PEREX - ECXAS
Library Documentation

Version 5.1.0

Lucía PÉREZ RAMÍREZ

September 4, 2023

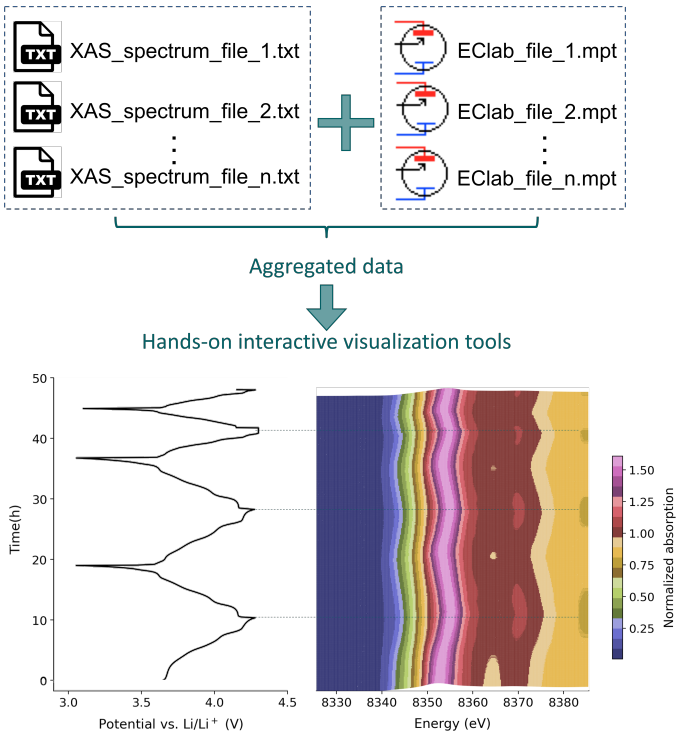
Contents

1 Introduction

In the context of the BIG-MAP (Battery Interface Genome Materials Accelerated Platform[1]) European project, a need for semi-automated data processing in battery characterisation was identified, specially for experiments concerning the use of large-scale facilities. The general efforts in synchrotrons and neutron sources to improve time and spectral resolution, along with the expanded use of multi-modal approaches (involving several techniques in parallel) has been resulting in ever-increasing volumes of data, impossible to be handled manually.

A simple example is the coupling of x-ray absorption spectroscopy (XAS) with electrochemical characterisation (EC). This is possible at ROCK beamline, in Synchrotron SOLEIL. In the current state, the data arrive out of sync at ROCK (for example, time series of absorption spectroscopy and electrochemical potential during a charge-discharge cycle of a battery), which makes their handling and representation complex. This has prompted the need of a tool that aggregates asynchronous data and allows simple visual representations.

ECXAS is a data aggregation tool for battery study in ROCK beamline, at Synchrotron SOLEIL.



[1] www.big-map.eu

2 Install

To install the ECXAS library simply open a terminal and run the command:

```
pip install git+https://github.com/GhostDeini/perex
```

3 The main functions

3.1 ecxas.getECdf

```
getECdf(*args, with_settings=0, acTime=1)
```

Function to get a pandas dataframe from one or several EC Lab files (must have an .mpt extension).

- *args: String(s) containing the complete path(s) of the file(s). Must be separated by commas.
- with_settings: Default set to 0. Set to 1 if you want an additional dataframe with the settings that are written in the file header.
- acTime: Usually, the EC Lab file has a line indicating when the acquisition started. If not, you must set it as a string with the format 'm/d/Y H:M:S.f'
- return: Returns a pandas dataframe with the data from EC Lab file(s).

3.2 ecxas.getXASdf

```
getXASdf(XAS_folder, filters=[], recursive=True)
```

Function to get a pandas dataframe with a list of the XAS files in a folder (files must have a .txt extension).

- XAS_folder: String with the complete path of the folder where the XAS txt files are.
- filters: A list of strings containing filter words to select the XAS filenames.
- recursive: Default set to True. This parameter sets if the function should find the files recursively (looking into folders and subfolders).
- return: Returns a pandas dataframe with the XAS filelist information.

3.3 ecxas.getRAMANdf

```
getRAMANdf(csv_path, filename_col='filename',
↪  datetime_col='acquisition_date_time')
```

Function to get a pandas dataframe with Raman data from files listed in the .csv reference file. The files in the list must be located in the same folder and must have a .txt extension.

