SAXS_TOOLBOX Documentation

SAXS_TOOLBOX is a software suite developped in Python for the reduction and processing of SAXS 2D images. Data reduction is directly performed on 2D images, including transmission correction, intensity and detector calibration. It is an alternative to Foxtrot and allows the conversion of detector images to 2D ascii format in the form of Qx, Qy, I for further use in other programs such as Sasview.

Since the code is not compiled yet, a python distribution (e.g. anaconda) must be installed on your computer. Please visit anaconda.org to download the latest version of Python.

I. Installation and start-up

To install SAXS_TOOLBOX , simply unzip the file in the directory of your choice.

IMPORTANT WARNING : some libraries used by SAXS_TOOLBOX do not accept the presence of 'space' character in the path of the directory where your data is saved. Make sure that the path do not contain space, or you may not have acces to some tools (e.g. pyFAI-drawmask)

I.1 Dependencies

SAXS_TOOLBOX has the following dependencies (python libraries) :

- fabio, pyFAI, numpy, matplotlib, scipy, sasview, sasmodels, sasdata

Some of those libraries might already be installed, particularly if you have installed python using an anaconda distribution. However, sasview libraries, fabio and pyFAI are not so common and should be installed manually.

To install those libraries on your python distribution, simply open a (anaconda) prompt and type

>pip install « library_name »

Once those libraries are installed, the program is ready to work.

I.2 Start-up

The procedure to start the program is the following :

- open a prompt (terminal on linux, anaconda prompt or shell in windows)

- browse to the directory where SAXS_TOOLBOX was unzipped using

> cd your/path/to/SAXS_TOOLBOX

- call python to execute the script

> python SAXS_TOOLBOX.py

The following window should appear :

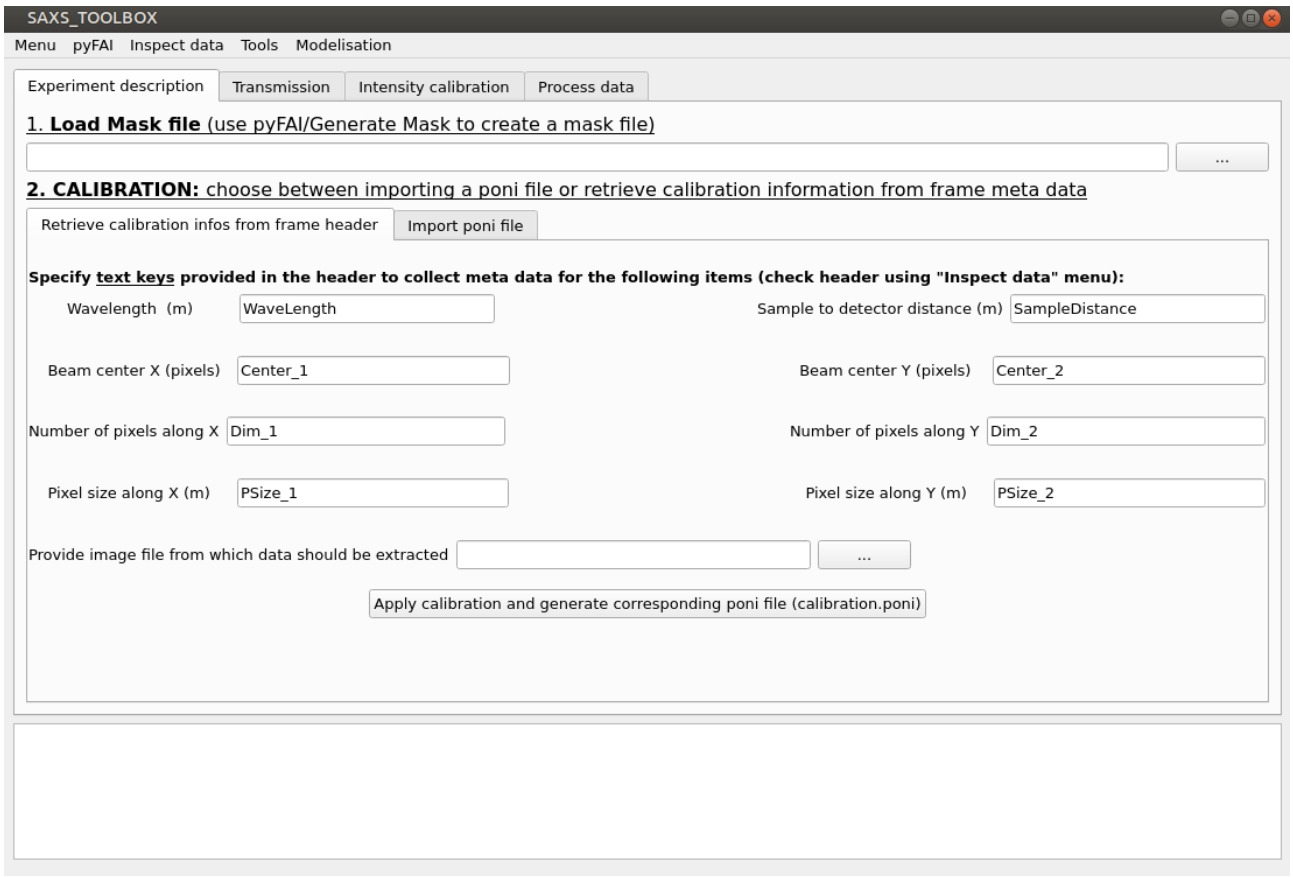


Figure 1 : SAXS_toolbox window

II.Drop-down menus description

II.1 Menu « Menu »

The SAXS_Toolbox « **Menu » (top left of the window)** gives you access to useful functions, such as :

- **Set Working Directory** : this function allows you to sselect the default directory that will be accessed by SAXS_Toolbox

- **Intialize form** : this function allows you to restart data reduction from scratch. Note that calibration factor can be input manually in the corresponding field.

II.2 Menu « pyFAI »

- **Detector calibration** : this function opens pyFAI-calib2 GUI for detector calibration (creation of a poni file, necessary to use the pyFAI suite). For more details, please visit pyFAI-calib2 documentation at the following web adress: https://pyfai.readthedocs.io/en/v2023.1/man/pyFAI-calib2.html?highlight=pyfai-calib2#.

