
Sakura Documentation

Release 0.3.1

Gary Ruben

August 10, 2015

1	Usage	3
1.1	On startup	3
1.2	On file load	4
1.3	Switching Absorption Edge Spectrum display mode	4
1.4	Mapping mode dataset spectrum view	5
2	Processing	7
3	Installation	9
3.1	Python package dependencies	9
3.2	Example 1. Windows installation using Enthought EPDFree (recommended)	9
3.3	Example 2. Windows installation using Python(x,y) (optional)	10
3.4	Example 3. Linux installation using Enthought Python Distribution (recommended)	10
3.5	Example 4. Linux installation using synaptic (experienced)	10
3.6	Example 5. Mac OSX installation (recommended)	11
4	About Sakura	13
5	Indices and tables	15



Sakura¹ is a data explorer for data generated at the XAS beamline at the Australian Synchrotron.

- Reads step mode and mapping mode data.
- Exports columnar ASCII.
- Cross-platform. Runs on Windows/Mac OSX/Linux
- Released under the Modified BSD-license

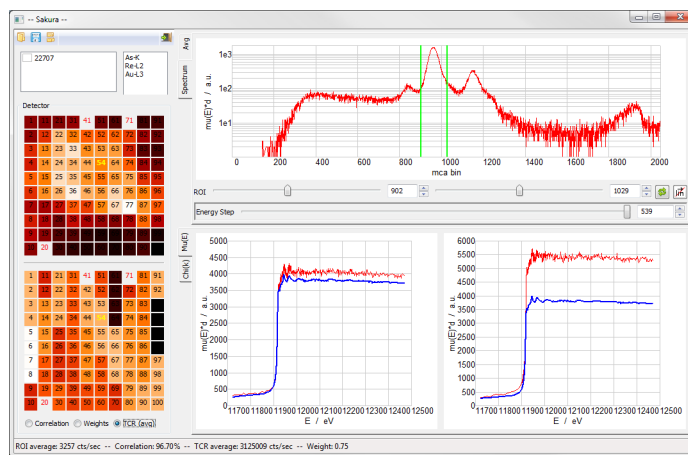


Fig. 1: Sakura screenshot.

¹ Sakura.

This page describes the usage of Sakura.

1.1 On startup

When started, Sakura presents the following empty interface with red overlays identifying the following parts of the user interface:

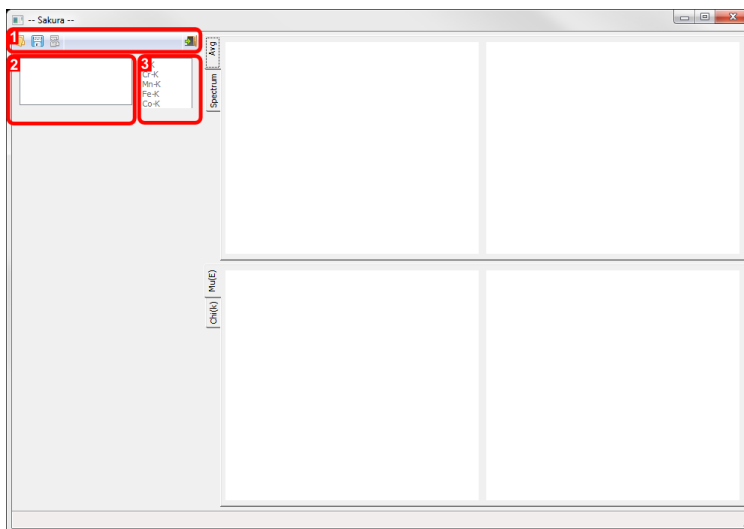


Fig. 1.1: Sakura (Windows version) on startup identifying parts of the user interface: 1. Toolbar, 2. File list, 3. Edge list.

1. **Toolbar.** From left-to-right, the toolbar buttons allow you to 1a) Open a scan dataset, 1b) Export reduced data as ASCII, 1c) Unload all loaded datasets, 1d) Exit the program.

1a) Open a scan dataset.

