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. The software therefore produces corrected and calibrated 2D images, and proposes tools to integrate them (raidal regrouping, sectors,…). 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 is also possible.
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

Some of those libraries might already be installed, particularly if you have installed python using an anaconda distribution. However, 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

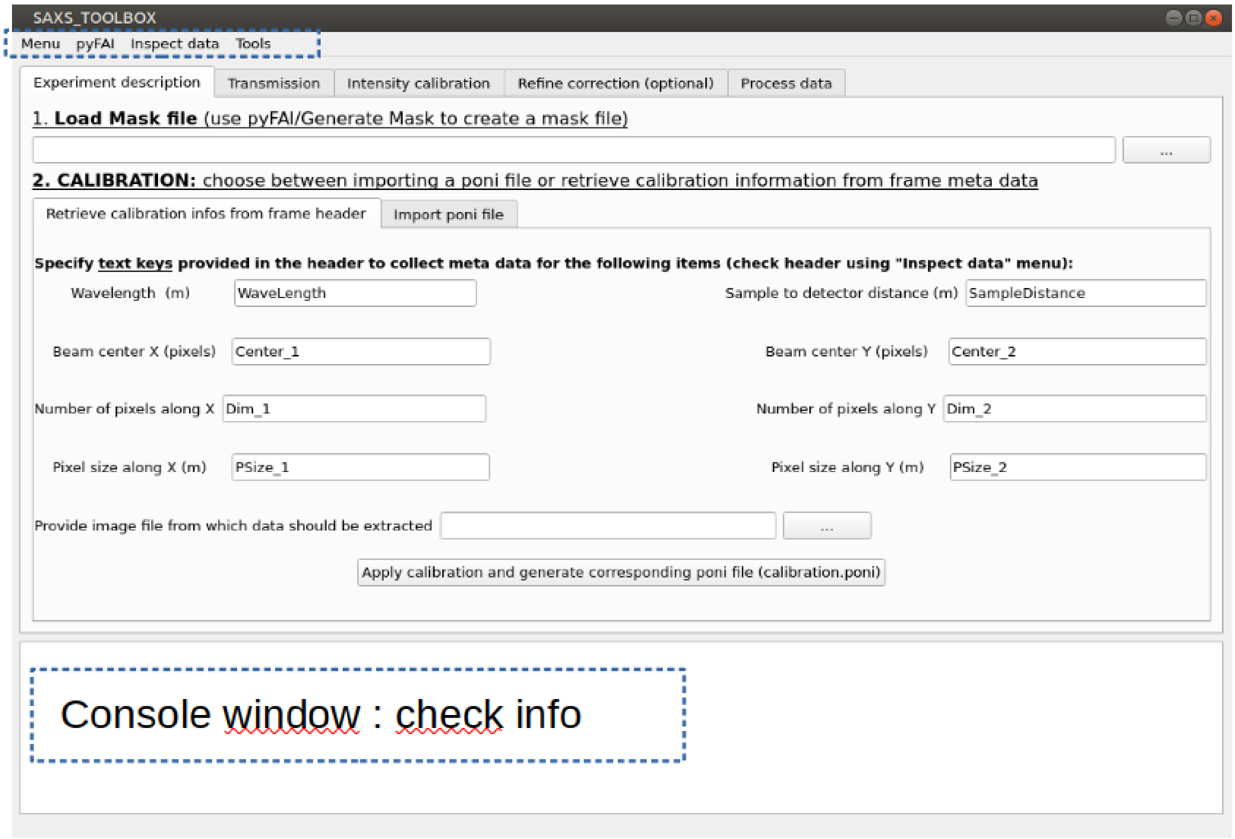
The procedure to start the program is the following :
- open a prompt (terminal on linux, anaconda prompt or shell in windows)
- browse to the diredtory where SAXS_TOOLBOX was unzipped using

```
> cd your/path/to/SAXS_TOOLBOX
```

- call python to execute the script

```
> python SAXS_TOOLBOX.py
```

The following window should appear :


SAXS_TOOLBOX
Menu
pyFAI
Inspect data
Tools
Experiment description
Transmission
Intensity calibration
Refine correction (optional)
Process data
Retrieve calibration infos from frame header
Import poni file
...
Apply calibration and generate corresponding poni file (calibration.poni)

1. **Load Mask file** (use pyFAI/Generate Mask to create a mask file)

2. CALIBRATION: choose between importing a poni file or retrieve calibration information from frame meta data

Specify text keys provided in the header to collect meta data for the following items (check header using "Inspect data" menu):

Wavelength (m): WaveLength

Sample to detector distance (m): SampleDistance

Beam center X (pixels): Center_1

Beam center Y (pixels): Center_2

Number of pixels along X: Dim_1

Number of pixels along Y: Dim_2

Pixel size along X (m): PSize_1

Pixel size along Y (m): PSize_2

Provide image file from which data should be extracted

Console window : check info

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 :

