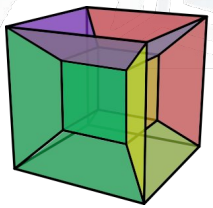


HyperSpy :

Your Multidimensional Data Analysis Toolbox

SciPy 2024

Tacoma, WA – Room 316

Josh Taillon

July 11, 2024

jat255

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

https://hyperspy.org/

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

Opinions are my own, not those of NIST or the U.S. Government (or the other maintainers!)

HyperSpy development team (as of v2.1.0)

https://doi.org/10.5281/zenodo.11148112

🇫🇷 **Francisco de la Peña**† 😇
🇫🇷 Pierre Burdet
🇳🇴 Magnus Nord
🇺🇸 Mike Sarahan
🇩🇪 Vadim Migunov
🇬🇧 Timothy Poon
🇺🇸 Suhas Somnath

🇬🇧 **Eric Prestat**
🇵🇱 Petras Jokubauskas
🇩🇪 Tomas Ostasevicius
🇺🇸 Joshua Taillon 👋
🇩🇪 Alberto Eljarrat
🇺🇸 Stefano Mazzucco
🇬🇧 Tom Slater

🇬🇧 Vidar Tonaas Fauske
🇬🇧 Tom Furnival
🇬🇧 Katherine MacArthur
🇬🇧 Thomas Aarholt
🇩🇪 Jan Caron
🇨🇭 Nicolas Tappy
🇫🇷 Michael Walls

🇩🇪 **Jonas Lähnemann**
🇺🇸 **Carter Francis**
🇬🇧 Duncan N. Johnstone
🇬🇧 Paul Quinn
🇯🇵 Takashi Nemoto
🇧🇪 Niels Cautaerts
🇺🇸 Hugh Ramsden

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

About Me (or, why might you consider trusting me?)

- Materials Data Scientist at NIST in Boulder, CO 🏔️🚵🥾
- 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 maintainer of HyperSpy project
- Software Carpentry instructor in Python, Git, bash, R, etc.
- Have led many HyperSpy tutorials

We've been here before…

"HyperSpy - How to Easily Bend Multi-Dimension Data to your (Analytical) Will"

Tomas Ostasevicius presented HyperSpy at SciPy 2016 in Austin, TX

Outline

- What is HyperSpy?
- What can you do/what has been done with HyperSpy?
- Fostering community in the HyperSpy project
- HyperSpy 2.0 (or, “breaking up is hard to do!”)

What *is* HyperSpy?

What is HyperSpy?

HyperSpy is an *open source* Python library designed for the interactive analysis of multi-dimensional data arrays, providing robust tools for navigation, visualization, curve fitting, and pattern recognition within high-dimensional datasets

It is *also* a vibrant and welcoming community of developers and users committed to producing a well-documented, powerful, and easy to use set of tools

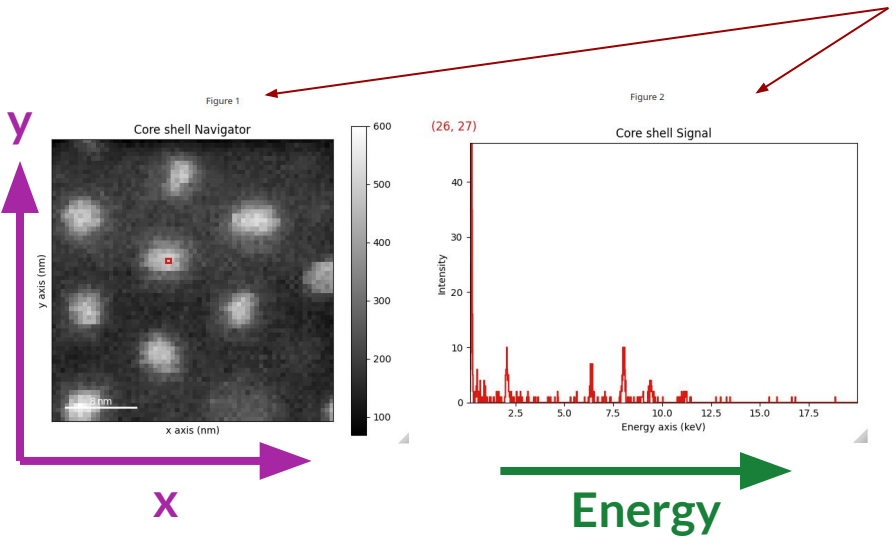
HyperSpy originated in the analytical electron microscopy community, but has evolved into a general purpose tool

Guiding principles of HyperSpy

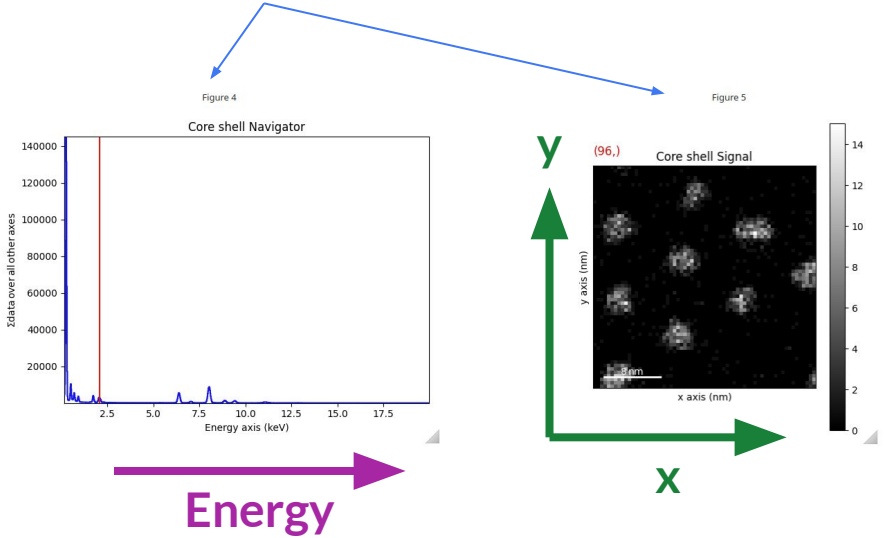
- Mission:
  - Facilitate the analysis of multi-dimensional datasets, which comprise signals of various dimensions typically collected over a range of parameters
- Principles:
  - Open-source
  - Community driven
  - Thoroughly documented
  - Powerful and scalable, with readable, efficient, and consistent syntax
  - Modular structure, making it highly extensible and adaptable to a variety of needs

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



vs.



