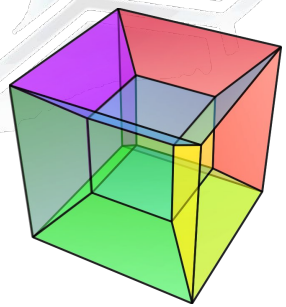


Open-source hyper-dimensional materials analytics using HyperSpy

MS&T21 - Materials Informatics for Images and Multi-dimensional Datasets

Joshua Taillon

October 20, 2021

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

MATERIAL MEASUREMENT LABORATORY

Disclaimer

Certain commercial equipment, instruments, materials, vendors, and software are identified in this talk for example purposes and to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose

HyperSpy development team (as of v1.6.4)

https://doi.org/10.5281/zenodo.592838

Francisco de la Peña†
Tom Furnival
Tomas Ostasevicius
Joshua Taillon 👋
Alberto Eljarrat
Ben Martineau
Nicolas Tappy
Gaël Donval
Eric Prestat
Petras Jokubauskas
Katherine E. MacArthur
Thomas Aarholt
Jan Caron
Suhas Somnath
Michael Walls
Vidar Tonaas Fauske
Jonas Lähnemann
Duncan N. Johnstone
Paul Quinn
Timothy Poon
Tom Slater
Niels Cautaerts
Pierre Burdet
Magnus Nord
Mike Sarahan
Vadim Migunov
Stefano Mazzucco
Carter Francis
Florian Winkler

† *Some slides in this presentation borrowed with great thanks from F. de la Peña*

About Me (or, why should you trust me?)

- Materials Data Scientist at NIST
- Extensive background in materials science and characterization
- TEM, SEM, EDS, EBSD, FIB, etc.
- Now focus on data science challenges in materials research
- Enjoy connecting scientists with novel analysis methods
- Regular user of and contributor to HyperSpy project
- Software Carpentry instructor in Python, Git, bash, R, etc.
- Have led many HyperSpy tutorials

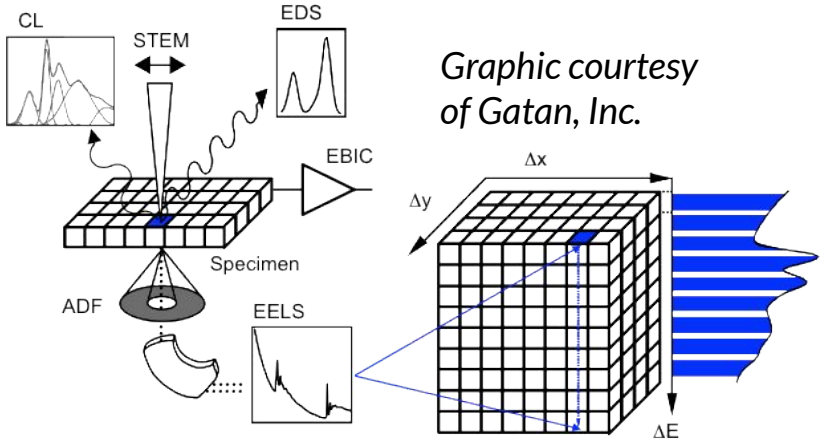
Outline

- What is HyperSpy?
- What can you do/what has been done with HyperSpy?
- The power of open-source in science
- How can *you* get started with HyperSpy?

What *is* HyperSpy?

What is HyperSpy?

HyperSpy is an *open source* and *open development* Python library for the interactive analysis of multi-dimensional datasets that can be described as multidimensional arrays of a given signal (for example, a 2D array of spectra, also known as a spectrum image).



Graphic courtesy of Gatan, Inc.

What is Python?

