

# RixsToolBox: software for the analysis of soft X-ray RIXS data acquired with 2D detectors

K. Kummer,<sup>a\*</sup> A. Tamborrino,<sup>a,b</sup> A. Amorese,<sup>a</sup> M. Minola,<sup>b,c</sup> L. Braicovich,<sup>b</sup>  
N. B. Brookes<sup>a</sup> and G. Ghiringelli<sup>b</sup>

<sup>a</sup>European Synchrotron Radiation Facility, 71 Avenue des Martyrs, CS40220, F-38043 Grenoble Cedex 9, France,

<sup>b</sup>CNR-SPIN and Dipartimento di Fisica, Politecnico di Milano, Piazza Leonardo da Vinci 32, I-20133 Milano, Italy, and

<sup>c</sup>Max-Planck-Institut für Festkörperforschung, Heisenbergstrasse 1, D-70569 Stuttgart, Germany.

\*Correspondence e-mail: kurt.kummer@esrf.fr

Received 6 September 2016

Accepted 17 January 2017

Edited by G. Grübel, HASYLAB at DESY, Germany

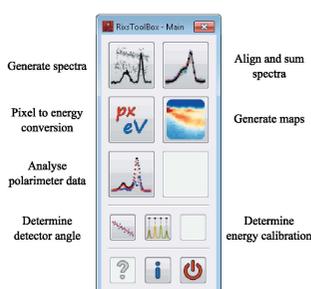
**Keywords:** *RixsToolBox*; resonant inelastic X-ray scattering; RIXS; software; ESRF.

A software with a graphical user interface has been developed with the aim of facilitating the data analysis for users of a new resonant inelastic X-ray scattering (RIXS) spectrometer installed at the ESRF beamline ID32. The software is organized in modules covering all relevant steps in the data reduction from a stack of several hundred two-dimensional CCD images to a single RIXS spectrum. It utilizes both full charge integration and single-photon centroiding to cope with high-flux and high-resolution requirements. Additional modules for further data analysis and the extraction of instrumental parameters are available. The software has been in routine use for about a year now and in that time many additional features have been incorporated. It now meets the users' need for an easy-to-use data analysis tool that allows looking at and understanding data as it is acquired and thus steering users' experiments more efficiently.

## 1. Introduction

Over the last decades resonant inelastic X-ray scattering (RIXS) has developed into an important tool for studying the electronic and magnetic structure and collective excitations of complex materials. Currently, several projects for new RIXS spectrometers in the soft X-ray range are ongoing, all trying to push the limits in terms of resolution, flexible scattering geometry and/or throughput. The first instrument of this new generation is operated at the ESRF soft X-ray beamline ID32 and has seen user operation for more than a year now (<http://www.esrf.fr/ID32>). It has replaced the AXES spectrometer at the former ESRF beamline ID08. With the better availability and the continuously improving performance of these new RIXS spectrometers, the user community will widen and more and more groups who are new to the technique will start to use RIXS for their research.