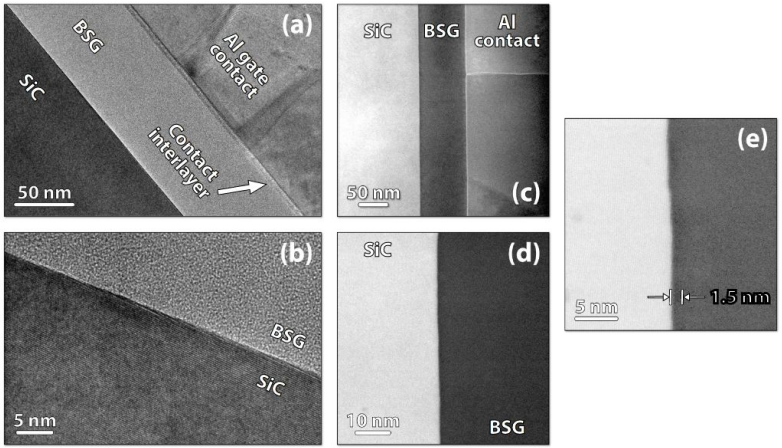
Examples of Multi-dimensional Datasets

	Signal Type	Dimensions (*navigation* | *signal*)
"Regular" imaging	Regular Image	$(\mid x, y)$
	Hyperspectral Imaging	$(\lambda \mid x, y)$
	Time Series Imaging	$(t \mid x, y)$
	Tomography	$(\Theta \mid x, y)$
Scanning Probe	Spectrum Imaging	$(x, y, \alpha, t \mid E)$
	4D-STEM (Diffraction)	$(x, y, \alpha, t \mid k_x, k_y)$

In the analytical characterization world, we often collect one (*or more*) signals over zero (*or more*) dimensions

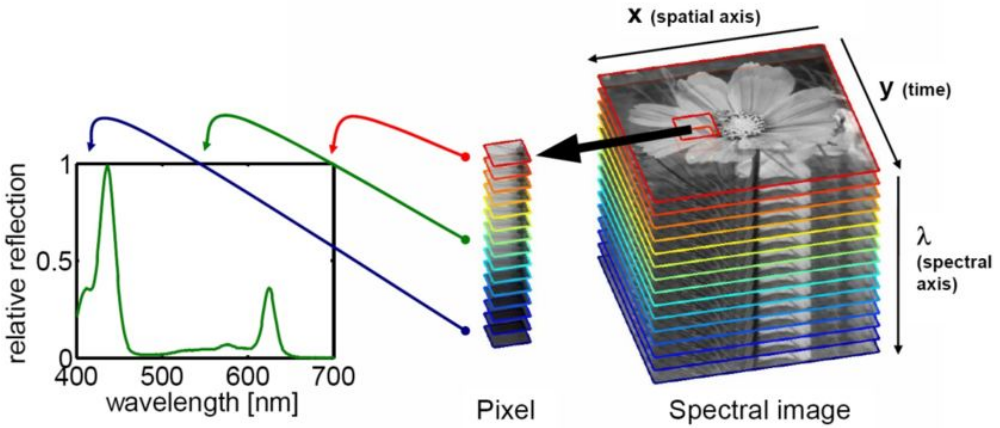
Examples of Multi-dimensional Datasets

Signal Type	Dimensions (*navigation* | *signal*)
Regular Image	(| *x*, *y*)
Hyperspectral Imaging	(λ | *x*, *y*)
Time Series Imaging	(*t* | *x*, *y*)
Tomography	(Θ | *x*, *y*)
Spectrum Imaging	(*x*, *y*, α, *t* | *E*)
4D-STEM (Diffraction)	(*x*, *y*, α, *t* | k_x, k_y)



1D Imaging (TEM)

(Taillon, 2016)

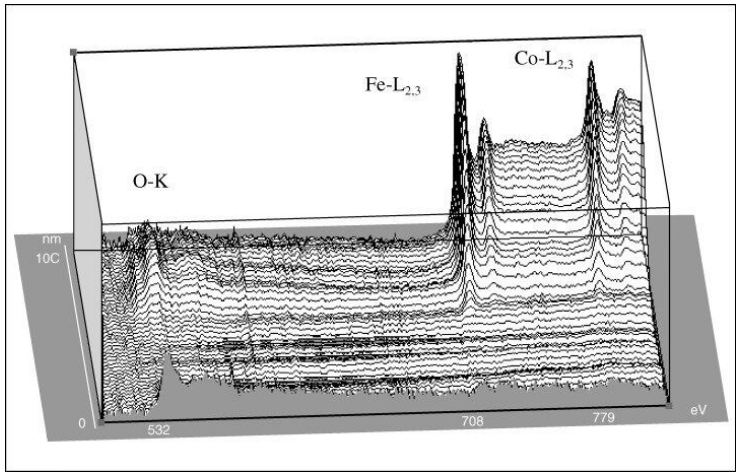


Hyperspectral Imaging

(Polder et al., *EFITA-WCCA-CIGR Conference*, 2013)

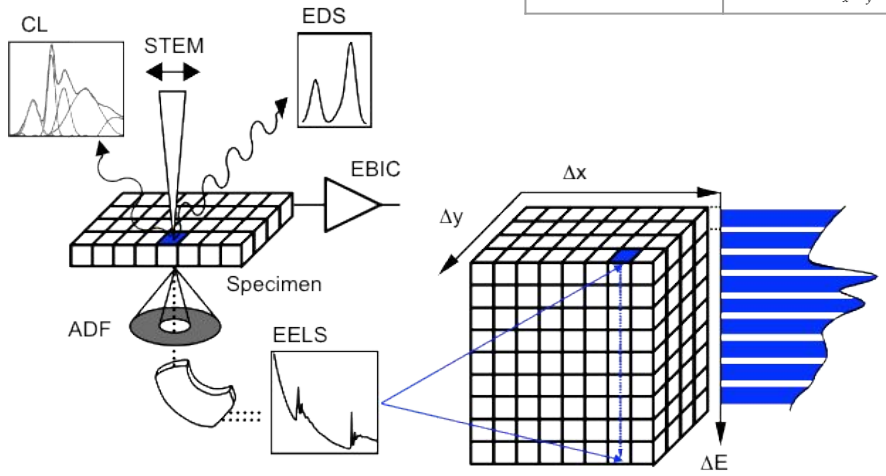
Examples of Multi-dimensional Datasets

Signal Type	Dimensions (*navigation* | *signal*)
Regular Image	(| x, y)
Hyperspectral Imaging	(λ | x, y)
Time Series Imaging	(t | x, y)
Tomography	(Θ | x, y)
Spectrum Imaging	(x, y, α, t | E)
4D-STEM (Diffraction)	(x, y, α, t | k_x, k_y)



1D Spectrum Imaging (Line Scans)

(Falqui, et al., *J. Microscopy*, 2003)
10.1046/j.1365-2818.2003.01177.x

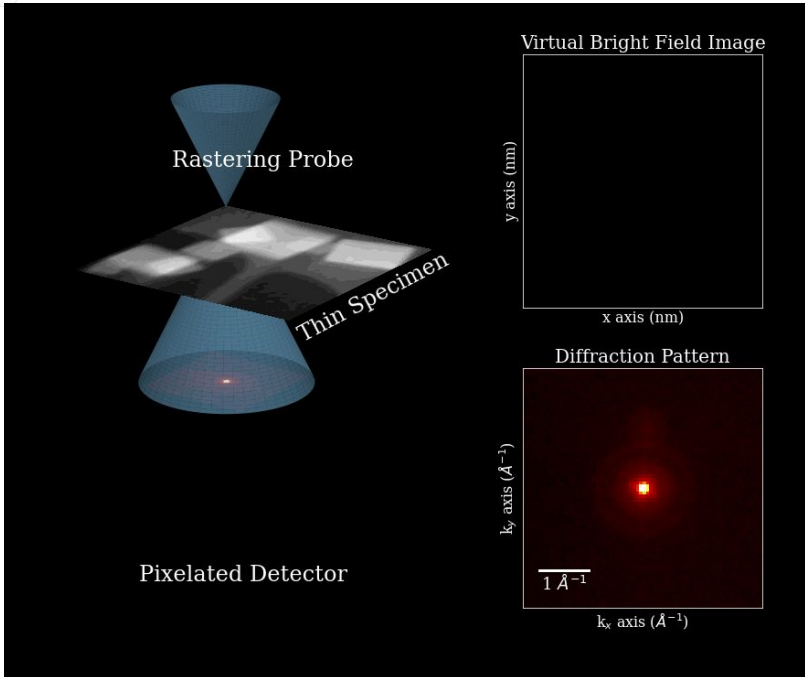


2D Spectrum Imaging

(Gatan, Inc.)