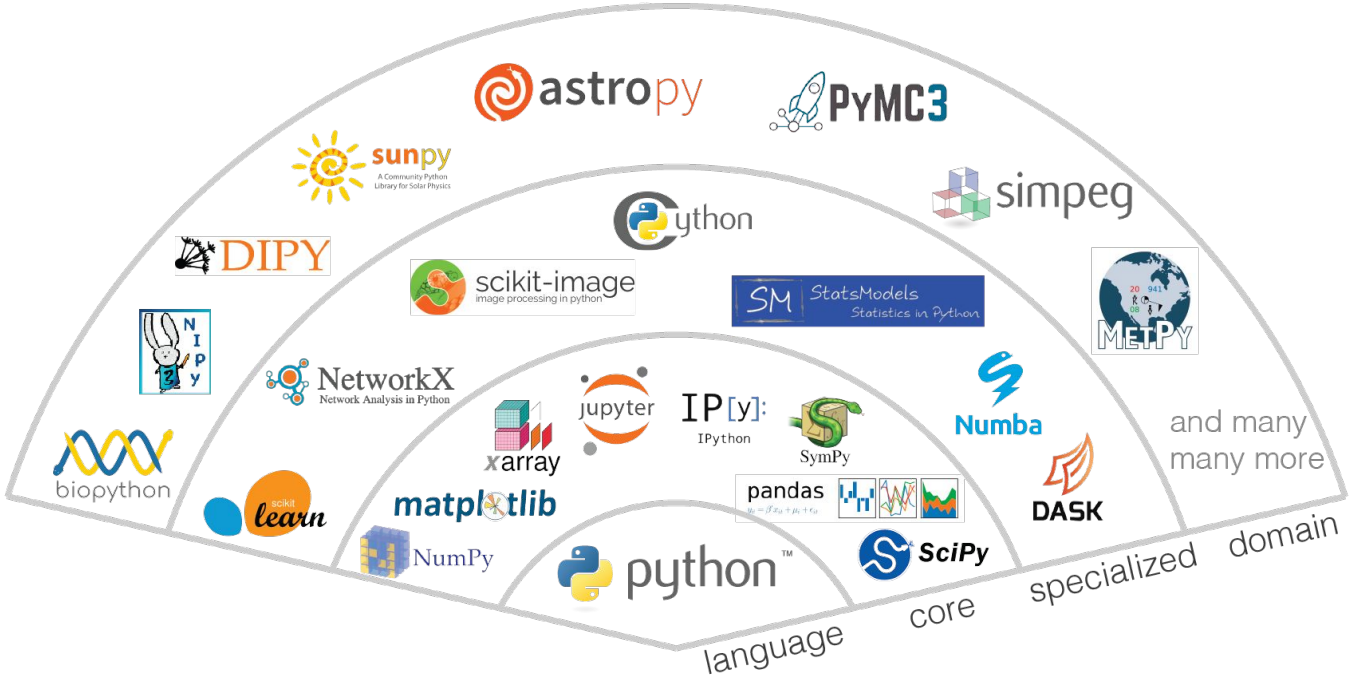
- High-level *general-purpose* programming language with “batteries included”
  - *i.e.*, it comes with enough features to do most basic programming tasks
- Simple enough for “quick & dirty” scripts; featured enough for complex projects
- Used extensively on the web, in applications, and throughout science
- Syntax and structure emphasizes readability and explicitness

```
print("Hello World!")
```

python™

A library? I thought HyperSpy was a program...