- csv_path: String with the complete path of the .csv reference file.
- filename_col: String with the complete path of the .csv reference file.
- datetime_col: String with the complete path of the .csv reference file.
- return: Returns a pandas dataframe with the Raman filelist information (the original csv tanle + the arrays from the Raman data).

3.4 ecxas.merge_dfs

```
merge_dfs(data1_df, data2_df,
↪  timecol_data1='acquisition_datetime', interp=False, tol='10
↪  min')
```

Function to merge EC and XAS-files dataframes or RAMAN and XAS-files dataframes based on datetime. Reference time comes from XAS experiment.

- data1_df: Pandas dataframe with the data from the EC Lab file or from a list of Raman spectra.
- data2_df: Pandas dataframe with the data from several XAS files.
- timecol_data1: Default set to 'acquisition_datetime' column.
- interp: Default set to False. If set to True, it interpolates all the values from the EC or the Raman dataframe to make a perfect time match.
- tol: Time tolerance for merging both dataframes if interp set to False. Default set to 10 min.
- return: Returns a merged dataframe with aggregated data from simultaneous experiments (EC+XAS, Raman+XAS).

4 The 'plot' functions

4.1 ecxas.plot.UI_vs_t

```
UI_vs_t(df, width=15, height=5)
```

Function to plot a potential/current vs time graph.

- df: Pandas dataframe with the data from the EC Lab file.
- width: Width of the graph.
- height: Height of the graph.
- return: Plot.

4.2 ecxas.plot.U_vs_capacity

```
U_vs_capacity(df, nb_cycle='all', width=8, height=5, mass=1000)
```

Function to plot a capacity vs potential graph.

- df: Pandas dataframe with the data from the EC Lab file.
- nb_cycle: List of the cycles you want to plot. Or number of the cycle you want to plot. Plots all by default.
- width: Width of the graph.
- height: Height of the graph.
- mass: Mass of active material in mg.
- return: Plot.

4.3 ecxas.plot.dQdU_vs_U

```
dQdU_vs_U(df, nb_cycle='all', reduce_by=1, boxcar=1, savgol=(1,
↪  0), colormap='plasma', width=10, height=6, dotsize=10,
↪  alpha=1, mass=1000)
```

Function to plot a capacity vs potential graph.

- df: Pandas dataframe with the data from the EC Lab file.

- nb_cycle: List of the cycles you want to plot. Or number of the cycle you want to plot. Plots all by default.
- reduce_by: Factor by which you want to reduce the number of points on your dataframe.
- boxcar: Factor indicating the size of the moving window of a moving average filter
- savgol: Tuple (x,y) with the parameters of a Savitzky-Golay filter.
- colormap: Name of the colormap you want to use for the plot. Default is set to 'plasma'. More options here.
- width: Width of the graph.
- height: Height of the graph.
- dotsize: Size of the dot of the scatter plot. Default set to 10.
- alpha: Opacity of the points. Default set to 1.
- mass: Mass of active material in mg.
- return: Plot.

4.4 ecxas.plot.all_XAS

```
all_XAS(df, nb_cycle='all', edge_intensity='inflection',
↪  intensity_col='', colormap='viridis', pre=20, post=40,
↪  width=7, height=4)
```

Function to plot all XAS spectra on the same graph.

- return: Plot.

4.5 ecxas.plot.XAS_vs_t_2D

```
XAS_vs_t_2D(df, nb_cycle='all', edge_intensity='inflection',
↪  intensity_col='', abstime_col='', colormap='turbo', width=7,
↪  height=4, plot_range=None)
```

Function to plot a 2D intensity graph of all the XAS spectra over time.

- df: Pandas dataframe with the data from the XAS spectra (either XAS alone or EC+XAS dataframe).

- edge_intensity: Intensity value (J) to get the edge energy value. Default set to inflection point.
- intensity_col: Name of the column with the intensity spectra.
- colormap: Name of the colormap you want to use for the plot. Default is set to 'tab20b'. More options here.
- width: Width of the graph.
- height: Height of the graph.
- plot_range: List [x,y] containing the energy range of the plot.
- return: Plot.