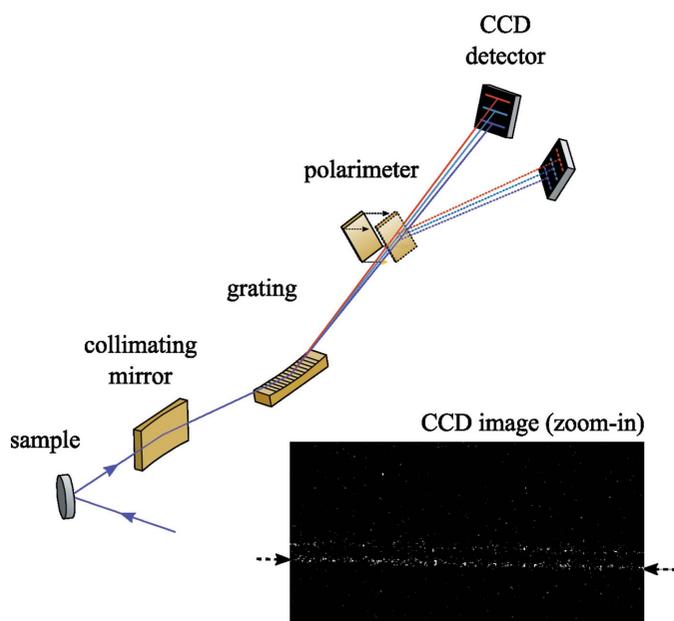
It became evident very early on during the commissioning of the ESRF spectrometer that user-friendly software for on- and off-line data analysis was needed in order to make efficient use of the instrument and to reduce the barriers to entry for first-time user groups. We have therefore developed a software package with a graphical user interface (GUI) that allows the recurring steps in data extraction and analysis to be performed conveniently and quickly. The software, written in Python, evolved from Matlab scripts developed by Politecnico Milano for the operation of the RIXS spectrometer at the former ESRF beamline ID08. It enables both expert and non-expert users to constantly visualize and assess the data they are obtaining and to steer their experiment accordingly. The



software has been in routine use for about a year now. In that time a number of feature requests that emerged during experiments have been incorporated and it has now matured into a somewhat steady state covering most of the needs of the users. The software is portable; executables for Windows and Mac OS X are available.

## 2. Instrumentation and raw data

The optical design of the ESRF ID32 spectrometer is shown in Fig. 1. Compared with previous instruments it has been extended by a parabolic collimating mirror and a graded multilayer for polarization analysis (Braicovich *et al.*, 2014) but otherwise it follows the classic scheme for high-resolution soft X-ray RIXS instruments (Dallera *et al.*, 1996; Ghiringhelli *et al.*, 2006). The sample is hit with a highly monochromatic polarized X-ray beam and the scattered photons are then analyzed in energy, momentum and possibly polarization state in order to learn about elementary excitations in the sample. Energy resolution is achieved by using a diffraction grating for dispersing the scattered light onto a two-dimensional (2D) CCD detector. Scattered photons of the same energy will hit the detector along isoenergetic lines perpendicular to the diffraction plane. Isoenergetic lines corresponding to different photon energies will be spatially separated from each other in the detector plane. Besides the quality of the optics, the energy bandwidth of the incoming beam and the beam spot size on the sample, the spatial resolution of the detector is therefore one of the factors that can limit the energy resolution of the spectrometer. Hence the choice of an appropriate detector and proper analysis of the acquired 2D images is crucial for



**Figure 1** Scheme of a classical soft X-ray RIXS spectrometer. The scattered beam is dispersed onto a 2D detector by a diffraction grating. Photons with the same energy will hit the detector along isoenergetic lines perpendicular to the diffraction plane. A zoom-in into a typical image on the detector is shown as an inset.

the performance of a RIXS spectrometer. Most of the operated and the upcoming soft X-ray spectrometers have the same or similar working schemes employing diffraction gratings to disperse the scattered light in one direction onto a 2D detector.