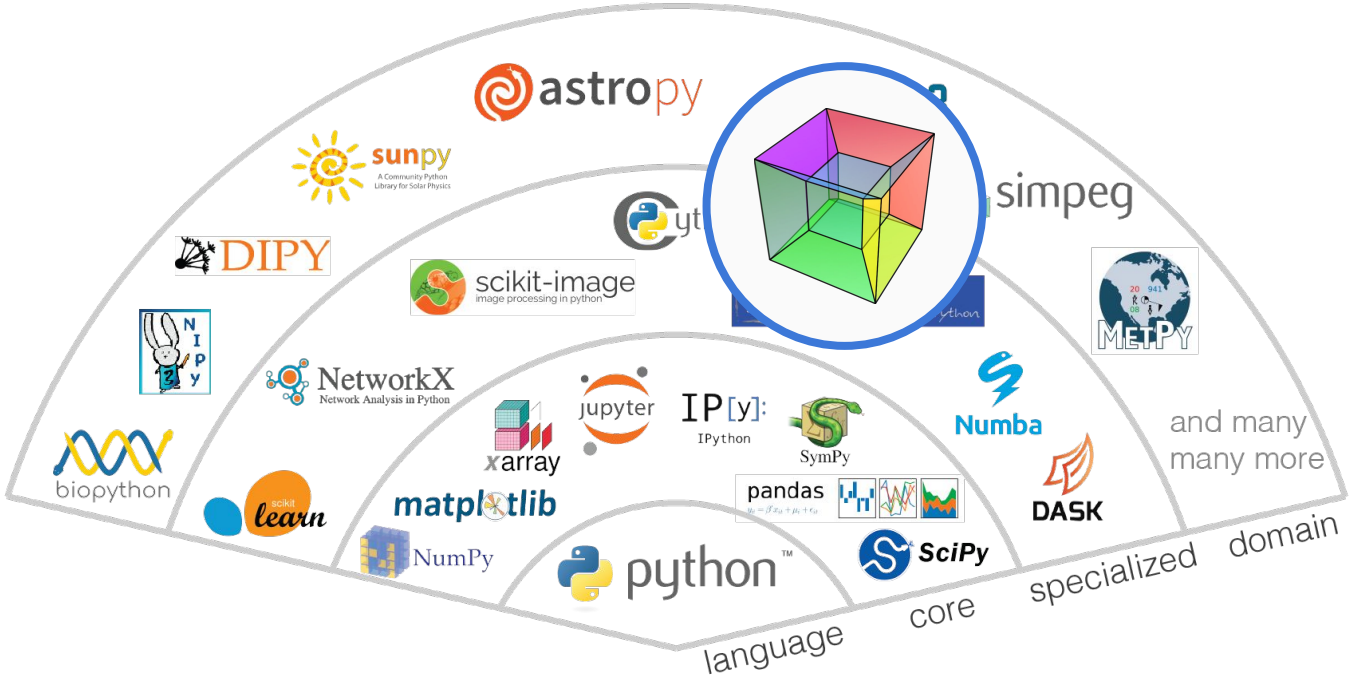
- HyperSpy is part of the “Scientific Python Ecosystem”
- Libraries extend the functionality of Python
- Works in concert with the entire ecosystem



https://jupytearth.org/jupyter-resources/introduction/ecosystem.html

Where does HyperSpy fit in the ecosystem?

- Has both domain-specific functionality and general purpose analysis tools
- Originally designed for electron microscopy, but much more broadly useful



https://jupytearth.org/jupyter-resources/introduction/ecosystem.html

A (Brief) History of HyperSpy

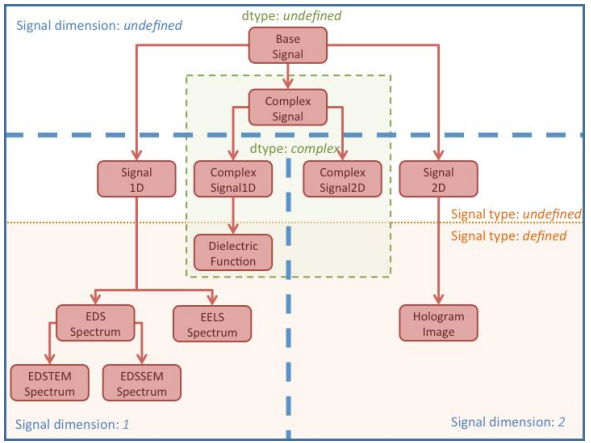
- HyperSpy developed by Francisco de la Peña 2007 - 2012 as part of PhD
  - Originally called EELSLab and focused on Electron Energy Loss Spectroscopy analysis
- Open sourced on Github in 2010
- Renamed to HyperSpy in 2011
- Now, over 700 citations, 43 version releases, 53 contributors and (at least) 90 other Github projects referencing Hyperspy

Core design principles of HyperSpy

- Data is organized into *signal* and *navigation* axes, and these are interchangeable!
  - The same 3D dataset could be a 2D array of spectra, or a 1D array of images; you get to decide