Note that when your SAXS images contain metadata with detector calibration information (frame size, pixel size, sample to detector distance, wavelength, beam center coordinates), this option may not be called and a poni file can be generated from the metadata. To do so, simply indicate int he « Experiment Decription » tab the keywords from your metadata to access the calibration information. When completed, click on the button « Apply calibration and generate poni file ».

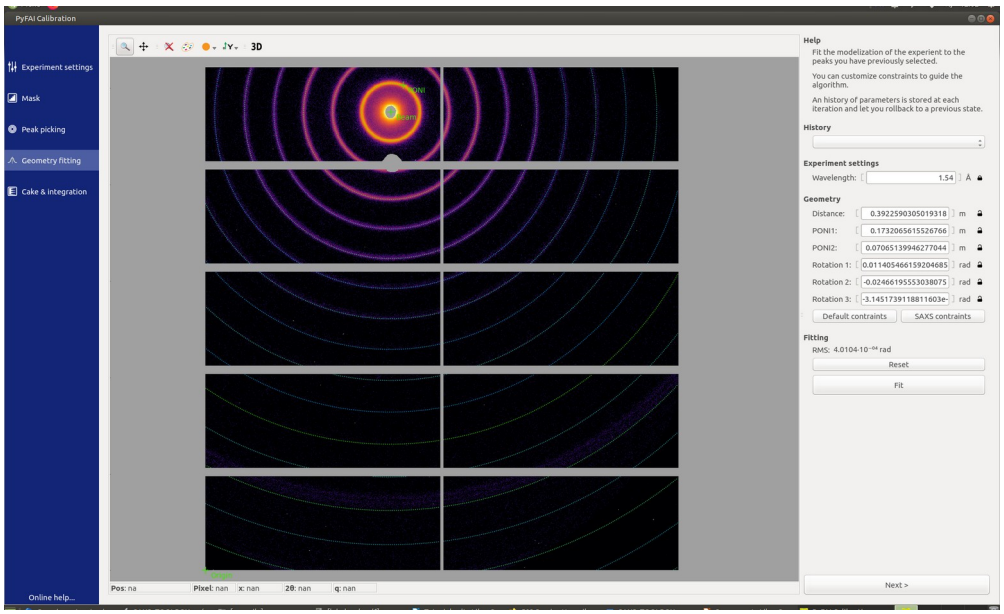


Figure 2 : Detector calibration using pyFAI-calib2

- **Frame integration :** this function opens pyFAI-integrate window for 2D frame integration to 1D file. To perform the integration, the user must provide a .poni file, that can be obtained using detector calibration. For more details on integration, please visit https://pyfai.readthedocs.io/en/v2023.1/man/pyFAI-integrate.html?highlight=pyfai-integrate#integration-tool-pyfai-integrate.

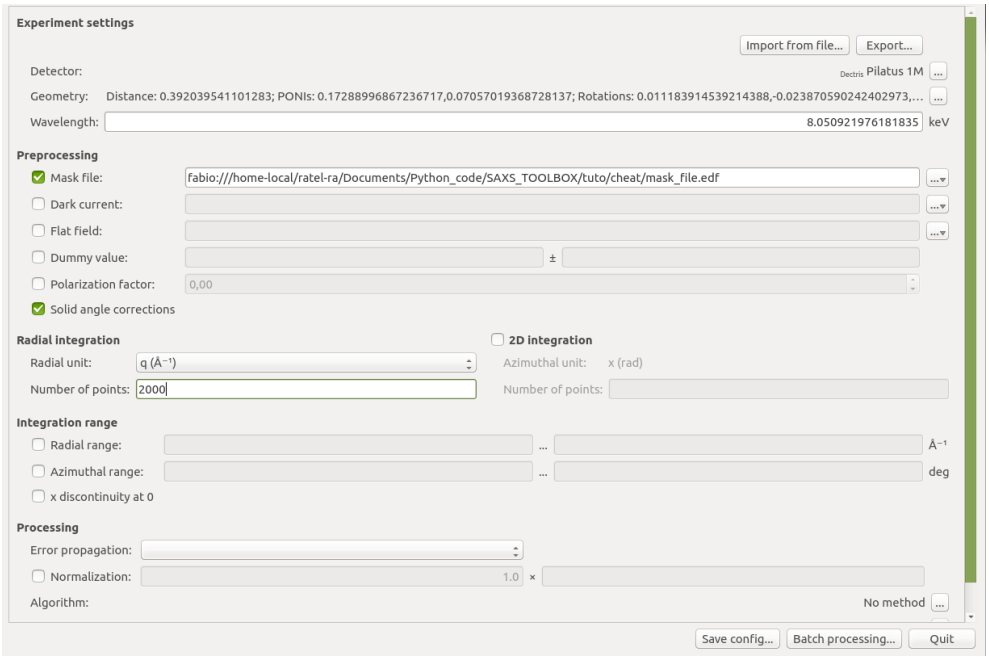


Figure 3 : Frame integration using pyFAI-integrate

- Generate a mask : 2 options are available

- using an image this function first ask you to open a frame file to be opened in pyFAI-drawmask tool. Although great care should be taken for this step, the generation of the mask can be relatively quickly done with the following steps :

- mask below 0 intensity (automatically masks the dead zones of the detector + some of the dead pixels – but not all of them)

- create a circular mask around the beamstop

- use pencil tool to mask additional aeras (remaining bad pixels, shadows,…)

- save your mask in edf format.