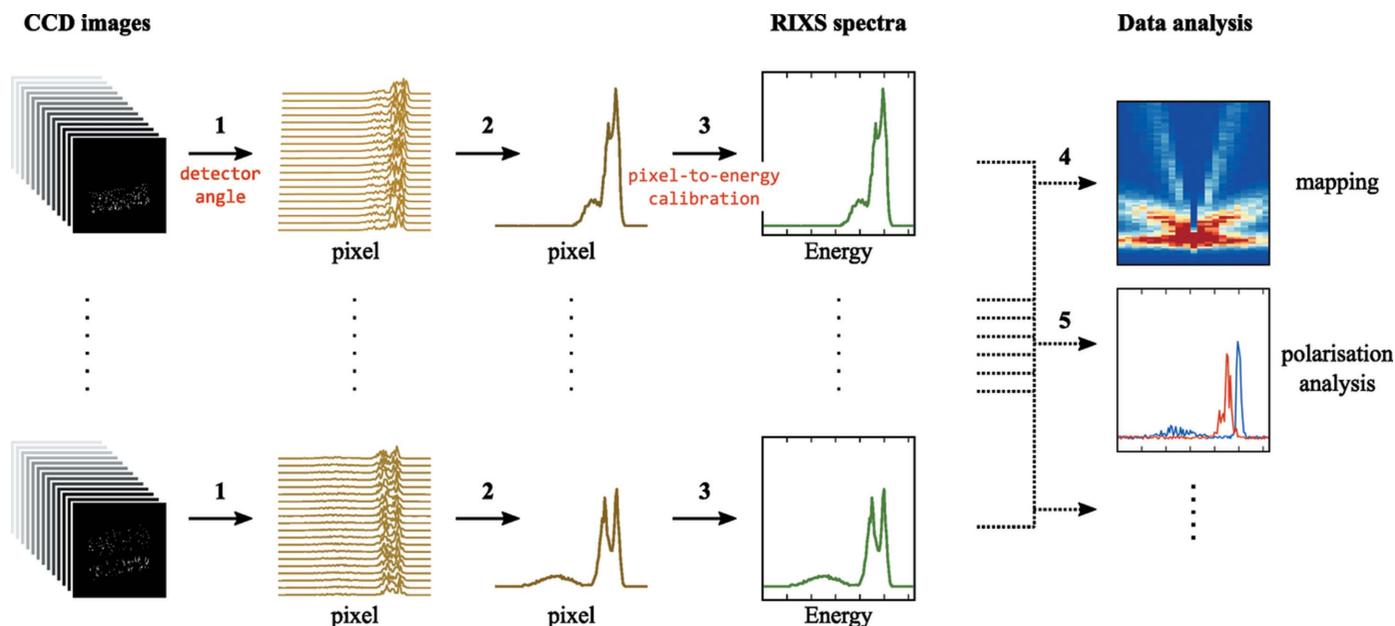
A zoom-in into a typical CCD image acquired at the ID32 spectrometer is shown as an inset in Fig. 1. Depending on the count rate, such an image is taken in a few to several tens of seconds in order to keep the number of photons per image low enough for single-photon-counting algorithms. The photons are distributed along isoenergetic lines going from the upper left to the lower right. The energetic lines are not running perfectly horizontal because the detector has been deliberately mounted with a small inclination in order to spread isoenergetic lines over several rows of the CCD. The line corresponding to zero energy loss, *i.e.* elastic scattering, is marked with arrows in Fig. 1.

## 3. Data analysis workflow

As part of the data analysis the information in the CCD images has to be translated into RIXS spectra, *i.e.* spectra showing the number of scattered photons as a function of energy loss,  $E_{\text{loss}} = h\nu_{\text{in}} - h\nu_{\text{out}}$ . In order to obtain spectra with sufficiently good statistics, data from several tens or hundreds of images taken under the same experimental conditions are combined. Fig. 2 shows the workflow that is commonly employed to reduce the stack of images to a single RIXS spectrum. This is done in three steps:

(1) Reducing each 2D image to a one-dimensional spectrum of scattered photons *versus* pixel. This can be done by simply integrating the intensity on the detector along the isoenergetic lines or by employing single-photon-counting algorithms that can dramatically increase the spatial resolution and the signal-to-noise ratio (*cf.* Amorese *et al.*, 2014). This step requires knowledge of the slope of the isoenergetic lines on the detector, *i.e.* the mounting angle of the detector as well as knowledge of detector properties like the CCD sensitivity, the base-mean level and the energy required to create a single electron–hole pair in order to correctly convert the analogue-to-digital counts of the CCD electronics into photon counts. All these parameters are fixed for a particular instrument and do not change during an experiment.

