainly files can be loaded into RAW. The filter can consist of any number of individual items, which are set as follows. First, chose whether an item is include (open only if) or exclude (don't open if). Then provide a string for RAW to match in the file name. Then specify where in the filename RAW should match the string (start, end, anywhere). This filter can be set in the *options window*.

5.8.18 File types

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

Output file types

RAW outputs the following file types:

.cfg – These are files containing all of RAW's settings (configuration files). This file type is not human readable.

.dat – These are files containing *three column scattering profile data* with a "header" at the start or end of the file for additional information. This file type is human readable.

.ift – These are files containing *three column BIFT data* followed by four column scattering profile data, with a "header" at the start or end of the file for additional information. This file type is human readable.

.msk – These are files containing a RAW mask. This file type is not human readable.

.out – These are files containing GNOM P(r) data and are written in the standard format described in the ATSAS manual for GNOM. This file is human readable.

.sec – These are files containing *SEC-SAXS curves*, which can be loaded back into RAW and contain all of the relevant scattering profiles and structural parameters. This file type is not human readable.

.wsp – These are files for *saved workspaces*. This file type is not human readable.

Input file types

In addition to the output file types described above, all of which can be read in by RAW, RAW can load the following types of input files.

ASCII files:

.csv – If data is in a 3 column csv format, it is loaded assuming the columns are q, I, Error.

.dat (2 column) – If a .dat file has exactly 2 columns separated by whitespace, it is loaded as q I.

.dat (4 column) – Produced by FoXs when fitting a theoretical curve to a scattering profile. Loads the experimental and simulated (_FIT) scattering profiles.

.fir – Output by various ATSAS programs such as DAMMIF. Loads the experimental and simulated (_FIT) scattering profiles.

.fit – Output by various ATSAS programs such as DAMMIF. Loads the experimental (often smoothed) and simulated (_FIT) scattering profiles.

.int – Output by CRYSOL if no fitting is used. Loads the scattering intensity in solution.

.rad – Depreciated file format for RAW saved scattering profiles.

.txt - 2 or 3 column data can be read in as if it was a .dat file.

Image files:

Image files are often distinguished not just by their extension, but by their header format. For example, many detectors produce tif files, but require different methods to read in the appropriate header data. RAW uses the fabIO python module to load images. The complete list o f images and extensions supported can be found here: http://pythonhosted.org/fabio/getting_started.html#list-of-file-formats-that-fabio-can-read-and-write

We reproduce that here with the detector/format followed by the file extension in parenthesis: * ADSC Quantum (.img)

- Bruker (.sfrm)
- CIF binary files (.cbf)
- EDNA-XML (.xml)
- ESRF EDF (.edf)
- Eiger (.h5)
- FReLoN
- Fit2D spreadsheet (spr)
- Gatan Digital Micrograph (.dm3)
- General Electric (.No?)
- HDF5 (Hierarchical Data Format 5) (.h5)

- Hamamatsu CCD (.tif)
- MarCCD/Mar165 (.mccd)
- Mar345 image plate (.mar3450)
- Nonius KappaCCD (.kccd)
- Numpy 2D array (.npy)
- Oxford Diffraction (.img)
- Pixi (.?)
- Pilatus Tiff (.tif or .tiff)
- Portable aNy Map (.pnm)
- Rigaku SAXS format (.img)
- Tagged Image File Format (.tif)

In addition, we have written custom methods to support the following detector/file formats:

- 32 bit TIF (.tif)
- FLICAM (.tif)
- ILL SANS D11
- Medoptics (.tif)
- Multiwire (.mpa)
- SAXSLab300

5.8.19 Automated saving

WARNING: The manual is current several versions out of date. While it may still be useful for some users, please refer to the tutorial for the most up-to-date information.

RAW can automatically saving the following types of files:

- Processed image files (after integration)
- Averaged data files
- Subtracted data files

To turn on any of these capabilities:

1. Open the *Save Directories* panel of the Options window.
2. For the appropriate file type (Processed, Averaged, or Subtracted), select a save directory by clicking the "Set" button.
3. Check the appropriate Auto Save checkbox.
4. Click "OK" to exit the Options window and keep the changes to the settings.

5.9 Changes

5.9.1 2.1.2

Release date: 2022-05-20

Overview

The RAW team is pleased to announce the release of RAW version 2.1.2. This version contains mostly minor bug fixes. Significant changes include:

- Added a prebuilt version for arm64 chips on MacOS (Apple Silicon).
- Removed Windows 7 and 8 support form the prebuilt version for Windows.
- Added support for Eiger2 images from BioCAT.
- Improved handling of multi-image files.
- Adds a new dependency on mmcif_pdbx to read mmcif files.

There are also numerous other small bug fixes and new features.

NOTE: As of now, RAW is not compatible with ATSAS 3.1.0, which is currently available for download as a pre-release. This is because ATSAS 3.1.0 is currently missing several important programs, such as some of the DAMAVER set of tools, so we can't test against that. Once ATSAS 3.1.0 is officially released we will update RAW to support it.

All changes:

- Fixed a bug that prevented normalization by a BioCAT header counter if it was the last counter in a line.
- Improved buffer region finding in the series analysis.
- Fixed a bug where the GNOM window raised an error if Dmax wasn't found automatically.
- Fixed a bug where a series buffer region wasn't found if there were no peaks present in the data set.
- Fixed a bug where REGALS results would show two lines for the same concentration curve when it wasn't supposed to.
- Fixed a bug where validation of the REGALS settings could cause an error.
- Fixed a bug where changing the range for REGALS in the SVD plot when there were previous REGALS results could cause an error.
- Fixed a bug where using the qmin or qmax boxes in the series adjustment panel could cause an error.
- Fixed a validation issue in REGALS.
- Added ability to read in Eiger2 images from BioCAT and associated headers.
- Added log binning for azimuthal integration.
- Fixed a bug in error propagation on binning.
- Fixed a bug where auto centering in the calibration panel didn't work on Eiger images.
- Fixed a bug were SASRES resolution couldn't be read out of the damsel.log file in newer versions of ATSAS (>=3.0.4).
- Improved azimuthal integration speed.

- Image display for Eiger2 and Pilatus images now ignores bad pixels and detector gaps when automatically setting min and max values.
- Fixed a bug where series buffer and sample ranges wouldn't print in the report if there wasn't also EFA or REGALS data.
- Fixed a bug where moving a polygon or square mask wouldn't regenerate the mask in RAW unless you saved and reloaded the settings.
- Fixed a bug where if a profile had a non-zero starting q index, autorg would return the wrong start index, which could cause it to fail in the GUI.
- Fixed some off by one errors in the start and end points for some of the ATSAS functions in the API.
- Improved speed of GNOM calculation in the GNOM window.
- Fixed a bug with the initial start point not getting set correctly for some IFTs in GNOM.
- Added an option (on by default) where masked pixels are excluded from automatic scaling in the image viewer.
- Improved file handling for multi-image files to significantly reduce memory usage.
- Removed GNOM options not available in gnom5.
- Fixed a bug where if alpha wasn't the default value the GNOM window didn't initialize with the correct alpha.
- Fixed a bug where the show plot buttons (1&2, 1, or 2) weren't working correctly with matplotlib >=3.4.
- Dropped support for Windows 7 and 8 in the prebuilt version.
- Added a prebuilt version for arm64 chips on MacOS (Apple Silicon chips).
- Updated DENSS to the current version.
- Made RAW mostly dark mode compatible on MacOS (requires wxpython>=4.1.1). May not work in the prebuilt version for Intel macs.
- Fixed a bug where DENSS symmetry settings other than 0 would cause an error when the DENSS window was opened.
- Added some initial compatibility with ATSAS 3.1.0, but this is not finished because the pre-release version of ATSAS doesn't have several of the necessary programs for testing. Main change is mmcif compatibility for dammif outputs.
- Added a new dependency on mmcif_pdbx for reading mmcif files.
- Fixed an issue with spin controller display sizes in GTK3 (some linux installs).

5.9.2 2.1.1

Release date: 2021-05-05

Overview

The RAW team is pleased to announce the release of RAW version 2.1.1. The major change in this version is:

- Fixed a serious bug that would cause RAW to crash on Ubuntu.

There are also several other small bug fixes and new features.

All changes:

- Fixed a serious bug that would cause RAW to crash on Ubuntu (and possibly other OSes).
- Fixed a bug where closing the Guinier window before autorg finished would result in an error.
- Fixed a bug that could cause an error if the auto_dmax function failed to return a result.
- Fixed a bug where using simple concentration regularizers would fail.
- Fixed a bug where caching of certain compiled functions wasn't working, leading to them having to be compiled every time RAW was opened.
- Tweaked when functions are compiled to try to speed up the user experience, particularly when opening the Guinier window, and either IFT window.
- Improved the speed of BIFT on Linux and Windows.
- Fixed an issue where RAW wouldn't work with matplotlib>=3.4.1.
- Fixed an issue where available fonts weren't properly displayed in the prebuilt versions.
- In the RAW API, BIFT now defaults to single processor (should be faster), and you can specify the number of processors to use if you use it in multiprocessor mode.

5.9.3 2.1.0

Release date: 2021-04-20

Overview

The RAW team is pleased to announce the release of RAW version 2.1.0. This version sees the release of two major new features:

- Analysis reports on your data can now be *saved as PDFs*.
- The release of a GUI for the REGALS technique, a new way to deconvolve overlapping LC-SAXS peaks. REGALS can be thought of as an extension and enhancement of EFA for other types of SAXS data, such as ion exchange chromatography, titration series, and time resolved SAXS. You can *read more about REGALS here*.

Additionally we've overhauled the auto_guinier function to improve accuracy and applicability to lower quality data. We've also added a new, more accurate method for automatically finding Dmax when using GNOM. Finally, there are the usual numerous small tweaks and bug fixes for the main RAW GUI and the API.

Special thanks to Steve Meisburger and Darren Xu for helping with the details of their REGALS algorithm and code, and testing the new REGALS GUI.

Note for MacOS users: In order to achieve full compatibility with MacOS 11, we have had to drop support in the prebuilt version for 10.9 and 10.10. The prebuilt version of RAW will now run only on 10.11 or later. Additionally, in 10.11-10.13 the main RAW windows will show some odd coloration (black bars near the top of various windows, for example), but all functionality seems to work fine. You can still build RAW from source on older versions of MacOS.

Also, we haven't been able to test on the Apple M1 chipset. RAW should work via the built in Rosetta 2 translation in MacOS 11, but it will not run natively. If someone wants to send us a Mac with an M1 chip to test on, we're happy to work on getting it to run natively.

All changes:

- Fixed a bug that was causing pyFAI to recreate the azimuthal integrator each time, slowing down radial averaging.
- Fixed a BioCAT specific bug where concentration would end up in the profile info when it wasn't actually known for that profile.
- Fixed a bug where series files couldn't be loaded or saved in python 3.8.
- Fixed a bug where if you declined to load a config when you started RAW, the ATSAS install location wouldn't be automatically found.
- Fixed a bug where matplotlib 3.3 would mess up the plot toolbars.
- Fixed a bug where you would see an error message if RAW failed to find a valid sample region in the LC series plot.
- Fixed a bug where doing EFA on a series that had the q range of the subtracted profiles truncated relative to the unsubtracted profiles would fail.
- Added the ability to generate PDF reports of analysis.
- Fixed a bug where the profile and ift line options dialogs couldn't be opened with matplotlib 3.3.
- Significant improvements to auto_guinier function for both the GUI and API, including better accuracy, better handling of low quality data, and better handling of poorly formatted data.
- Fixed a bug where if previous EFA ranges were available they wouldn't be properly set when the EFA window was opened.
- The RAW DENSS results .csv file now indicates if a refinement was run.
- Fixed a bug where RAW could fail to load a .out file.
- Fixed a bug where aborting in the middle of a DENSS average could cause an error.
- Fixed a bug where opening the GNOM window if the profile had a non-integer Dmax value caused an error.
- Added the REGALS technique.
- Added an enhanced way to automatically find Dmax when using GNOM.
- Fixed a bug where running GNOM when RAW was run with python 3.8 could fail.
- Added ability to read in a fourth dQ column in .dat files, preserve the dQ values through analysis and saving. Note that merging and interpolating do not preserve the dQ values at this time.
- Added ability to read in CRYSOL 3 .int files.
- Fixed an off by one bug that could affect SVD/EFA/REGALS
- Fixed a bug where settings could fail to save or load on Windows if they included non-ascii characters
- Fixed a bug where calculating the corrected Porod volume MW could return an error
- EFA now remembers the force positive settings for concentration.
- Fixed a bug where if an EFA range was listed high to low it would cause an error.
- Fixed a bug where scaling q by 10x or 0.1x only worked once (i.e. you could scale to 0.1x or 10x, but not 100x, and couldn't go back to 1x after applying a scale).
- Fixed a bug where the linear baseline wasn't getting a good start value.
- GNOM window now 'truncates for dammif/n', which truncates to the smaller of 8/Rg or 0.3.
- Guinier window now opens faster.