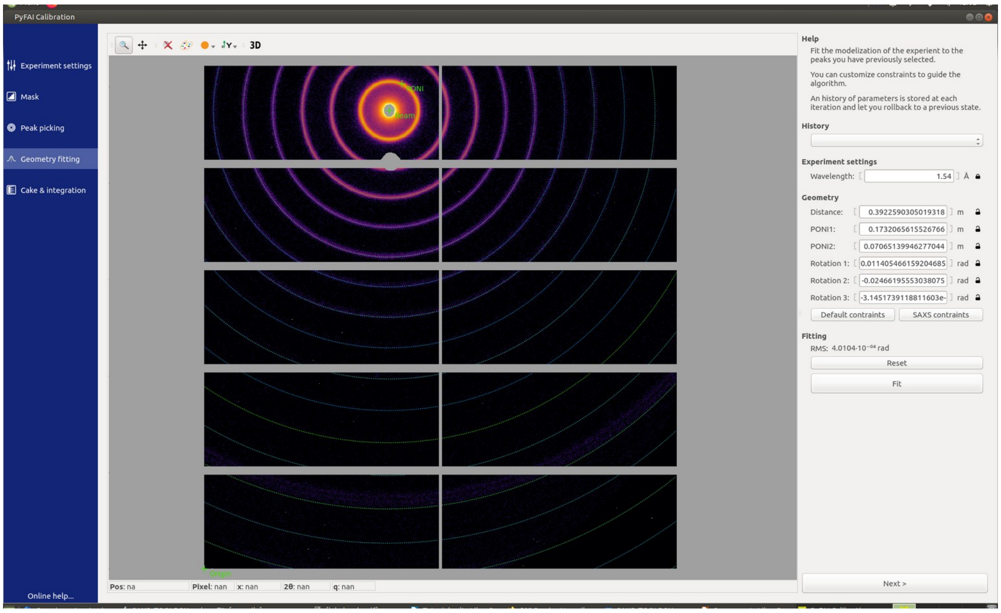
- **Set Working Directory** : this function allows you to sselect the default directory that will be accessed by SAXS_Toolbox

- **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 ».



Help

Fit the modelization of the experient to the peaks you have previously selected.

You can customize constraints to guide the algorithm.

An history of parameters is stored at each iteration and let you rollback to a previous state.

History

Experiment settings

Wavelength: 1.54 Å

Geometry

Distance: 0.3922590305019318 m

PONI1: 0.1732065615526766 m

PONI2: 0.07065139946277044 m

Rotation 1: 0.011405466159204685 rad

Rotation 2: -0.02466195553038075 rad

Rotation 3: -3.1451739118811603e- rad

Default contraints | SAXS contraints

Fitting

RMS: 4.0104·10⁻⁰⁴ rad

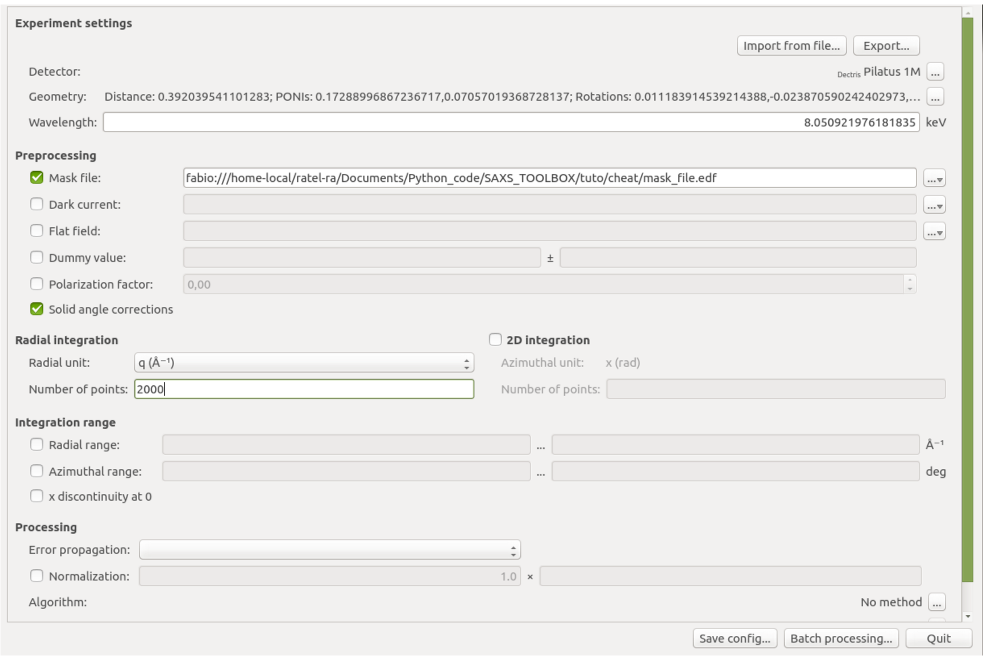
Reset

Fit

Pos: na Pixel: nan x: nan 2θ: nan q: nan

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.



Experiment settings

Detector: Dectris Pilatus 1M

Geometry: Distance: 0.392039541101283; PONIs: 0.17288996867236717,0.07057019368728137; Rotations: 0.011183914539214388,-0.023870590242402973,...