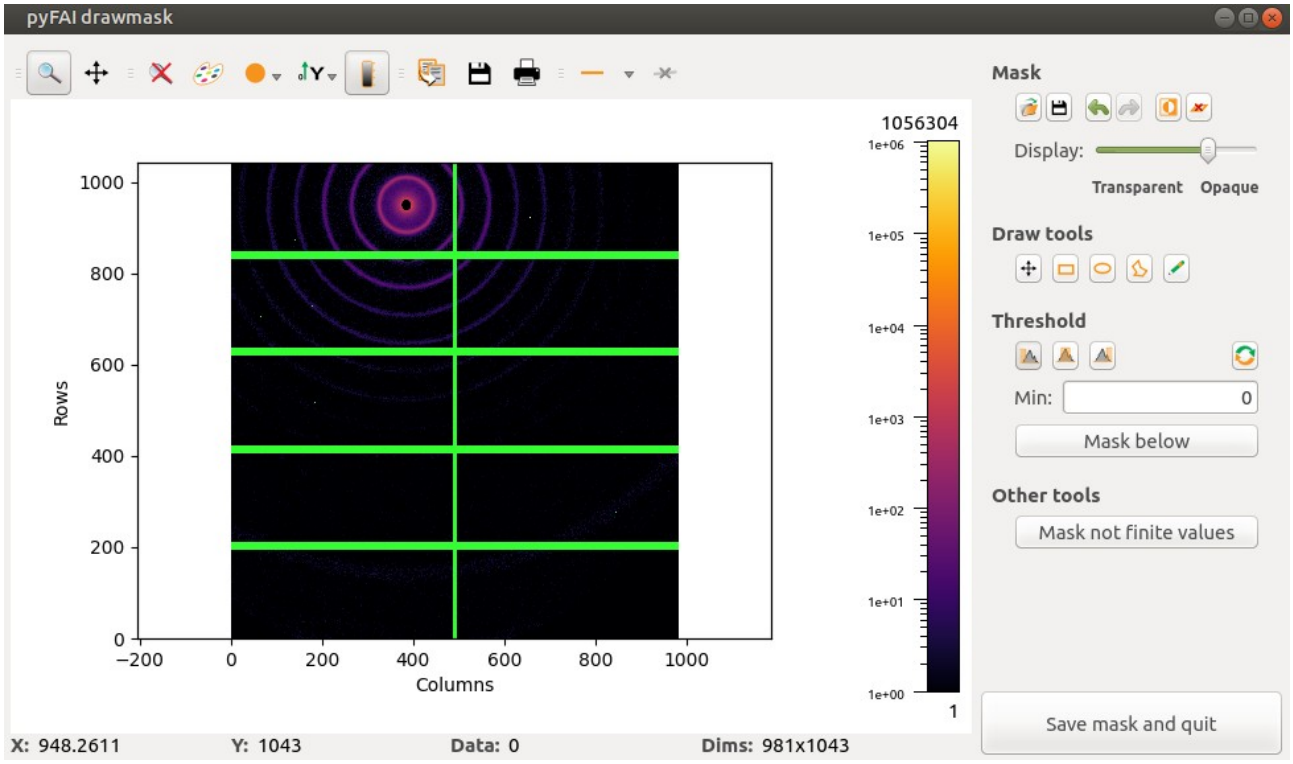


Figure 4 : View of pyFAI-drawmask window

- **convert foxtrot mask** file *.txt to *.edf format for further compatibilty in SAXS_toolbox suite

II.3 Menu « Inspect data »

This menu gives you access to tools that can be used for 2D/1D data inspection.

- **Open and view image** : this function gives you access to fabio_viewer, a Python library to view images of different formats. Please visit fabio documentation (http://www.silx.org/doc/fabio/dev/getting_started.html#)for more details.

- **plot 1D data file(s) :** This function allows the inspection of several 1D data file(s). All files should be located in the same directory. The user can choose between loglog plots (useful for SAXS data) or linear plots.

II.4 Menu « Tools »

- **Average frames :** this tool calls pyFAI-average to calculate the mean frame of several frames provided in input.

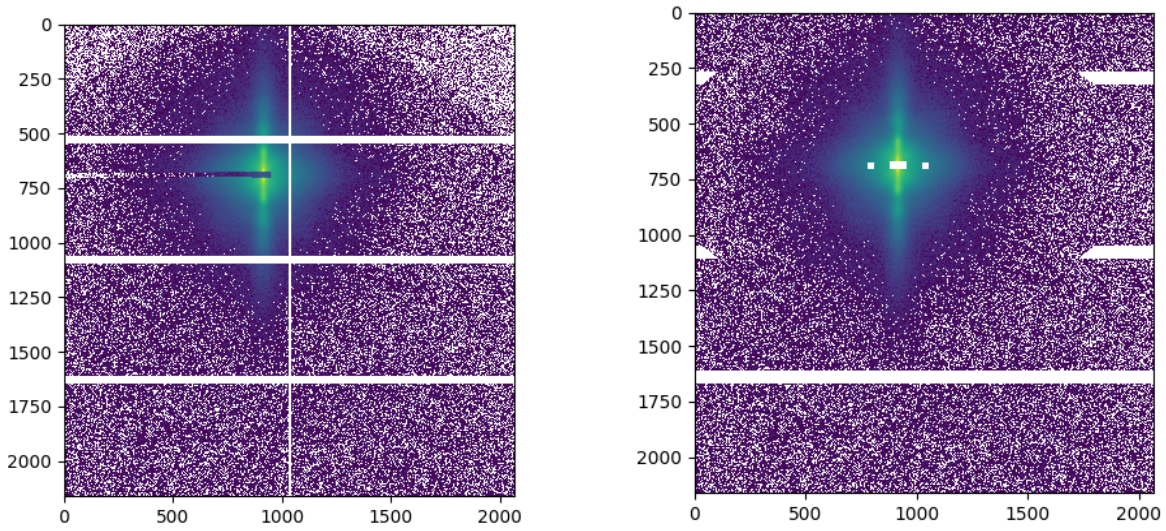
- **Convert .nxs/h5 :** Because fabio library does not support .nxs or .hdf5 format, conversion of .nxs file to .edf format may be required. This function allows the extraction of individual frames

contained in .nxs files, together with required meta data (if available). An average frame (with suffix _ave in the filename) is alos generated.

In the rare cases when meta data do not contain detector calibration information, the detector calibration can be provided using pyFAI-calib2 tool (see pyFAI menu), via the .poni file generated by pyFAI-calib2. In that case, a Warning will appear in the Console Window.

- **Batch radial_profile_plot :** by activating this command, the user is prompted to provide 2d data files (e.g. corrected frames). This function will **compute and plot** the radial profile of the supplied images (applying the mask if specified on the 'Experiment description' tab), and save them in *_radial_profile.dat files.

-Cave frames : This tool can be used to suppress the masked zones of the detector (e.g. deadzones of the detector, bad pixels, beamstop,…). In practice, the user must provide mask file and 2D frames to be caved. The algorithm simply replaces masked pixels by symetrically equivalent ones. The symetries considered are inversion center together with horizontal and vertical mirrors with respect to beam center.



a) before caving b) after caving

Figure 5 : Influence of caving on saxs frames

II.5 Menu « Modelisation »

- **Compute I(q) from pdb file:** this tool calls sasview to compute I(q) curve generated by a structure provided by the user in the form of a pdb file. Default q range is 0.001-2 $Å^{-1}$. The corresponding I(q) data file « *_sasview.iq » is also stored in the directory of the pdb file.