Signal Type	Dimensions
Regular Image	$(\mid x, y)$
EDS/EELS	$(x, y, \alpha, t \mid E)$
4D-STEM (Diffraction)	$(x, y, \alpha, t \mid k_x, k_y)$

- Data in HyperSpy is held in the `Signal` class; specific subclasses enable other features
  - Each subclass of Signal contains the actual data (as `numpy`, `dask`, or (soon) `cupy` arrays), and methods specific to the type of data
  - Functions general to all types of data are inherited as part of `BaseSignal`



HyperSpy extensions

- The core functionality of HyperSpy can be extended by registering new types of **signals**, model **components**, and interactive **widgets**
- Extensions keep the core of HyperSpy more manageable
- Focuses developer communities into areas of domain expertise
- Smaller specific communities generally more welcoming to new users
- Increases visibility (and credit, if that's important) of domain expert contributors

(Officially known) HyperSpy extensions

https://github.com/hyperspy/hyperspy-extensions-list

Package Name	Description
hyperspy-gui-ipywidgets	Interactive “in-browser” widgets for HyperSpy
hyperspy-gui-traitsui	Interactive “desktop” widgets for HyperSpy
KikuchiPy	Processing and analysis of electron backscatter diffraction patterns
LumiSpy	Analysis of luminescence spectroscopy data
pyxem	Multi-dimensional diffraction microscopy (4D-STEM)
Atomap (not officially, but very related)	Analysis of atomic resolution scanning transmission electron microscopy images

What can you do with HyperSpy?

Primary features

- Input/output of many file formats - mostly electron microscopy
- Interactive visualization, cropping, ROI analysis, etc.
- “Lazy” and parallel processing built-in (can handle big data)
- Multi-dimensional curve fitting with custom components
- Basic ML built in - signal decomposition, clustering, blind source separation
  - Advanced ML available through scikit-learn
- Specialized tools and signals for EM signals: EELS, EDS, etc.

Why might anyone want use HyperSpy?

- Makes it easy to operate on multi-dimensional arrays of data as you would a single spectrum (or image)
- Easy access to cutting-edge signal processing tools (both within HyperSpy and the greater Python ecosystem)
- Modular structure makes it easy to add custom features or extend into dedicated packages
- Use of “Jupyter Notebooks” encourages reproducible and sharable analyses
- It’s free! (more on this topic later…)

What about for Electron Microscopists?

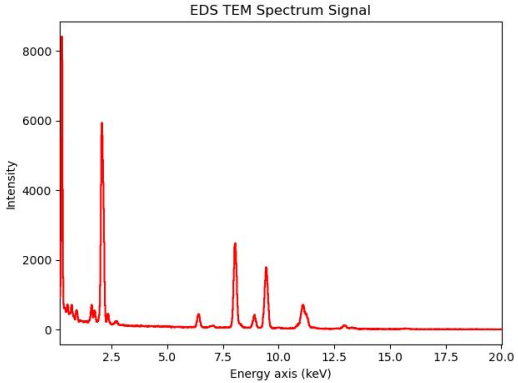
- Support for a wide range of open and proprietary data formats
- Generic signal processing, curve fitting, signal modeling, etc.
- Specialized tools for analytical EM:
  - EDS
  - EELS (including dielectric analysis)
  - Electron holography
- Advanced methods:
  - Multivariate statistics
  - Dimensionality reduction/signal separation

Gatan `.dm3/.dm4`	DENS heater log
FEI `.emi/.ser`	Bruker's `.bcf/.spx`
Image: e.g. `jpg`, `png`, `tif`	EMD (NCEM)
HyperSpy `.hspy`	EMD (Velox)
TIFF	Protochips log
MRC/MRCZ	EDAX `.spc/.spd`
EMSA/MSA	h5USID `.h5`
NetCDF	Phenom `.elid`
LISPIX (`.rpl/.raw`)	DigitalSurf's `.sur/.pro`
SEMPER `.unf`	Nexus `.nxs`
Blockfile	EMPAD `.xml`