Wavelength: 8.050921976181835 keV

Preprocessing

☑ Mask file: fabio:///home-local/ratel-ra/Documents/Python_code/SAXS_TOOLBOX/tuto/cheat/mask_file.edf

□ Dark current:

□ Flat field:

□ Dummy value: ±

□ Polarization factor: 0,00

☑ Solid angle corrections

Radial integration

Radial unit: q (Å⁻¹)

Number of points: 2000

□ 2D integration

Azimuthal unit: χ (rad)

Number of points:

Integration range

□ Radial range: ... Å⁻¹

□ Azimuthal range: ... deg

□ χ discontinuity at 0

Processing

Error propagation:

□ Normalization: 1.0 ×

Algorithm: No method

Figure 3 : Frame integration using pyFAI-integrate

- **Generate a mask : 2 options are available**

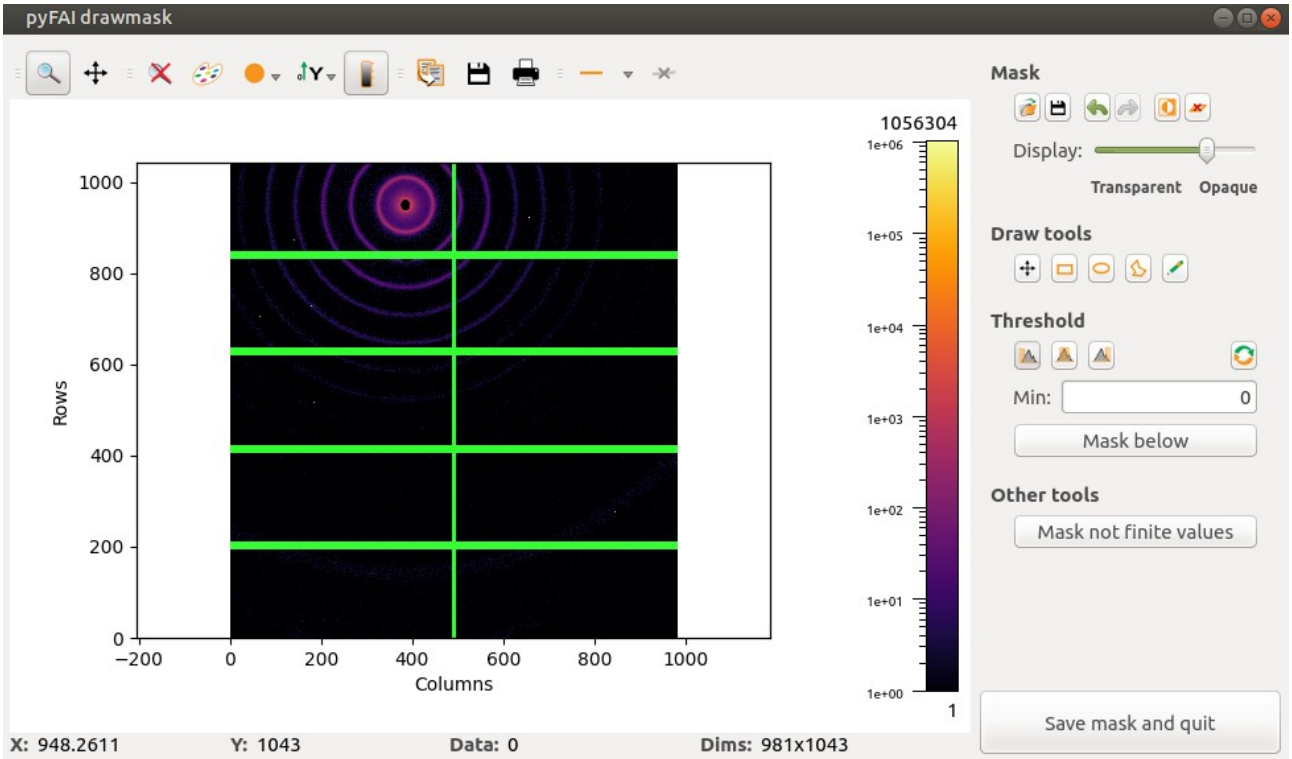
- **using an image** this function first ask you to open a frame file to be opened in pyFAI-drawmask tool. Although great care should be taken for this step, the generation of the mask can be relatively quickly done with the following steps :

- 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.



Min: 0

X: 948.2611 Y: 1043 Data: 0 Dims: 981x1043

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.