- Updated DENSS to version 1.6.3, which includes the possibility of doing DENSS on a GPU (requires RAW to be built from source).
- Fixed a bug where multiple DENSS windows couldn't be used at the same time.
- Fixed a bug where subtracted and baseline corrected profiles from the LC Series Analysis window would have the prefix of the individual profiles in the series, rather than the series itself.
- Fixed some possible memory leaks related to dialog creation/destruction.
- Added a number of new tests.
- Added compatibility with new string handing in h5py version 3.
- Full compatibility with MacOS 11, which fixes several graphics glitches in 10.15 and 11. This required dropping support in the prebuilt version for MacOS 10.10 and earlier.
- Fixed a bug with the API where loading multiple images from a single file wasn't working properly.
- Added the ability to abort DAMMIF/N and related functions and DENSS runs in the API.
- Made SECM and RAWSettings objects picklable, so they can be passed through a multiprocessing queue.
- Fixed an API bug where saving a series would fail if you didn't set a filename.
- Fixed an API bug where saving the GNOM results to a profile was saving the wrong qmax value.
- Fixed a bug in the API where Dmin and Dmax zero conditions weren't getting set correctly for GNOM.
- Added a feature to the API to truncate an IFT for dammif using either 8/rg or 0.3, whichever is smaller.
- Fixed a bug in the API that could cause GNOM to fail to run.
- The DENSS function in the API now returns chi squared, rg, and support volume as a function of iteration so you can check convergence.
- Fixed a bug in the API where running EFA would change the associated ranges.
- Fixed a bug in the API that could cause BIFT to fail.
- Fixed a bug in the API that could cause the auto_guinier function to fail.
- Fixed a bug in the mw_vp API function.

5.9.4 2.0.3

Release date: 2020-08-11

Overview

The RAW team is pleased to announce the release of RAW version 2.0.3. While this is only a point release, we are releasing a major new feature for RAW. There are only minor modifications to the RAW GUI, but we are releasing an entirely new RAW API. This means that RAW can now be imported as a python package and you can call RAW functions in your own scripts. The API is fully documented, and the *documentation plus install instructions and some examples are available on the RAW website.*

A short example of the usage would be:

```
import bioxtasraw.RAWAPI as raw

#Load the settings
settings = raw.load_settings('./standards_data/SAXS.cfg')
```

(continues on next page)

(continued from previous page)

```
#Load the profile of interest
profile_names = ['./reconstruction_data/glucose_isomerase.dat']
profiles = raw.load_profiles(profile_names)

gi_prof = profiles[0]

#Automatically calculate the Guinier range and fit
(rg, i0, rg_err, i0_err, qmin, qmax, qrg_min, qrg_max, idx_min, idx_max,
    r_sq) = raw.auto_guinier(gi_prof, settings=settings)
```

The API should be considered in beta right now. It is tested, but based on further testing and user feedback the API may still change significantly. If you use the API, please let us know if you encounter any bugs, incomplete (or inaccurate) documentation or examples, or have suggestions for changes or additions.

There are also several small bug fixes for the main RAW GUI.

All changes:

- First release of the RAW API
- Added new unit tests for the API.
- Improved backwards compatibility of the RAW series .hdf5 files.
- Fixed some depreciation warnings.
- Fixed a bug where returning to the first EFA panel from the last and changing the number of significant singular values, then returning to the third panel would result in an error.
- Fixed several bugs that could cause multiprocessing calculations to lock up.
- Fixed several bugs in BIFT to make it more robust for poorly formatted data.
- Fixed a Guinier fit bug where the fit could fail if a point in the fit had an uncertainty value of 0.
- Fixed a bug where DATCLASS M.W. calculation could fail with an error.
- Fixed a bug where if the estimation of the Rg error failed the Rg results would fail to save with the profile when the Guinier window was closed.
- Fixed a bug where axes for the IFT profile plot couldn't be changed.
- The P(r) fit is now plotted on top of the IFT data.
- Fixed a bug where workspaces with IFTs couldn't be loaded.
- Modified the BioCAT header load function to parse a single field spread out over multiple lines in the header.

5.9.5 2.0.2

Release date: 2020-07-09

Overview

The RAW team is pleased to announce the release of RAW version 2.0.2. The major change in this version is:

- Fixed a bug where SAXSLAB images couldn't be loaded.

There are also several other small bug fixes and new features.

All changes:

- Can now load .dat files from WAXSiS, .dat files that are comma separated.
- Fixed a bug where the Vc integrated intensity plot was blank if the scattering profile had saved M.W. results.
- Improved rebinning functions, particularly the log binning function.
- Fixed a bug where error wasn't interpolated properly when interpolating a profile.
- Fixed a bug where pixel size from header bindings was in the wrong units.
- Fixed a bug where series type could get lost in certain operations.
- Fixed a bug where centering and enantiomorph selection options were ignored in the DENSS alignment panel.
- Fixed a bug where series buffer range finding was scaling profiles before testing for similarity.
- Fixed a bug where SAXSLAB images failed to load with pillow version 7.
- Fixed a bug where SAXSLAB images failed to load.
- Fixed a bug where images wouldn't load if there was no beamstop mask.
- Fixed a bug that could result in calculated data not displaying on the series plot.
- Fixed a bug where series saved as .hdf5 with EFA analysis would fail to open the EFA window when reloaded.
- Fixed a bug where datgnom couldn't be run on truncated profiles.
- Fixed a bug where opening the advanced settings window from the GNOM window didn't properly update changed settings in the GNOM window.
- Fixed a bug where .hdf5 files couldn't be plotted by dragging and dropping or using the 'Plot Series' button.
- Fixed a bug where running a dammin refine with too long a filename would fail (previously thought to be fixed in 2.0.0).
- Fixed a bug that could prevent DENSS from starting on Windows and Linux.
- Fixed a bug that could prevent auto determination of number of components in EFA.

5.9.6 2.0.1

Release date: 2020-06-01

Overview

The RAW team is pleased to announce the release of RAW version 2.0.1. This version fixes several serious bugs in the previous version, including:

- A bug where some of the M.W. calculations failed for profiles with a maximum q value greater than 0.5.
- A bug where the standalone DENSS alignment window failed to run on Windows and Linux.
- A bug where killing the prebuilt version of RAW on Linux would fail to delete temporary files, which could lead to the /tmp partition filling up.

There are also several other small bug fixes and new features.

All changes:

- Fixed a bug where the 'More Info' button didn't work for DATCLASS M.W.
- Fixed a bug where Bayes and DATCLASS M.W. weren't saved when saving all analysis info.
- Fixed a bug where RAW woud fail to load in .out files if they were missing values for any of the perceptual criteria.
- Fixed a bug where an error message was shown whenever a profile with qmax > 0.5 was opened in the M.W. window.
- Fixed a bug where the Vp M.W. extrapolation range warning could be shown even if the qmax selected was inside the extrapolation range.
- Added zero lines to DAMMIF, DENSS residual plots.
- Fixed a bug where running DENSS without averaging could result in an error message.
- Fixed a bug where running DENSS without averaging or alignment would result in an error message.
- Fixed a bug where when there was more than one profile in the normalized Kratky plot the dashed lines to guide the eye were not removed when switching from dimensionless Rg to other plots, which would throw off the scale of the plot.
- Fixed a bug on Windows where the standalone DENSS alignment window didn't work.
- Fixed bugs where the advanced options couldn't be shown in the SUPCOMB or DENSS alignement standalone windows on Windows.
- RAW now catches SIGINT and SIGTERM and tries to exit gracefully. This mostly fixes an issue with the prebuilt .deb installer where the temp files created when starting RAW don't get deleted.

5.9.7 2.0.0

Release date: 2020-05-07

Overview

The RAW team is pleased to announce the release of RAW version 2.0.0. This version brings a number of exciting changes, including:

- Python 3 compatibility!
- Conversion to pyFAI for radial integration
- A .deb prebuilt installer for Debian/Ubuntu
- A new series save format, .hdf5, that can be easily read by other programs. This new format is also ~50-70% smaller than the previous format.
- New information windows above the control panel to display all your analysis
- Ability to align DENSS and DAMMIF/N outputs to PDB files
- Works with ATSAS 3 on all platforms
- New Series adjustment panel that lets you adjust the scale, offset, and q range for all profiles in a series easily.

There are also a number of smaller new features, and a ton of bug fixes and tweaks.

With this release, we recommend that all users running RAW from source start using Python 3 instead of Python 2.

Important notes:

- RAW configuration files from 2.0.0 will not necessarily be back compatible with previous versions of RAW. You will get a warning if you load a .cfg file from RAW 2.0.0 in a previous version. Old configuration files can be loaded in RAW 2.0.0.
- RAW series .hdf5 files from 2.0.0 cannot be opened by previous versions of RAW. The old series .sec files can be opened by 2.0.0.
- This is the last release that is guaranteed to be Python 2 compatible, since Python 2 hit end of life in January 2020. While we will not intentionally break compatibility with Python 2, we don't have the resources to test on both Python 2 and 3.

Many thanks to the beta testers who helped test this release:

- Norm Cyr
- Richard Gillilan
- Rob Miller

All changes:

- All RAW code is now compatible with both Python 2 and 3
- Conversion to pyFAI for radial integration, including support for using detector angles, polarization correction, dark correction, flatfield correction, and dezingering.
- A new .deb prebuilt installer for Debian/Ubuntu
- A new series save format, .hdf5, which is easily read by any program that can load HDF5 data.
- Modified Control panel button layouts.
- Added a requirements.txt file for easy pip installation.
- Added ability to align DENSS results to a PDB or MRC file either from the main DENSS window or an auxiliary window.
- Updated DENSS to version 1.5.0.
- Added ability to view help documents in program on MacOS and Windows.
- Changed a number of default windows sizes and tweaked layouts to be better on all operating systems.
- Added all new info panels for Profiles, IFTs, and Series that display all relevant analysis information about a selected item.
- Added ability for users to permanently add new hdf5 file definitions to RAW from inside the program.
- Added a q cutoff to the volume of correlation calculation.
- Fixed a bug where the q cutoff for the porod volume MW calculation wasn't being used for Series files.
- Fixed a bug where the q cutoff for the porod volume MW calculation wasn't getting set properly in default mode when the MW window opened.
- Changed how imports are done on startup so that RAW can be run from the command line in two number of ways, including `python <path-to>RAW.py`, and as a module as `python -m bioxtasraw.RAW`
- Added ability to use SUPCOMB to align dammif/n output with a PDB file, either directly from the main dammif/n window from from a separate window.
- Fixed a bug where not all of the program output would get written to the log window for damaver, damclust.
- Added error handling for the main thread, which will reduce amount that RAW freezes from unexpected errors.

