

SOFTWARE

STXMPy: a new software package for automated region of interest selection and statistical analysis of XANES data

Tamás Haraszti^{*1,2}, Michael Grunze^{3,4} and Michael G Anderson^{5,6}

Abstract

Background: Soft X-ray spectromicroscopy based absorption near-edge structure analysis, is a spectroscopic technique useful for investigating sample composition at a nanoscale of resolution. While the technique holds great promise for analysis of biological samples, current methodologies are challenged by a lack of automatic analysis software e. g. for selection of regions of interest and statistical comparisons of sample variability.

Results: We have implemented a set of functions and scripts in Python to provide a semiautomatic treatment of data obtained using scanning transmission X-ray microscopy. The toolkit includes a novel line-by-line absorption conversion and data filtering automatically identifying image components with significant absorption. Results are provided to the user by direct graphical output to the screen and by output images and data files, including the average and standard deviation of the X-ray absorption spectrum. Using isolated mouse melanosomes as a sample biological tissue, application of STXMPy in analysis of biological tissues is illustrated.

Conclusion: The STXMPy package allows both interactive and automated batch processing of scanning transmission X-ray microscopic data. It is open source, cross platform, and offers rapid script development using the interpreted Python language.

Background

Scanning transmission X-ray microscopy (STXM) is a synchrotron based technique for the investigation of sample structure and composition with nanoscale (c. a. 30 - 50 nm) resolution [1,2]. High resolution X-ray microscopy is based on X-ray absorption spectroscopy and X-ray absorption near-edge structure analysis (XANES) which provides the chemical information about the specimen.

Compared to electrons soft X-rays have excellent tissue penetrating capability. Using photon energies in the so called "water window" between the carbon and oxygen K-shell absorption edges, STXM allows imaging of naturally occurring absorption contrast differences within biological samples. The spectral information of soft X-ray XANES combined with the high spatial resolution of STXM near the carbon or the oxygen K-shell energy

(about 284 eV or about 533 eV) holds promise for discovering and studying chemical changes underlying a wide-range of biological phenomenon and disease states.

One challenge in the biological application of these techniques pertains to sample variability within and between individual preparations. Biological samples tend to be highly heterogeneous. Accordingly, biological applications of STXM and XANES require larger number of analyses in order to perform experiments with statistical significance. Currently, analysis of STXM data is typically completed using software packages such as the one created by the X-ray physics group of the Stony Brook University or the aXis2000 software provided by the McMaster University. Both packages are written in the interactive data language (IDL, Visual Information Solutions) and offer many powerful tools such as automatic stack alignment. Unfortunately, spectral data averaging of both packages is based on image areas selected manually by the user. Thus, neither are ideal for biological samples requiring analysis of many regions of interest and both

* Correspondence: tamas.haraszti@uni-heidelberg.de

¹ Biophysical Chemistry, University of Heidelberg, Im Neuenheimer Feld 253, 69120 Heidelberg, Germany

Full list of author information is available at the end of the article



are subject to potential user bias in selection of regions of interest.

Here, we present a new software package for analyzing STXM data based upon a simplistic analysis approach, and including a line-by-line absorption conversion tool. By automating the selection of regions of interest, the approach empowers analyses of large biological data sets. In developing this software, we analyzed melanosomes, the sub-cellular organelle responsible for melanin pigment production. As expected, the variability within data from melanosomes was found to be very high. However, the high number of data points analyzed through use of the STXMPy [Additional file 1] software package empowered a statistically meaningful analysis to be performed and was able to identify spectral differences between organelles isolated from mice with known genetic differences.

Implementation

All the programs described below were written in the interpreted language Python, and are based on three main libraries: the NetCDF library pycdf from Unidata, the numpy library [3] and the matplotlib plotting library [4]. For testing and development the ipython interface was used, which allows command history and history recording [5].

The hierarchy of algorithms is organized into three packages (Figure 1). 1) All fundamental image processing is done by the ImageP package. This package was originally developed to collect various functions related to image processing and contains several functions beyond what can be described here. 2) The sm package collects the basic wrapper object for the STXM images, stack loading and a normalization function, specific to the data

from the X1A microscope. 3) The xanesP package collects various tools (functions) for processing the image stack, such as absorption conversion and stack alignment. In addition, scripts were written to use the available functionality in batch mode processing of large data sets, including biological data. By default, the STXMPy package is currently configured for use with the X1A STXM located at the National Synchrotron Light Source at Brookhaven National Laboratory (Upton, USA), but by replacing the sm module it can be adapted for use with instruments located elsewhere.