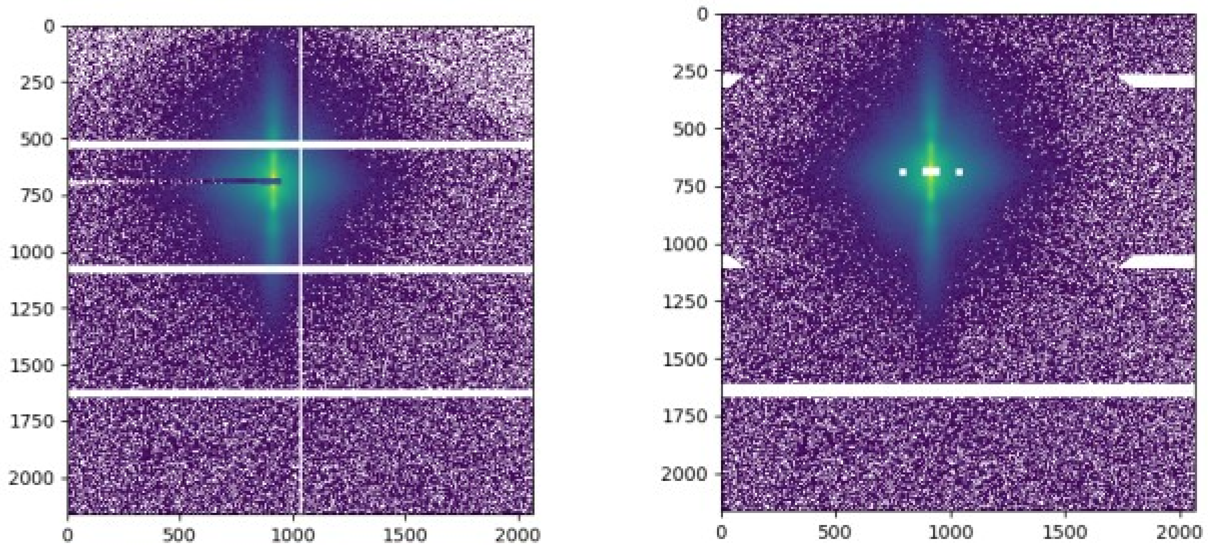
- **plot 1D data file(s) :** This function allows the inspection of several 1D data file(s). All files should be located in the same directory. The user can choose between loglog plots (useful for SAXS data) or linear plots.

II.4 Menu « Tools »

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

- **Convert .nxs to .edf :** 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). 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.

- **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

- **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.
- **Batch_horizontal_regrouping :** Similar to Batch_radial_profile_plot, but performs the regrouping in the azimuthal range from 170° to 190° (180±10).
- **Batch_vertical_regrouping :** Similar to Batch_radial_profile_plot, but performs the regrouping in the azimuthal range from 80° to 100° (90±10).

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 corrected for transmission. Note that in all cases, 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, thus giving access to 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/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.

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,

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

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.

III.2 Data correction

Transmission correction

In SAXS experiments, experimental data must be corrected by their transmission in order to account for specimen absorption. 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 :

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

In the case of data acquired with the Xeuss, transmission information are provided in the image meta-data. In this case, o**nly the empty beam transmission Io must be provided by the user (manual input).**

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 (0.9Å -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.