III. Principles behind SAXS_toolbox

SAXS toolbox must be seen as a series of tools made avalaible for data reduction and processing of 2D images from SAXS or diffraction experiments. Depending on the type of experiments, several calibration and correction steps must be performed. In the following, we will consider the most

complex case of SAXS data, for which data must be calibrated and a reference signal may be subtracted. Note that whatever the type of data to be analysed, a mask file must be provided and the detector calibration must be performed.

III.1 Data calibration

In SAXS toolbox, the following calibrations are available :

- **detector calibration** : beam center definition, distance from sample to detector, solid angle correction. Calibration information are stored in a *.poni file, as done in pyFAI tools. This calibration is performed in the « Experiment description » tab, following 2 different protocols, depending on your data:

Case 1. the meta data of your saxs images contain calibration information. In that case, the calibration can be performed directly by specifying the keywords used in the meta data of your images in the « Experiment description/Retrieve calibration inf from header » tabs. Then simply provide an image file for reading the data, check the keywords of the metadata, and click on the « Apply Calibration and generate poni file » button.

Case 2. The meta data are inexistent or don't contain calibration information. In such case, the mesurement of a sample of well known crystalline structure (Silver behenate for SAXS experiments, and LaB6, corundum, etc. for diffraction experiments) must be provided. The first step is to generate the poni file using pyFAI-calib2, accessible through pyFAI/Detecto Calibration menu. The second step is to load the poni file in the « Experiment Description/Import poni file » tab. Informations such as image dimension and pixel size must be provided by the user.

- **intensity calibration** : scattered intensity is calibrated in absolute value through the measurement of an incoherent scatterer of known scattering cross section I_{th} (water, hexane, glassy carbon…). This step requires the measurement of the empty container (i.e. the cell without the calibrant), and a measure of the calibrant inside the container. The calibration coefficient CF is calculated using the following equation,

$$CF = \frac{I_{th}}{\left(\frac{T_0}{T_{cal}} . I_{cal}\right) - \left(\frac{T_0}{T_{EC}} . I_{EC}\right)} \quad \text{equ.1}$$

where T_0 , T_{cal} and T_{EC} are the respective transmitted intensities of the empty beam, calibrant and empty container. I_{cal} and I_{EC} are the respective scattered intensities of the calibrant and empty container. With modern detectors, intensities contained in the data file are directly corrected for transmission. Equation (1) is therefore simplified by replacing all transmission values by 1.

III.2 Data correction

Transmission correction

In the case of SAXS experiments, experimental data must be corrected by their transmission. For this reason, each intensity (i.e. each pixel value) is mutliplied by the coefficient T_0/T, where T_0 is the empty beam transmission, and T is the transmission of the sample of interest (can be a reference measurement).

When a reference signal I_{ref} (e.g solvant contribution) must be substracted from the sample signal I_{sample}, the corrected intensity is given by the following equation :

$$I_{corr} = CF * \left(\frac{T_0}{T_{sample}} . I_{sample} - \frac{T_0}{T_{sample}} . I_{ref} \right) \quad \text{equ. 2}$$

In the case of diffraction experiments, where transmission corrections are optional, all the transmission values should therefore be set to 1 (i.e. their default values). This can also be done by clicking the « Set transmissions to 1 » button in the 'Process data' tab.

Reference subtraction

When a reference signal I_{ref} (e.g solvant contribution) must be substracted from the sample signal I_{sample}, the corrected intensity is given by the following equation :

$$I_{corr} = CF * \left(\frac{T_0}{T_{sample}} I_{sample} - \frac{k * T_0}{T_{ref}} * I_{ref} \right) \quad \text{equ. 3}$$

where CF is the intensity calibration factor, and k is a coefficient that can be automatically computed by checking the « Refine reference substraction » checkbox ob the « Process data » tab. The algorithm performs to following operations to determine k :

-each data (sample and reference) are smoothed and transmission corrected.

- k is the value that brings the reference signal to the level of the sample signal. The algorithm uses the first diffraction peak of the solvent as a reference to determine k value. It will therefore look for a maximum value in intensity, above a given q value (0.9$Å^{-1}$ by default, but can be modified by the user).

Figure 6 shows the strong influence that can have this optimization step. It can be explained by differences in capillary diameter induced by the low reproducibility of the manufacturing process of the glass capillaries.