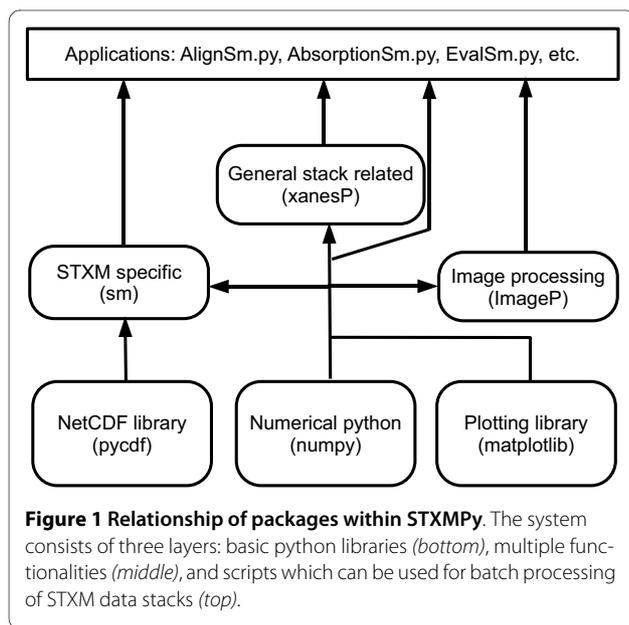
The basic sm object

In order for data to be interpreted, image data and related parameters must first be integrated. To achieve this, the basic sm object is located in the sm package and acts to load and store a STXM image and stores the energy of the image or the minimum and maximum energy if those are different from one another. The physical extents of the image are calculated and stored, along with all other parameters loaded from the NetCDF file. The snippet below presents an example of picking up an STXM image from a typical experiment recorded using the proportional counter of the X1A STXM. (The 'xlaos_20060412_0162.sm' file is available in the additional files of this paper [Additional file 2 and 3].)

```
from sm import sm
from matplotlib import pyplot as pl
stxmImage = sm('xlaos_20060412_0162.sm')
imagedata = stxmImage.get_image(-1)
print "Energy: %.2f eV" %stxmImage.E
pl.clf()
pl.imshow(imagedata, origin='lower',\
interpolation='nearest',\
extent=stxmImage.extent)
pl.show()
```

Reading and aligning stacks

Because STXM based XANES utilizes a stack of STXM images taken at various energies, the achievable spatial resolution can be limited by alignment imperfections. Stack alignment is therefore critical to successful implementation. Based on the xanesP and ImageP packages, the AlignSm.py script presents an application of the available parts, reading and aligning a list of STXM images. For simplicity the list of image files is stored in a single text file one filename per line. This list is read, the files imported, and the images are aligned using a convolution comparison. The program does not modify the sequence of the images, except to invert the sequence if the first energies are in a decreasing order. The script accepts several command line parameters, allowing the entry of arbitrary file name as the list of data files, selection of frames



to be used as reference image, and defining a region of interest for the reference image or stack.

For example, to select the text file containing the names of the data files to be 'lst.lst', align the stack relative to image number 50 in this list, and 'despike' the stack before aligning, use:

```
AlignSM.py -f lst.lst -i 50 -d
```

The resulting stack will subsequently be displayed on screen and stored on the hard disk together with the list of energies.

Alignment is achieved by the `AlignStack()` function from the `xanesP` package. This functionality uses the image convolution built into the `ImageP` package. This is a standard, fast Fourier transform (FFT) based convolution algorithm [1,2,6,7] with padding (either padding to the added size of the two images as a minimal padding, or alternatively to the next power of 2) [8]. Many biological applications, including our experiments with isolated organelles, will contain relatively few small absorbing features against a large bright background devoid of features. Therefore for the convolution the images are inverted in intensity by default. For images with a large area of absorbing features, inversion can be shut off to achieve a better alignment. If requested, the convolution image is also displayed, so that the user may visually check the result. Because images realigned by subpixel accuracy using a quadrilinear interpolation formula [1] also apply smoothing, it is advantageous to use pixel level alignment. To achieve this, shifted images were padded by zero for those pixels which were unknown due to the shift, and then the image stack was cropped to the smallest common area containing measurement data. In an ideal case, approximately 1 pixel of uncertainty is expected due to the round off error of the positioning, with greater uncertainty when low contrast or high image noise are present in the images.