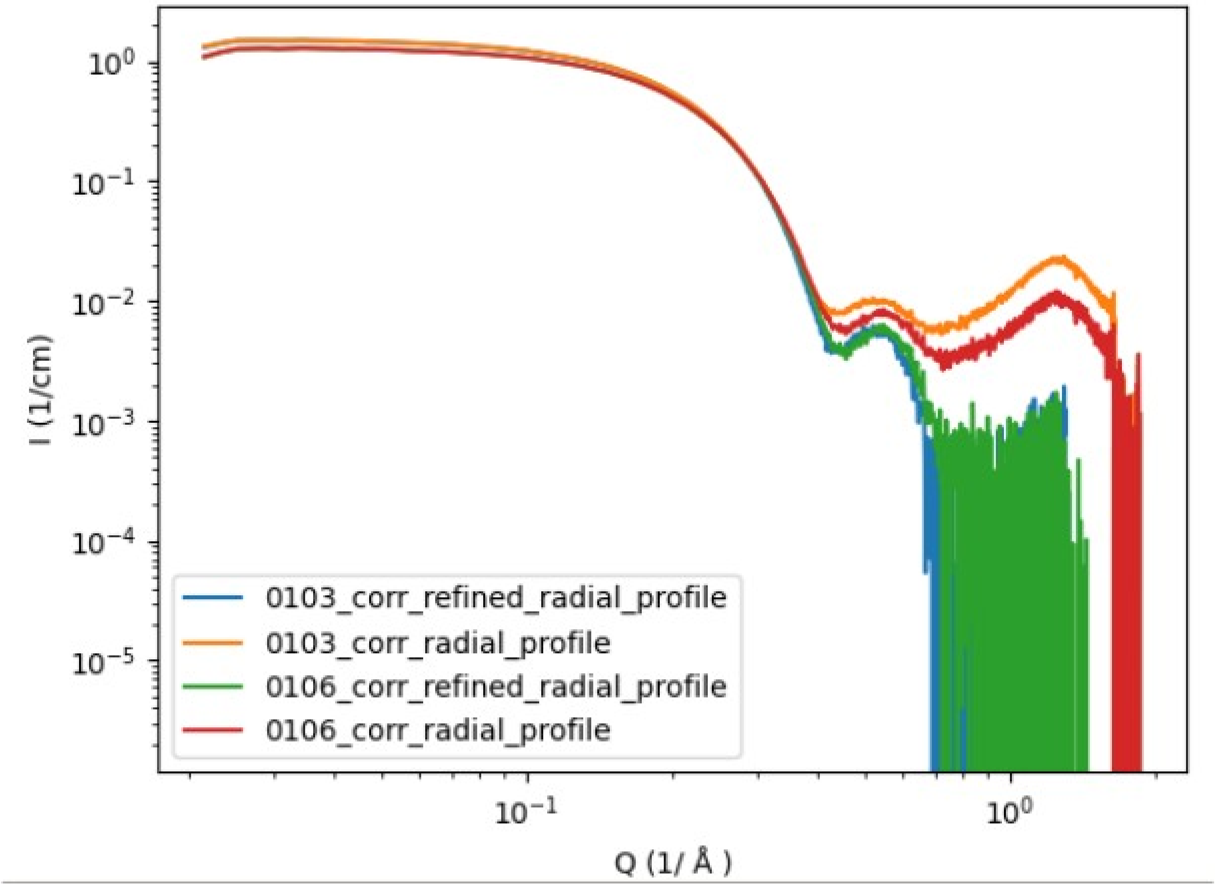
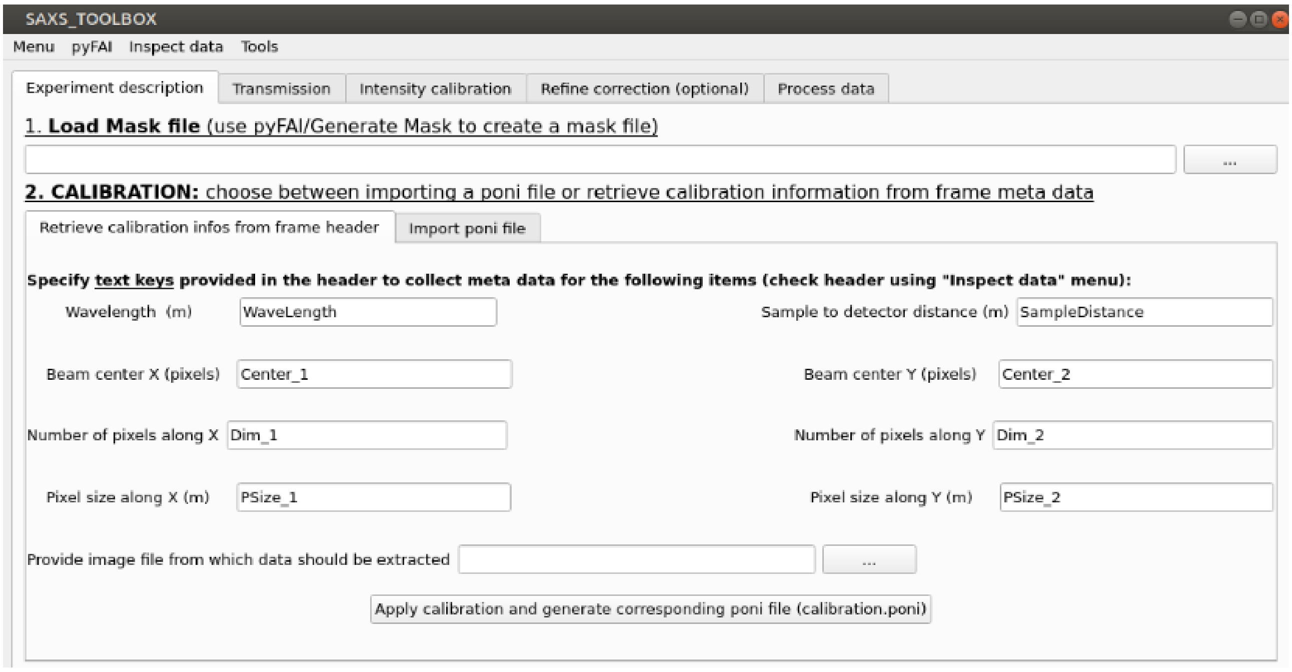


Figure 6 : influence of reference subtraction optimization

IV Description of the graphical interface

IV.1 Experiment description tab


SAXS_TOOLBOX
Menu
pyFAI
Inspect data
Tools
Experiment description
Transmission
Intensity calibration
Refine correction (optional)
Process data
...
Retrieve calibration infos from frame header
Import poni file
...
Apply calibration and generate corresponding poni file (calibration.poni)

1. **Load Mask file** (use pyFAI/Generate Mask to create a mask file)

2. CALIBRATION: choose between importing a poni file or retrieve calibration information from frame meta data

Specify text keys provided in the header to collect meta data for the following items (check header using "Inspect data" menu):

Wavelength (m): WaveLength

Sample to detector distance (m): SampleDistance

Beam center X (pixels): Center_1

Beam center Y (pixels): Center_2

Number of pixels along X: Dim_1

Number of pixels along Y: Dim_2

Pixel size along X (m): PSize_1

Pixel size along Y (m): PSize_2

Provide image file from which data should be extracted

Figure 6 : View of the Experiment description tab

This tab gives you access to two tasks :