Simple syntax to work with data of all dimensions

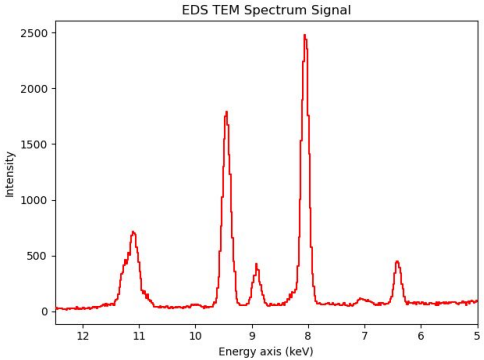
Load and plot a spectrum:

```
import hyperspy.api as hs
s = hs.load("spectrum.dm3")
s.plot()
```



Crop, reverse energy axis, and plot:

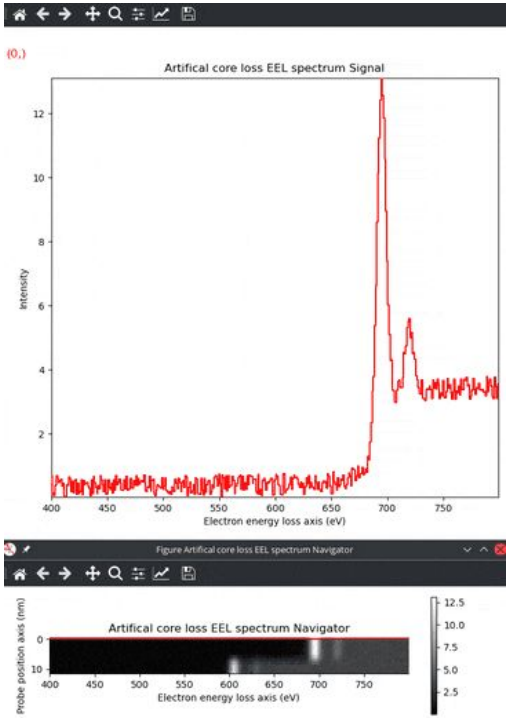
```
s.isig[5.0:12.5].isig[::-1].plot()
```

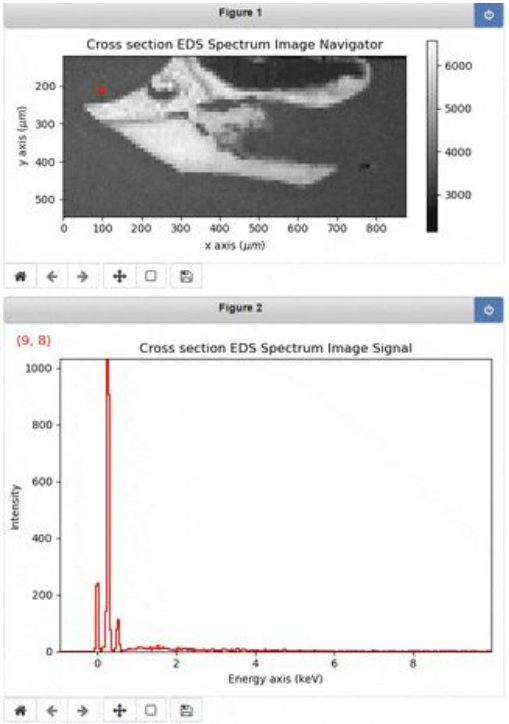


Interactive visualization of multi-dimensional data

Visualizing 1D and 2D spectrum images:

- Uses the same `s.plot()` syntax
- HyperSpy figures out “best” visualization for data shape
- “Navigator” can be easily customized