4.6 ecxas.plot.XAS_vs_t_3D

```
XAS_vs_t_3D(df, nb_cycle='all', edge_intensity='inflection',
↪  intensity_col='', abstime_col='', arrows_cycles=True,
↪  colormap='viridis', colormap2='plasma', width=12, height=10,
↪  alpha=0.5, plot_range=None)
```

Function to plot a 2D intensity graph of all the XAS spectra over time.

- df: Pandas dataframe with the data from the XAS spectra (either XAS alone or EC+XAS dataframe).
- edge_intensity: Intensity value (J) to get the edge energy value. Default set to inflection point.
- intensity_col: Name of the column with the intensity spectra.
- colormap: Name of the colormap you want to use for the plot. Default is set to 'tab20b'. More options here.
- width: Width of the graph.
- height: Height of the graph.
- plot_range: List [x,y] containing the energy range of the plot.
- return: Plot.

4.7 ecxas.plot.Eshift_vs_U

```
Eshift_vs_U(df, nb_cycle='all', edge_intensity='inflection',
↪  intensity_col='', option=1, colormap='plasma', width=7,
↪  height=4, dotsize=30, alpha=0.8, guideline=False)
```

Function to plot edge shift vs potential graph.

- df: Pandas dataframe with the data from the EC Lab file merged with the XAS files data.
- nb_cycle: List of the cycles you want to plot. Or number of the cycle you want to plot. Plots all by default.
- edge_intensity: Intensity value to get the edge energy value. Default is the inflection point.
- intensity_col: Name of the column with the intensity values.
- option: Parameter to choose the style of the graph. Default set to 1, meaning the figure will have a simplified color code with charge subcycles in red, discharge subcycles in blue, and different markerstyles for each cycle. If set to 2, the figure will have a different color for each cycle (according to a chosen colormap). If set to 3, x axis will be extended (1 sub-figure for each cycle).
- colormap: Chosen colormap if option set to 2. Default set to 'plasma'. More options here.
- width: Width of the graph.
- height: Height of the graph.
- dotsize: Size of the dot of the scatter plot (option 2).
- alpha: Opacity of the points (option 2).
- return: Plot.

4.8 ecxas.plot.Eshift_vs_t

```
Eshift_vs_t(df, nb_cycle='all', edge_intensity='inflection',
↪  intensity_col='', width=10, height=4, colormap='plasma',
↪  option=1, dotsize=10, alpha=0.5, guideline=False,
↪  hspace=0.0)
```

Function to plot edge shift vs time.

- df: Pandas dataframe with the data from the EC Lab file merged with the XAS files data.
- nb_cycle: List of the cycles you want to plot. Or number of the cycle you want to plot. Plots all by default.
- edge_intensity: Intensity value to get the edge energy value. Default set to inflection point.
- intensity_col: Name of the column with the intensity values.
- width: Width of the graph.
- height: Height of the graph.
- option: Parameter to choose the style of the graph. Default set to 1, meaning the figure will have a simplified color code with charge subcycles in red, discharge subcycles in blue, and different markerstyles for each cycle. If set to 2, the figure will have a different color for each cycle (according to a chosen colormap).
- colormap: Chosen colormap if option set to 2. Default set to 'plasma'. More options here.
- guideline: If True adds a grey line connecting the scattered points.
- return: Plot.

4.9 ecxas.plot.Eshift_vs_x

```
Eshift_vs_x(df, nb_cycle='all', edge_intensity='inflection',
↪  intensity_col='', option=1, colormap='plasma', width=7,
↪  height=4, dotsize=30, alpha=0.8, guideline=False)
```

Function to plot edge shift vs x graph.

- df: Pandas dataframe with the data from the EC Lab file merged with the XAS files data.
- nb_cycle: List of the cycles you want to plot. Or number of the cycle you want to plot. Plots all by default.
- edge_intensity: Intensity value to get the edge energy value. Default is the inflection point.

- intensity_col: Name of the column with the intensity values.
- option: Parameter to choose the style of the graph. Default set to 1, meaning the figure will have a simplified color code with charge subcycles in red, discharge subcycles in blue, and different markerstyles for each cycle. If set to 2, the figure will have a different color for each cycle (according to a chosen colormap). If set to 3, x axis will be extended (1 sub-figure for each cycle).
- colormap: Chosen colormap if option set to 2. Default set to 'plasma'. More options here.
- width: Width of the graph.
- height: Height of the graph.
- return: Plot.

5 Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957189. The project is part of BATTERY 2030+, the large-scale European research initiative for inventing the sustainable batteries of the future.