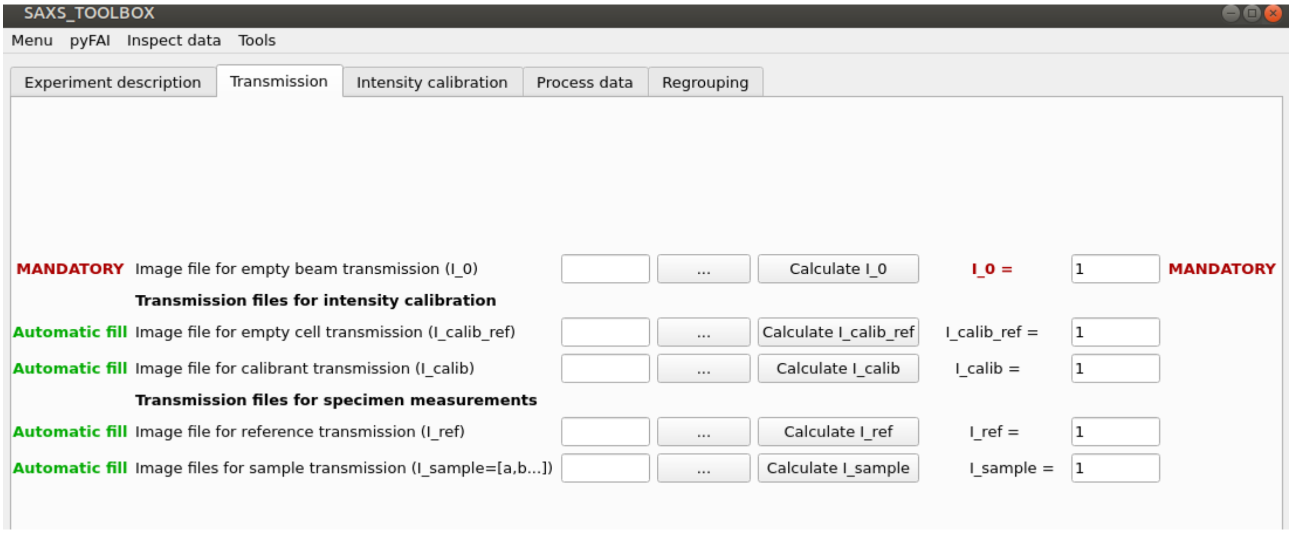
1. assign mask file (mandatory). Simply browse to your mask file, either generated using an image or after conversion of a foxtrot mask. The mask file must be in *.edf format (masked pixels are assigned 0 value, as done in pyFAI).

2. Detector calibration : 2 options

-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



MANDATORY	Image file for empty beam transmission (I_0)	I_0 =	1
	Transmission files for intensity calibration		
Automatic fill	Image file for empty cell transmission (I_calib_ref)	I_calib_ref =	1
Automatic fill	Image file for calibrant transmission (I_calib)	I_calib =	1
	Transmission files for specimen measurements		
Automatic fill	Image file for reference transmission (I_ref)	I_ref =	1
Automatic fill	Image files for sample transmission (I_sample=[a,b...])	I_sample =	1

MANDATORY

Figure 7 : 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 on the tab) support multiple file input. For this reason, transmission values calculated for sample transmissions are stored in an array, whose ith element corresponds to the transmission of the ith 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 8.

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.

In the case of data acquired with the Xeuss, transmission information are provided in the image meta-data. In this case, o**nly the empty beam transmission Io must be provided by the user (manual input).**

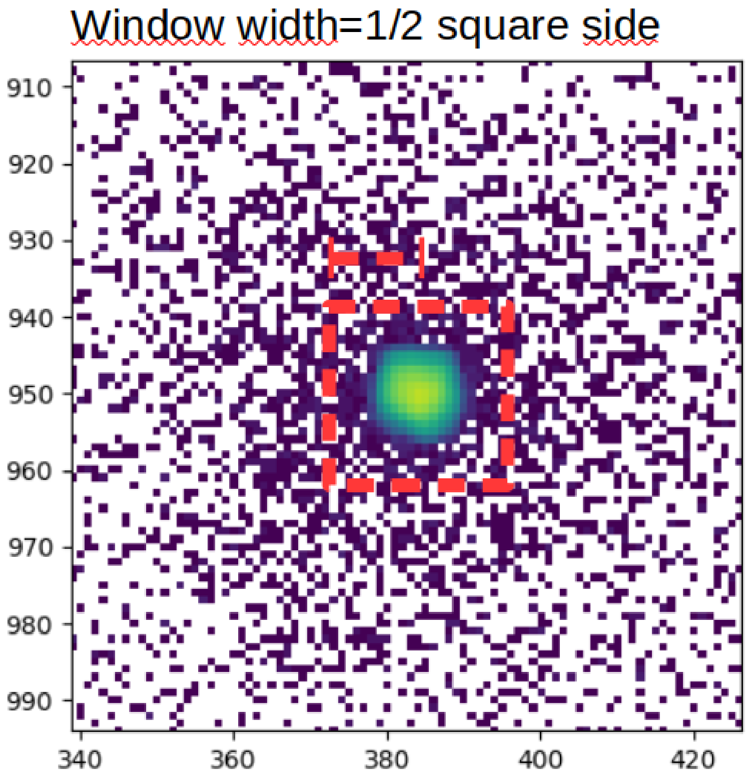
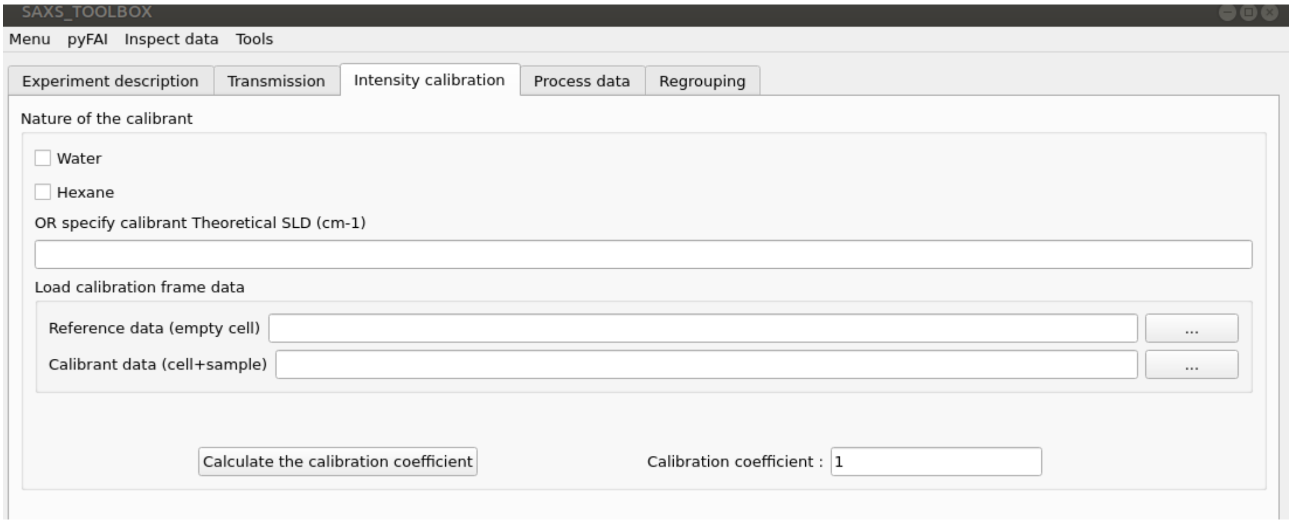