(2) Aligning and summing the individual spectra. In the second step the individual spectra are all summed in order to obtain a high-statistics spectrum. Prior to summing, alignment of the spectra relative to each other can be necessary. This is because a single high-statistics spectrum may require several hours of image acquisition. Depending on the stability of the instrument (beamline and spectrometer), small continuous drifts of the isoenergetic lines over the detector may be observed and must be corrected to not deteriorate the instrumental resolution. Using a cross-correlation algorithm for aligning the spectra before summing yields good results typically. The biggest source of drifts at the ESRF ID32 spectrometer were the liquid-nitrogen CCD detectors that had to be refilled every 12 hours and showed continuous thermal



**Figure 2**

Workflow for the analysis of RIXS data collected with 2D detectors. To obtain a RIXS spectrum from a stack of images one has to (1) reduce each 2D image into a one-dimensional spectrum 'pixel versus photons', (2) align the individual spectra and sum, and (3) translate pixel into an energy scale. In a typical experiment, several RIXS spectra are collected, for instance, as a function of momentum transfer, photon energy and temperature, with and without the polarimeter, and then further analyzed (4, 5).

expansion in between moving the spectrum by up to ten pixels in 12 h. These detectors have been replaced now by Peltier-cooled detectors and the stability of the instrument is now improved to less than typically one pixel drift over many hours of acquisition. Correction of even such small drifts is required in the data analysis to preserve the ultimate instrumental resolution.

(3) Converting pixel units into energy loss. After aligning and summing, a high-statistics spectrum of photon counts versus pixel is obtained. Knowledge of the position of the pixel corresponding to  $E_{\text{loss}} = 0$  and the energy dispersion on the detector plane allows pixels to be converted into energy loss. Both parameters can be determined experimentally, the 'zero' pixel by measuring the non-resonant purely elastic scattering,  $E_{\text{in}} = E_{\text{out}}$ , from another material like amorphous carbon at the same photon energy as the sample. The correspondance meV-per-pixel can be determined by using again an elastic scatterer and looking at the shift of the elastic line in pixels when varying the incoming photon energy.

During an experiment many RIXS spectra are measured and then further analyzed. The most common experimental case is to look at changes in the RIXS spectra as a function of momentum transfer, *i.e.* scattering geometry, or other parameters such as incoming photon energy or sample temperature. Visualizing several RIXS spectra against some parameter in a waterfall plot or a 2D map is therefore often the next step in the data analysis. Extracting information about the polarization state of scattered photons by looking at two RIXS spectra obtained with and without the polarizing multilayer in the scattered beam is another. Beyond that is where the detailed analysis starts which can be different for each

experiment and for which the user groups usually use their favourite software.

#### 4. Description of the software and its functionality

*RixsToolBox* has been written entirely in Python. For the GUI we are using the *PyMca5* library for Python (<https://github.com/vasole/pymca/>) with no other dependencies which keeps maintenance low and makes the GUI look familiar to most ESRF users who are already using the *PyMca* data viewer (Solé *et al.*, 2007). *RixsToolBox* is designed to run on a desktop PC with no special requirements. Executables are available for Windows and Mac OS X.

The objective of our data analysis software was to offer convenient tools for each step in the data analysis. At the same time the user should be free to choose at any moment in the analysis to change to their own data analysis software. The software is therefore organized in small tools each for one particular step in the analysis. Data exchange between the modules is done using ASCII files in the ESRF \*.spec data format. These files can be easily read by other data analysis software and, with *PyMca*, the ESRF standard viewer for spectroscopic data, a powerful viewer is already available.

Fig. 3 shows the main interface of the software. Each button will open a separate window with a GUI-based tool for the specific task. There are tools for the three steps to reduce the stack of CCD images to a single RIXS spectrum as described in the previous section. For further analysis a tool to generate waterfall plots and 2D maps and a tool to analyze data acquired with the polarimeter are available. In addition, there are two auxiliary tools that can be used to determine input