Sakura allows processing of step-mode and mapping-mode datasets. In the case of step-mode data, all data is stored in an individual .mda file. In the case of mapping-mode data, a .mda file contains experiment metadata process variables (PVs), whereas the raw spectral data is stored in accompanying netCDF files. In both cases, the data is loaded by opening the appropriate .mda file in the open file dialog.

1b) Export reduced data as ASCII.

When clicked, any processed data in any checked datasets in the File list box are written to ASCII files with the same name as the .mda file, but with a .asc filename extension.

1c) Unload all loaded datasets.

All data is unloaded. Equivalent to exiting and restarting Sakura.

1d) Exit the program.

Quits Sakura. The same as trying to close the main window.

2. **File list** Currently loaded files appear here alongside a checkbox that is used to choose datasets for export.

Step-mode and mapping-mode datasets cannot be simultaneously loaded.

3. **Edge list** Candidate edges are listed here. Selecting one ???

1.2 On file load

Once a file is loaded, the following display areas become active:

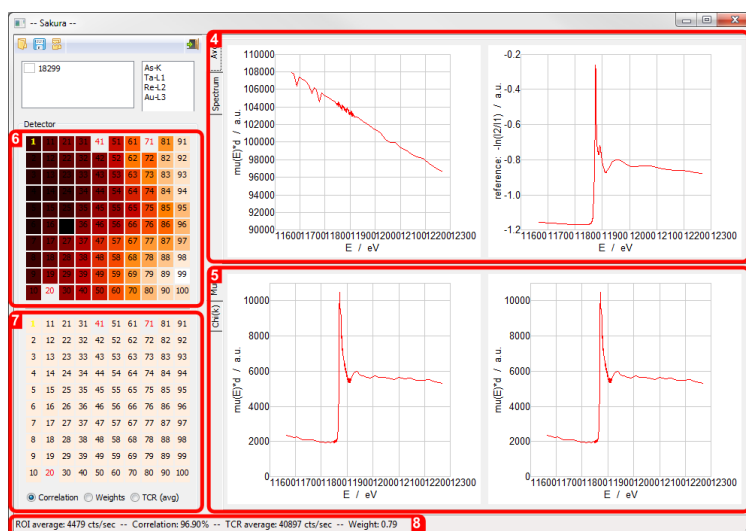


Fig. 1.2: Sakura screen areas activated by file load: 4. Average Absorption Edge Spectrum and ??? plots, 5. Absorption Edge Spectrum [$\mu(E)*d$ mode], 6. 100-element detector selection widget, 7. 100-element detector selection widget/statistics map, 8. Numerical statistics report.

4. **Average Absorption Edge Spectrum and ??? plots** Blah ???

5. **Absorption Edge Spectrum [$\mu(E)*d$ mode]** ???

6. 100-element detector selection widget

7. 100-element detector selection widget/statistics map

8. Numerical statistics report

1.3 Switching Absorption Edge Spectrum display mode

9. Absorption Edge Spectrum [$\chi(k)$ mode]

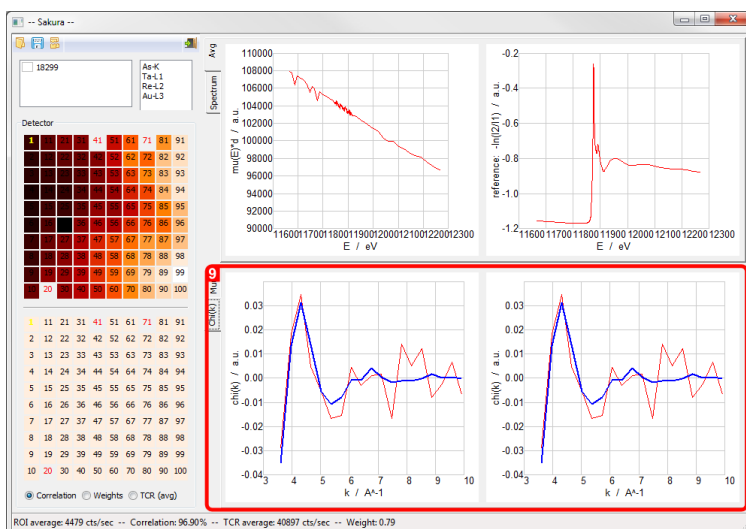


Fig. 1.3: The reciprocal Absorption Edge Spectrum display is accessed via the Chi(k) tab.