Examples of Multi-dimensional Datasets

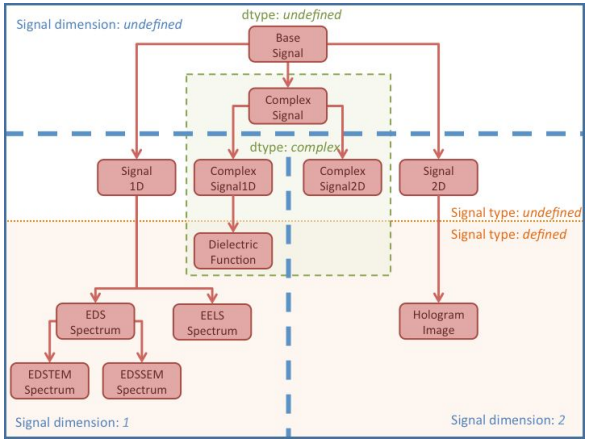
Signal Type	Dimensions (*navigation* | *signal*)
Regular Image	(| x, y)
Hyperspectral Imaging	(λ | x, y)
Time Series Imaging	(t | x, y)
Tomography	(Θ | x, y)
Spectrum Imaging	(x, y, α, t | E)
4D-STEM (Diffraction)	(x, y, α, t | k_x, k_y)



4D-STEM (Scanning Transmission Electron Microscopy)
(Carter Francis, 2023)

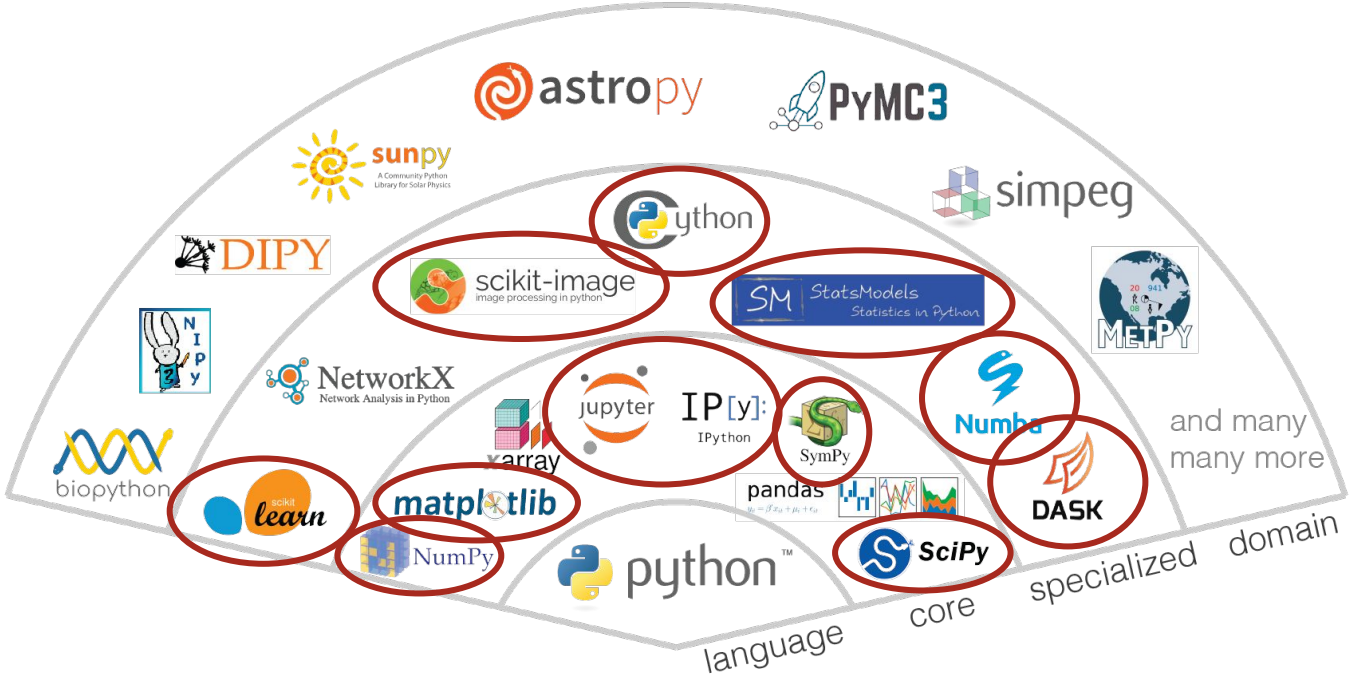
Core design principles of HyperSpy

- Data in HyperSpy is held in the `Signal` class; specific subclasses enable other features
  - Contains the actual data (as `numpy`, `dask`, or `cupy` arrays)
  - Subclasses enable methods specific to the type of data
  - Functions general to all types of data are inherited as part of `BaseSignal`
  - Domain-specific further subclasses provide other features



Part of the wonderful Scientific Python Ecosystem

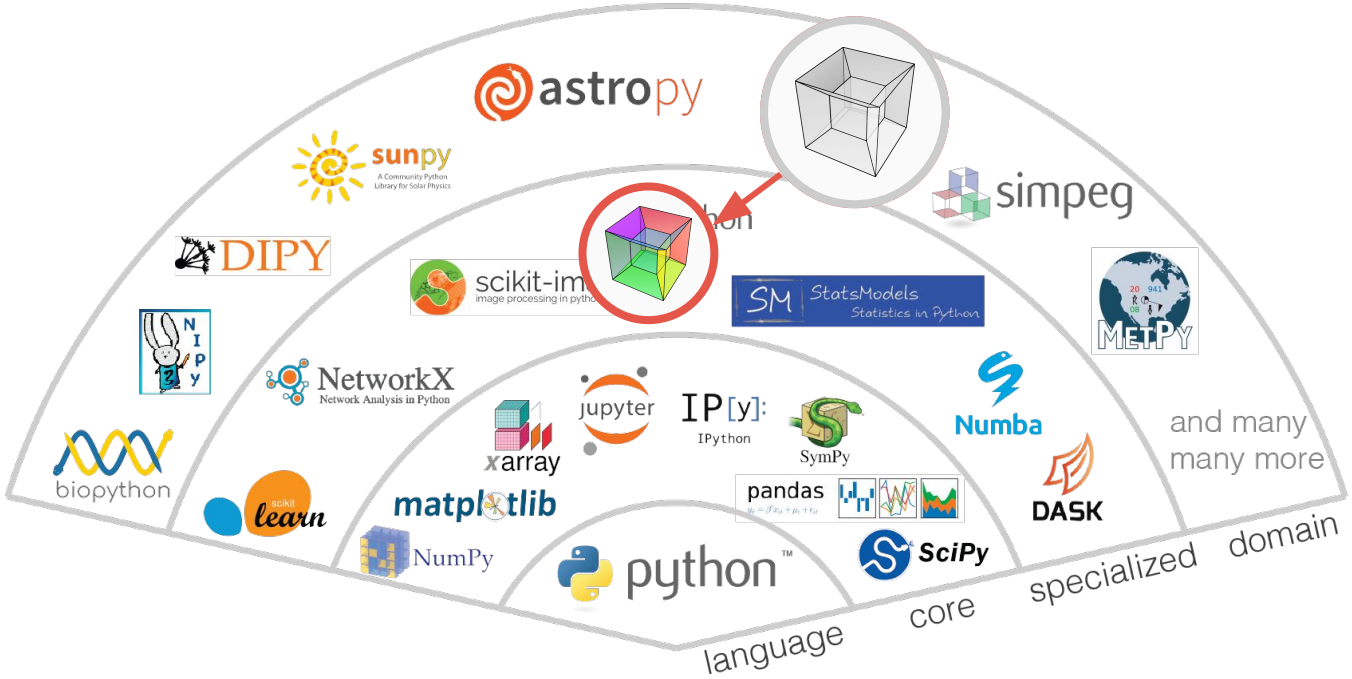
- HyperSpy is part of the greater Scientific Python Ecosystem
- “Stand on the shoulders of giants”
- Plays nicely with the entire ecosystem