Absorption calculation

Measuring the absorption of materials in X-ray spectroscopy is usually done by recording the transmitted intensity of X-rays through the sample and then through a blank reference sample providing the background I_0 data. This approach, i. e. recording reference measurements before/after the sample, is however not feasible when spectroscopy is derived from STXM image stacks, because recording such stacks takes several hours. Instead, one can envision measuring a reference point before/after each image. This latter scenario is equivalent to recording images with areas void of the sample in interest, providing simultaneous intensity measurement of the supporting substrate and avoiding a high accuracy repositioning of the sample between each step. This approach is generally accepted for STXM stack based spectroscopic analy-

sis and requires the identification the background areas reliably in order to access the background intensity data.

The `AbsorptionSm.py` script converts the aligned stack data to absorption values and performs normalization (using the `NormAStack` function from the `sm` package). The script offers several options, allowing choice between absorption conversion methods (line or image), thresholding values for these methods, and parameters to be passed to the normalization routine. The data are taken from the results of the `AlignSm.py` and stored again automatically. The script also displays the resulted stack of absorption images. The example below converts the intensity stack to absorption images using a histogram based image conversion and a relative threshold of 0.6. Intensities above 0.6 times the maximum intensity are used to calculate the bright background.

```
AbsorptionSm.py -a -t 0.6
```

The script has two implementations of the absorption calculation: an overall conversion and a line-by-line conversion.

In the standard GUI based STXM analysis the background intensity I_0 is either estimated from averaging the intensity on an image area selected by hand, or a user defined part of the intensity histogram [1]. Following this, the intensity (I) of manually selected image areas are used to calculate the absorption, according to equation (1).

$$\alpha = -\log\left(\frac{I}{I_0}\right) = -\log\left(\frac{I}{\langle I \rangle_0}\right) \quad (1)$$

In standard spectroscopy the logarithm is often used with a base of 10, but in order to be more consistent with Beer's law, the `AbsorptionSm.py` script uses the natural base. To automate the process and remove possible user bias, a simple, statistics based approach was used. The background intensity I_0 is calculated using the mean value of the higher intensity pixels from the image. These pixels are selected by thresholding the image. By default a dynamic thresholding is used based on Otsu's method and implemented as `graythresh()` in the `ImageP` package [9]. The function uses a histogram with a given number of bins (50 by default) to calculate the maximum of the interclass variance. Requesting verbose output will plot the interclass variance data to the screen.

```
from ImageP import graythresh
from matplotlib import pyplot as pl
th = graythresh(imagedata, bins = 50,
verbose = True)
print th
#to display which part of the image is
above threshold:
dispimage = imagedata - image-
data.min()
```

```
pl.figure() #open a new figure to display the image
pl.clf()
pl.imshow(imagedata*(dispimage >
th*dispimage.max()),\
origin='lower',\
interpolation='nearest',\
extent=stxmImage.extent)
pl.show()
```

The returned value is a floating point number, the relative position of the threshold between the minimum and maximum intensity of the image.

Absorption line-by-line

A complicating feature of X-ray absorption spectroscopy based on STXM data is that the intensity of the beam reaching the samples sometimes fluctuates during image recording. The observed fluctuation has two characteristic time scales. In one case fluctuation occurs within single scan lines. This happens only rarely, causing small local perturbations (a few pixel in a line) in the images that are negligible in scale compared to the data of the entire image. More frequently, regional fluctuations occur which change the overall intensity of larger areas in the images. The standard practice in such instances is to drop the image and the given energy value from the image stack. To avoid this loss of data, a line-by-line absorption calculation was introduced.

The algorithm is based on two decisions: 1) selecting lines which contain absorptive features (objects), and 2) selecting pixels to be used for the zero intensity calculation.

To select lines containing features a simple statistics based approach was utilized. The program first analyses each line by taking a simple running average filter, typically with 5 point averaging range. Comparing the standard deviation of the original data to the smoothed data, lines with features must have a smaller reduction in their standard deviation than those containing only fluctuations caused by the beam noise. Lines reaching a critical difference are treated as empty and I_0 is calculated as their mean average.