1.4 Mapping mode dataset spectrum view

10. Multichannel analyser plot with ROI selection

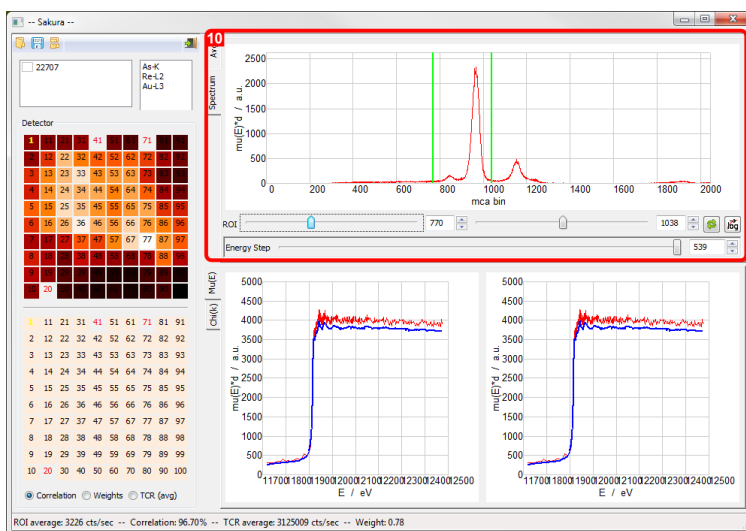


Fig. 1.4: For netCDF-based mapping-mode datasets, clicking the Spectrum tab in the upper display accesses the MCA spectrum display.

- The Spectrum display allows the ROI limits to be set.
- To lock the Spectrum display to show a particular detector element, left-click the mouse on the detector element of interest. The element number changes to Bold Yellow. Repeat this action to unlock the element.
- After setting the ROI limits using the ROI setters, clicking the Reprocess button will recompute the Absorption edge displays.

TBC

Processing

This page describes the processing that Sakura performs.

TBD

Installation

For MS Windows users, a standalone installer is available from the [main site](#). Download and install this to run Sakura.

Linux and Mac users or Windows users wishing to install from source must install ‘**Python 2.7.x**<http://python.org>’ and the packages listed below. Sakura is written in Python (2.7 at the time of writing). The easiest way to satisfy the Python module dependencies is to install one of the [Enthought Canopy Distributions](#). The older Enthought EPDFree distribution (also available from the [main site](#)) also contains all the modules on which Sakura depends. A typical user can just install one of these, then download and unpack the Sakura source code and run the sakura.py file.

Some users might prefer alternatives to EPDFree or Canopy. Other popular Python distributions are [Python\(x,y\)](#) and ActiveState’s [ActivePython](#), both of which will require installation of at least some of the following dependencies:

3.1 Python package dependencies

Sakura depends on the following Python modules being installed in the Python environment

- [Python 2.7.x](#)
- [numpy](#)
- [scipy](#)
- [wxPython 2.8](#)

Those wishing to build the documentation from source will also need [Sphinx](#). For those familiar with installing Python packages, the dependencies can be found in the [central repository of Python modules](#). For Microsoft Windows users, Christoph Gohlke (C.G.) maintains a useful [repository of Windows installers](#) containing the required packages. Linux users can typically find the dependencies using their package manager (e.g. synaptic or yum). Mac users should visit the individual sites linked above for instructions. In all cases, be careful to choose the package versions compiled for Python 2.7.x and choose either the 32bit or 64bit package versions to match the version of Python you installed.

3.2 Example 1. Windows installation using Enthought EPDFree (recommended)

1. Visit the [Sakura download page](#) and follow the instructions to obtain the EPDFree Python installer and the Sakura file archive. You may wish to visit the [Enthought website](#), directly, and choose one of Enthought’s Canopy Python distributions. However, these include wxPython 2.9, which may cause incompatibility problems with Sakura.