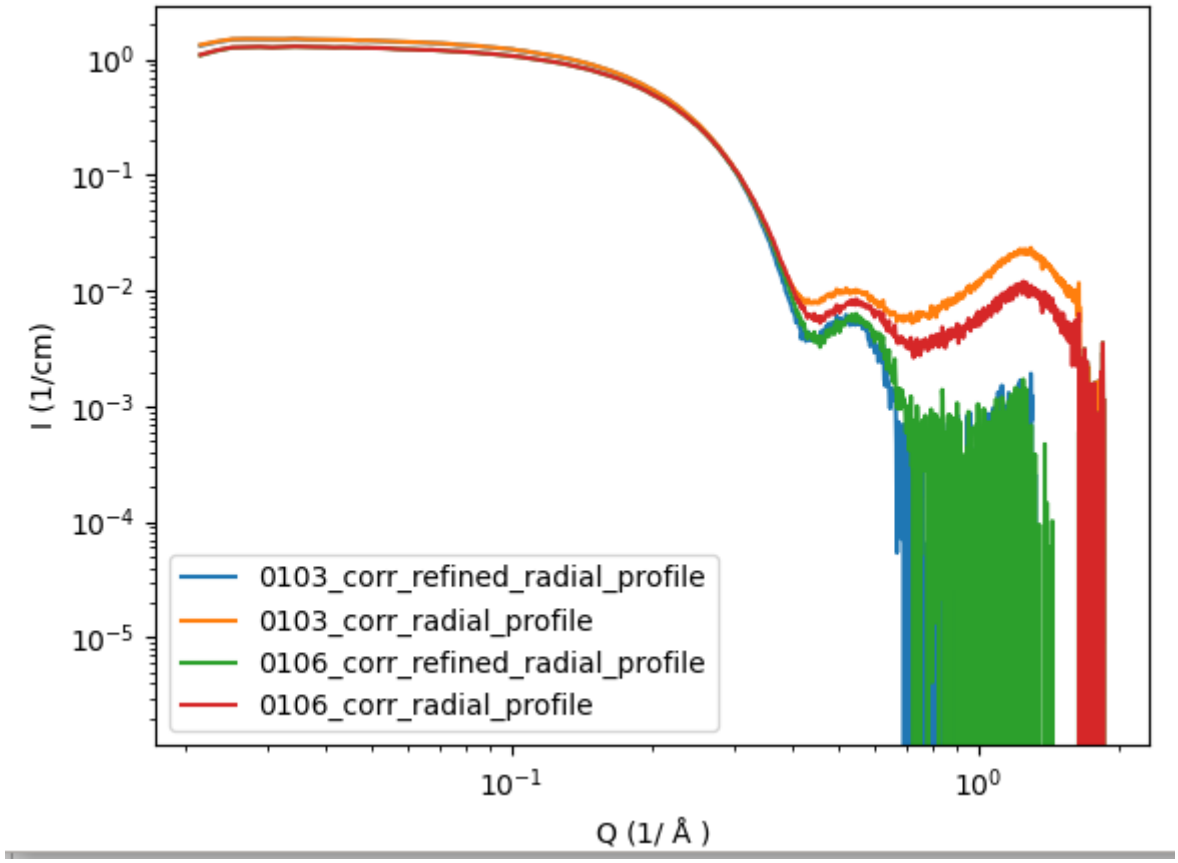


Figure 6 : influence of reference subtraction optimization

IV Description of the graphical interface

IV.1 Experiment description tab

This tab gives you access to two tasks :

1. **assign mask file (mandatory) :**

Simply browse to your mask file. If you don’t have any, iti is possible to generate a mask file using pyFAI-drawmask, which can be accessed via the menu pyFAI/Generate Mask. From there it is possible to either generate the mask file using an image, or through the conversion of a foxtrot mask. The mask file must be in *.edf format (masked pixels are assigned 0 value, as done in pyFAI).

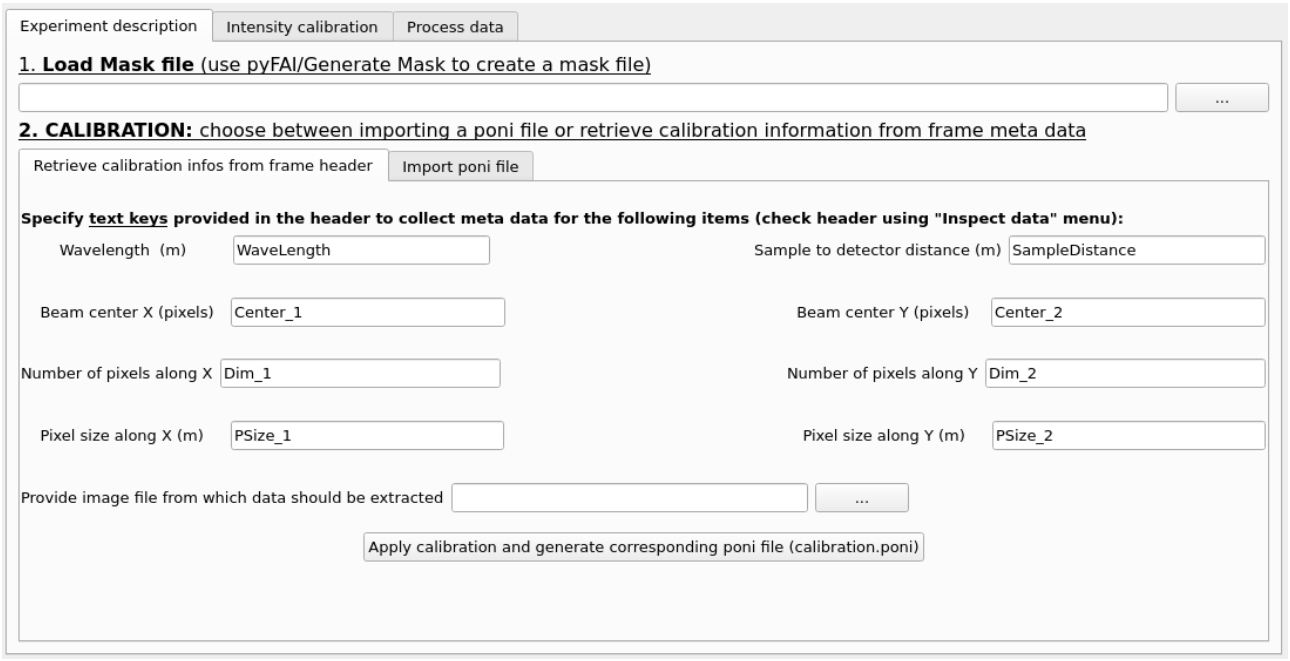


Figure 7 : View of the Experiment description tab

2. **Detector calibration** : As discussed earlier, there are 2 options to perform the detector calibration

2.1 retrieve calibration information from image meta data.In that case, the calibration can be performed directly by specifying the keywords used in the meta data of your images in the « Experiment description/Retreive calibration inf from header » tabs. Then simply provide an image file for reading the data, and click on the « Apply Calibration and generate poni file » button.

2.2 assign *.poni file when meta data do not contain calibration information (distance, beam_center, …). Informations such as image dimension and pixel size must be provided by the user.