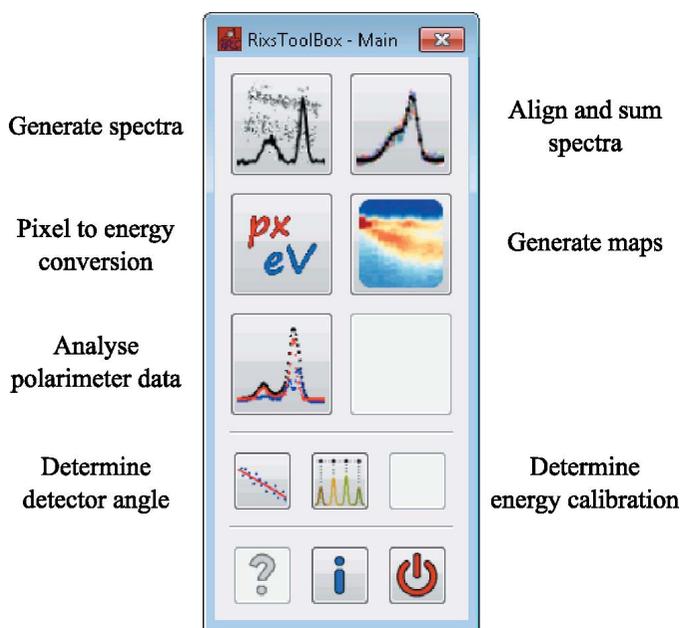


Figure 3 Main interface of the RixsToolBox software.

parameters needed to generate the RIXS spectra, the detector angle and the energy dispersion on the detector.

Fig. 4 shows as an example the interface of the tool to align and sum the spectra. The left half of the window is for input file, output file and data selection. The right half has, at the top, a parameter panel for user inputs on how to align and sum the spectra and, below, two plot windows that show the spectra at each step and the calculated and the applied shift in pixels for each spectrum. The GUIs of the other tools follow that general layout with data input/output selection on the left and parameter input and visualization on the right in order to confront users with a consistent interface.

The workflow typically follows the scheme shown in Fig. 2. The tool for spectra generation can be used to extract the

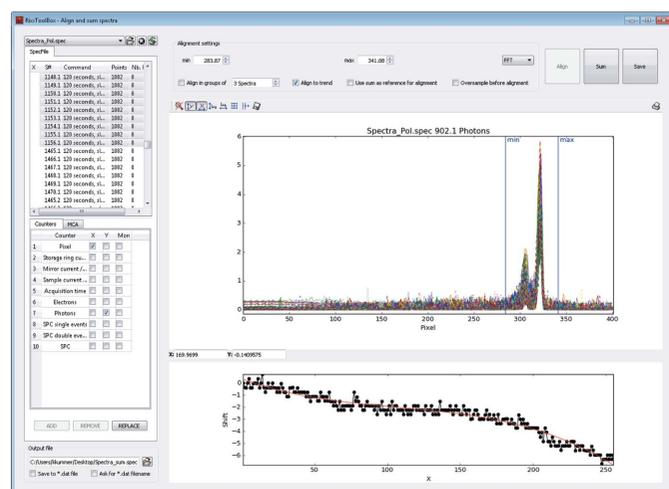


Figure 4 Screenshot of the GUI tool for aligning and summing a stack of spectra generated from CCD images.

spectra from the CCD images and saved into an ASCII file. This file is then loaded into the tool to align and sum the spectra and the result saved in an ASCII file, ready for the pixel-to-energy conversion using the respective tool. It should be noted that, after the spectra have been generated from the CCD images, the user is free to alter the workflow as required. For instance, the energy conversion could be done before aligning and summing the spectra, or the spectra could be only aligned but not summed or only summed without alignment. Here we give a brief description of each tool.