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

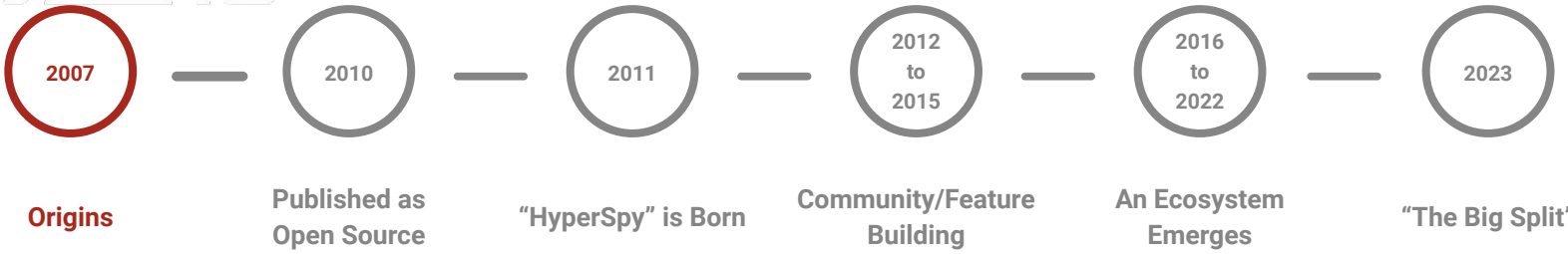
Where does HyperSpy fit in the ecosystem?

- Started as domain-specific functionality and general purpose analysis tools
- Originally designed for electron microscopy, but much more broadly useful
- Significant work recently to move towards the center



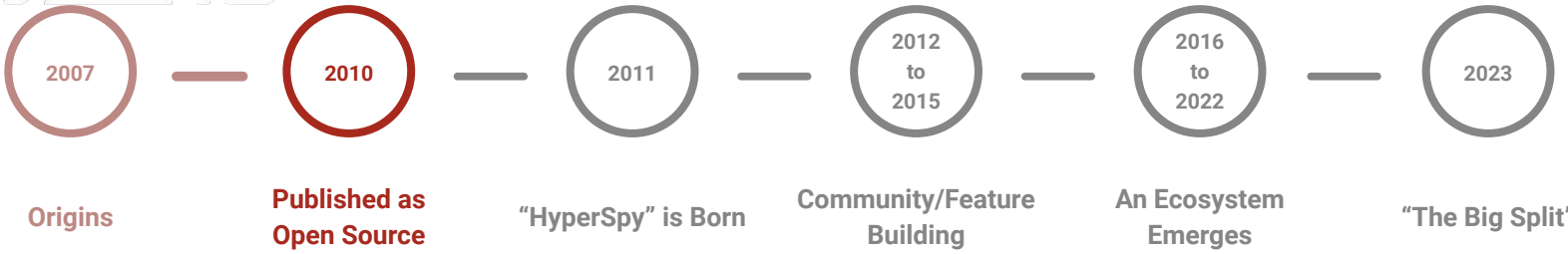
https://jupytearth.org/jupyter-resources/introduction/ecosystem.html

An Early History of HyperSpy



- Origins of HyperSpy date to Francisco de la Peña’s PhD at Paris-Sud
  - Originally called “EELSLab” and focused on Electron Energy Loss Spectroscopy analysis
  - Personal analysis functions grouped into classes useful for EELS work

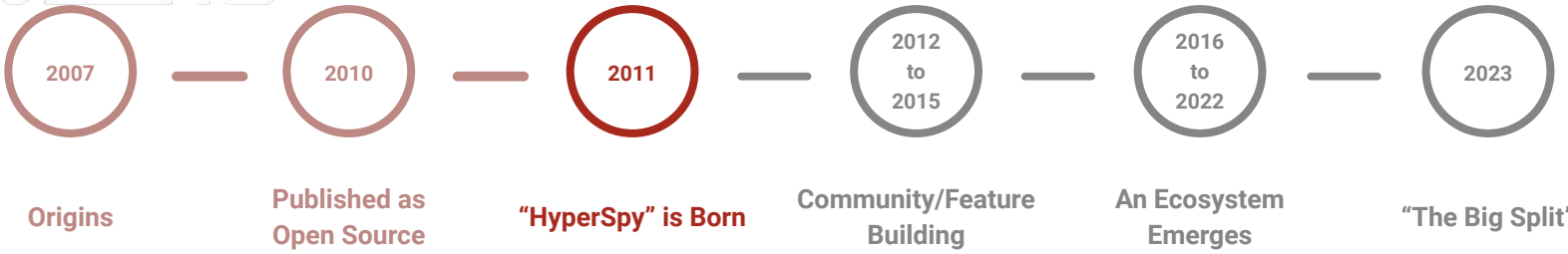
An Early History of HyperSpy



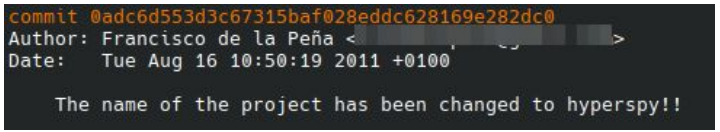
- Towards the end of Francisco’s PhD:
  - Multivariate analysis routines were added
  - EELSLab released under GPLv2 license (now v3)
  - First “other” users; tutorials in Paris and Oxford

```
commit 10dff1efbc83ed2560348eb594791fabf3a4ffd8
Author: Francisco de la Peña <[redacted]>
Date:   Mon Nov 15 10:36:27 2010 +0100

    Initital commit
```

An Early History of HyperSpy



- Renamed to “HyperSpy”:



  - New name better represents general utility
  - Development team expands to three
  - Refactored to support N-dimensional datasets

What can you do with HyperSpy?

Primary features (including basic extensions)

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

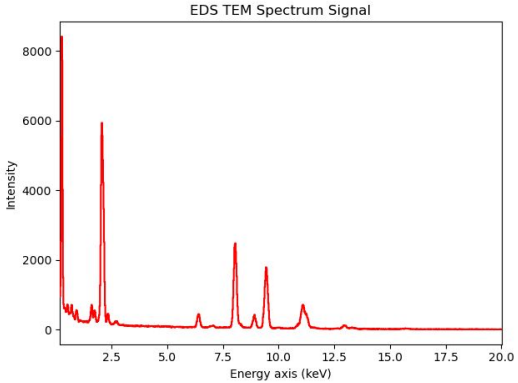
Why might you want use HyperSpy?