Figure 8 : Sample transmission calculation

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 case another calibrant, is used, the user is invited to provide the expected theoretical scattering length density ρ_{th} in the appropriate field.



Nature of the calibrant

□ Water

□ Hexane

OR specify calibrant Theoretical SLD (cm-1)

Load calibration frame data

Reference data (empty cell) ...

Calibrant data (cell+sample) ...

Calculate the calibration coefficient

Calibration coefficient : 1

Figure 9 : Intensity calibration tab

The calibration is performed as follow :
- select the nature of the calibrant (or specify is SLD)
- provide the experimental data files corresponding to calibration measurements (calibrant + its reference, reference only)
- press the « Calculate the calibration coefficient » button,
Each scattering profile is computed from the provided frames. A mean intensity value is calculated for each frame for q values below 0.5Å. From those mean intensity 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.

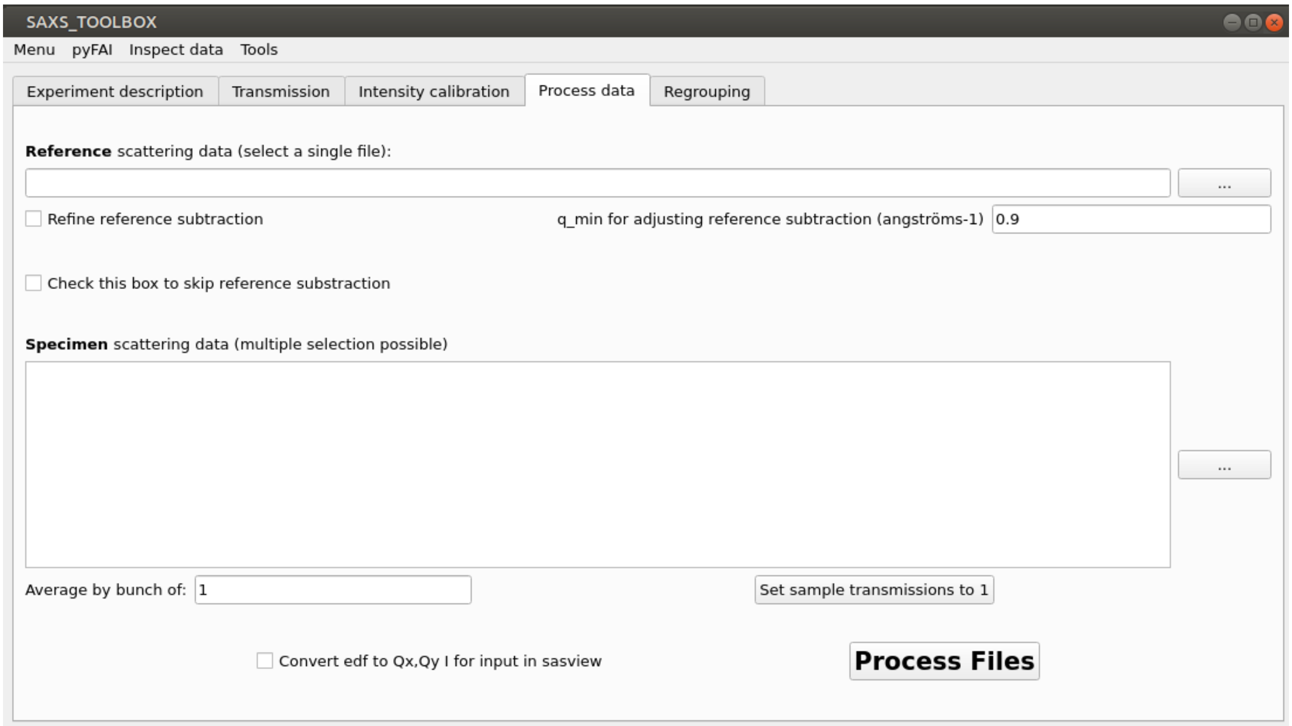
IV.4 Process data tab

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 used 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 0,9Å -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 use 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 field « Average by bunch of ».
A command button is available to set sample transmissions to 1. This button should be clicked when no transmission corrections are applied.