- Changes to prevent/reduce flickering on Windows.
- You no longer see two overwrite prompts when saving multiple items at once.
- Fixed a couple of options in the view menu that didn't work.
- Added a feature where if a user tries to load more than 100 profiles individually they are asked if they instead want to load them as a series.
- EFA ranges can now span the whole dataset, allowing better fits for components that are still eluting at the end of the EFA range.
- Added ability to easily apply the 8/Rg cutoff for dammif in the GNOM panel.
- Fixed a bug where RAW was not checking for unsaved IFTs on exit.
- Fixed a bug where IFTs would show as having unsaved changes when there were no unsaved changes.
- Changed Manipulation and Main Plot names to Profiles.
- Added ability for RAW to prevent the computer from going to sleep during long calculations, such as DENSS or DAMMIF/N.
- Fixed a bug where view menu items were not getting properly selected on startup.
- Added IFT plot axes to the view menu.
- Added LC Series Analysis to the Tools menu.
- Fixed a bug where an error message was displayed when quitting RAW.
- Fixed a bug where running DENSS on a BIFT P(r) function wasn't using the correct q & i vectors for the refinement.
- Changed item highlight color.
- Made masking, centering, and all options panels scrolling panels.
- Added busy dialogs for saving files.
- Fixed a bug where failing to save a file could crash RAW.
- Added a zero line to the normalized Kratky plot.
- Fixed a bug that could cause dammif/n, damaver, and damclust to stop running on Windows.
- Fixed a bug where new versions of ATSAS (>=3.0) wouldn't be found on Windows.
- Fixed a bug where if filenames got too long and were truncated for dammif the refine step failed.
- Fixed a bug where making lots of masks could result in an 'out of window ids' error.
- Fixed a bug where masks didn't stay inverted upon resizing.
- Fixed a bug where changing the qmin or qmax while BIFT was running an initial search gave an error.
- BIFT now saves the set q range and reopens the window with that range.
- BIFT and GNOM windows now default to the min q of the Guinier range if no analysis has previously been done.
- GNOM analysis now saves alpha value, restores it when window is reopened.
- MW analysis now saves Porod cutoff choice, density, and VC mol type choice and restores when the window is reopened.
- Fixed a bug where if you ran dammif or denss again in the same window the results summary wouldn't display properly.

- Fixed several bugs related to running ATSAS by properly setting the environment ATSAS and PATH variables.
- Fixed a bug where the wrong version of ATSAS could be fond on MacOS.
- Added ability to display P(r) functions on an I(0) normalized plot (set as default).
- Fixed a bug where custom toolbar buttons didn't display as toggled properly on MacOS.
- Added ATSAS MW methods Bayes and DATCLASS (Shape&Size) to the MW panel.
- Fixed a bug where running datgnom didn't respect the q ranges set in the GNOM window.
- Changes for compatibility with wxpython 4.1.0.
- Fixed a bug where moving a profile between plots didn't preserve all of the line/marker style settings.
- Fixed several bugs related to displaying and moving a legend on the plots.
- New Series adjustment panel that lets you adjust the scale, offset, and q range for all profiles in a series easily.
- Fixed a bug where the end point for profiles used in GNOM and BIFT was one data point earlier than specified in the controls.

5.9.8 1.6.4

Release date: 2020-03-10

Overview

The RAW team is pleased to announce the release of RAW version 1.6.4. This version adds in a new header type for the MacCHESS SAXS beamline Eiger 4M detector. There are also a few other minor bug fixes.

All changes:

- Fixed a bug where negative values for error would cause points to not be read from .dat files.
- Fixed a bug where the BIFT window wouldn't open if the profile and n min or n max for the q vector set to other than 0 and the length of the q vector.
- Fixed a bug where looking for bind list keywords that don't exist in the RAW settings would prevent a file from loading.
- Fixed a bug where damaver didn't run with symmetry even if dammif did.
- Added a name to the dock/menu bar icon.
- Fixed a bug with moving masks.
- Added CHESS EIGER 4M to counter file reader options.
- Fixed type-casting issues for max/min in polygonmasking that caused errors on some older systems.

5.9.9 1.6.3

Release date: 2019-11-01

Overview

The RAW team is pleased to announce the release of RAW version 1.6.3. This version fixes a critical bug where when average detected different files, regardless of user choice it would always average all files (selecting just similar files didn't work). There are also a few other minor bug fixes.

All changes:

- Fixed a bug where if you averaged, subtracted, or merged two items with analysis done on them, you could end up with partial analysis info in the resulting file that would cause errors opening analysis windows.
- Fixed a critical bug where when average detected different files, regardless of user choice it would always average all files (selecting just similar files didn't work).
- Fixed a bug where the MW window wouldn't open if the Guinier fit hadn't been done.
- Fixed a (Debian specific?) bug where wx.CallAfter used with wx.MessageBox wasn't threadsafe and could cause RAW to crash (use wx.MessageDialog).

5.9.10 1.6.2

Release date: 2019-10-28

Overview

The RAW team is pleased to announce the release of RAW version 1.6.2. This version fixes several critical bugs that could prevent DENSS from running. There are no other changes.

5.9.11 1.6.1

Release date: 2019-10-21

Overview

The RAW team is pleased to announce the release of RAW version 1.6.1. This version contains numerous small bug fixes and improvements, particularly for the new series analysis methods released in version 1.6.0.

We anticipate this will be the last release before RAW version 2.0, which will be our first python 3 compatible release. We're aiming to release version 2.0 sometime around the end of the year.

Note: as part of this release we did test with ATSAS 3.0 (pre-release version), and RAW seems to be compatible with it.

All changes:

- Fixed a bug where opening the Ambimeter panel could fail if ATSAS was installed in a directory with a space in the path name.
- Fixed a bug where if RAW didn't load a settings file when opened it wouldn't automatically find the ATSAS directory on startup.
- Fixed a possible bug where using the LC Series Analysis panel on series data being loaded in online mode could fail.

- Added intensity type selector for the series panel when sending data to the main plot.
- Fixed a bug where automatic integral baseline start/end region determination could set the wrong control limits.
- Fixed a bug where zero lines on various plots (like the GNOM P(r)) plot weren't getting redrawn when necessary.
- Fixed a bug with autoscaling in the LC Series Analysis plots when changing the data type displayed.
- Fixed a bug where sample and buffer region comparison in the LC Series Analysis panel could return the wrong indices for dissimilar profiles.
- Fixed a bug where profile comparison in LC Series Analysis would skip the first profile.
- Fixed a bug where a very short series (<22 frames) could cause errors when opening the LC Series Analysis panel.
- Improved automatic buffer search in the absence of major peaks.
- Improved automatic buffer search to search on the right side of the main peak if it doesn't find a good buffer region on the left side.
- Removed a bias to the left side of the peak in the automated sample region search.
- Removed actual baseline correction values from being saved in file history, as history could get very long (>100000 lines).
- Added a cutoff for header length, at which point RAW will stop saving file history. This avoids saving extremely large text files.
- Added compatibility for pyFAI 0.18 (note: on linux and python 2 pyFAI 0.18 seems to be broken, stick with 0.17).
- Added a new way of loading HDF5 files with definitions done in external files.
- Added ability to load HDF5 files from LiX.
- Fixed a bug where identical selected regions in the LC Series Analysis window didn't count as overlapping.
- Fixed a couple of typos in messages.
- Fixed a bug where Ambimeter and GNOM couldn't run if the current working directory was read only.
- Improved how ATSAS programs are called, and added use of temporary file names and directories.
- Fixed a bug where when dragging the image plot with the masking showing you could sometimes get an unexpected error.
- Added the name of the series to the LC Series Analysis panel.
- Fixed a bug where baseline subtracted profiles were improperly being skipped when calculating Rg etc in the LC Series Analysis panel.
- Fixed a bug where if a series was loaded with a baseline already calculated, changing buffer range or other parameters wouldn't properly recalculate baseline corrected values.
- Fixed a bug where if you set a baseline, then set it back to none, when exiting the LC Series Analysis window the baseline calculated values would be saved instead of the regular subtracted values.
- Fixed a bug where if you loaded in a series curve with baseline correction, then turned off baseline correction, it wouldn't have any calculated values.
- Fixed a bug where the LC Series Analysis panel would resize itself when any of the collapsible panes were collapsed or expanded.
- Fixed a bug in the LC Series Analysis panel where if you had a range that was one frame long, when you closed and reopened the analysis window you couldn't adjust the range.

- Fixed a bug in the LC Series Analysis panel where if you had a one frame long range you could get a reported correlation in the range.
- Fixed some issues with window height where windows weren't opening large enough for all of their contents.
- Fixed some issues with window size where windows could open up bigger than the screen.
- Fixed a bug where in certain circumstances opening the SVD and EFA windows could fail.
- Added compatibility with numba >= 0.44

5.9.12 1.6.0

Release date: 2019-06-07

Overview

The RAW team is pleased to announce the release of RAW version 1.6.0. This version contains several major changes:

- Completely new and improved SEC-SAXS processing, including new automated buffer and sample region selection and baseline correction. There are also significant speed improvements for SEC-SAXS processing, in addition to the new features.
- Completely redone BIFT, which fixes several bugs (both minor and major), and adds Monte Carlo error estimation and extrapolation fit of data to I(0).
- RAW now preserves matching metadata across processes like averaging and subtraction. Metadata is now saved with keywords compatible with the SASBDB to make uploading there easier for users. Users can now also provide arbitrary metadata during data reduction.
- All new icons which are compatible with retina displays, including changing out the check mark for showing/hiding data on plots for an eye, which we hope will be more intuitive.
- RAW now loads the last used config, rather than the last saved config, when it starts up.
- Any analysis window (Guinier, MW, GNOM, etc) can now be opened more than once, allowing easy comparison or side-by-side processing of data sets.

You can see the full set of changes below.

We also want to note that we're not anticipating any other major feature releases this year. With the upcoming end of life for python 2 at the end of 2019, we need to focus on making RAW work with python 3. Once that is done we will start doing major feature development again.

All changes:

- Updated DENSS to have the latest features, including refining averaged structures, symmetry constraints, and the 'Membrane' protein mode.
- Completely redid BIFT code from the ground up. This fixes several bugs, and now includes Monte Carlo estimation of errors.
- Completely redid series analysis for SEC-SAXS data. Now includes automated buffer and sample region determination and baseline correction.
- Added residual plots to GNOM, BIFT, and DAMMIF/N results.
- Fixed several bugs related to setting error bar line styles.
- Added ability to add arbitrary metadata to a file header when an image is processed by RAW

- Updated the adjusted Porod volume MW method to match the newly published MoW2 approach.
- Fixed a bug where info panel data could get improperly set
- All appropriate fields in MW panel now editable.
- You can now open any analysis window more than once (previously only one instance of each window was allowed).
- Fixed a bug where in the GNOM window changing q_min or q_max didn't update the IFT results.
- RAW now loads the last used config (saved or loaded) by default rather than the last saved.
- RAW now preserves all shared header values when averaging, subtracting, or merging datasets.
- Added visual guidelines to the dimensionless Kratky plot.
- Added option to display normalized residuals, now on by default.
- Added Rigaku HiPix to known images (requires Fabio 0.9.0)
- Guinier panel can now export Guinier fit data so users can make the Guinier plots in their plotting software of choice.
- RAW's file list no longer displays hidden files.
- Can now read in time of each data point for BioCAT data.
- Fixed a bug where closing the BIFT window with BIFT running would crash RAW.
- Better formatting for numbers displayed in the status bar.
- Fixed a bug where windows could be too large on low resolution displays.
- Fixed a bug where series plot calculated data were not highlighted by the locater button.
- Fixed a bug where markers were not highlighted by the locater button for any plot.
- Fixed a bug where when selecting a line by clicking on it the plot markers were not highlighted.
- Fixed a bug where selecting a line on the IFT plot didn't work.
- Can now display unsubtracted, subtracted, or baseline corrected intensity in the main series plot.
- Fixed a bug where series data could be truncated when exporting.
- Fixed a bug where the SVD window wasn't doing the SVD on non-error-normalized curves.
- Moved cormap to cython for speed, increased by at least 5x.
- Modified layout of the repository to standardize.
- Autorg now uses numba for just-in-time compiling. Speed increase of 2 orders of magnitude.
- Fixed bugs that would occur when quick reduce, plot, plot series, or show images were used on folders, '..', or with no files selected.
- Added ability to plot intensity over a q range for series plots.
- All-new icons that work with retina displays, including a new 'eye' for show/hide instead of a check box (hopefully more intuitive).
- Fixed a few bugs in the DAMMIF/N GUI.

5.9.13 1.5.2

Release date: 2019-04-04

Overview

The RAW team is pleased to announce the release of RAW version 1.5.2. The only change is this version is a modification to how BioCAT header files are read in, to accommodate a new header file format at that beamline.