- 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! (was once an important consideration in the EM community…)

Simple syntax to work with data of all dimensions

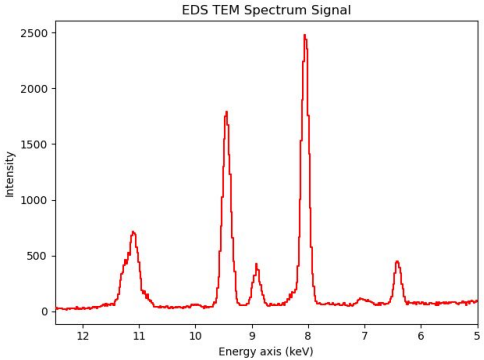
Load and plot a spectrum:

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



Crop, reverse energy axis, and plot:

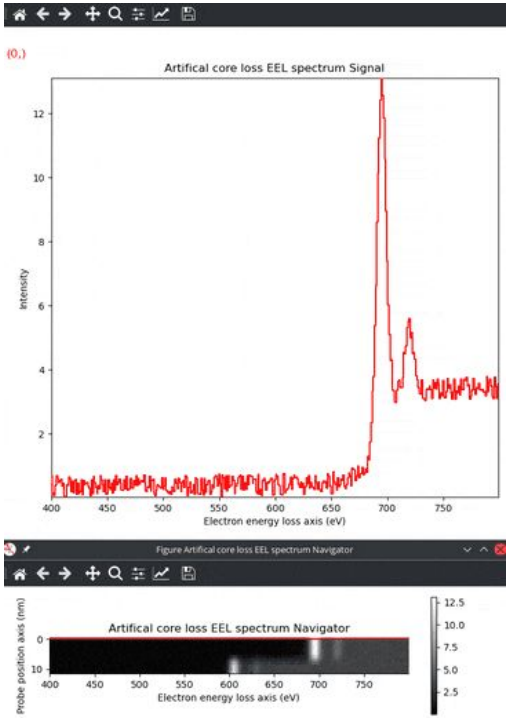
```
s.isig[5.0:12.0].isig[::-1].plot()
```

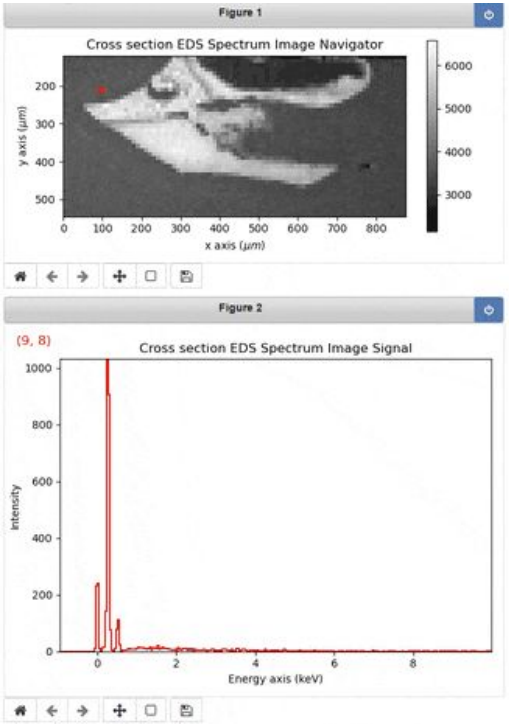


Interactive visualization of multi-dimensional data

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





Intuitive axes handling

```
s.axes_manager
```

< Axes manager, axes: (64, 64|992) >

Navigation axis name	size	index	offset	scale	units
x	64	26	0.0	0.5	nm
y	64	27	0.0	0.5	nm

Signal axis name	size		offset	scale	units
Energy	992		0.16	0.02	keV

Editing dataset calibration:

```
s.axes_manager['x'].scale *= 1000
s.axes_manager['y'].scale *= 1000
s.axes_manager['x'].units = 'pm'
s.axes_manager['y'].units = 'pm'
s.axes_manager['Energy'].scale *= 1.6022E-16
s.axes_manager['Energy'].units = 'J'
```

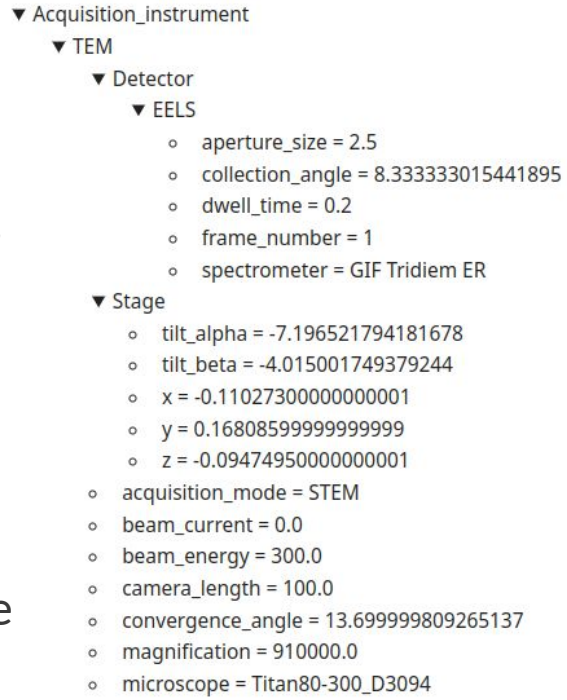
Pixel- or unit-based indexing:

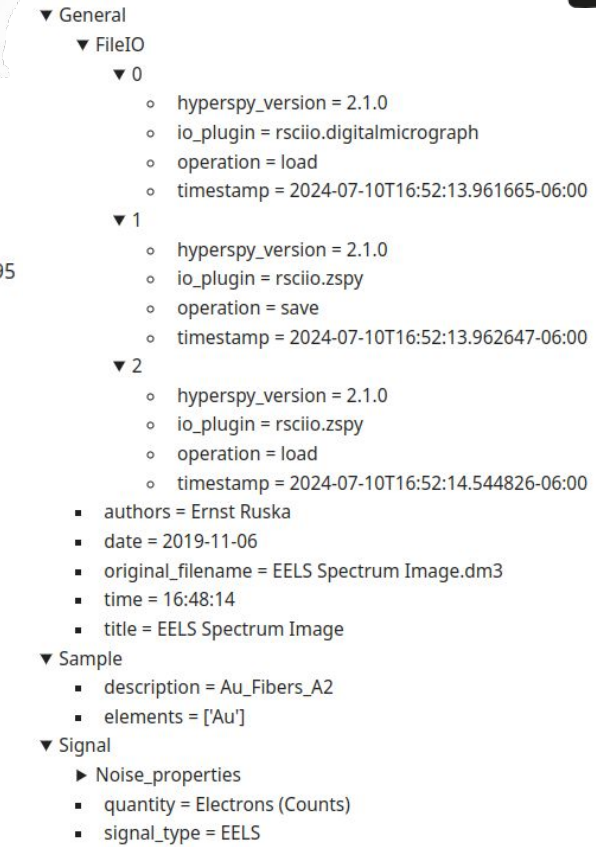
```
s.inav[0:5, :]      # index by pixels

s.inav[0.1:4.2, :]  # index by calibrated
                    # axis units (nm)
```

Dataset metadata

`s.metadata`

- HyperSpy readers parse metadata from many proprietary data formats
- Information about signal operations also stored in metadata tree
- Native data formats (`.hspy`/`.zspy`) preserve metadata