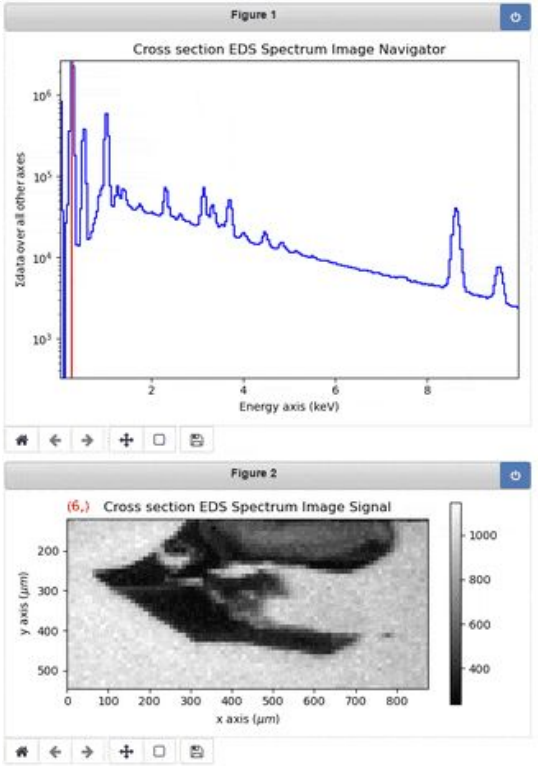




Simple manipulation of dimensionality

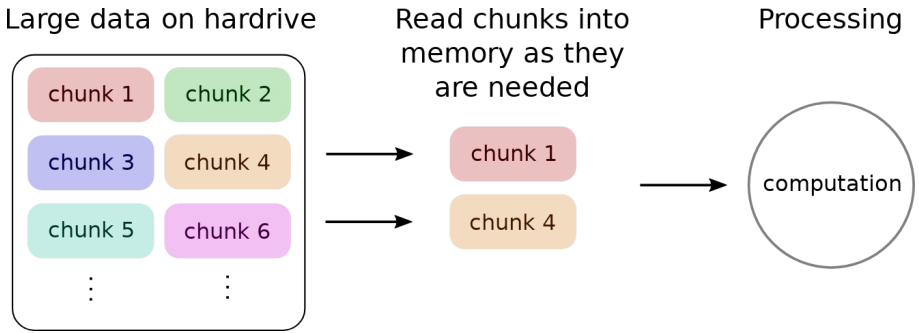
Datacubes can be transposed to easily view spectrum images as “energy level image stacks”

```
s.as_signal2D(image_axes=(0, 1)).plot()
```



Support for “Big Data”

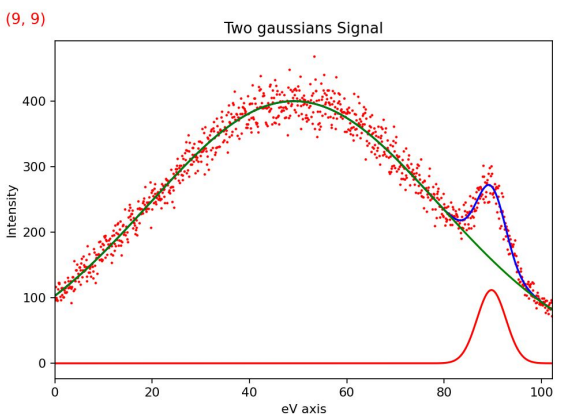
- In normal operation, data is held in memory as `numpy` arrays
- Most HyperSpy functions accept the `lazy=True` argument, which uses `dask` to “chunk” the data and only load from disk as needed
- Allows for computation on datasets of arbitrary size (larger than available memory)