5.9.14 1.5.1

Release date: 2018-11-01

Overview

The RAW team is pleased to announce the release of RAW version 1.5.1. This version contains several small bug fixes. Normally we might wait to release these until more major changes had happened, but there is a workshop using RAW soon and we wanted these bugs fixed before then. Significant issues that have been eliminated:

- An issue where the electron density output from DENSS could fail to load into pymol correctly because the default scaling was too small (still loaded into Chimera fine). Strictly speaking I think this is a workaround for a bug in pymol . . .
- Several bugs with running GNOM, including using data with minimal sampling (<100 points).
- Fixed a bug where the .app package for Mac wasn't displaying natively on retina displays, so the text was fuzzy.

You can see the full set of changes below.

All changes:

- Fixed a bug where automatic loading of BioCAT SEC data wouldn't work if there was more than one underscore in the filename.
- Fixed a bug where automatic loading of BioCAT SEC data wouldn't work if there existed another file with the same name but different extension as one of the image files.
- Added parsing of damsup.log file for bead models, which allows highlighting of the representative model in the dammif summary.
- Dammif results summary now saved by default.
- DENSS results summary now saved by default.
- Fixed several bugs in the GNOM interface that could cause it to fail.
- Fixed a bug that prevented some .fit files from being loaded.
- Fixed a bug where the Rg for BIFT was being calculated incorrectly.
- Fixed a bug where temporary files (with a .tmp prefix) would mess up SEC autoupdates.
- Fixed an issue where you couldn't run DENSS twice without closing the panel between each run.
- Fixed an issue where the default scaling for DENSS was too small, and caused issues loading the electron densities into pymol.
- Fixed a bug text in some items and list controls would display 'fuzzy' on high dpi monitors. This is still an issue for the plot labels.
- Added the ability to run damaver and damclust on the same set of reconstructions.
- Fixed a bug where the .app package for Mac wasn't displaying natively on retina displays, so the text was fuzzy. Note that in order to fix this, even after you install the new version you may have to do the following:

1. Enter the following commands in your terminal:

```
/System/Library/Frameworks/CoreServices.framework/Frameworks/LaunchServices.
↪framework/Support/lsregister -f /Applications/RAW.app
/System/Library/Frameworks/CoreServices.framework/Frameworks/LaunchServices.
↪framework/Support/lsregister -kill -seed
/System/Library/Frameworks/CoreServices.framework/Frameworks/LaunchServices.
↪framework/Support/lsregister -f /Applications -all local,user
```

2. You may then have to right click on RAW.app, select 'Get Info' and uncheck the box 'Open in low resolution mode'

5.9.15 1.5.0

Release date: 2018-08-23

Overview

The RAW team is pleased to announce the release of RAW version 1.5.0. This version focused on several significant updates that will be invisible to most users. Namely:

- RAW is now compatible with wxpython4
- RAW no longer uses weave, which has been essentially unsupported for years, to compile code. It now uses the numba just-in-time compiler.

This will make it much easier for us to support RAW, and should make it easier for users to install RAW from source on any platform. It also prepares us for the inevitable transition to Python 3 that has to happen in the next several years.

In addition to a range of bug fixes and small enhancements detailed below, RAW also now incorporates the new DENSS alignment code. This is all done in python, in RAW. This removes the dependency on EMAN2, and means that all parts of density reconstructions work on Windows!

Finally, RAW is now saving configuration files in JSON format. This is human readable, and makes the RAW configuration files more open and accessible for other programs to use. However, this does mean that earlier versions of RAW will not be able to open configuration files created with version 1.5.0 or later. However, configuration files created in earlier versions of RAW ARE compatible with version 1.5.0.

All changes:

- Fixed a bug where if atsas is in the path but not installed RAW will still find the directory from the path.
- ATSAS filepaths and filenames should be able to deal with spaces.
- Fixed various strange threadsafe bugs on debian 8.
- Weighted average now checks for similarity
- Fixed a bug where the dammif results window wouldn't work when you did only one dammif run and had damaver checked.
- Fixed a bug where dammin in normal mode wouldn't work on windows.
- Fixed a bug where dammif/n wouldn't abort on windows.
- Added in new expected shape parameter for dammif in custom mode.
- Fabio, hdf5plugin, and pyfai are now required dependencies, rather than optional dependencies

- Mode all previously compiled code into using the numba just-in-time compiler. This is important because the previous code was compiled with weave, which has been unsupported for years.
- Fixed a bug where users could give dammif/n file prefixes that were too long for damaver.
- Fixed a bug where canceling out of the color change dialog didn't cancel the color change.
- Made the plot options box resizable (important for computers with large font size).
- Fixed a bug where the sec plot right axis framestyle wouldn't properly restore if you canceled out of the plot options dialog.
- Significant code restructuring and cleanup.
- EFA calculations are now in a thread, so it might not freeze the whole GUI.
- Circle and rectangle masks are now resizable.
- Added ability to automatically mask pixels at/above/below a given threshold.
- Added ability to automatically mask images based on known detector panel gaps.
- Added ability to create predefined size/location circle and rectangle masks.
- Added ability to control detector image left-right flip and up-down flip.
- Fixed a bug where RAW could crash under certain conditions when exporting analysis info.
- Fixed a bug where the Guinier window would give an error under certain circumstances.
- GNOM and BIFT windows now show scattering profiles on log-lin axes.
- RAW is now wxpython4 compatible.
- Added alpha as an available setting in the GNOM window.
- Fixed several bugs in the GNOM window that caused RAW to unnecessarily calculate the P(r) function, slowing down the program.
- Added drag and drop file loading for both the plot and control panels.
- Settings are now saved in JSON format, which is human readable, to increase compatibility and ease of use by other programs. This means that settings saved from RAW 1.5.0 are not compatible with previous versions of RAW. Settings saved from previous version of RAW ARE compatible with RAW 1.5.0.
- DENSS now uses custom python code for aligning and averaging density. This removes the requirement on EMAN2, which means all parts of DENNS will work on Windows.
- The image plot now maintains the same zoom when you change images. Previously it would zoom back out to the full image whenever you showed a new image.
- Fixed a bug where the SVD would sometimes not open correctly.
- Fixed a bug where if there was one pixel in the q bin during integration the error would be set to 0 instead of the square root of the value
- Fixed a bug where nans or infinities in the SVD matrix would break SVD/EFA without an appropriate error message.

5.9.16 1.4.0

Release date: 2018-03-20

Overview

The RAW team is pleased to announce the release of RAW version 1.4.0. This is a major feature release for us! The big new feature is that RAW can now use the DENSS method to calculate electron density from SAXS scattering! You can read more about this at http://denss.org/.

To fully use this new feature (for density averaging and enantiomer filtering) you have to install EMAN2 (http://blake.bcm.edu/emanwiki/EMAN2/Install) which, sadly, doesn't work on Windows. Windows users can still generate densities, but they won't be able to average them. A new tutorial on DENSS in RAW is now available in the documentation (https://bioxtas-raw.readthedocs.io/en/latest/tutorial/s2_denss.html).

The other feature many folks will be interested in is the new error calculation for Guinier fits, which is a much requested feature. This is now available whenever you open the Guinier panel, and saves and exports with the rest of the analysis information as expected.

We've also done the usual set of bug fixes and tweaks. You can find a full list of changes below.

All changes:

- Added DENSS method for calculating electron density from SAXS profiles
- Added support for EMAN2 averaging and enantiomer testing of DENSS results
- Fixed a bug where the GNOM window could fail to exit and save the .out file to the IFT tab
- Changed the default DAMMIF mode to slow.
- Changed when the 'please wait' message appears when loading SEC-SAXS files in autoupdate mode. Now it only shows up if more than 5 files are loaded at once.
- Fixed a bug where advanced options for GNOM and DAMMIF couldn't be set while the respective analysis windows were open.
- Fixed a bug where the spectral color map couldn't be displayed, breaking the image control panel.
- Fixed a bug where ambimeter would try to run in the DAMMIF window even if ambimeter wasn't available.
- Fixed a bug where if files were averaged or subtracted and had analysis history, that analysis would get transfered to the new file.
- Fixed a bug where Guinier fit limits would be improperly displayed on the plot when the Guinier window was first opened.
- Fixed a bug where calls to set up the DAMMIF results window could be non thread safe.
- Added estimate of the parameter (Rg and I0) errors for a Guinier fit.
- Reformatted the MW display to make it more compact.
- Changed how numbers are displayed in all of the analysis windows, to better handle very large or very small values.
- GNOM, Ambimeter, DAMMIF windows now open much faster.
- Added support for BioCAT header files (new style).
- Added support for autoloading of BioCAT Series curves.
- Added GNOM P(r) parameters (Rg, I0) errors to the GNOM window, and the estimated Guinier errors.
- Guinier parameter errors and GNOM P(r) parameter errors are now saved with profiles, and with analysis info spreadsheets.
- Fixed bugs where spin controls could raises errors if a user entered a blank value.

- Values from analysis windows are now saved with more precision.
- Rearranged the manipulation item right click menu to make it more compact, put some less-used items on sub-menus.
- Changed 'SEC' labels to 'Series' labels.
- Fixed an off by one error in SEC autoupdate that could occur for certain file names.
- Renamed and rearranged some menu items in the IFT item right click menu.
- Added universal newline support when loading in scattering data.
- Fixed a bug where averaging could fail if all the averaged files were different form the first file.
- Fixed a bug where similarity testing could fail with an overflow error if there were too many points in the scattering profile.
- Minor improvements to plotting speed with large numbers of files.
- Fixed a bug where having no positive values in a curve displayed on a log-y axis would cause an error.
- Updated the documentation to include a DENSS tutorial. Updated various other parts of the documentation, including the images, to reflect other new features.
- Updated all of the installation documentation.
- Removed the RAW-Windows-Source-Install-Essentials file from the downloads.

5.9.17 1.3.1

Release date: 2017-11-01

Overview

The RAW team is pleased to announce the release of RAW version 1.3.1. This is a very minor release. Several small bugs have been fixed, and we have updated the citations in the program to reflect the release of the new RAW paper. Most of the major work in this release went into updating the documentation, which we have already released on the new website: https://bioxtas-raw.readthedocs.io/

All changes:

- Made RAW compatible with pyFAI 0.14 (not back compatible with 0.13)
- Improved the multiwire loading function
- Updated some citations and error messages in the program
- Revamped and updated all of the documentation and tutorials. It is now in sphinx format, in the RAW SVN for better tracking.
- Updated the RAW citation to reflect the newly released RAW paper.
- Updated the .app build on mac.

5.9.18 1.3.0

Release date: 2017-08-19

Overview

The RAW team is pleased to announce the release of RAW version 1.3.0. This release is a major feature release, and we're very excited that you get to use all of the fun new stuff we've added in! The major new features are:

- Similarity testing for scattering profiles using the CorMap test. This allows statistical testing of whether or not profiles are similar. This is done automatically when averaging profiles or picking a buffer region of a SEC curve, and is also available in the right click menu for profiles, IFTs, and sec files. In the automatic check, if it detects files that may be different, you'll see a message asking you how you want to proceed.
- Normalized Kratky plots can now be made, and are accessible through the right click menu.
- We've added a results summary panel for dammif/n reconstructions that shows the NSD, resolution (if SASRES is installed), and statistics about the individual reconstructions including chi squared, rg, dmax, excluded volume, and molecular weight. There is also a new dammif results viewer panel that lets you get a basic look at the reconstructions (this panel is still very simple).
- Absolute scaling can now be done using the NIST glassy carbon standard SRM 3600.

In addition to all of these major changes, we've made the usual range of small tweaks, bug fixes, and enhancements. See the full list of changes below.

Finally, we're happy to announce that we're also releasing a new tutorial, that has been updated to include tutorials for all of the new features mentioned above!