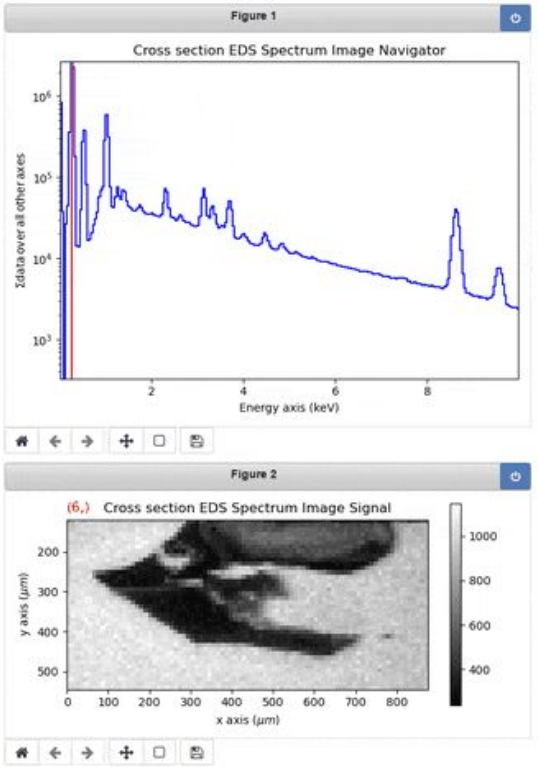




Simple manipulation of dimensionality

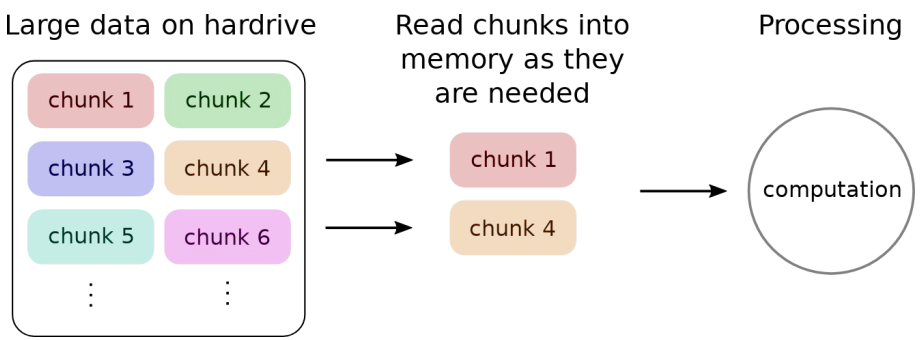
Datacubes can be transposed to easily view spectrum images as “energy level image stacks”

```
s.T.plot()
```



Built-in 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)

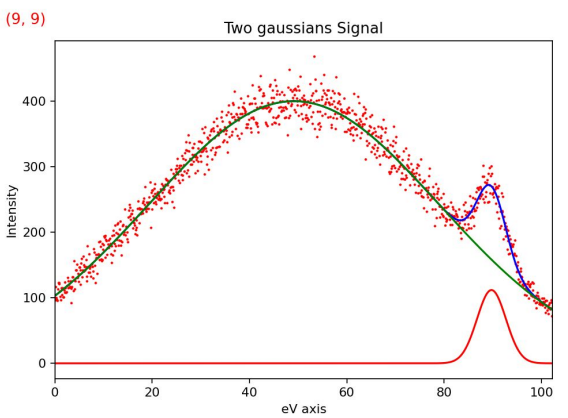


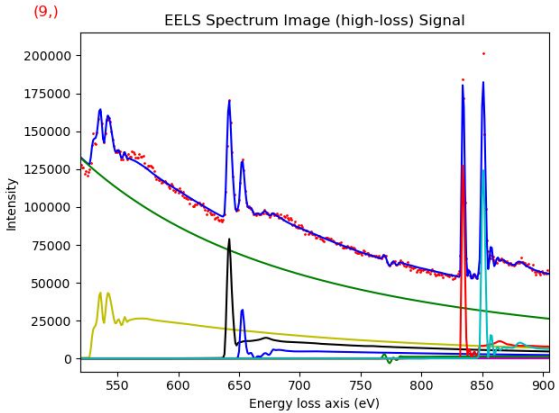
https://hyperspy.org/hyperspy-doc/current/user_guide/big_data.html

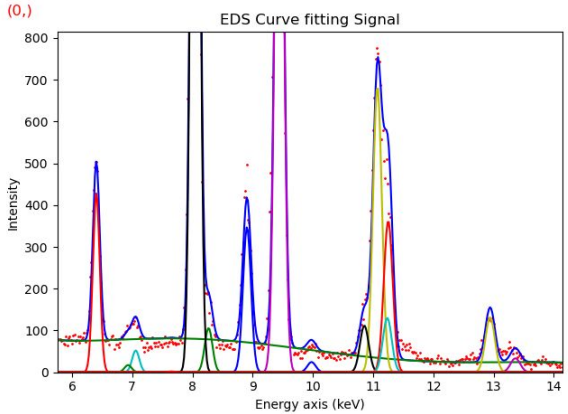
Curve/Model fitting

- Can fit arbitrary models to any type of signal with predefined or custom components
- Uses any algorithm supported by `scipy.optimize.minimize`
- Smart Adaptive Multi-dimensional Fitting (SAMFire)

https://doi.org/10.17863/CAM.15486



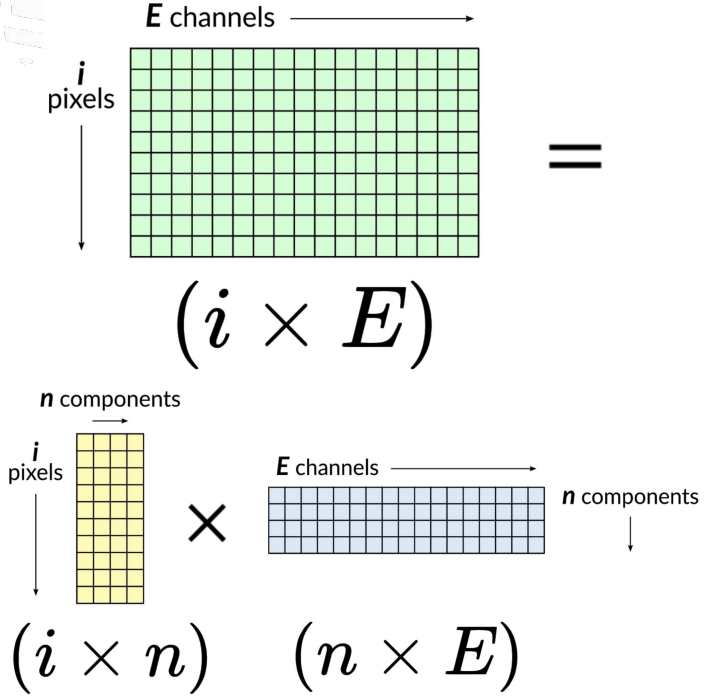




https://hyperspy.org/hyperspy-doc/current/user_guide/model/

Unsupervised machine learning

- HyperSpy has built-in tools for *unsupervised* machine learning
  - Signal separation, clustering, decomposition, etc.
- Identifies constituent signals and where they are located in the dataset
- Various algorithms implemented within HyperSpy core:
  - Principal component analysis (PCA), non-negative matrix factorization (NMF), Independent component analysis (ICA), clustering (*k*-means etc.), and more