To select pixels for the zero intensity calculation, lines having features are treated to a second manipulation. The goal is to identify an intensity limit above which value the pixels can be taken for calculating the zero intensity. Because each line contains less than 200 data points, histogram based zero intensity determination is not practical. Two alternative approaches were initially tested. In the first approach, all intensities above the mean value of the line were taken into account. In the second approach, only intensities which were in the upper 20% of the intensity range of the data were taken into account. The latter

approach proved superior, though there is a potential for "spikes" (single pixels with very high intensity) to corrupt the results. Therefore, the software generates a warning if the data set used to calculate I_0 is less than 3 pixels long. The zero intensity (I_0) is calculated as the mean value of the selected pixels and then used to convert the whole line to absorption. For the sake of completeness the algorithm is also applicable with various polynomial fittings to reduce known drift within the image lines, as it is a common feature for data processing in atomic force microscopy images and was incorporated for possible use in future instruments.

Spectra normalization

Traditional STXM data analysis strongly relies on definition of the area of interest (ROI), and therefore the pixels from which the absorption or intensity spectra can be averaged. Averaging of absorption data can be done before [10] or after normalization.

During normalization, the linear aspect of the pre-edge spectra (below 282.0 eV for the carbon K-shell absorption edge) is frequently subtracted as a background in order to remove the contribution of other elements from the spectrum. The resultant spectra can subsequently be directly fitted with a set Gaussian or Voigt line profiles and a step function to follow the given absorption edge. Normalization can also be achieved either by using this height of the fitted absorption edge or by using the post absorption edge slope of the data [7,11-13].

Because biological samples are typically of non-uniform thickness and heterogeneous, the preferred methodology employed by the STXMPy package was to perform normalization before averaging. Because high beam noise above 300 eV made the post edge region inaccessible in our experiments with the X1A STXM [14], the broad absorption peak at 288 eV (aliphatic CH, carboxylic and amide groups [12]) was chosen as a representative of the organic content of the melanosomes. The peak value was evaluated using a second order polynomial fit in the 287 - 289 eV range. In order to avoid noise amplification with the normalization and background subtraction, each spectrum was first characterized by its mean value and variation. When the characteristic edge step was not found (the spectrum did not increase with at least 0.1 between the normalization range and the background, thus from 280 eV to approximately 288 eV), the image point was considered as a background pixel and its spectrum was left unaltered. The resultant normalized data stack was then saved for later use.

Spectral evaluation

The EvalSm.py program picks up the normalized spectral set resulted by the AbsorptionSm.py, and evaluates the

spectra based on comparison to a smooth fit (6th order polynomial). Spectra which have a fitting noise below threshold are averaged, the average and the standard deviation values are stored into a tabulated text file, and plots are generated.

Further tools

All the above described scripts, which process STXM stacks save their data back to the working folder where they were called from. It is desirable to compare and average these data sets in a flexible way, for which another script is available. The XAS-averager.py accepts a list of folder names in an input file, and creates the average and standard deviation of the spectral data from these folders. The script checks the energy of each data and eliminates errors caused by missing energy values.

Speed of execution

Interpreted programming languages may provide execution speeds inferior to optimized, compiled languages, such as C. To improve work flow performance, the data processing was split into various steps, storing the data between in predefined data files. In this manner, some of the slower steps, such as the alignment of the stack, did not have to be repeated when experimenting with various analysis parameters for absorption conversion or when spectral averaging is executed.

The current, pure Python code has three slow steps: 1) despiking takes approximately 2 minutes for 135 images, which is the slowest step. This could be improved by employing a less general algorithm, which would however limit the broad scope of the script. 2) Alignment, which is based on the already optimized built in FFT from the numpy package, takes 30 - 80 seconds running time. 3) The line-by-line absorption conversion takes approximately 45 seconds, which is acceptable taking into account the simultaneous image feedback provided.

All other processing steps are performed within a few seconds, resulting an acceptable execution speed on a modern personal computer (Pentium class, dual core machine with 2 GB RAM). Increasing the performance of the STXMPy package may be achieved by adding external, compiled alternatives of the various functions. This step may improve the running speed up to two times, depending on the complexity of the algorithm, and may present a part of future development.

Results and Discussion

Imaging of highly purified and freeze dried melanosomes was carried out as described in reference [14] at the carbon K-shell energies in a He jet atmosphere using the STXM of the X1A beamline of the National Synchrotron Light Source (NSLS), Brookhaven National Laboratories,

Upton, NY USA) [15]. Energies 280 - 310 eV were used with up to 0.1 eV resolution.

Comparing the newly introduced line absorption algorithm to its classical counterpart, we were interested in evaluating two aspects of its performance: 1) how the algorithm handles energy shifts within an image and 2) whether the algorithm provides similar results to those of the classical absorption conversion.