The bottom section contains one check box, 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 therfore be activated only if necessary. Those dat files are automatically saved in a directory named 2D_corr_dat_files.

The « Process Files » button applies equation 2 to the selected files. Corrected frames are automatically saved in a folder named « corrected_edf_files ».



Reference scattering data (select a single file):

□ Refine reference subtraction

q_min for adjusting reference subtraction (angströms-1): 0.9

□ Check this box to skip reference substraction

Specimen scattering data (multiple selection possible)

Average by bunch of: 1

□ Convert edf to Qx,Qy I for input in sasview

Figure 10 : Process data tab

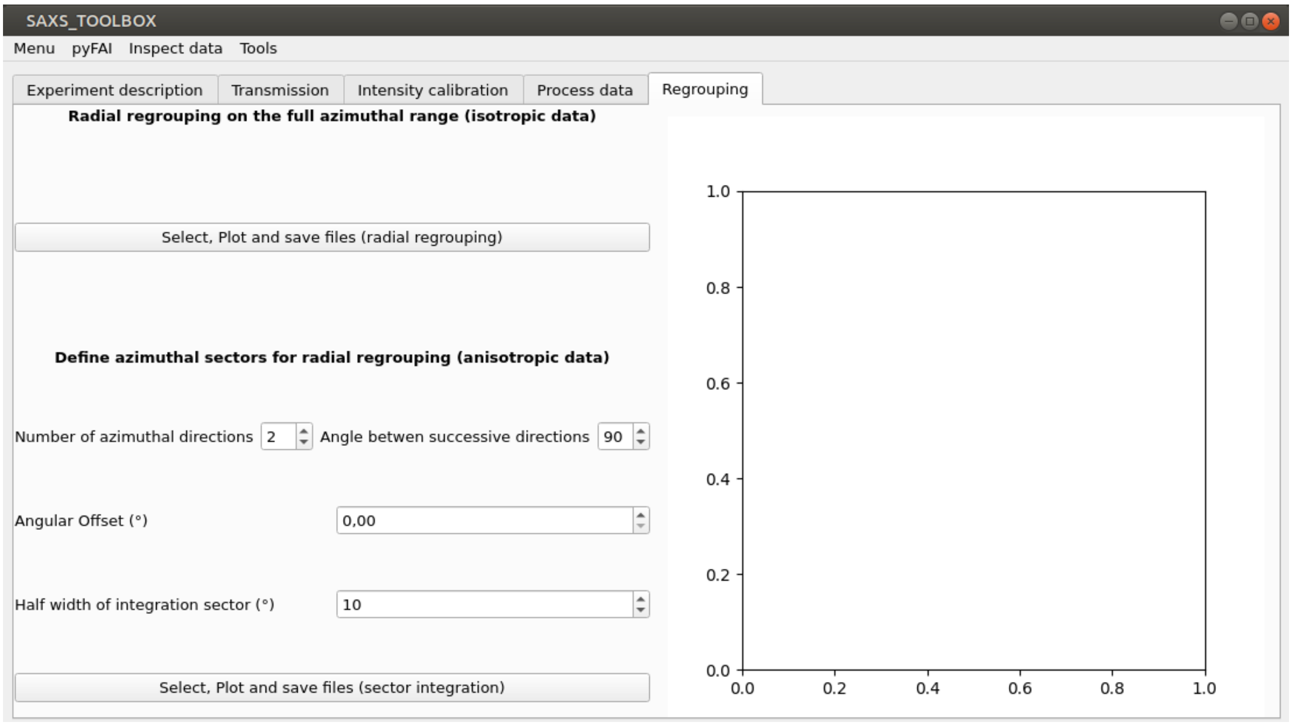
IV.5 Regrouping tab

2D images (corrected or not) can be integrated using the tools proposed in the « Regrouping tab ». On top of the tab, a button allows the selection of 2D images to be radially regrouped, as standardly done with isotropic data. The program then plots the data and creates corresponding 1D data files (two columns q (1/Å), I(cm^{-1}).

In the case of anisotropic data, sector integration can be performed to visualize diffrences between different azimuths. The sectors can be defined by the following parameters :

- number of azimuthal directions
- angle between successive directions
- angular offset
- half width of integration sector

Defaut parameters are set for perfectly horizontal and vertical directions. The angular offset allows for small corrections.



Radial regrouping on the full azimuthal range (isotropic data)

Define azimuthal sectors for radial regrouping (anisotropic data)

Number of azimuthal directions 2 Angle betwen successive directions 90

Angular Offset (°) 0,00

Half width of integration sector (°) 10

Figure 11 : Regrouping tab.