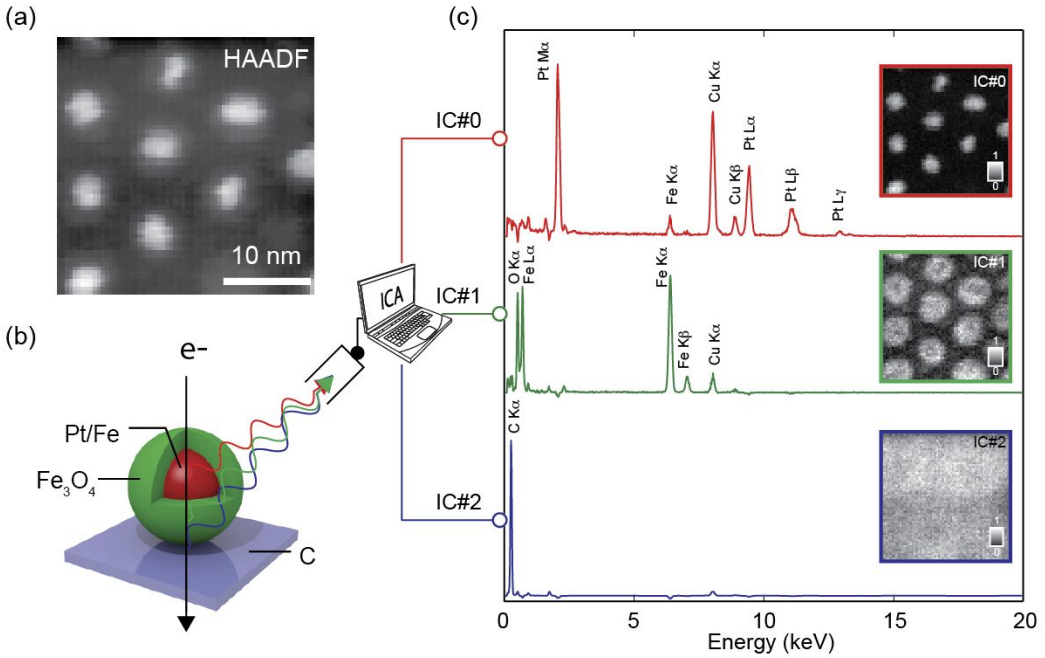
https://hyperspy.org/hyperspy-doc/current/user_guide/mva/

A quick example of ML in EDS analysis

- D. Rossouw, *et al.*, *Nano Letters*, **15**, 2716–2720, 2015

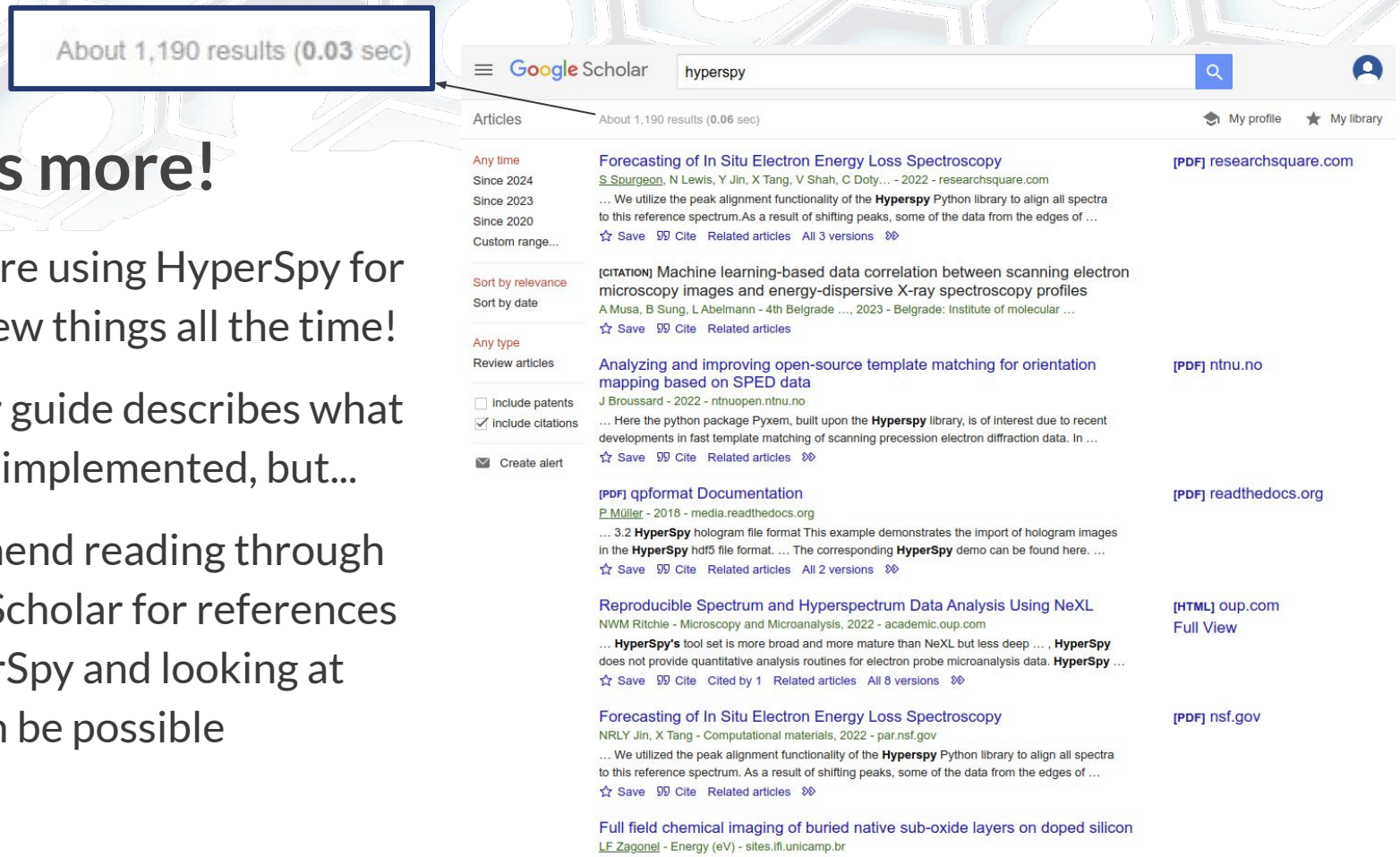
  https://doi.org/10.1021/acs.nanolett.5b00449

- Pt/Fe core surrounded by iron oxide shell
- Identified meaningful spectral components in a sea of overlapping signals
- Composition of the core can be measured with ICA, even though it's measured through the shell
- Try it yourself! https://github.com/hyperspy/exspy-demos

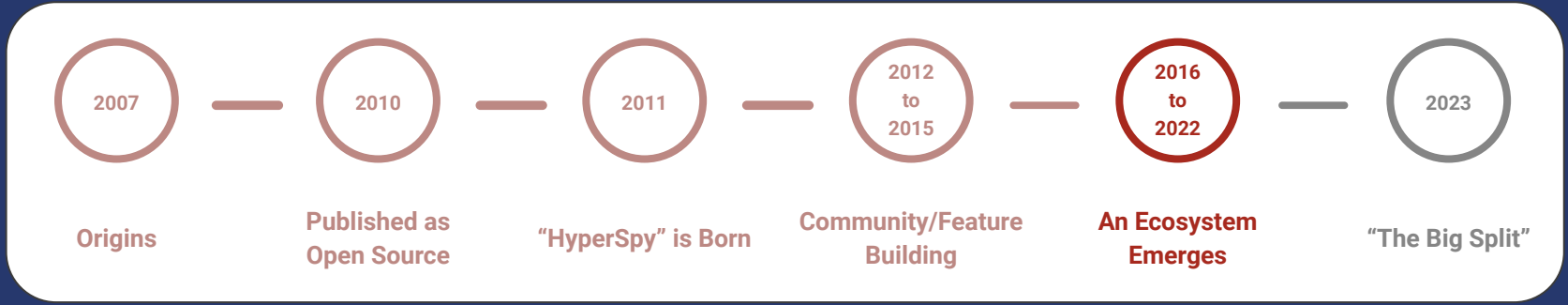


And lots more!