Energies near the carbon absorption edge yielded sufficient contrast differences to allow imaging of melanosomes (Figure 2). As expected, several experiments were disrupted by regional and local intensity fluctuations in the incoming light (Figure 2a). The corresponding absorption image with dynamic thresholding (Figure 2b) also shows the fluctuation inherent from the intensity image. However, with the use of the line-by-line absorption correction (Figure 2c), the influence of regional intensity fluctuations was eliminated. Thus, the algorithm was able to accommodate intensity shifts within data that previously would have been dropped from analysis (if it was noticed) or a source of data corruption (if it was unnoticed). Although small in-line fluctuations were not removed by the analysis, they presumably represent a

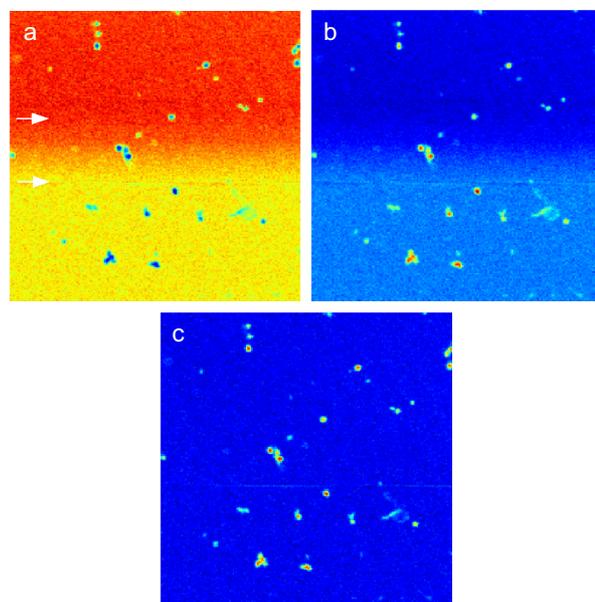


Figure 2 Image processing performed by STXMPy. (a) Unprocessed intensity image of isolated mouse melanosomes (image size approximately $20\ \mu\text{m} \times 20\ \mu\text{m}$). Note there are two fluctuations in beam intensity, one with regional influence (*upper arrow*) and one influencing only a single line (*bottom arrow*). All images have been pseudocolored so that blue represents low absorption and red high absorption. (b) Absorption image generated by AbsorptionSm.py script using the graythresh algorithm. (c) Correction for beam fluctuations performed with line-by-line absorption calculation. An intensity threshold of 80% was used to reject pixels with lower intensities and the mean value of the selected pixels was used as I_0 . Note that the influence of the regional fluctuation in beam intensity has been negated.

negligible amount of error. Comparing the absorption data of an average image between the line absorption and the traditional absorption conversion using automated thresholding, we found less than 2% relative deviation (data not shown) within an image. Combined, these results indicate that the algorithm was able to accommodate intensity shifts within an experiment and yielded results similar to those of the classical absorption conversion.

Two characteristic output of the STXMPy system are a sum image (typically generated in the 283-289 eV energy range) and an image of excluded pixels (Figure 3a,b). The former indicates the general region of interest for the spectral analysis and the latter is characteristic to the alignment accuracy. To test the influence of various thresholding and absorption algorithms on the average

absorption spectrum, individual data sets were repeatedly subjected to different analyses. Comparisons were first made for data collected with uniform beam intensities (Figure 3c) [Additional file 2]. In most instances, changes in thresholding resulted in negligible changes in the average spectra. The sole exception being the instance of an intensity threshold of 0.8, which influenced the estimation of the background intensity, I_0 (data not shown). In analyzing data collected with significant fluctuations in regional beam intensity (Figure 3d) [Additional file 3], the line-by-line method corrected the error presented by the classical image based conversion, which otherwise would have resulted in up to 100% higher absorption at the energy of 285.8 eV.

These results indicate that the overall output, the obtained XANES spectrum, is robust against various