#### 4.1. Spectrum generation

A screenshot of the tool for spectrum generation is shown in Fig. 5. This tool allows spectra to be generated from CCD image files and saved into an ASCII file. The possibility of choosing regions of interest (ROIs) allows different parts of the CCD image to be analysed separately. Both methods, the conventional integration along isoenergetic lines and single-photon counting with centroiding, have been implemented and are always done in parallel. So, for every defined ROI the user always obtains the result of both methods and can easily compare. A detailed description of the single-photon-counting algorithm is given by Amorese *et al.* (2014) where the increase in spatial resolution when using centroiding is demonstrated. All relevant parameters for the spectrum generation, like the CCD parameters, the detector angle, the sampling rate (data points per pixel) and the threshold for the different algorithms, are preset to meaningful values for the ESRF ID32 instrument but can be modified by the user at any moment in the main interface or dialog window shown in Fig. 5 on the right. As the spectrum generation can be performed without any user input, we have automatized this step in our data acquisition chain and the spectra are generated as the images are acquired using the same algorithms that are used by the GUI tool. Therefore, in practice, this tool will only be interesting to users if spectra should be re-extracted from the raw images using different parameters. At the moment, the only accepted input data format for CCD images is the ESRF data

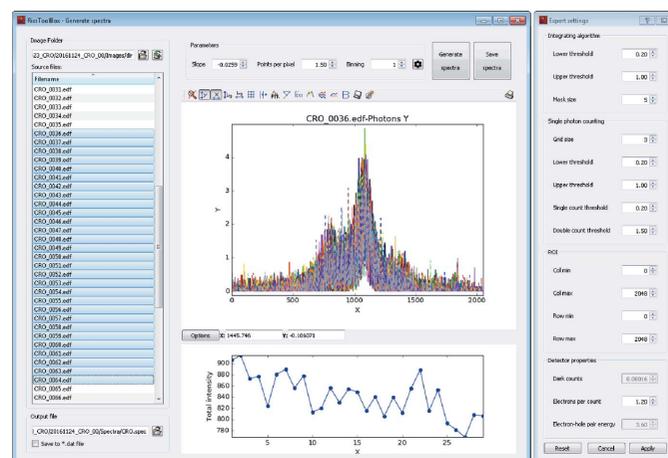


Figure 5 Screenshot of the GUI tool for generating spectra from CCD images.

format (.edf), the standard data format for 2D images at the ESRF, but other file formats could be added easily if needed.

#### 4.2. Aligning and summing spectra

This tool allows long-term drifts of the instrument to be corrected during the acquisition of an image stack and summing the spectra in the stack to one single high-statistics spectrum. After selecting a group of spectra all spectra in the group can be aligned to the first spectrum using different algorithms including cross-correlation and peak fitting. The part of the spectra which should be used for the alignment can be easily selected with sliders and input boxes. In order to reduce the effect of statistical noise in the spectra on the alignment, several methods are available. Several spectra can be summed together in a group of user-defined size and then groups can be aligned with respect to each other. The better statistics in these groups will make the alignment less prone to noise but the time that is required to acquire such a group of spectra should not exceed the instrument stability timescale. It is also possible to apply a band-pass filter to calculated shifts in order to separate the long-term drifts of the instrument from the short-term fluctuations of the alignment algorithm. Again, all the relevant parameters for the alignment can be changed in the main interface. The spectra can be only aligned and saved, only summed and saved without alignment, or aligned, summed and then saved. We refer again to Fig. 4 for a screenshot of this tool.

#### 4.3. Pixel-to-energy conversion

This tool allows an energy-loss axis to be added to the spectra. As inputs it requires the zero-loss pixel and the energy dispersion in  $\text{meV pixel}^{-1}$  on the detector. The zero-loss pixel can be determined directly inside the tool by fitting the spectrum of an elastic reference or just be input by hand.

#### 4.4. Generating maps and waterfall plots

Spectra can be displayed in a waterfall plot or a 2D colormap as a function of another parameter, like the momentum transfer  $Q$ , photon energy or sample temperature. It is possible to select and normalize all spectra to the average or maximum value in a certain region. Results can be saved as images or ASCII data files.