IV.2 Transmission tab

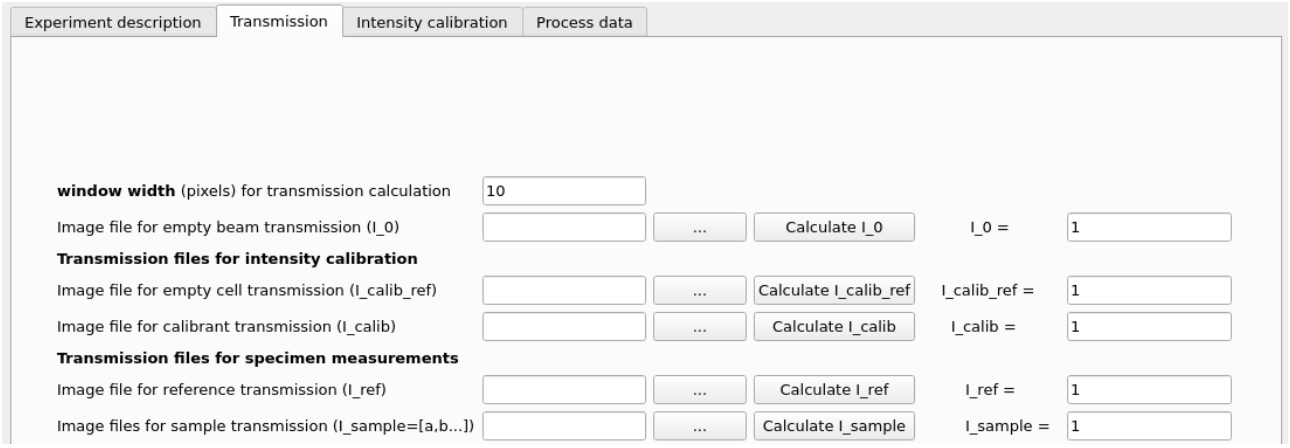


Figure 8 : View of the transmission tab

The « Transmission » tab allows you to attribute your transmission data files to measurement type (reference, calibration, …). Simply indicate the path to your 2D data file and click the associated « calculate transmission » button. Except for specimen transmissions, only one file can be provided by type of measurement (it is not possible to provide 2 files for empty beam transmission).

Sample transmissions fields (last line in the batch) support multiple file input. For this reason, transmission values calculated for sample transmissions are stored in an array, whose i^{th} element corresponds to the transmission of the i^{th} file in the data file list.

In practice, the calculated transmission is computed as the sum of the pixels intensity included in a square around the beam center. The size of the square (in pixels) can be adjusted through the window width field located at the top of the transmission tab. Note that the size of the square is 2*window_width, as shown in Figure 9.

Default transmission values have been set to 1, except those of specimen data. This can be done however after the selection of the input data files (scattering data, not transmission data), by clicking on the button « Set sample transmissions to 1 » in the specimen data tab.

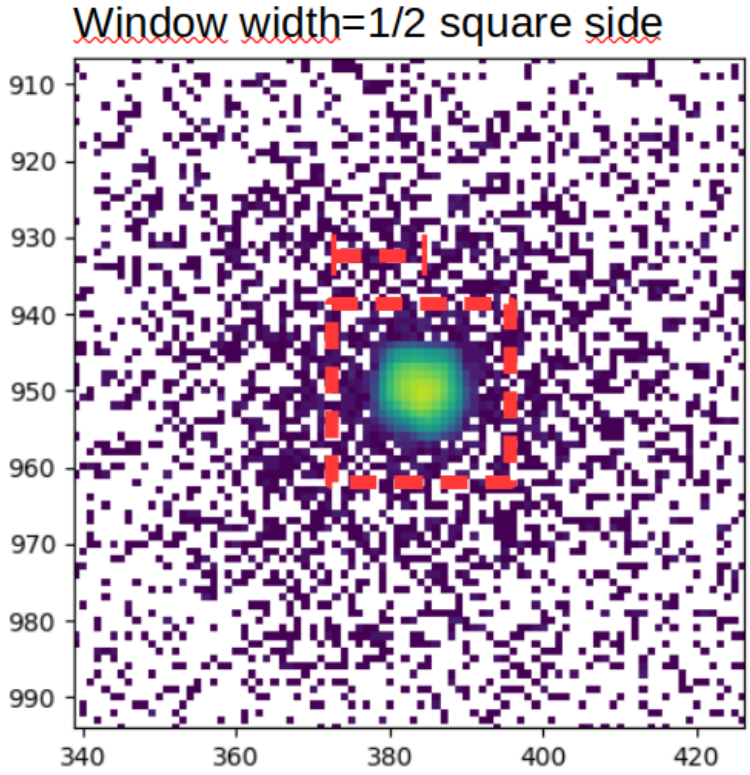


Figure 8 : Sample transmission calculation

IV.3 Intensity calibration tab

The intensity calibration tab is used to calculate the calibration coefficient, using the scattering of an incoherent scatterer (water, hexane, glassy carbon,…).

SAXS_TOOLBOX supports water and hexane as calibrant, with respective scattering length densities of 1.6E-2 and 2.8E-2 cm^{-1}. Checkboxes are availaible in the GUI for the user to specify which calibrant is in use. In the case when another calibrant, is used, the user is invited to provide the expected theoretical scattering length density ρ_{th} in the appropriate field.