All changes:

- Fixed a bug where switching between linear and log scale in the image display could change the overall scaling of the image without changing the displayed limits in the dialog.
- Added a new dammif/n results summary panel.
- Added a new dammif/n results viewer panel.
- Added a new normalized kratky plot panel
- Changed how multiple images in a single file are deal with when loaded as a sec curve (now each is loaded as an individual point on the curve).
- Added a new check for statistical similarity between profiles (or IFTs or SEC curves).
- Now on average, RAW automatically checks whether the profiles are statistically similar.
- Fixed a bug where the first image loaded from a file with multiple images in t was flipped left-to-right relative to the rest.
- Fixed a bug where if a configuration file is loaded and doesn't contain certain setting keys (a configuration made with a previous version where those settings don't exist, for example), those settings are now set to default, rather than left as whatever is loaded in RAW.
- Added ability to view all images in a single file if the file contains more than one image.
- Added ability to use glassy carbon (NIST SRM 3600) to calibrate absolute scale.
- Fixed a bug in subtraction that could result in the q and i vectors being rounded.
- Fixed a bug where if the beam center was in the masked region of the image it could be assigned a non-zero value.
- Fixed a bug where a RAW setting for a choice type with default value of None could cause an error when trying to set the field in the Advanced Options window.
- Added a check for syncing items to make sure that an item is starred and an item is selected.

- Added ability to reset all settings to default values from the advanced options panel.
- Marker face, marker edge, and error bar colors are now saved when you save a workspace.
- Error bars now show up correctly for Guinier, Kratky, and Porod axes in the Main Plots.
- Added ability to use error weighting in fits, and ability for user to toggle that on and off in the advanced options panel. Fitting is now by default done with error weighting.
- RAW can now load .txt files.
- Fixed a bug where on a single core machine there would be no default selection for the number of simultaneous runs in the dammif/n window.
- Font list now includes matplotlib fonts
- Changed LaTeX symbols to default to regular instead of italics.
- Fixed a bug where line size on a plot would change when opening/closing the line properties window without making any changes to the line size in the window.
- Added ability to use fractional line sizes.
- Fixed a typo in the readme
- Removed a message asking if you're sure you want to load the workspace.
- RAW now checks whether or not you're saving something when it quits. If it is saving something, it warns you that you might now want to quit.
- Legend labels are now saved with a workspace.
- Fixed a bug where the legend label for IFT items would get changed from the default when you opened the line properties window.
- Fixed a bug where the calculated markers for a SEC item would show when loading a workspace even if the item wasn't supposed to be visible.
- Added sync and superimpose to the right click menu, tools menu.
- Added the program version to integrated dat files history.
- Added integration method and calibration parameters to the integrated dat files history.
- Fixed a bug where a dammin refine would try to run even if damaver didn't run.
- Fixed a bug where superimpose could break for different q vectors.
- Fixed a bug where the slider and custom color boxes in the color dialogs didn't change line/marker colors.
- Fixed a bug where in autoupdate mode the SEC plot could fail to switch between rg, mw, i0 on the right axis.
- Fixed a bug where you couldn't resize custom question dialogs.
- Fixed a bug where SVD/EFA wouldn't work with some sec data loaded in autoupdate mode.
- Fixed a bug where when updating the SEC data in autoupdate mode, an improper q value could be used when getting the intensity at a given q.
- Fixed a bug where if improper values were entered in the buffer range or window size and the set/update parameter button was pressed, if autoupdate mode was on it would stop.
- Removed the error printing on startup that backup.ini file could not be found.
- Fixed a bug where carrying out EFA to panel 3, then going back to panel 1 and changing the frame range used, then carrying out EFA again could cause an error in the rotation.

- Fixed a bug where for unsubtracted profiles from images, EFA would use the full profile rather than the appropriately truncated profile.
- Fixed a bug where the options panel couldn't be opened twice in windows.
- Added a check to prevent errors with missing lines when changing plot type in the main plots.
- Added a check to prevent index errors when setting the q range of a sasm.
- Fixed a bug where online mode would show an error if the directory being watched was removed.
- Added a choice in the GNOM panel to force dmax to zero or not.
- Added ability to use superimpose to find scale, offset, or scale and offset.
- Fixed a bug where EFA results wouldn't export due to getting the wrong q values from the scattering profiles.
- Fixed a bug with new versions of numpy not integrating images correctly. (actually fixed in 1.2.3 rerelease)
- Changed the generic error message. (actually fixed in 1.2.3 rerelease)
- Fixed a bug where temporary files that vanish in the online directory could raise an error. (actually fixed in 1.2.3 re-release)
- Fixed a bug that could cause intensity integration to fail in the sec plot. (actually fixed in 1.2.3 re-release)
- Fixed a bug where calculating the scale constant of water could cause the main thread to lock up if it had an error.
- Verified compatibility with ATSAS 2.8.2.
- Fixed a bug where in the prebuilt windows version any plots not in the main window (for example, Guinier plots) couldn't be saved.
- Fixed a bug where line colors didn't reset properly when canceling out of any of the line properties dialogs.
- Fixed a bug where the SVD window could have no default selection for type of profile to use.
- Fixed a bug where the advanced options window didn't open properly centered on the parent window.
- Minor speed improvements from code streamlining.

5.9.19 1.2.3

Release date: 2017-05-08

Overview

The RAW team is pleased to announce the release of RAW version 1.2.3. The release again mostly focuses on bug fixes, speed improvements, and other small improvements to the user experience. There is one bit of exciting news: we are releasing a prebuilt version for Mac! Users can now download a .dmg with a RAW.app in it. This can be installed via drag-and-drop, like other app files, and run just like any other app. We hope this will make installation much easier for mac users. If you want to try this, the download is available in the usual area, and the mac install instructions have been updated.

In addition to the new prebuilt version, we've also made errors more obvious, now if there is an unhandled error in the program, rather than failing silently it will pop up a dialog box to let you know. We're hoping this is seen as an improvement!

All changes:

- Made numerous changes to fix strange behavior in frozen version on mac
- Created instructions for building a frozen version on mac
- RAW icon now shows up in the dock instead of the top bar on mac
- Fixed how RAWWorkDir is used in the program, and how it gets set. It now gets set appropriately for each type of OS
- Switched to using an embedded version of the BioXTAS logo, for easier packaging
- Changed the default directory for RAW if there is no previous directory. It now uses the documents directory
- Fixed a bug in the Porod volume calculation that in some cases could extrapolate to q<0
- Added Guinier extrapolation to the volume of correlation molecular weight calculation
- Changed how Guinier extrapolation is done for the adjusted porod volume method
- Updated some of the text in the More Info buttons of the MW panel
- Fixed a bug where changing the q vector of a scattering profile would print an error in the console
- Updated the A and B coefficients for the adjusted porod volume method to perfectly match those used in the paper
- Updated GNOM and BIFT windows to both report reduced chi squared values
- Fixed a bug where having an ROI mask set could prevent loading image headers in the calibration section of the advanced settings
- Tweaked the MW, GNOM, and BIFT GUIs
- Fixed a bug where GNOM wouldn't run on SL6 with ATSAS 2.7.2
- Fixed a bug where rescaling profiles wouldn't work on a kratky plot
- Fixed a bug where IFT data plot could display the wrong scale for Guinier and Porod plots
- Changed how Guinier plots are displayed from I vs. q2 on a loglin scale to log(I) vs. q2 on a linlin scale, to match with labels shown on the plot
- Fixed a bug that could cause autorg to crash
- Changed the circle masking tool to draw more quickly/smoothly
- Improved responsiveness of dragging masks on an image
- Fixed a bug where the beam center wouldn't turn off if the masking panel was closed
- Improved responsiveness of updating positions of calibrant rings and beam center when working in the centering panel
- Fixed some bugs that could happen when switching between calibration and masking windows without hitting the okay or cancel buttons first
- Fixed a bug where VC integration was highly unstable in some cases, required switching from simpsons method to trapezoid method for numerical integration
- Attempted to fix a not reproducible bug where clicking the clear all button could cause a segfault on linux
- Fixed a bug so that the info panel is only cleared if the user actually decides to clear all items when clicking the clear all button
- Fixed a bug where loading FoXS files with fits would not load the fit

- Fixed a bug where PIL.Image couldn't load files (prevented loading of SAXSLAB300 images)
- Fixed a bug where if an image load returned no header, RAW could crash
- Did some futureproofing in the code
- Fixed a possible memory leak when loading certain image types
- Attempted to fix an irreproducible bug where masking would fail because pixel positions were floats instead of ints
- Error bars, if shown, now move properly with the line on scale and offset
- Fixed a bug where the Guinier window didn't respect the q limits set on the manipulation panel
- Set the default plot type to log-lin instead of lin-lin
- Fixed a bug where using the next/previous image buttons would cause the image to flicker if a fixed range were set for the color scale
- Fixed a bug where scaling q didn't mark the item as modified
- Fixed a bug where online mode loading more than one image didn't update the image plot
- Changed the green for the average file name text from green to forest green, which may be easier to read
- Fixed a bug where the centering panel being displayed without an image loaded could cause an error.
- Fixed a bug where the ATSAS 2.8.0 GNOM wouldn't run if an Rg for the profile had not been calculated.
- Fixed a bug where DAMCLUST wouldn't run.
- Added a global error handler to pop up a dialog for unhandled errors.
- Attempted to fix a bug where the program could run out of control ids on mac, causing a crash.
- Fixed a bug where damclust and dammin refine could both be selected in the advanced options window.
- Fixed a bug where dammin refine could be selected without damaver being selected in the advanced options window
- Fixed a bug where setting a flatfield image could fail if there wasn't an absolute scale normalization factor set
- Fixed a bug where GNOM and BIFT autosaving could be turned on without directories selected.
- Fixed a bug where switching from linear to log scale or vice versa with limits locked in the image display would set the slider bar maximum value incorrectly.
- Removed tifffile.py (no longer used).
- Fixed a bug where automated centering wouldn't work with the newest pyFAI
- Fixed a bug where typing an incomplete LaTeX expression in the plot label could cause an error.
- Added some error checking to running GNOM/DATGNOM in case it fails for some reason.

5.9.20 1.2.2

Release date: 2017-03-10

Overview

The RAW team is pleased to announce the release of RAW version 1.2.2. This release mostly focuses on bug fixes, speed improvements, and other small improvements to the user experience. However, there are several changes/new features we think many of our users may want to know about:

1. RAW now has the ability to do weighted averages of scattering profiles (accessible by the right click menu in the main control panel)
2. RAW is now compatible with ATSAS version 2.8.0.
3. You can now run DAMMIN from RAW (previously on DAMMIF was available). This includes using DAMMIN to refine the damstart.pdb file output from DAMAVER, which is now the default option.
4. RAW can now handle files with multiple images in them, such as Eiger hdf5 files. This is an ongoing project, so some features, such as image viewing and SEC plotting do not yet handle these types of files perfectly.
5. We have changed how the show/hide and collapse/expand buttons work. Previously they affected all items. Now if no items are selected they affect all items, otherwise they affect the selected items. We hope that once users are accustomed to this change they will find it useful.
6. RAW has a new header type available, P12 Eiger header files.

Additionally, RAW users should be aware that we have added an additional dependency, the weave package (to replace scipy.weave, which was removed in scipy version 0.19), and that RAW is not yet compatible with matplotlib version 2.0 (released January 2017). We are working on updated install instructions to reflect these changes, and those will be available (hopefully) next week.

As always, we appreciate user feedback, as that is how we improve the program. If you have questions, need help, or want to report a bug, please contact us!

All changes:

- Added ability to do a weighted average in RAW, using either error based weighting per q point or weighted by a counter value.
- Many small changes to the code to streamline how plotting works, which should results in modest speed improvements, particularly when working with large numbers of plotted files.
- If autoscaling is on for plots, plots should now autoscale in all appropriate instances (previously they didn't autoscale when moving items between plots, rescaling the q range, and a few other instances)
- Trimmed out many dead functions to make the code easier to maintain.
- Changed how the visibility check box for control panel items works, which improved show/hide speed for a single item when lots of files were loaded by a factor of 2.
- Improved speed for certain actions that resulted in marking lots of items as modified.
- Fixed a bug in autorg where error for the rg value could fail to be calculated
- Fixed a bug in running GNOM for ATSAS <2.8 where certain advanced settings couldn't be used.
- Fixed a bug where flatfielding would fail when using pyFAI to integrate images (not yet publicly available)
- Fixed a bug where using the roi_counter would fail when using pyFAI to integrate images (not yet publicly available)
- Fixed a bug where dezingering would fail using python for integration (instead of the compiled c++ modules)
- Removed the SASIft.py file that was unused.
- Fixed a bug where having nothing entered for limits in the plot options panel (such as when typing a new limit) would print an error message in the console.
- Fixed a bug where loading a roi_counter header value with no image header would give an error.
- Fixed a bug where legend position wasn't maintained when all items were removed or hidden on a plot.