#### 4.5. Analyzing polarimeter data

For polarization analysis, the different reflectivity of the multilayer for  $\sigma$  and  $\pi$  polarized light (Kohn, 1995) can be exploited in soft X-ray RIXS to discern the two components in the scattered beam (Braicovich *et al.*, 2014). To this end, the same spectrum needs to be taken with and without the additional reflection by the multilayer (see Fig. 1). The intensity in the two cases is given by

$$I_{\text{direct}} = I_{\sigma} + I_{\pi} \quad \text{and} \quad I_{\text{ML}} = R_{\sigma} I_{\sigma} + R_{\pi} I_{\pi}, \quad (1)$$

with  $R_{\sigma}$  and  $R_{\pi}$  the reflectivity of the multilayer (ML) for  $\sigma$  and  $\pi$  polarized light. From that,  $I_{\sigma}$  and  $I_{\pi}$  can be calculated as

$$I_{\pi} = \frac{I_{\text{ML}} - R_{\sigma} I_{\text{direct}}}{R_{\pi} - R_{\sigma}} \quad \text{and} \quad I_{\sigma} = \frac{I_{\text{ML}} - R_{\pi} I_{\text{direct}}}{R_{\sigma} - R_{\pi}}. \quad (2)$$

This tool can be used to perform a polarization analysis using the above relationship. The only required inputs are the reflectivity  $R_{\sigma}$  and  $R_{\pi}$  of the multilayer for  $\sigma$  and  $\pi$  polarization, respectively, which can be determined experimentally. After selecting a RIXS spectrum acquired with and without the polarizing multilayer, the  $\sigma$  and  $\pi$  polarization channels are computed and displayed. The two experimental spectra can be preprocessed prior to the calculation in order to account for slightly different energy resolution or an energy shift between the two channels. After the calculations a Gaussian smoothing filter can be applied on the calculated  $\sigma$  and  $\pi$  spectra in case they are very noisy.

#### 4.6. Determining the detector angle

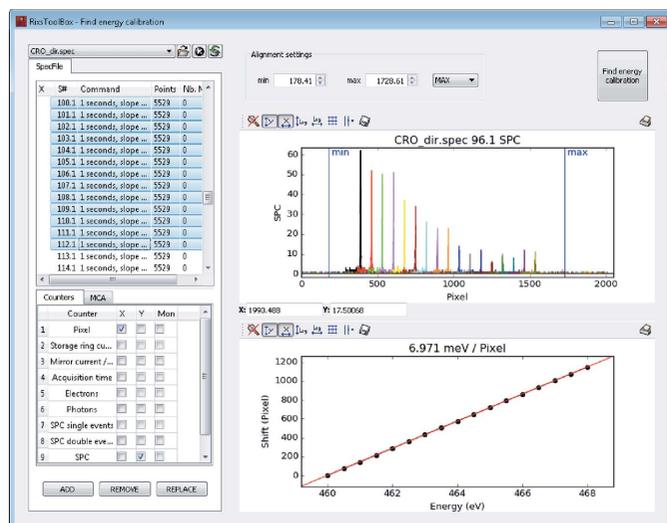
For the generation of the spectra the angle by which the detector is inclined needs to be known. This tool allows this angle to be determined directly from the images. Inputting images with only an elastic line will usually yield the best results but any kind of image can be used. Multiple images can be viewed at the same time to reduce the uncertainty in the calculated detector angle. This tool is mostly for convenience during commissioning of new detectors as the detector angle is fixed during the physical mounting of the detector and will not change during the user operation of the spectrometer.

#### 4.7. Determining the energy dispersion on the detector plane

The energy dispersion on the detector plane is needed to translate pixel units into more meaningful energy units. The conversion factor  $\text{meV pixel}^{-1}$  is required in step (3) of the data analysis workflow (see Fig. 2). It is determined at the beginning of each experiment, after the initial alignment of the spectrometer, by changing the incident photon energy in small steps and measuring the shift of an elastic line in pixels on the detector. The respective spectra can be loaded into this tool and the energy dispersion is calculated automatically from the observed shift of the spectrum with incident photon energy. A screenshot of this tool at work is shown in Fig. 6. In the shown example, the pixel-to-energy conversion at the oxygen  $K$ -edge has been determined.