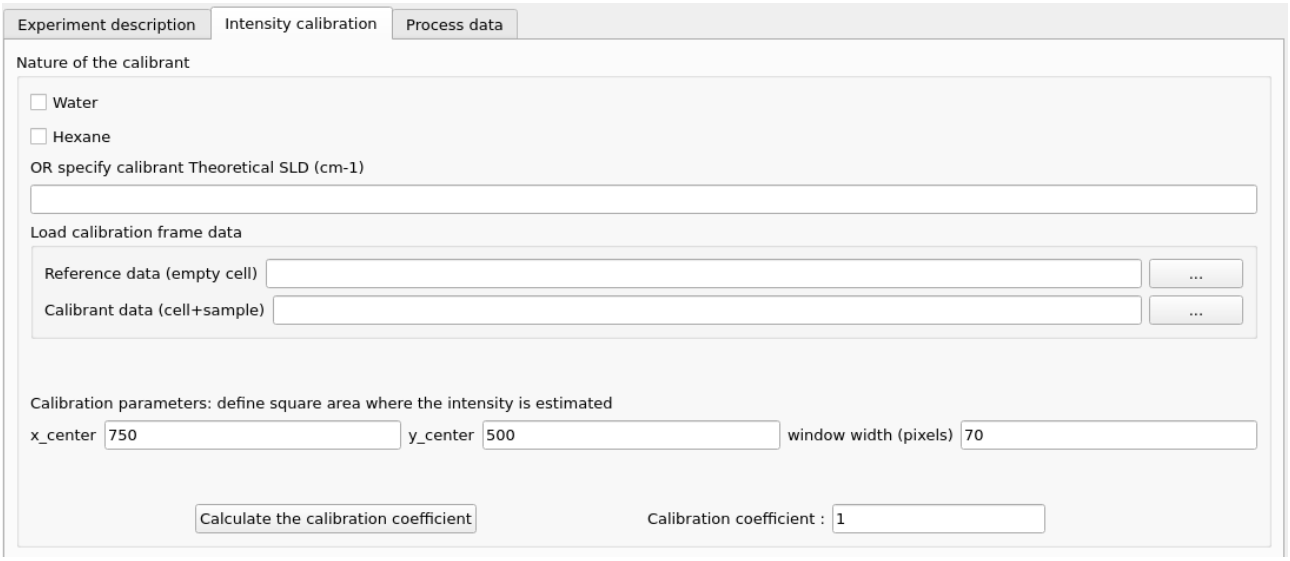


Figure 10 : Intensity calibration tab

The calculation of intensities for calibration is performed by averaging the pixels intensity in a zone of the image defined by the user. This zone has a default square shape. The user can define the position of the square center using x_center and y_center fields in calibration parameters (at the bottom of the tab), and the width of the square with the window width parameter (quantities expressed in pixels). A schematic view is provided in Figure 11.

Calibration is performed as follows :

1. provide the experimental data files corresponding to calibration measurements,

2. define the calibration parameters (nature of calibrant, region of the image to analyze)

3. press the « Calculate the calibration coefficient » button

The average intensities and their standard deviation are calculated for each image. From those values, the calibration coefficient CF is computed using equation (1).

Note that the calibration coefficient can also be input manually by the user in the corresponding field. This feature is particularly useful when treating successive data files sharing a similar experimental setup.

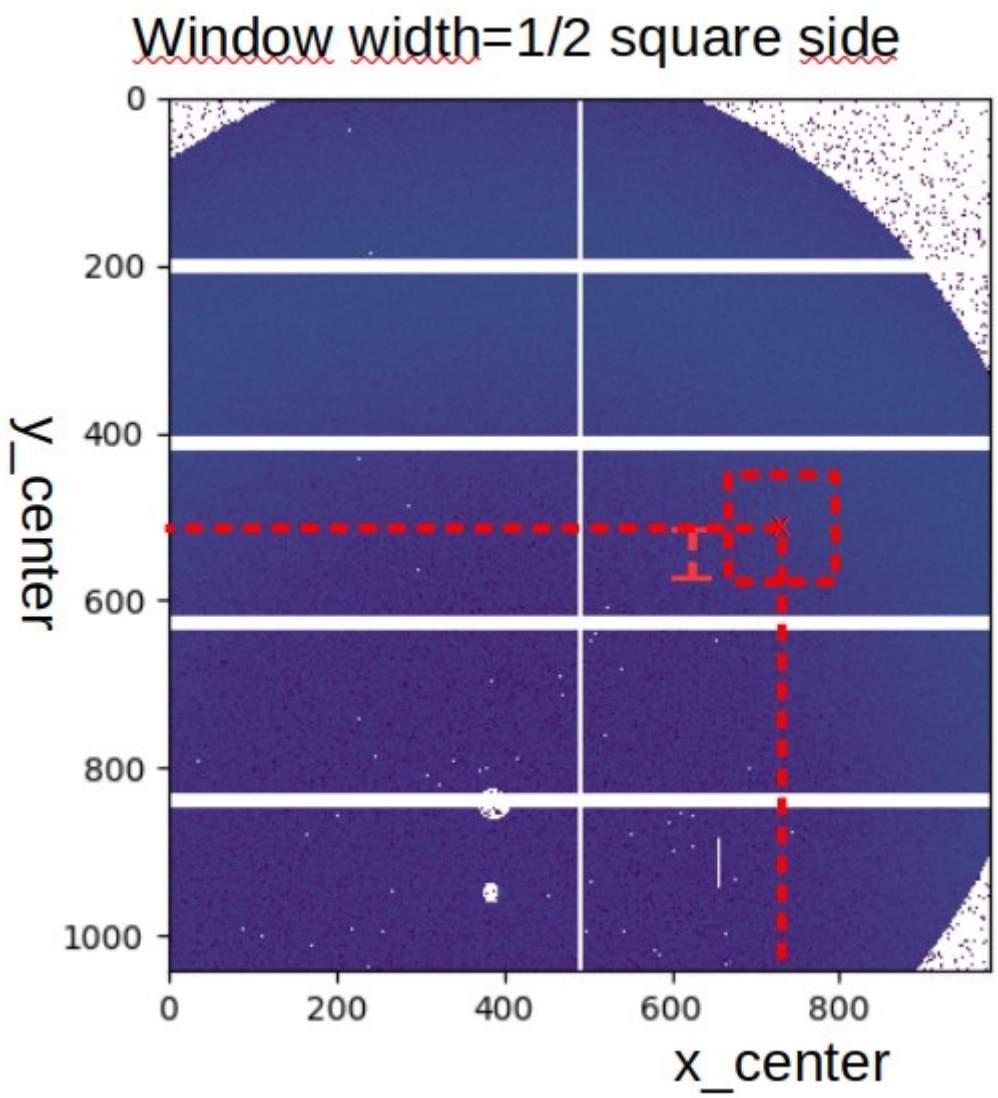


Figure 11 : Schematic view of intensity calculation

IV.5 Process data tab

The Process data tab can be divided in 3 sections.

The top section concerns the reference data, where the user can specify which file should be use as reference scattering data. If no reference is to be substracted, an option is available to skip reference substraction (check the box).

Another checkbox is available to refine reference subtraction. This option allows to properly subtract the reference from each sample data frame provided by the user. The optimization procedure for reference subtraction is the following : the algorithms accounts for the scattered signal in the q-range above 0,9$Å^{-1}$ (first diffraction peak of the solvent), and calculates the coefficient that brings the maximum value of the reference signal to that of the sample (data smoothing is involved in the process, to reduce the impact of noise).

In the middle section, the user can specify which files to be used as specimen scattering data. An « average » option is avalaible to average data files in sequence of n files. Let’s consider the case when 10 samples are measured, with 5 frames per specimen. In that case, data files can be averaged

in sequence of 5 files, thus producing 10 average files (1 per specimen). In that case, the number 5 should be place in the filed « Average by bunch of ».