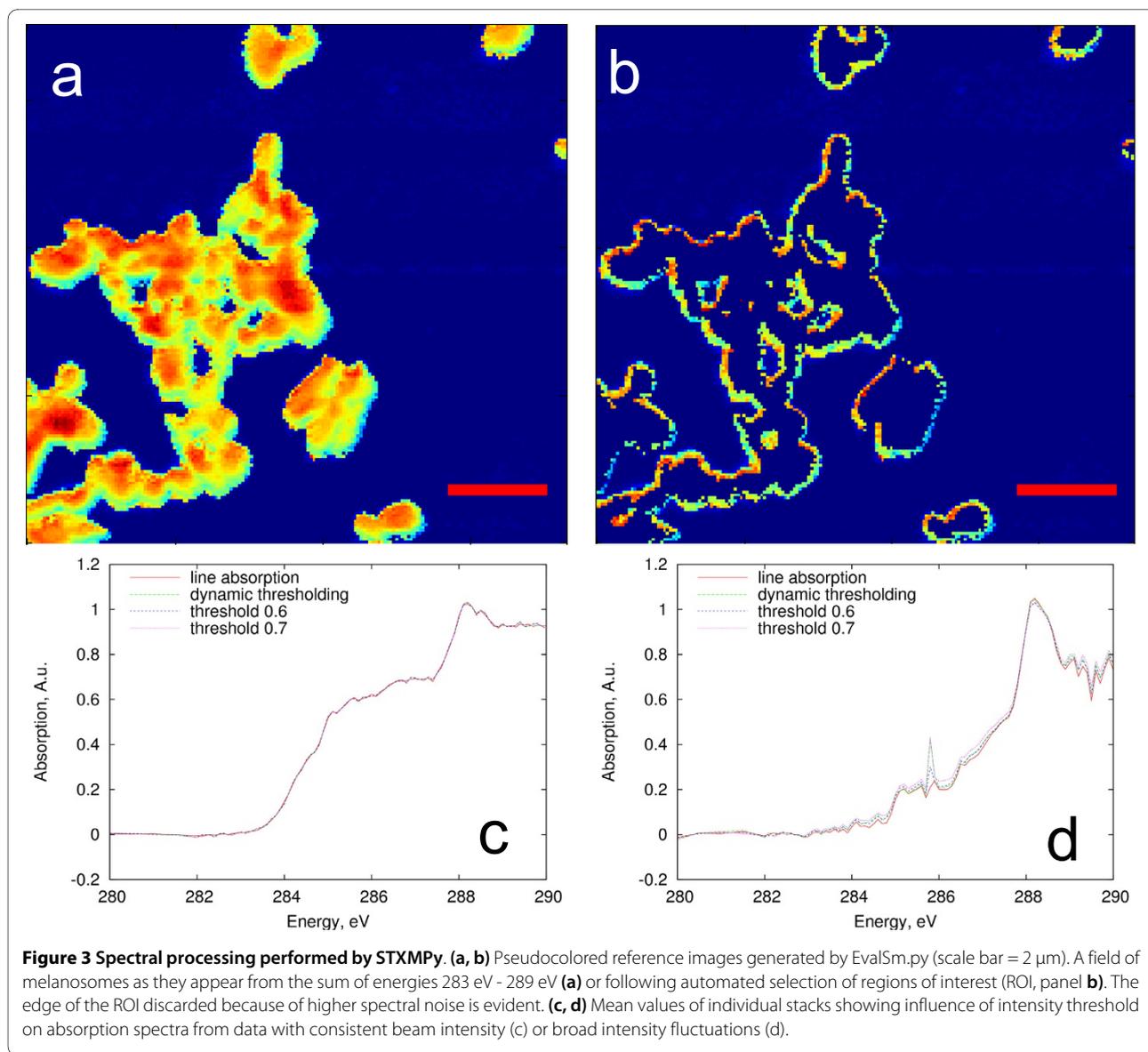


Figure 3 Spectral processing performed by STXMPy. (a, b) Pseudocolored reference images generated by EvalSm.py (scale bar = 2 μm). A field of melanosomes as they appear from the sum of energies 283 eV - 289 eV (a) or following automated selection of regions of interest (ROI, panel b). The edge of the ROI discarded because of higher spectral noise is evident. (c, d) Mean values of individual stacks showing influence of intensity threshold on absorption spectra from data with consistent beam intensity (c) or broad intensity fluctuations (d).

thresholding methods. Furthermore, the absorption conversion line-by-line performs similarly to the classical, whole image conversion method (compare dynamic thresholding to line-absorption in Figure 3c,d).