2. Run the `epd_free-*.msi` installer to install EPD Free (or install using the Canopy installer).
3. Verify that Python is running correctly. e.g. for Windows 7, click on Start Menu|All Programs|Accessories|Command Prompt. At the `>` prompt type `python -c "print 'hello world'"` noting single and double quotes. Verify that `hello world` is displayed.
4. Download the Sakura source code bundle and uncompress it.
5. Double-click `sakura.py` to run Sakura.

3.3 Example 2. Windows installation using Python(x,y) (optional)

1. Visit the [Python\(x,y\) downloads page](#) and install a distribution.
2. Verify that Python is running correctly (see *Example 1. Windows installation using Enthought EPDFree (recommended)*).
3. Visit the [Sakura download page](#) and follow the instructions to obtain the Sakura setup file.
4. Download the Sakura source code bundle and uncompress it.
5. Double-click `sakura.py` to run Sakura.

3.4 Example 3. Linux installation using Enthought Python Distribution (recommended)

1. Visit the [Sakura download page](#) and follow the instructions to obtain the EPD Free Python installer and the Sakura .zip package for Linux. You may wish to visit the [Enthought website](#), directly, and choose one of Enthought's other Python distributions. Note, [EPD Free](#) satisfies the dependencies listed above.
2. Run the `epd_free-*.sh` shell script to install EPD Free.
3. Verify that Python is running correctly. e.g. for Ubuntu, open a terminal. At the `$` prompt type `python -c "print 'hello world'"` noting single and double quotes. Verify that `hello world` is displayed.
4. The main Sakura application file is `sakura.py` in the directory into which Sakura was unpacked.
5. Start Sakura by running `python ./sakura.py`.

3.5 Example 4. Linux installation using synaptic (experienced)

Note: untested.

This description is for Ubuntu Linux. yum packaged names in Fedora Linux flavours should have similar names.

1. First, verify that Python2.7 is running correctly. e.g. for Ubuntu, open a terminal. At the `$` prompt type `python -c "import sys; print sys.version"`. Verify that a string displays identifying a 2.7 branch version of Python.
2. Using synaptic or `apt-get install <package>` install the following packages: `python-numpy`, `python-scipy`, `python-wxgtk2.8`
3. Visit the [Sakura download page](#) and follow the instructions to obtain the package.
4. The main application file is `sakura.py` in the directory into which Sakura was unpacked.
5. Start Sakura by running `python ./sakura.py`.

3.6 Example 5. Mac OSX installation (recommended)

Note: untested.

1. Visit the [Sakura download page](#) and follow the instructions to obtain the EPD Free Python installer and the Sakura .zip package for Mac OSX. You may wish to visit the [Enthought website](#), directly, and choose one of Enthought's other Python distributions. Note, [EPD Free](#) satisfies the dependencies listed above.
2. Run the `epd_free-*.dmg` installer to install EPD Free.
3. Move the .zip package to the Applications folder.
4. Double click the Application .zip package to unpack it the first time.
5. At a command prompt, start Sakura by running `python ./sakura.py`.

About Sakura

Please send bug reports and suggestions to sakura_at_synchrotron.org.au

Software authors:

Peter Kappen peter_dot_kappen_at_synchrotron.org.au, Australian Synchrotron

Gary Ruben gary_dot_ruben_at_synchrotron.org.au, Victorian eResearch Strategic Initiative (VeRSI)

Software home:

<http://www.synchrotron.org.au/sakura>

Software source:

<http://github.com/AustralianSynchrotron/sakura>

Gary Ruben acknowledges funding for Sakura development from NeCTAR.

Recognition of NeCTAR funding:

The [Australian Synchrotron](#) is proud to be in partnership with the National eResearch Collaboration Tools and Resources ([NeCTAR](#)) project to develop eResearch Tools for the synchrotron research community. This will enable our scientific users to have instant access to the results of data during the course of their experiment which will facilitate better decision making and also provide the opportunity for ongoing data analysis via remote access.



Indices and tables

- [genindex](#)
 - [modindex](#)
 - [search](#)
-



nectar



VeRSI