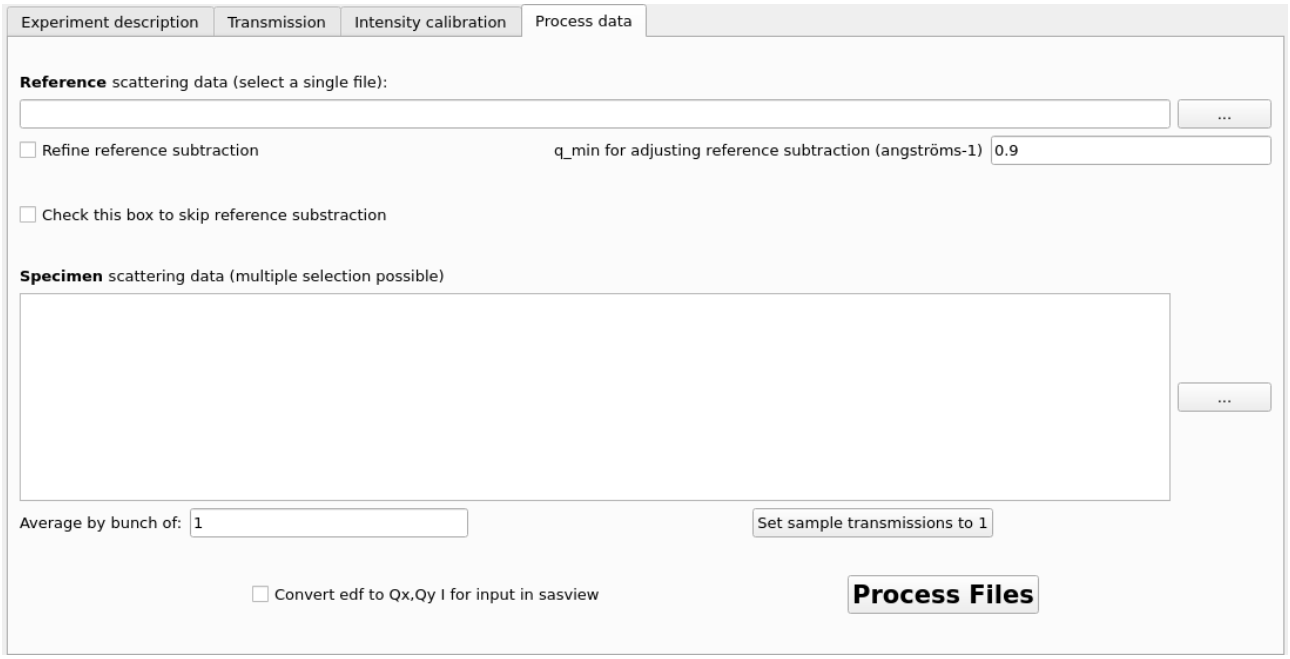


Figure 12 : Process data tab

The bottom section contains one check box « Convert edf to Qx,Qy,I for input in sasview », that should be checked if the user wants to create *.dat files in the form of Qx, Qy, I for further input in sasview. Note that this option strongly increases the processing time. This option should therefore be activated only if necessary. Those data files are automatically saved in a directory named 2D_corr_dat_files. Original frames should advantageously be caved prior the final conversion step .

The « Process Files » button applies equation 2 to the selected files. If the checkbox « Refine reference » substraction is checked, then reference substraction is optimized, as described in section III.2.

Corrected frames are automatically saved in a folder named « corrected_edf_files ».

The figure below gives a comparison between data processing with a manual refinement of the correction coefficient with Foxtrot, and data processed with SAXS_toolbox with the application of the « refine reference subtraction option ». Results show an excellent agreement.

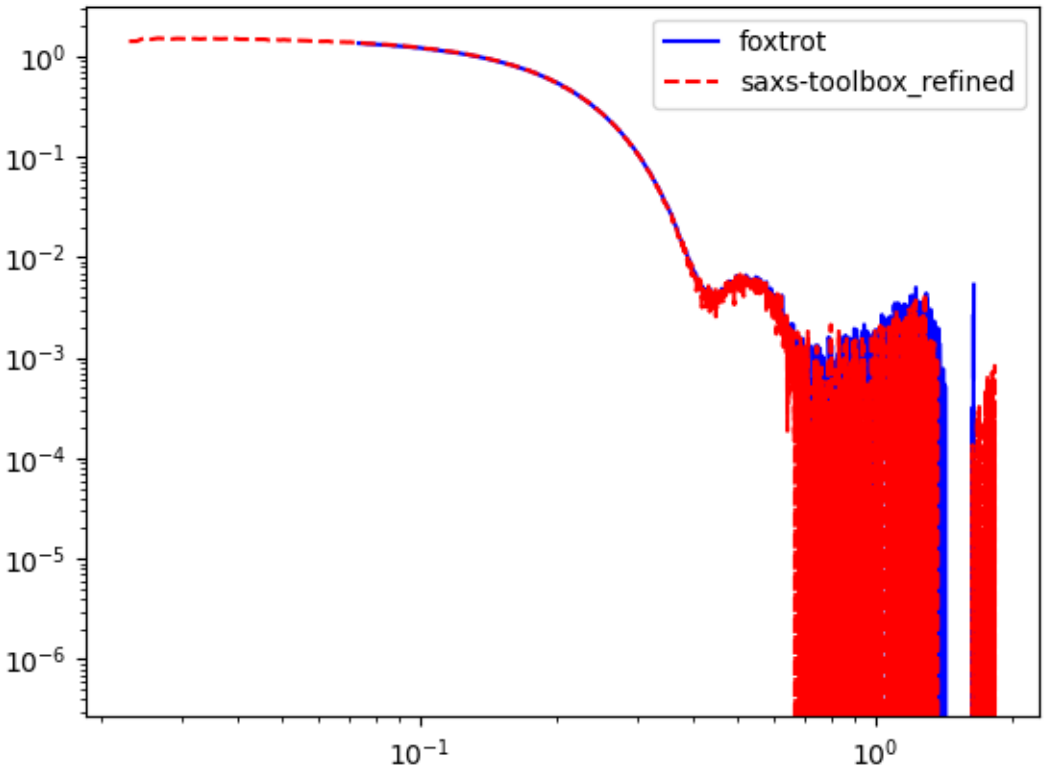


Figure 13: Comparison of data processed with Foxtrot (manual optimization of reference subtraction) and SAXS_toolbox (automatic optimization of reference subtraction)