

---

# **Xanespy Documentation**

***Release 0.1***

**Mark Wolf**

**Jan 08, 2017**



## CONTENTS:

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	X-Ray Absorbance Basics . . . . .	3
1.2	Example Workflow . . . . .	3
<b>2</b>	<b>Importing Data into Xanespy</b>	<b>5</b>
2.1	TXM from 8-BM (APS) and 6-2c (SSRL) . . . . .	5
2.2	Ptychography from 5.3.2.1 (ALS) . . . . .	5
<b>3</b>	<b>Analyzing the Data</b>	<b>7</b>
3.1	Frame Alignment . . . . .	7
3.2	Subtracting Surroundings . . . . .	7
3.3	Spectrum Fitting - K-Edge . . . . .	7
3.4	Spectrum Fitting - L-Edge . . . . .	7
<b>4</b>	<b>Visualization of Results</b>	<b>9</b>
4.1	Plotting . . . . .	9
4.2	Interactive Viewer (Qt) . . . . .	9
<b>5</b>	<b>Accessing the Data Directly</b>	<b>11</b>
<b>6</b>	<b>Indices and tables</b>	<b>13</b>



Python library for analyzing **X-Ray absorbance spectroscopy** data.

**Warning:** This documentation is still under development. The code it represents could change at any time.



## INTRODUCTION

Xanespy is a toolkit for interacting with X-ray microscopy data, most likely collected at a synchrotron beamline. By collecting a set of frames at multiple X-ray energies, spectral maps are reconstructed to provide chemical insight. Multiple framesets can be collected sequentially as part of an *operando* experiment and analyzed simultaneously in python. Slow operations take advantage of multiple cores when available.

### 1.1 X-Ray Absorbance Basics

### 1.2 Example Workflow

A typical procedure for interacting with microscope frame-sets involves the following parts:

- Import the raw data
- Apply corrections and align the images
- Calculate some metric and create maps of it
- Visualize the maps, static or interactively.

Example for a single frameset across an X-ray absorbance edge:

```
import xanespy

# Example for importing from SSRL beamline 6-2c
xanespy.import_ssrl_frameset('<data_dir>', hdf_filename='imported_data.h5')

# Load a pre-defined XAS edge or create your own subclass xanespy.Edge
edge = xanespy.k_edges['Ni_NCA']
# Now load the newly created HDF5 file and the X-ray absorbance edge
fs = xanespy.XanesFrameset(filename='imported_data.h5', edge=edge)

# Perform automatic frame alignment
fs.align_frames(passes=5)
# Fit the absorbance spectra and extract the edge position (SLOW!)
fs.fit_spectra()

# Inspect the result with the built-in Qt5 GUI
fs.qt_viewer()
```





## IMPORTING DATA INTO XANESPY

### 2.1 TXM from 8-BM (APS) and 6-2c (SSRL)

### 2.2 Ptychography from 5.3.2.1 (ALS)



## ANALYZING THE DATA

### 3.1 Frame Alignment

### 3.2 Subtracting Surroundings

### 3.3 Spectrum Fitting - K-Edge

### 3.4 Spectrum Fitting - L-Edge



## VISUALIZATION OF RESULTS

### 4.1 Plotting

### 4.2 Interactive Viewer (Qt)



## ACCESSING THE DATA DIRECTLY

While the `XanesFrameset` class has methods for common tasks, sometimes it is necessary to access the data directly, as either numpy arrays or h5py datasets. The `xanes_frameset` has a `store()` method that returns an interface (`TXMStore`) to the underlying HDF5 file.

**Warning:** The `TXMStore` created by `xanes_frameset.store()` is attached to an open HDF5 file. It is strongly recommended to use the `with` statement described below. Otherwise make sure to call the store's `close()` method in a `try...except` block. File corruption is possible if not opened in this manner.

Call the following to get access to the associated datasets. Properties of the interface will return an HDF5 dataset in most cases.:

```
import xanespy as xp
frameset = xp.XanesFrameset(...)

# Open the TXMStore interface
with frameset.store() as store:
    # For example, the images are in (timestep, energy, row, column) order
    assert store.absorbances.shape == (10, 62, 1024, 1024)
    # Energies are in (timestep, energy) order
    assert store.energies.shape == (10, 62)
```





## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`