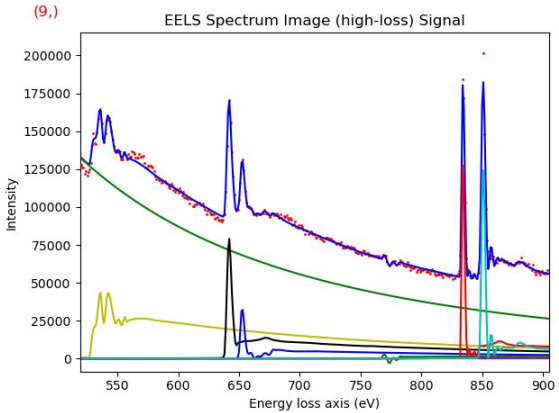


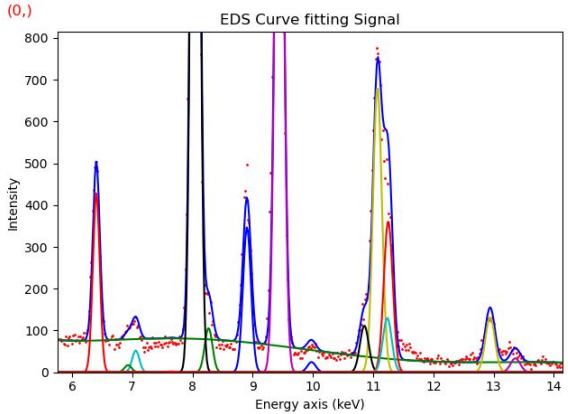
http://hyperspy.org/hyperspy-doc/current/user_guide/big_data.html

Curve/Model fitting

- Can fit arbitrary models to any type of signal with predefined or custom components
- Uses any algorithm supported by `scipy.optimize.minimize`
- Optimized fitting for EDS and EELS data



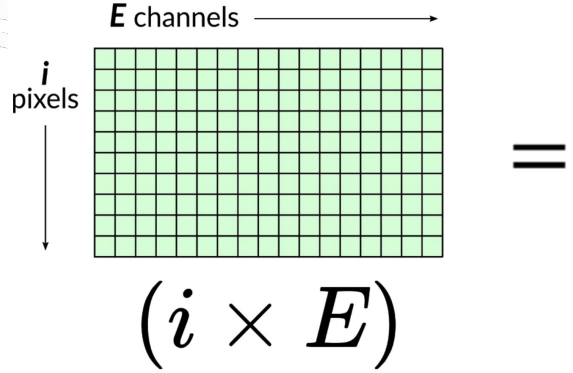


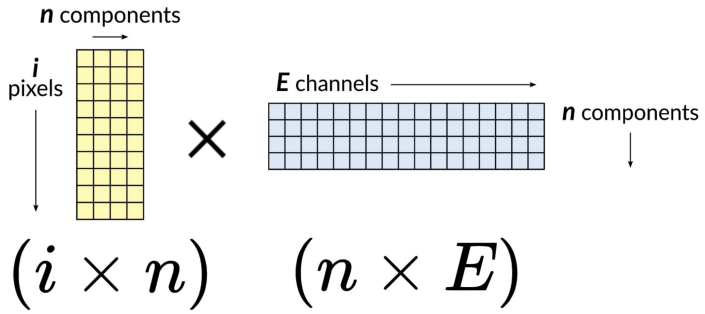


http://hyperspy.org/hyperspy-doc/current/user_guide/model.html

Unsupervised machine learning

- HyperSpy has built-in tools for *unsupervised* machine learning
  - Signal separation, clustering, etc.
- Helps you find “sources” of signals within your overlapping data
- Identifies constituent signals and where they are located in the dataset
- Various algorithms implemented within HyperSpy core:
  - Principal component analysis, non-negative matrix factorization, Independent component analysis, clustering, etc.