An important question relevant to data interpretation pertains to significance of changes in absorption spectra between data sets. In general, manual data processing only considers average values of data and largely ignores variation. As expected, our biological data sets exhibited substantial deviation within individual image stacks (Figure 4). This level of deviation was consistently observed in all of our biological samples. By allowing this variability to be easily measured without potential user bias in selecting regions of interest, STXMPy thus empowers analysis allowing biological comparisons and tests of whether defined genes or genetic contexts significantly alter composition of cells or organelles. In applying STXMPy to analysis of biological samples, it will be important to consider statistical ramifications of experimental design. Because a typical STXM field will contain several hundred or thousand pixels of data points that will all be included in analysis, only a few percent relative change of the mean value would likely be significant. However, until appropriate procedures of multiple tests employed based on relevant biological hypotheses, there is also an increased opportunity for generating misleading conclusions. In the meantime, we suggest that STXMPy be used to generate averages from individual fields of sample, with multiple fields from biological and technical replicates analyzed to estimate variability. Thus, by measuring the "average of averages" from data sets with limited numbers of fields ($n = 3 - 10$), a reasonable degree of reproducibility could be evaluated with fewer statistical comparisons and without requiring overly large amounts of beamtime.

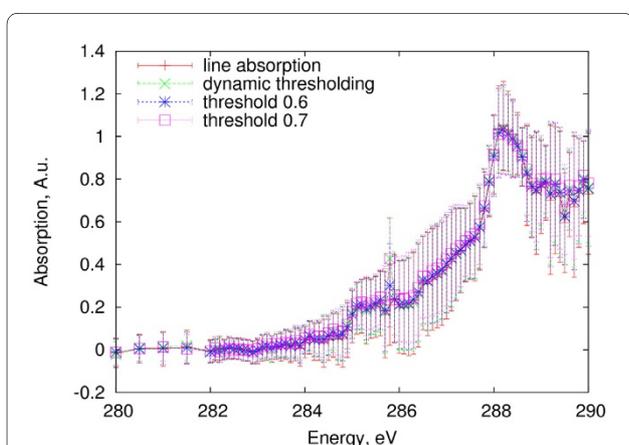


Figure 4 Statistical considerations. Error level of absorption spectra for data presented in Figure 3d. Mean \pm standard deviation.

Conclusion

Here, we have presented a semiautomated data processing method for extracting XANES data from STXM data sets. The analysis allowed average information of samples to be analyzed and statistically compared. Separate programs were written in the Python programming language to process and visualize data sets both in interactive and batch processing. The system includes a line-by-line conversion of images to absorption data and outputs images that can be evaluated as data and quality controls. This algorithm is envisioned to be generally useful in XANES analysis of biological samples and any STXM system where external noise source contributes variations in the intensity of the incoming beam or where automated region of interest selection would be desired.

Availability and requirements

Project name

STXMPy: image manipulation for STXM images

Project home page

<http://www.rzuser.uni-heidelberg.de/~ge6/Programing/STXMPy.html>

Operating system

Linux, but all implementations are platform independent

Programming language

Python

Other requirements

Python 2.4 or higher (not tested with Python 3)

Python imaging library (not used in this work, but required by some parts of ImageP)

Numerical Python: numpy for array handling and math
Matlab like plotting library: matplotlib for data visualization

NetCDF wrapper from Unidata: pycdf for accessing the sm file format from the X1A beam line

License

GNU LGPL 3

Additional material

Additional file 1 The STXMPy program suite. This file contains the sm and xanesP python libraries and various data processing scripts, described in this paper. A Readme.txt file is also included, describing the installation and some testing examples.

Additional file 2 Example data set 1. STXM images of freeze dried melanosomes, recorded at the NLSL X1A beam line. Provided to test the program installation and functionality, presenting an example of uniform illumination in Figure 3a-c.

Additional file 3 Example data set 2. Another STXM image set, this time with noisy data. Provided to demonstrate the line absorption algorithm. Further usage is described in the STXMPy.zip, in a Readme.txt file. This data presents intensity fluctuations, the source data of Figure 3d and Figure 4.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MGA provided the melanosome samples and oversaw biological aspects of the project. TH performed the STXM experiments, software development and data processing. MG supervised the biophysical aspects of the project. All authors contributed to the writing of this manuscript.

Acknowledgements

The authors would like to thank to the National Synchrotron Light Source for the beam time and Dr. Sue Wirick for support at the beamline and many helpful discussions. We thank Drs. Colleen Trantow and Sachiko Iwashita for preparing melanosomes and Randy Nessler of the University of Iowa Central Microscopy Research Facility for technical assistance with freeze-drying of melanosome samples. The Central Microscopy Research Facility is supported by the Office of the Vice President for Research and Grant Number CA086862-10 to the Holden Comprehensive Cancer Center from the National Institutes of Health and National Cancer Institute. Biological aspects of this project were supported by National Institutes of Health grants to MGA, including Grant Number AR055697 from the National Institute of Arthritis and Musculoskeletal and Skin Diseases and EY017673 from the National Eye Institute. STXM imaging and associated travel were supported by University of Maine, the University of Heidelberg and the German Federal Ministry of Education and Research (BMBF) (project number 05KS7VH1) to MG and TH.

