



Luminescence Spectroscopy Data Analysis in Python Using  **HyperSpy**
multi-dimensional data analysis

Jonas Lähnemann, Jordi Ferrer Orri, and the HyperSpy and LumiSpy developers

2nd Workshop on CL-EBIC, March 21, 2023

Today: short intro –
complemented by interactive
tutorial on Friday morning

Why use python for scientific data analysis?

- Both a genuine programming and a scripting language
- **Free** as in speech (open source) and as in beer (no fees)
- **Portable**: Runs on any environment (Win, Mac, Linux)
- **Intuitive** and readable code

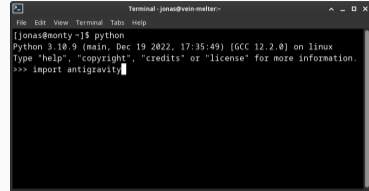
Why use python for scientific data analysis?

- Both a genuine programming and a scripting language
- Free as in speech (open source) and as in beer (no fees)
- Portable: Runs on any environment (Win, Mac, Linux)
- Intuitive and readable code
- **Vast and unmatched range of scientific libraries:**
 - numpy + scipy + matplotlib = Matlab replacement
 - SymPy, scikit-image, scikit-learn, ...

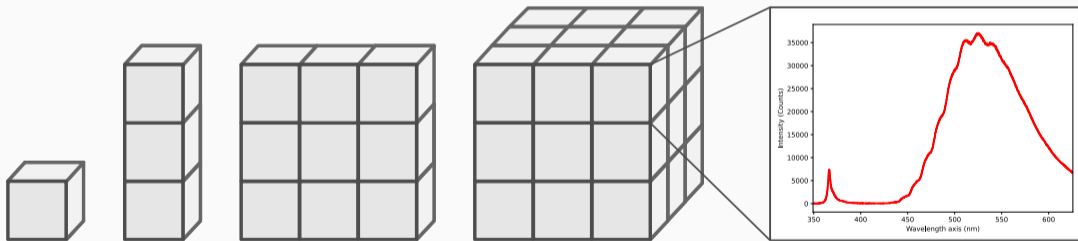


Why use python for scientific data analysis?

- Both a genuine programming and a scripting language
- Free as in speech (open source) and as in beer (no fees)
- Portable: Runs on any environment (Win, Mac, Linux)
- Intuitive and readable code
- Vast and unmatched range of scientific libraries:
 - numpy + scipy + matplotlib = Matlab replacement
 - SymPy, scikit-image, scikit-learn, ...
- Different environments:
 - Use interactive console: python, ipython
 - Run script files: silly-walk.py
 - Matlab-style integrated desktop environments: e.g. spyder
 - Mathematica-style notebooks: Jupyter
 - Dedicated frontends: e.g. hyperspyUI

A terminal window titled "Terminal - jonas@vein-melter" showing the execution of the Python command 'python'. The output displays the Python version (3.10.9) and the GCC version (12.2.0) on Linux. The prompt '>>>' is followed by the command 'import antigavity'.

- Open-source **python** library for interactive **analysis of multi-dimensional data**
- Easily extend operations from a single spectrum/image to multi-dimensional arrays



- Open-source `python` library for interactive **analysis of multi-dimensional data**
- Easily extend operations from a single spectrum/image to multi-dimensional arrays
- Easy access to cutting-edge **signal processing tools**
- **Community-driven**: Developed by scientists for scientists



- Open-source `python` library for interactive **analysis of multi-dimensional data**
- Easily extend operations from a single spectrum/image to multi-dimensional arrays
- Easy access to cutting-edge signal processing tools
- Community-driven: Developed by scientists for scientists
- **Extension framework for different signal types:**
 - EELS and EDX (currently still in the main package)
 - Holography (currently still in the main package)
 - `pyxem`: STEM data from pixelated electron detectors
 - `Kikuchipy`: EBSD patterns
 - `LumiSpy`: Luminescence spectroscopy



- **2007-2012:** Developed by Francisco de la Peña in his PhD (Univ. Paris-Sud) as EELSlab
- **2010:** Renamed to HyperSpy and open-sourced on GitHub
- **Now ...**
 - over 850 citations
 - 46 releases
 - 61 contributors
 - 415 stars on GitHub (users with GitHub account)
 - used in (at least) 115 other GitHub projects
 - and still growing!



- Tools for loading/saving various **data file formats** (soon new library RosettaSciIO)
- **Analytical tools** that exploit the multidimensionality of datasets
- User-friendly and powerful framework for **model fitting**: provides many standard functions, easily extended to custom ones
- **Machine learning algorithms**, useful for e.g. denoising data
- Efficient handling of **big datasets**
- **Data visualization**: evaluate datasets during the analysis, provide interactive operation for certain functions, and plotting
- **Extracting subsets** of data from multidimensional datasets: regions of interest and a powerful numpy-style indexing mechanism
- Handling of **non-uniform data axes**



- **HyperSpy** extension for luminescence spectroscopy data
- Provides the framework to work with photoluminescence (PL), cathodoluminescence (CL), electroluminescence (EL) and Raman spectroscopy
- Adds **specific functions**
- Work in progress ... open for other contributors

- Specific signal types (python classes) for **luminescence spectra and transients**
- Transformation to electron volt and wavenumbers/Raman shift (non-uniform signal axes)
- **Tools** such as data normalization/scaling
- **Utility functions** useful in luminescence spectroscopy data analysis: joining multiple spectra, unit conversion, etc.

- Specific signal types (python classes) for luminescence spectra and transients
- Transformation to electron volt and wavenumbers/Raman shift (non-uniform signal axes)
- Tools such as data normalization/scaling
- Utility functions useful in luminescence spectroscopy data analysis: joining multiple spectra, unit conversion, etc.
- All of the built-in HyperSpy tools!



- Website: <https://hyperspy.org>
- User guide: <https://hyperspy.org/hyperspy-doc/current/>
- Tutorial notebooks: <https://github.com/hyperspy/hyperspy-demos>
- Gitter chat: <https://gitter.im/hyperspy/hyperspy>



- User guide: <https://lumispay.org>
- Tutorial notebooks: <https://github.com/LumiSpy/lumispay-demos>

Anyone can make LumiSpy/HyperSpy better! **Don't be shy!**

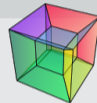
- Community driven development
- Report bugs on the github issue tracker
- Improve documentation
- Help review pull requests on github
- Contribute new functionalities:
 - Include tests and documentation
 - More effort than scripting for personal use
 - Shared development (avoid reinventing the wheel)

Interactive tutorial presented as Jupyter Notebook

Two ways to follow on your own PC:

Install python and relevant libraries on your Laptop:

- See LumiSpy User Guide: <https://lumispay.org>
- If you have python: get them from **pip** and **conda**
- We recommend installing the **HyperSpy-Bundle**
- A number of colleagues here can help you in the coffee breaks to get started!



Run the interactive web-based LumiSpy demo-notebooks on binder:

- <https://mybinder.org/v2/gh/lumispay/lumispay-demos/main>
- Easier start, but might be slower



PS: Find this presentation and relevant links on the workshop website!