- Fixed a bug where the legend wouldn't go away if all items on the SEC plot were hidden and there had previously been a legend.
- Updated how legend settings are handled in plot options to improve speed and maintainability.
- Fixed a bug where plot titles and axes labels didn't reset appropriately when using the clear all button.
- Fixed a bug where the plot options font selector boxes didn't work.
- Fixed a bug where not all settings were restored to previous values when canceling out of the plot options dialog.
- Fixed a bug where the Porod volume calculation was not getting properly interpolated to q=0.
- Fixed a bug where hitting the next/previous image buttons in the RAW Image plot would throw an error and crash RAW if the image currently displayed wasn't in the current working directory of the Files panel.
- Fixed a bug where saving items wasn't threadsafe on scientific linux 6.
- Fixed a bug in how the error bars for log(I) were calculated in the autorg function.
- Switched the autorg to calculate the Guinier fit without error weighting, to match how it is done in the Guinier panel.
- Fixed a bug where the how to cite button in the dammif frame wasn't getting properly placed in wxpython < 3.0.
- Addeed the ability to run dammin from the DAMMIF (now DAMMIF/N) window.
- Added the ability to use dammin to refine damstart files from dammin/f in the DAMMIF window.
- Fixed a bug where autoMW, autoRG did not respect the limits set for the scattering profile in the manipulation controls.
- Changed how the show/hide and collapse/expand buttons work. Previously they affected all items. Now if no items are selected they affect all items, otherwise they affect the selected items.
- Added compatibility for gnom5 from ATSAS 2.8.
- Counters available for normalization now show up in the combo box in the normalization list panel.
- Made some progress fixing a windows specific bug having to do with hitting enter after clicking a button in another panel.
- Fixed some bugs on windows where the mouse would get captured and not released by txtctrl boxes.
- Fixed a bug where the rename option wasn't working in the file overwrite dialog.
- Moved the version number into the RAWGlobals.py file.
- Improved speed of saving items from RAW, by a factor of ~160x for a large number of files on my test machine.
- Fixed a bug in the Guinier panel where the maximum point shown in the plot and used for the fit was one less than the maximum point shown in the spin control.
- Tweaked the autorg function to allow some intervals with qmaxRg > 1.3 (up to 1.35) to improve fitting.
- Fixed a bug where interpolate did not work on multiple selected scattering profiles.
- Fixed a bug where interpolate was giving the interpolated file the wrong name.
- Fixed a bug where writing the header could cause RAW to crash due to improper json serialization.
- Changed how normalization deals with zero values. Instead of raising an error it prints a warning.
- Added the GNU disclaimer at the top of all .py files that didn't have it.
- Added a header type for P12 Eiger, Petra III
- Updated image loading and all associated functions to handle multiple images in a single file, for example eiger files.

- Added filtering of headers so that () and [] characters are removed, as header names with these characters could not be used for normalization.
- Fixed a bug where image and other headers were getting filtered differently.
- Added some new file types to the TestData folder.
- Added error catching for json formatting of file headers upon save. If the header can't be serialized properly, the files saves without a header (used to cause a crash).
- Fixed a bug where ambimeter could fail if there were spaces in the filename.
- Fixed a bug where with older versions of wxpython and matplotlib, failure to find points in the autocentering mode could cause RAW to freeze.
- Fixed a bug where quick reduce would crash if it couldn't find the header.
- Replaced the dependency on scipy.weave with the weave package (which is a fork of scipy.weave), as scipy.weave is removed in scipy 0.19.

5.9.21 1.2.1

Release date: 2016-12-02

Overview

The RAW team is happy to announce the release of RAW version 1.2.1. This version focuses on bug fixes and small improvements to the user experience. There were a few significant changes:

1. In addition to numerous bug fixes, the EFA technique can now be used with explicit, iterative, or hybrid methods for computing the concentration profiles of the components. Previously, only the iterative approach was available.
2. We added a new automated centering and calibration routine using the pyFAI library, for better determination of beam center and sample-detector distance.

In addition to a new version of RAW, we have also released new installation instructions for all platforms.

As always, we appreciate user feedback, as that is how we improve the program. If you have questions, need help, or want to report a bug, please contact us!

All changes:

- Updated online mode so RAW only plots files if there are files to plot. This prevents some flickering when files enter the directory but are not plotted for any reason (such as not being suitable images).
- Updated online mode so that the "Processing incoming file…" status doesn't linger forever after an image is processed, but goes away suitably quickly.
- Fixed a bug that prevented EFA from running on scattering profiles that don't use the full range of their q vector.
- Fixed a bug where concentration wasn't saved when the 'save all analysis info' option was used.
- Fixed a bug where changing SEC plot axes while SEC live update is going could cause a crash
- Fixed a bug where Normalization information got saved in the scattering profile processing parameters twice, once with a capital N, once with a lowercase n.
- Fixed a bug where the wrong upper limit was getting set for the end of range controls in the third EFA control panel.

- Fixed a bug where if no normalizations were set in the normalization list, the solid angle correction would not be saved in the normalization history list for the scattering profile.
- Made a change where if EFA has a converged solution, if the ranges are changed it uses that solution as a starting point. This leads to faster convergence to the new solution.
- Added ability to display calibration rings from any calibrant in the pyFAI library.
- Fixed a bug where plotting certain scattering profiles on a Kratky plot would cause RAW to crash
- Fixed a bug where having the SEC plot set to display the intensity at a particular q value would prevent structural parameters from being calculated, and in some cases could prevent new SEC items from being plotted.
- Fixed a bug where the plot legend wasn't updated if the plot was turned on, then off, and then items were removed from the plot.
- Added an energy box in the centering and calibration window, so that if energy is entered, wavelength is automatically calibrated, and vice versa.
- Fixed a bug where changing centering values with no centering values selected could crash RAW.
- Added ability to explicit calculation of concentrations for EFA, as opposed to currently iterative method.
- Added ability to use a hybrid method for calculation EFA, using the explicit calculation as a starting point, then refining iteratively.
- Added ability to chose rotation method for EFA in the third EFA control panel.
- Fixed a bug where the range plot in the third EFA panel was not refreshing properly when the number of significant values was changed.
- Fixed a bug where the info panel was not updated when a scattering profile was selected by clicking on it on the main plot.
- Updated build commands for making a windows installer, including adding some explicit hooks for pyFAI and pyinstaller.
- Added the optional use of the hdf5plugin to RAW to support eiger images.
- Fixed a bug in the image display where the dialog box could fail to open because the maximum value in the image was greater than 2^31-1 (the maximum value a wx slider can handle).
- Added a feature for automatic centering and fitting of the beam center and sample to detector distance. Requires pyFAI to be installed.
- Added a header reader for g1 eiger files, which have the spec header file one level up from the image files.
- Fixed a bug where the RAW ROI could not consistently be used for normalization.

5.9.22 1.2.0

Release date: 2016-10-25

Overview

The RAW team is very pleased to announce the release of version 1.2.0. We've added two major new features, the first of which is the ability to perform SVD on a set of scattering profiles, IFTs, or a SEC-SAXS curve. We've also implemented the exciting new evolving factor analysis (EFA)[1] method for deconvolving overlapping data. This is primarily intended to be applied to SEC-SAXS data, but it is implemented so that it can be applied to any set of scattering profiles or IFTs. We want to note that while EFA is an exciting new technique, it is still in ongoing development. We intend continuing development on the stability and utility of the algorithm.

We will release an updated tutorial document and dataset which includes examples of doing SVD and EFA soon.

As always, we appreciate feedback from users, either positive or negative.

The RAW Team

[1] Steve P. Meisburger, Alexander B. Taylor, Crystal A. Khan, Shengnan Zhang, Paul F. Fitzpatrick, and Nozomi Ando. Journal of the American Chemical Society 2016 138 (20), 6506-6516.

All changes:

- Added the solid angle correction to the normalization parameters in the sasm history, so that if it is used, that use is recorded.
- Fixed a bug where SAXSLAB images could not be loaded when using version 3.0 or newer of the pillow library.
- Added in the ability to use a RAW defined beamstop mask in addition to a SAXSLAB beamstop mask for SAXSLAB data.
- Fixed a bug (on OSX, wxpython 3.0) where clicking the OK button in the Masking Panel was returning the plot window to the IFT panel instead of the Main Panel.
- Added in some dialog boxes letting users know they can’t modify the SAXSLAB header mask in RAW. Previously, the Remove and Set buttons in the masking panel appeared to work for the SAXSLAB header beamstop mask, but in reality did nothing. Now they still do nothing, but pop up a dialog letting the user know that nothing has happened (and no longer appear to do anything).
- Added a molecule type choice to the SEC calculate parameters panel, so that the user no longer has to change the default molecule type in the mol weight options panel.
- Fixed a bug where the Clear All button was not properly clearing some fields in the SEC control panel.
- Added SVD capability.
- Fixed a bug which prevented some .sec curves from being loaded.
- Added overwrite checking to the .sec saving function.
- Fixed a bug where the SEC item filename didn’t change when the item was saved with a different name.
- Made how SEC names are deal with consistent with how scattering profile names are dealt with.
- Added overwrite checking to the Export data option for SEC curves.
- The parameters on a SEC plot now default to markers, not lines.
- Fixed a bug where in a 3 column data file with no non-data first line (empty or otherwise), the first data point would get cut off.
- Added evolving factor analysis (EFA) capability
- Added ‘How To Cite’ buttons for the RAW functions that incorporate other people’s work, so that they can correctly cite the methods.
- Added in backwards compatibility for loading .sec files from previous versions of RAW, and workspaces with saved .sec files from previous versions of RAW.
- Saving/Loading a workspace now preserves the file order in the workspace.
- Fixed a bug where selecting log axes would crash RAW if you tried to do so before loading any data.
- Fixed a bug where the legend label for ift and sec items got set when it didn’t need to be.

5.9.23 1.1.0

Release date: 2016-08-22

Overview

The RAW team is happy to announce the release of version 1.1.0. While there are several significant new features, the major milestone that pushed us into version 1.1 is the integration (after almost a year) of the RAW code that has been available on this website and the RAW code improvements made by Soren Skou for use with the SAXSLAB home-source machines. All of RAW is now unified, and we intend to have only one development trunk for the foreseeable future (though we may have temporary branches for major feature development).

We have also added in a solid angle correction for integrating images into scattering profiles. This correction accounts for the change in solid angle of a pixel as you change q. We have tested it against the solid angle correction implemented in pyFAI, and found that the results are identical. This effect will get stronger at higher q, and cause an overall increase in intensity of integrated profiles. On a Pilatus detector, the solid angle correction has a ~0.5% effect on integrated intensity at q=0.25 A^-1 and ~4% effect at q=0.75 A^-1.

Major new features include:

- The solid angle correction mentioned above
- Improved speed when calculating Rg, MW, and I(0) for SEC-SAXS curves (up to a factor of 7 faster in our limited testing)
- Ability to read in multiwire (.mpa) files
- Ability to read in headers from SAXS beamline BL19U2 at the Shanghai Synchrotron Radiation Facility
- Merging, rebinning, and interpolating now all save history information like averaging and subtracting have
- Scattering profile history (either: averaging, subtracting, merging, rebinning, and interpolating, or information about loading in and normalization) can now be viewed within RAW by right clicking and selecting 'Show history'
- RAW is now (mostly) compatible with wxpython 3.0 on Linux

Beyond these changes, there are numerous small improvements, visual tweaks, and bug fixes. You can find a full list of those below.

Simultaneous with this release we are also releasing updated installation guides for all platforms. We are happy to say that we are confidant enough in our ability to produce prebuilt windows installers that we now recommend that windows users install from the .msi files unless they know that they need to compile from source.

As always, we appreciate any feedback (positive, or, especially, negative), bug reports, and suggestions for new features!