Author Details

¹Biophysical Chemistry, University of Heidelberg, Im Neuenheimer Feld 253, 69120 Heidelberg, Germany, ²Department of New Materials and Biosystems, Max-Planck Institute for Metals Research, Heisenberg str. 3, 70569 Stuttgart, Germany, ³Applied Physical Chemistry, University of Heidelberg, Im Neuenheimer Feld 253, 69120 Heidelberg, Germany, ⁴Institute for Functional Interfaces (IFG) at the Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany, ⁵Department of Molecular Physiology and Biophysics, The University of Iowa, Iowa City, Iowa 52242, USA and ⁶Ophthalmology and Visual Sciences, The University of Iowa, Iowa City, Iowa 52242, USA

Received: 29 January 2010 Accepted: 4 June 2010

Published: 4 June 2010

References

1. Jacobsen C, Wirick S, Flynn G, Zimba C: **Soft X-ray spectroscopy from image sequences with sub-100 nm spatial resolution.** *J Microsc* 2000, **197**:173-184.
2. Kirz J, Jacobsen C, Howells M: **Soft X-ray microscopes and their biological applications.** *Q Rev Biophys* 1995, **28**(1):33-103.
3. **Numerical Python** [<http://www.numpy.org>]
4. **Matplotlib** [<http://matplotlib.sourceforge.net/>]
5. **IPython** [<http://ipython.scipy.org/>]
6. Smith SW: **The Scientist & Engineer's Guide to Digital Signal Processing.** California Technical Publishing: San Diego, CA; 1997.
7. Lerotic M, Jacobsen C, Schäfer T, Vogt S: **Cluster analysis of soft X-ray spectromicroscopy data.** *Ultramicroscopy* 2004, **100**(1-2):35-57.
8. Press WH, Flannery BP, Teukolsky SA, Vetterling WT: **Numerical Recipes in C++ (Second Edition).** New York: Cambridge University Press; 2002.
9. Otsu N: **A Threshold Selection Method from Gray-Level Histograms.** *IEEE Trans Sys Man Cyber* 1979, **9**(1):62-66.
10. **Axis 2000 software** [<http://unicorn.mcmaster.ca/aXis2000.html>]
11. Petterson LGM, Agren H, Schürmann BL, Lippitz A, Unger WES: **Assembly and Decomposition of Building Blocks to Analyze Polymer NEXAFS Spectra.** *Int J Quant Chem* 1997, **63**(3):749-765.
12. Benzerara K, Yoon TH, Tyliszczak T, Constantz B, Spormann AM, Brown GE Jr: **Scanning transmission X-ray microscopy study of microbial calcification.** *Geobiology* 2004, **2**:249-259.
13. Hitchcock AP, Morin C, Heng YM, Cornelius RM, Brash JL: **Towards practical soft X-ray spectromicroscopy of biomaterials.** *J Biomater Sci Polym Ed* 2002, **13**(8):919-937.
14. Anderson MG, Haraszti T, Petersen GE, Wirick S, Jacobsen C, John SWM, Grunze M: **Scanning transmission X-ray microscopic analysis of purified melanosomes of the mouse iris.** *Micron* 2006, **37**(8):689-698.

15. Winn B, Ade H, Buckley C, Feser M, Howells M, Hulbert S, Jacobsen C, Kaznacheyev K, Kirz J, Osanna A, Maser J, McNulty I, Miao J, Oversluisen T, Spector S, Sullivan B, Wang Y, Wirick S, Zhang H: **Illumination for coherent soft X-ray applications: the new X1A beamline at the NSLS.** *J Synchrotron Radiat* 2000, **7**(Pt 6):395-404.

doi: 10.1186/1752-153X-4-11

Cite this article as: Haraszti et al., STXMPy: a new software package for automated region of interest selection and statistical analysis of XANES data *Chemistry Central Journal* 2010, **4**:11

Publish with **ChemistryCentral** and every scientist can read your work free of charge

"Open access provides opportunities to our colleagues in other parts of the globe, by allowing anyone to view the content free of charge."

W. Jeffery Hurst, The Hershey Company.

- available free of charge to the entire scientific community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours — you keep the copyright

Submit your manuscript here:
<http://www.chemistrycentral.com/manuscript/>


ChemistryCentral