## 5. Outlook

For the future a number of possible further developments could be imagined both from a merely technical point of view as well as what concerns the use at other instruments. On the technical side, the ESRF is currently moving to new data formats and in the long term we will adapt the software to also work with HDF5 files. For the moment, however, the plain ASCII format \*.spec seems to be most compatible with the users' needs. For the GUI, we reused the developments for the *PyMca* software based on *PyQt* as much as possible. In the future, we will probably move to a dedicated Python toolkit for the development of data analysis applications, like for



**Figure 6** Screenshot of the GUI tool for determining the energy dispersion on the detector.

instance the *silx* toolkit currently being developed at the ESRF (<https://github.com/silx-kit/silx>).

It could also be envisaged to further automate and group some of the data analysis steps shown in Fig. 2. In particular for first-time users of the RIXS spectrometer, reducing the number of steps that require user input could be desirable. However, our experience up to now is that almost each experiment is different in terms of how and what kind of datasets are acquired. We have not identified a common workflow that would work for more than a few experiments. But in order to allow further automation, the data analysis steps are organized in a modular fashion and kept as much as possible independent of the GUI. Therefore, they could easily be scripted in order to increase the degree of automation. At the moment, we are still thinking about the best way of increasing automation without losing flexibility.

The software could easily be used with other RIXS spectrometers that have a similar data analysis workflow. Compared with its current state, only slight additions are needed in order to accommodate image file formats other than ESRF data format (\*.edf). From our point of view, using and developing a standard software together with other RIXS facilities around the world would be largely beneficial to both the users and the user facilities. Maybe this project could be a

starting point for such a development. Maybe it animates the open development of another software with similar or more extended capabilities. In either case, the authors are very open to discussions and exchange about technical aspects and where to go in the future.

## 6. Summary

A portable software for the analysis of RIXS data acquired with 2D detectors has been developed. It is now in routine operation serving the users of the ESRF ID32 RIXS spectrometer in the on-site and off-site analysis of their data. A GUI mimicking that of the ESRF standard data viewer *PyMca* makes the software very accessible with practically no learning curve. The software could be easily adapted and extended to meet further needs and to work with other RIXS spectrometers based on 2D detectors. Packaged versions of *RixsToolBox* for Windows and MacOS X are available for download (<https://github.com/kkummer/RixsToolBox/releases>). Future developments and addition of further tools will be largely driven by new requirements coming up during experiments and requests by the user community. For feature requests the authors can be contacted directly.

## Acknowledgements

The authors would like to thank G. Dellea, Y. Peng, C. Langini, M. Conni, F. Yakhou-Harris and D. Betto for their feedback on the software, and V. A. Solé, C. Ferrero and R. Wilcke from the ESRF data analysis unit for valuable discussions and help with packaging the software.

## References

Amorese, A., Dellea, G., Braicovich, L. & Ghiringhelli, G. (2014). *arxiv:1410.1587*.  
 Braicovich, L., Minola, M., Dellea, G., Le Tacon, M., Moretti Sala, M., Morawe, C., Peffen, J.-C., Suprunangnet, R., Yakhou, F., Ghiringhelli, G. & Brookes, N. B. (2014). *Rev. Sci. Instrum.* **85**, 115104.  
 Dallera, C., Puppini, E., Trezzi, G., Incorvaia, N., Fasana, A., Braicovich, L., Brookes, N. B. & Goedkoop, J. B. (1996). *J. Synchrotron Rad.* **3**, 231–238.  
 Ghiringhelli, G., Piazzalunga, A., Dallera, C., Trezzi, G., Braicovich, L., Schmitt, T., Strocov, V. N., Betemps, R., Patthey, L., Wang, X. & Grioni, M. (2006). *Rev. Sci. Instrum.* **77**, 113108.  
 Kohn, V. G. (1995). *Phys. Status Solidi B*, **187**, 61–70.  
 Solé, V., Papillon, E., Cotte, M., Walter, P. & Susini, J. (2007). *Ar. Spectrosc.* **62**, 63–68.