All changes:

- Fixed a bug that prevented BIFT from running in uncomplied mode
- AutoRG now runs automatically when the Guinier window opens, assuming there is no previous Guinier analysis
- Fixed a bug where BIFT failing to find a solution caused RAW to crash
- If autosave is active, and a the folder vanishes, autosave now detects that, and is disabled, instead of crashing RAW
- When RAW settings are loaded, all folders and files in the settings (autosave directory, online directory, flatfield file) are checked. If they cannot be found, these settings are disabled, and the user is notified.

- Visual improvements of the BIFT window, DAMMIF window, and some options windows
- Fixed a bug where analysis windows would show up behind the main window, where you could move them by dragging the title bar without losing focus on the analysis windows, and where you could bring them to the front without first clicking on the main window
- Changed the layout in the SEC tab to be more descriptive, and to save space
- Changed welcome dialog info
- Fixed display problems of the Guinier and GNOM windows under wxpython 2.8 on Ubuntu
- Added the ability to start online mode at startup with a predefined directory
- Added the option of automatic saving of BIFT and GNOM results
- Updated save functions in RAW so that files that RAW saves are not automatically loaded back into RAW
- Added in option (on by default) to apply a solid angle correction to the integrated data to account for change in solid angle of the pixels with q
- Fixed several small bugs with the online mode: crashing when the online mode directory ceased to exist, online mode being able to start without selecting an online directory
- All counters and image header parameters now automatically have any spaces in the file name replaced with underscores, so that they do not crash the normalization
- DAMCLUST is now available as an alternative to DAMAVER after running DAMMIF
- Merging, rebinning, and interpolation now add to the file history in the same way that subtracting and averaging have
- Added a new feature to view the file manipulation history or load history within RAW (right click on a scattering profile in the manipulation list and select 'Show history')
- Added a sorting function to the .dat file saving so that file parameters should always appear in the same order in the saved file
- Fixed a bug where a tiff file with the wrong header getting read in as a Pilatus tiff file would cause RAW to hang up
- Added extra error catching to the file header load function
- Sped up calculation of SEC-SAXS Rg, MW, and I0 by adding a threshold function. The threshold checks the ratio of integrated sample intensity (or whatever intensity is being used for the SEC plot) of the average buffer to the average sample files. If the intensity is not above the set threshold (1.02 by default), it does not try to calculate the parameters. This means all of the buffer curves are automatically skipped, and calculation is much faster. It depends on the threshold and the data, but I saw speed increases of up to ~7x. This can be set by the user in the new SEC-SAXS panel in the Advanced Options window.
- Changed how normalization information is saved when a .dat file is saved. Now, normalization information is only saved when it is applied. The absolute scale factor applied is also now saved
- Added more files to the list of files that can be loaded in online mode
- Updated sync function so that files are only marked as modified when something is changed during the sync
- Modified how the centering arrows work to catch faster clicks, and to (mostly) prevent two moves with one click (noticed on a mac)
- Masks with zero area are no longer saved as masks
- Added the ability to load some multiwire detector files (.mpa files)
- Added the ability to read in the header for files from BL19U2 at the Shanghai Synchrotron Radiation Facility

- If the image or beamline header contains a concentration key word, that is now set as the sample concentration in RAW when the image is loaded
- Fixed a problem where ambimeter in the ATSAS 2.7.2 package could not be run
- Fixed numerous small and large visual problems with running RAW on linux with wxpython 3.0. I now believe that RAW can be considered compatible with wxpython 3.0 on all platforms, but there are still occasional sizing issues on Linux that it does not handle perfectly
- Fixed a bug where damaver and damclust would not run if the directory path contained a space

5.9.24 1.0.3

Release date: 2016-07-20

Overview

We're releasing the latest version of RAW, 1.0.3 today. This includes several minor bug fixes. The timing of the release is done so that the version being demoed at the ACA meeting (http://www.amercrystalassn.org/2016-scientific-program#SAXS) will be identical to the latest release.

All changes:

- Fixed a bug where saving a mask without an image loaded would cause an error.
- Fixed a bug where attempting to show a SAXSLAB BS Mask without a SAXSLAB image loaded would cause an error.
- Fixed a bug where autosaving for files (processed image files, averaged files, subtracted files) could be turned on without a valid save directory selected.
- Added a feature so that when an autosave directory is cleared, autosave for that file type is turned off.
- Fixed a bug where the final lines of the damaver output were not being shown in the dammif window.
- Added some extra information to the two most common error messages we get contacted about: inability to load an image type, and inability to load a header file.
- Fixed an error where if an image header contained non-unicode characters, when a scattering profile generated from that image header was saved it would crash RAW. Fixed the same error if the header was shown.
- Removed some unused settings values.
- Removed the brightness bar in the image settings pop up window, as it was currently disabled. This may be re-enabled in the future.
- Set the image settings pop up window to have the default upper value be the max pixel value, rather than 65535.
- Fixed a bug where starting two dammif runs in the same window (running it again after either aborting or letting the current runs finish) did not clear the old log tabs.
- Fixed a bug where entering a wavelength longer than ~115 A resulted in an error. Now a window pops up informing you of the error and you have to re-enter the wavelength value.
- Fixed a bug where the quick reduce dialog was not displaying, and thus quick reduce could not be used.
- Profiles reduced using quick reduce will now have a q range corresponding to the start/end skip points in RAW, consistent with items loaded into RAW and saved from there.
- Fixed a bug where certain .fit files and FoXS .dat files with 4 columns would not plot properly.

- Fixed a bug where the x and y axis values of the Guinier plot were not updating when the data range was changed
- Relabeled the residual plot in the Guinier window with the correct axis labels.
- Updated how GNOM, BIFT, an Guinier plots are refreshed for improved speed and to remove certain display glitches.
- Changed the header display in the image panel to be read only (since changes there were not saved).
- Removed the automation and SANS options panels, as they had no effect. These may be reenabled in the future.
- Changed the default bin size in RAW for q spacing from 2 to 1.
- Removed some extraneous print statements.
- Cleaned up RAWAnalysis.py code and some code in SASFileIO.py
- Added ability to load .fir files.
- Fixed a bug where most of the new image types added in RAW 1.0.2 were not being recognized by RAW.

5.9.25 1.0.2

Release date: 2016-06-22

Overview

We're happy to announce that we're releasing RAW 1.0.2. This is another version focusing on small bug fixes and speed improvements, to try to increase the stability and usability of the software. As always, please report any bugs you find to us, so we can fix them!

The one major change is the inclusion of the fabIO package (https://pypi.python.org/pypi/fabio) for opening images. This has allowed us to support a number of new image types. RAW now supports images in the following formats:

- Pilatus TIff
- CBF
- SAXSLab300
- ADSC Quantum
- Bruker
- Gatan Digital Micrograph
- EDNA-XML
- ESRF EDF
- FReLoN
- Nonius KappCCD
- Fit2D spreadsheet
- FLICAM
- General Eelctric
- Hamamatsu CCD
- HDF5
- ILL SANS D11

- MarCCD 165
- Mar345
- Medoptics
- Numpy 2D Array
- Oxford Diffraction
- Pixi
- Portable aNy Map
- Rigaku SAXS format
- 16 bit TIF
- 32 bit TIF

All changes:

- Removed tifffile warnings upon opening RAW
- Improved the SEC-SAXS online mode based on user feedback to make it easier to work with.
- Fixed an issue where active masks could be removed from memory when saving config files.
- Fixed an issue where no warning was being displayed when config files failed to save properly.
- Improved the speed of selecting large numbers of manipulation, IFT, and SEC items by at least 3 orders of magnitude.
- Updated how loading and plotting works to improve speed by a factor of ~2.5 for both loading and subtracting large numbers of items.
- Updated the Plot Sec button to improve the speed of file loading in certain cases.
- Fixed a bug where FLICAM images could no longer be loaded due to changes in how tiffs are loaded in pillow >=3.0
- Removed some possible issues with loading items where files were not getting closed correctly.
- Fixed an error where rebinning an item under certain conditions could crash RAW.
- Added a warning if a users tries to update or send frames from a hidden SEC curve (assumes that they forgot to change their selection)
- Fixed a big where sending the same frames twice to the main plot from a SEC curve would cause various problems with RAW.
- IFT items are now marked as modified when they are renamed.
- Fixed an error caused by clicking on the top item of the advanced options configuration tree
- Fixed an error in the Image tab where selecting the pan/zoom buttons wouldn't always properly toggle the button in the toolbar.
- Fixed a bug where the popup menu for inverting the mask couldn't show.
- Fixed a bug where panning or zooming when centering would turn off the silver behenate centering rings
- Fixed a bug in OS X where holding down the centering arrows didn't continuously move the beam center position
- Fixed a bug where the centering arrows wouldn't move the beam center in smaller than integer steps (when holding them down).

- Updated the sync function to greatly increase speed when used with lots of items.
- Updated the superimpose function to greatly increase the speed when used with lots of items.
- The file panel now automatically refreshes when you switch to the file tab.
- Added the ability to use the common keyboard shortcut ctrl-A to select all items in the manipulation, IFT, and SEC lists.
- Fixed an issue with the beam center indicator in the masking panel vanishing when it should not.
- Fixed a bug where error bar color was not maintained when moving a line between different plots.
- Fixed a bug where the error bar color selector for the manipulation and IFT line properties displayed the wrong color in the line properties box.
- Added the ability to change the calculated line name in the SEC line properties box.
- Fixed an issue where, if the legend position had been changed, it reset to the default position when the legend was updated.
- Fixed an issue where the legend shadow went away when legend was updated.
- Added ability to load many more image types using the fabIO library.
- Fixed a bug where the wrong legend label would sometimes be used for SEC curves in windows.

5.9.26 1.0.1

Release date: 2016-05-23

Overview

We're very happy to announce that we are releasing RAW 1.0.1. This is a minor release, concentrating on bug fixes and small changes to the user interface.

There is one very exciting piece of news, which is that this release comes with a prebuilt windows installer (.msi file)! This should make it much easier for those on windows to install the program. We're currently working on a similar thing for OS X.

We are also happy to announce that, to the best of our testing, RAW is compatible with wxpython 3.0 on OS X and Windows (Linux is still a work in progress).

All changes:

- Fixed a bug where online mode without an online filter would load files twice.
- Fixed a bug which caused dammif to crash when run in a directory where the path contained a space.
- Masking panel now defaults to the beamstop mask, not the ROI mask.
- Fixed a bug where if OS X preview files became visible on another system, loading them would crash RAW.
- Fixed an intermittent bug where in scientific linux 6 and wxpython 2.8, occasional calls to the File List would crash RAW.
- Added in error catching, so attempting to load bad .cfg files (either corrupted, or non-RAW files with the same extension) doesn't crash RAW.
- Added in automatic verification of saved .cfg files, to check they can be loaded back into RAW.

- Scrolling with the third mouse button in the Image plot panel, but outside of an image, no longer produces errors in the console.
- Moving manipulation items between plots now respects visibility of the manipulation items.
- The plot axes now automatically refresh when the scale or offset of an item is changed if the axes are set to autoscale.
- Tool tips now work in wxpython 3.0 on OS X
- Selecting the “remove” option in a right click context menu in the Manipulation, IFT, or SEC control tabs no longer causes a seg fault in wxpython 3.0 and OS X.
- Removed MM and conc from Guinier panel, to unify GUi so that MW information is only in the MW panel.
- Added ability to change online mode directory without going offline and back online.
- Added a sort to the online mode, so that files should load in order if multiple files are detected in a given online mode load sequence.
- Added a size check to the online mode load, so that if a file fails to load because it hadn’t finished writing/copying, it should load when it is finished.
- Removed the Load button in the SEC control panel .SEC items are now loaded automatically once the file is selected.
- Added an online mode for SEC-SAXS
- Fixed a bug in how SEC-SAXS data was updated when no parameters were being calculated.
- Added a feature so that RAW’s online mode will not load in files that RAW saves in the online directory.
- Fixed a bug occasionally preventing the ATSAS directory from being automatically detected.
- Changing control tabs now automatically clears/loads the info window as appropriate.
- Fixed a bug with running datgnom from inside RAW that caused it to fail in certain circumstances.