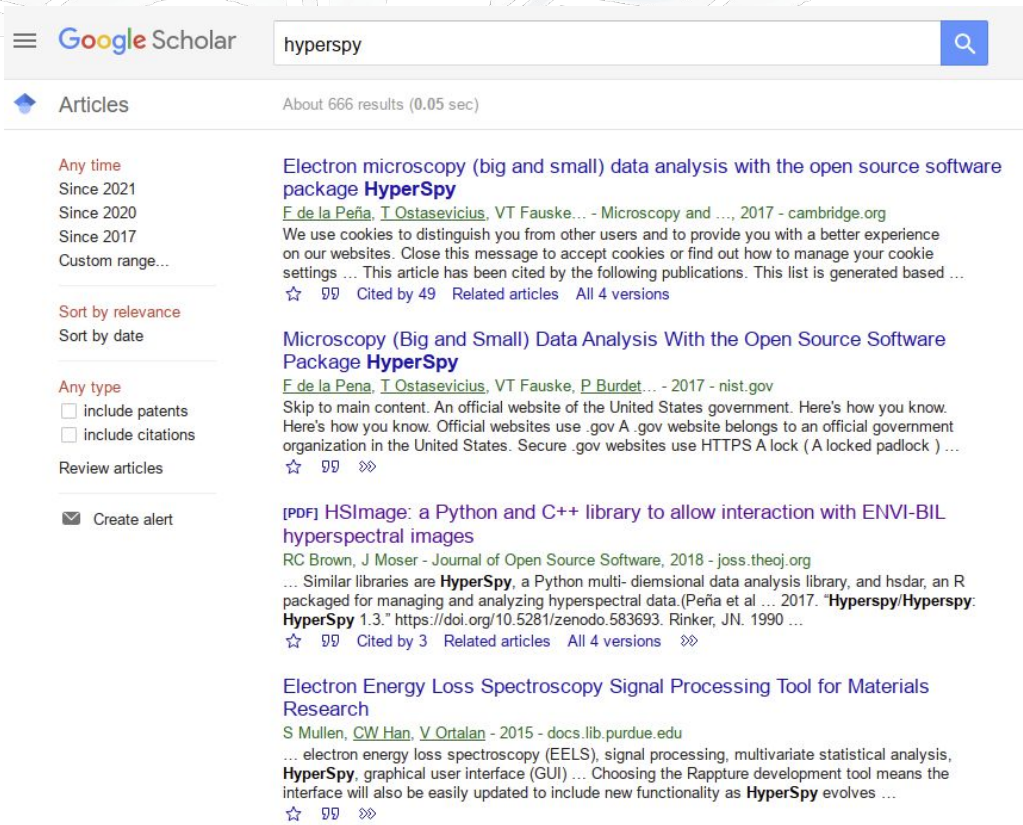




http://hyperspy.org/hyperspy-doc/current/user_guide/mva.html

And lots more!

- People are using HyperSpy for lots of new things all the time!
- The user guide describes what we have implemented, but...
- Recommend reading through Google Scholar for references to HyperSpy and looking at what can be possible



Why open source software for science?

What is open source?

- The “source code” that runs your program is publicly accessible, and people are allowed to modify, share, and redistribute it
- Usually “free” (zero cost), but more importantly “free” (you can do what you want with it)
- Encourages the reuse of ideas, implementations, and expertise
  - “Many hands make light work”
  - “Sunlight is the best disinfectant”
- Helps avoid “reinventing the wheel”, and focuses developer time and expertise into “standard” packages that do their job well

Relationship to the scientific enterprise

- In its ideal form, science should be done in an open and reproducible manner, where all components of the research enterprise are transparent †
  - There should be no “secret sauce” and everyone should be able to reproduce the result that you obtained
- Open source software is a tool to help us obtain that ideal
- Low (or no) cost analysis tools democratize science; grant dollars should not affect the quality of science you can do

† *Sentiment expressed very well by Marcus Hanwell at https://opensource.com/resources/open-science*

We all stand on the shoulders of giants

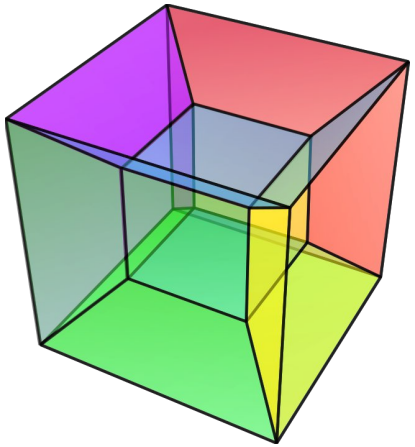
Our work is hard enough already!

We should make it easier where we can by helping each other

How can you get started?

Resources

- Best place to start is the HyperSpy website:
  - www.hyperspy.org
- Development happens in the open on Github:
  - https://github.com/hyperspy
- Extensive user guide and documentation:
  - http://hyperspy.org/hyperspy-doc/current/user_guide/index.html
- Tutorials and demos:
  - https://github.com/hyperspy/hyperspy-demos
- Chat room for developers and users:
  - https://gitter.im/hyperspy/hyperspy



Questions?

joshua.taillon@nist.gov