5.9.27 1.0.0

Release date: 2016-05-06

Overview

Very exciting news, we’re moving the project out of beta! That doesn’t mean there aren’t still bugs, or that we’re done adding features. But it does mean that we’re happy with the current build (and that we ran out of numbers to increment in beta).

The major new features in this release:

- Added support for running GNOM from RAW
- Added support for running DAMMIF from RAW
- Added support for running DAMAVER from RAW
- Added support for running AMBIMETER from RAW
- Major overhaul of the IFT panel, so it actually works, which involved changing how BIFT is run.

All other changes:

- Added support for reading in FoXS .dat files that have both experimental and model intensities in them
- Fixed a bug where after using the Clear SEC data button RAW could still think there were unsaved changes in the SEC panel
- After removing an item from a plot, the plot axes will automatically resize (unless automatic axes size is turned off in plot options)
- Added a README file in the RAW directory with information on installation and getting help
- Fixed an issue with the porod volume MW calculator crashing if the scattering profile extended to q greater than 0.45 A^-1
- Fixed a bug where MW for RNA was not properly calculated in the SEC plot
- Added ability to save all integrated scattering profiles from a SEC curve as dats
- Fixed an issue where header for save analysis csv files was not using the correct delimiter
- Fixed an issue where beam center did not initially show up correctly in the centering/calibration panel
- Fixed a bug where changing font size for the plot title and axis labels had no effect
- Fixed an issue where the home button in the sec plot didn't work if the calc data existed but was not shown
- Added complied windows 8 exentions, updated compiled windows 7 extensions.
- Various other small bug fixes.

5.9.28 1.0.0b

Release date: 2016-03-24

Overview

We are proud to announce that RAW version 1.0.0b has been released for download! This version includes a huge number of new features and bug fixes.

Our favorite new features are:

- Easy processing of in-line SEC-SAXS data
- New molecular weight panel for calculating mol. weight from the volume of correlation, adjusted porod volume, and absolute scaling.
- AutoRG now available.
- Uncompiled running, which allows RAW to run as long as the appropriate python packages are installed, even if the extension files cannot be compiled.
- Files saved as .dats now automatically save all analysis information in the header, and reload it into RAW when loaded again.

We have also made significant improvements to speed and responsiveness:

- Sped up loading and plotting for large numbers of files on a test machine by a factor of ~30
- Sped up subtraction of large numbers of scattering profiles by a factor of ~4
- Improved responsiveness when large numbers of scattering profiles are plotted.

Also, there are new, up-to-date install guides available for Windows, Mac, and Linux. Check them out in the files tab.

Finally, we have cleaned up both the code repository and the files area.

If you have questions, or feedback, please contact us!

All changes:

SEC-SAXS data processing:

Added capability to process SEC-SAXS data. This included adding a new SEC tab in the control panel, a new SEC plot, and a new SECM data class.

SEC-SAXS data is collected by continuous framing of the detector while sample is being pumped through a column. The output of that column is connected to the SAXS cell. The new RAW addition allows users to load all of the detector images collected during column elution into a new data type, the SECM. The overall frame intensity is plotted vs. frame number or time, and this should look very similar to the UV-chromatograph that an FPLC produces. The users can then select a range of frames from this curve, and send them to the main plot for processing as normal.

Additionally, the users can select a specified buffer range, and an average window size. The window is then slid across the curve, and the scattering profiles within the window are averaged. The averaged buffer is subtracted from the curves in the window, and radius of gyration, molecular weight, and I(0) are automatically calculated. These are then plotted on the same plot as the 'SAXS chromatograph' (intensity vs. frame #), allowing users to quickly get a feel for what is in each peak they measured.

Major code additions:

- There is now a SEC Panel, SEC Item panel, and SEC Control panel class, based on the Manipulation panel and Item Panel in RAW.py.
- There is now a new plot class in RAWPlot.py, the SECPlot, which allows for multiple axes on the same plot, and handles the various plotting options.
- There is a SECM class in SASM.py, which is the data structure for this new thing.
- There is a new SASCalc.py file, which contains the autorg and automw functions. The autorg is pure python, based on the ATSAS package autorg function. It could probably use some tuning of the various parameters. The automw is also purely python, and based on the Rambo & Tainer correlation volume method for determining molecular weight.
- There is a new save/load format, extension .sec, for saving SEC objects.
- The SEC data is saved when the workspace is saved.
- Various bits and pieces everywhere have been adjusted to accommodate these new panels.

Online mode filtering:

- Added an online mode section in the advanced option panel. This allows you to turn on online filtering, and give a set of strings that allow you to ignore certain files when they enter the watched folder. You can either set a list of strings in the file name to ignore, or a list of strings that must be in the file name, or some combination. You can also set the location where these strings must occur: at the start, end, or anywhere.

MW Panel:

- Added a new analysis panel for finding MW. It has methods for MW by I(0) ratio (also in Guinier plot), MW by the Rambo & Tainer method of the volume of correlation, MW by the Porod volume (corrected by the method of Fisher), and MW by absolute intensity.
- Users can modify default calculation values for the MW in the advanced options MW panel.

Speed improvements:

- Changed the loadAndPlot function so that it only updates the curves on the plot every 20 curves loaded (and at the end), and only updates the legend after all the curves are loaded. On my machine, for ~400 data files (pilatus 100K tiffs) this sped up loading and plotting by ~30x (~40 s vs. 20 minutes & 15 s).
- Changed the subtractItems function so that it only updates the curves on the plot every 20 curves loaded (and at the end), and only updates the legend after all the curves are loaded, as with the _loadAndPlot function. On my machine, this sped up subtraction by ~4x (1 min 7 s vs. 4 min 5 s for ~400 manipulation items).
- Updated online mode to take advantage of the faster plotting, by passing all of the files to be plotted to 'loadAndPlot' at once, rather than one at a time (will only matter if files are coming in faster than the online mode update timer)
- Changed the legend to be off by default (since it significantly hinders performance). Changed the update legend and the legend plot options dialog functions so that this all still works. This seemed to improve load in performance for ~400 data files by ~15% (35 s compared to 40 s).

Uncompiled running:

- Removed all attempts to compile unused extensions.
- Added in try/except cycles for importing and compiling compiled extensions.
- Rewrote compiled extensions scipy.weave code (essentially c code) as pure python.
- Set it so that if RAW is unable to compile extensions, it displays a warning message to users on startup, and then runs with the pure python versions.
- Compilation is particularly an issue on windows, so hopefully this will make deploying to windows much easier (even though the program will run slower). Particularly for versions where a windows installer is not available.
- This required the inclusion of a RAWGlobals file, which contains a variable that notes whether or not the compiled extensions were successfully imported.

Minor changes:

- Switched from PIL to pillow. PIL is not longer under active development, pillow is a fork of PIL that is still supported. Also, pillow is included in the default enthought python installation, while PIL no longer is.
- Fixed an issue where integrated scattering profiles could end up with different numbers of points. This was simply disabling the zero trim command in the integration routine.
- Added in an option to skip points at the end of a scattering profile (identical to the skip at the beginning). This was needed after the removal of the zero trim command when you have entire range of high q masked out (such as to eliminate shadowing from the beam pipe). This setting is accessible in the advanced options calibration dialog.
- Added in a parse function and header profile for log files from the BioCAT beamline.
- Removed the requirement that the beam position be an integer.
- Added in the ability to add a 'zero line' to the main plots (a horizontal line at y=0), in the plot options dialog.
- Fixed the plot options dialog so that it can be opened when no items are loaded in the plot.
- Fixed how the plot options dialog handles legend settings, so it doesn't break if there are no curves already plotted.
- Fixed plot options so that setting x limits and y limits when auto limits is not checked actually affects the graph. Also fixed the limits so that they properly acquire the current axis limits when plot options is opened.
- Made it so that turning auto fitting axes back on forces the plot to autofit the axes when the plot option dialog is closed with the okay button.
- Fixed a bug where the legend would turn on when an item was hid/shown in the manipulation panel, even if the legend was previously turned off.

- Fixed a bug where error bars didn't turn off when an item was hidden in the manipulation panel with error bars turned on.
- Made it so that the borders check boxes in the plot options window actually cause the borders (and tick marks) to turn on and off in the plot.
- Changed the Guinier plot panel so that it automatically updates the MW when the concentration is entered (instead of needing one of the up/down arrows to be hit in the spin control)
- Fixed a bug where the MW of a SASM object wasn't updated when the SASM object was set as a MW standard.
- Fixed a bug in the menu creation of the file browser pane where the right click menu wouldn't open on a mac (wxpython >=2.9.2.4)
- Fixed a bug where the concentration of a sasm object was getting improperly set when the clearinfo function in the information panel was called.
- Made the info panel Rg, MW, and I(0) boxes read only, since user modified values in those boxes aren't saved
- Made the info panel conc box update whenever it gets text, so that if you update the concentration and click on another sasm it still saves the concentration.
- Fixed the options window not opening at the right size.
- Switched to using json to save/load sasm parameter dictionary contents in .dat files. This allows easy saving and loading of dictionaries in human readable format. So now all parameters (header, counters, analysis, etc) are loaded. NOTE: THIS IS NOT STRICTLY BACKWARD COMPATIBLE. RAW can still load old .dat files (and primus .dat files), but it cannot load analysis information out of the old files. This doesn't really affect anything, as for the old files the analysis information didn't load anyways.
- Modified how saving of averaged files history is done. Added in saving of subtracted files history. Now all of the averaging and subtracting manipulation history of a file is saved in the history entry of the parameters dictionary. This works even when you average or subtract files that are already averaged or subtracted. It is mostly human readable in the saved .dat file (though as you get more layers deep in averaging or subtracting it gets hard to tell what is what).
- Fixed a bug where the correct qmin and qmax weren't loading in the Guinier window when a previous Guinier analysis had been done.
- Changed it so that when guinier or mol. weight analysis is done, if the results are different from previous analysis, the scattering profile is marked as modified to denote that the results are not saved.
- Fixed a bug where plot axes didn't auto resize when curves were moved from the top main plot to the bottom main plot and vice versa.
- Fixed a bug where selecting 'Help!' in the help menu crashed RAW. No in-program help is yet available, but a message dialog now tells the user to look for help on the raw project homepage.
- Set 'okay' button to be selected by default in the welcome window.
- Fixed a bug where on mac, last saved settings wouldn't load from the dialog on startup (this may have also been affecting other OSes).
- Enabled normalization by ROI counter using an 'ROI counter mask' (formerly called a transparent beamstop mask).
- Fixed a bug where minor tick marks weren't turning off for log axes that weren't shown (such as top and right) (I believe this was introduced by an updated version of matplotlib, I don't remember seeing it before).
- Fixed the logarithmic image scale display in the image panel. It works now, and is enabled.
- Disabled nexus support to remove error on starting raw (can be easily re-enabled, it is simply commented out in a couple places in SASFileIO).

- Updated the manipulation and IFT item saves so that it offers the choice to rename the file when saving a single file, and so that there are more explicit instructions when saving multiple files.
- Fixed a bug in the rebin function, where it wasn't setting the qrange according to the original sasm.
- Fixed a bug where comparison of q vectors to test for subtraction was done by length rather than elementwise by q.
- When scattering profiles with different q vectors are subtracted, choosing to force the subtraction now actually carries out the subtraction (with appropriate matching/rebinning of the q vectors).
- Fixed a bug so that the average function now tests the q vectors point wise, rather than by length, to make sure they actually match.
- Added a feature to export all analysis information from sasm objects as an alternative to selecting which analysis features you want to save.
- Update the old save analysis feature to be called 'save item info' in the menu, since it can save things that aren't analysis. Updated the layout of that window a little bit, and added ability to save the new MW analysis info into the item.
- Added scattering profile manipulation into the tools menu: average, subtract, merge, rebin, interpolate, normalize by concentration, change q scale, set as MW standard.
- Upon quitting, RAW now checks whether there are unsaved changes to manipulation or SEC items, and asks for confirmation of quitting if there are.
- Added show image, show data, show header options to the view menu.
- File list maintains sort order upon refresh.
- Doing a Guinier fit on a scattering profile that is all zeros no long crashes RAW.
- Subtraction can handle mismatched q vectors.
- Autosave for averaged and subtracted files now available.
- Features supporting SAXSLab300 image format now available.

Python Module Index

Index