- 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



The HyperSpy Community



Some stats about HyperSpy

- ~1,200 “citations”/references in Google Scholar
- Over 60 individual contributors over 14 years
- 500 GitHub stars
- 200 dependent repositories (according to GitHub “insights”)

Community-driven development

- Since the beginning, written by users, for users (no outside funding)
- A focus on documentation and lowering the barrier for contributions
- An active interactive chat room / GitHub discussions
- Regular running of in-person and virtual tutorial sessions at conferences and “summer schools”

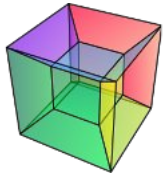
👋 We are also just a friendly group!

https://hyperspy.org/hyperspy-doc/current/user_guide/
https://hyperspy.org/hyperspy-doc/current/dev_guide/

https://gitter.im/hyperspy/hyperspy
https://github.com/hyperspy/hyperspy/discussions

https://hyperspy.org/news/

Personal aside about the power of community…



Computational Materials Science Research

What's a compiler?
What's Emacs?
Why does my pinky hurt?

2010

2013

Electron Microscopy Grad Student

I guess I should try this "Python" thing I've heard about.

You mean I can analyze my data beyond what the company allows me to do in their GUI?

2018

Gainfully Employed!

It turns out this Python thing is actually pretty useful, and contributing to open-source is a pretty good way to learn it

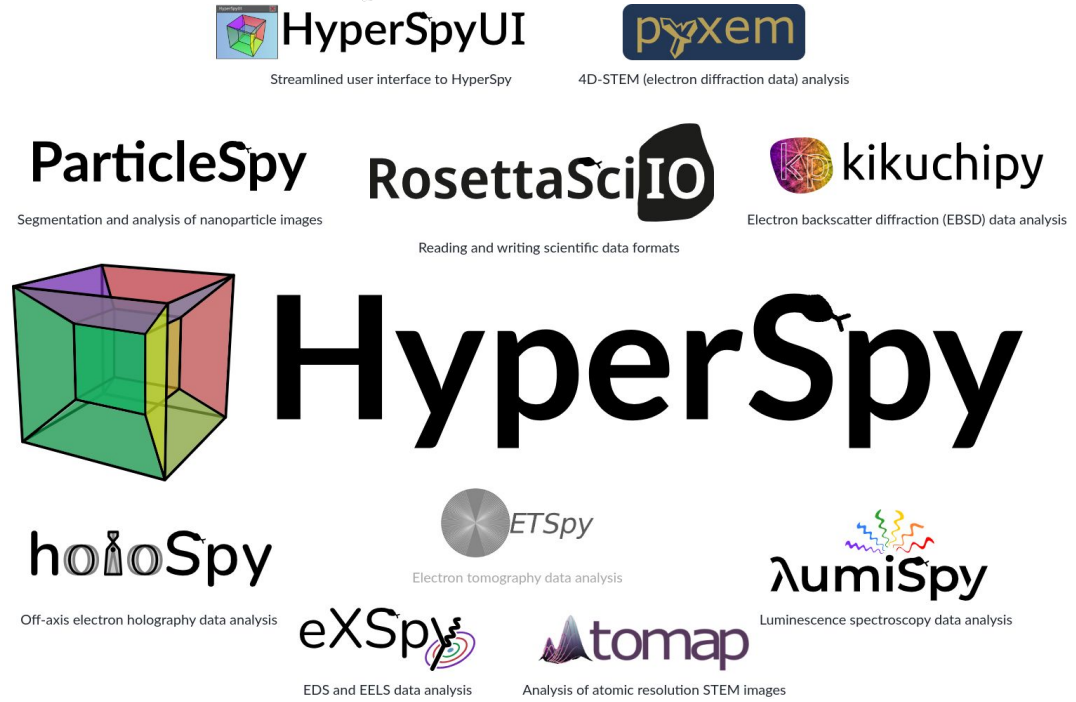
2024+

My First SciPy!

Maybe I shouldn't have waited so long…

A multi-dimensional analysis ecosystem

- As the user base has grown, so have the use cases
- Extension registration mechanism introduced in v1.5.0 (2019)
- “Lean” HyperSpy core with domain-agnostic functionalities
- Currently ~10 known registered extensions



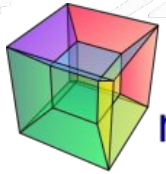
HyperSpy extensions

- The core functionality of HyperSpy can be extended by registering new types of **signals**, model **components**, and interactive **widgets**
  - https://hyperspy.org/hyperspy-doc/current/dev_guide/writing_extensions.html
- 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

HyperSpy 2.0


2007
Origins
2010
Published as Open Source
2011
“HyperSpy” is Born
2012 to 2015
Community/Feature Building
2016 to 2022
An Ecosystem Emerges
2023
“The Big Split”

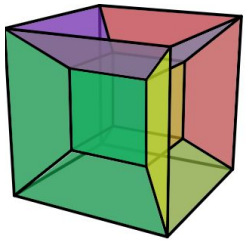
Breaking up into a leaner core



HyperSpy

multi-dimensional data analysis

≤ 1.7.6



HyperSpy 2.X

+

holoSpy

Off-axis electron holography data analysis

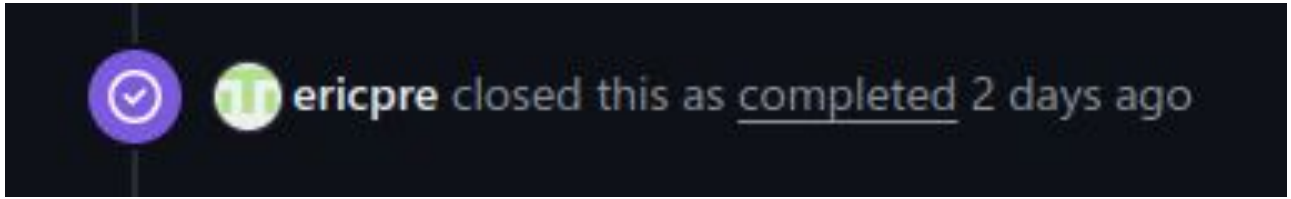
eXSpy

EDS and EELS data analysis

RosettaSciIO

Reading and writing scientific data formats

Idea proposed over 8 years ago…



Technically, 2.0 release was Dec. 2023

Why “split” the package?

- Since its origins, HyperSpy had included file I/O, EELS, EDS, and holography code in its core; incompatible with being “general purpose”
- Many dependencies (especially for I/O)
- Unnecessary complications for development/high barrier to entry for contributors
- Better positions HyperSpy for future development
- Opportunity to “clean out the cruft” (40% LoC reduction)

What took so long?

- Lots of work – little immediate benefit to the project
- Limited development budget
- “Extension mechanism” had to be implemented first
- Wrapping of big changes into API-breaking 2.0 release

Final thoughts

What about  ?

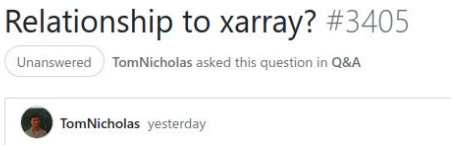
Big caveat!

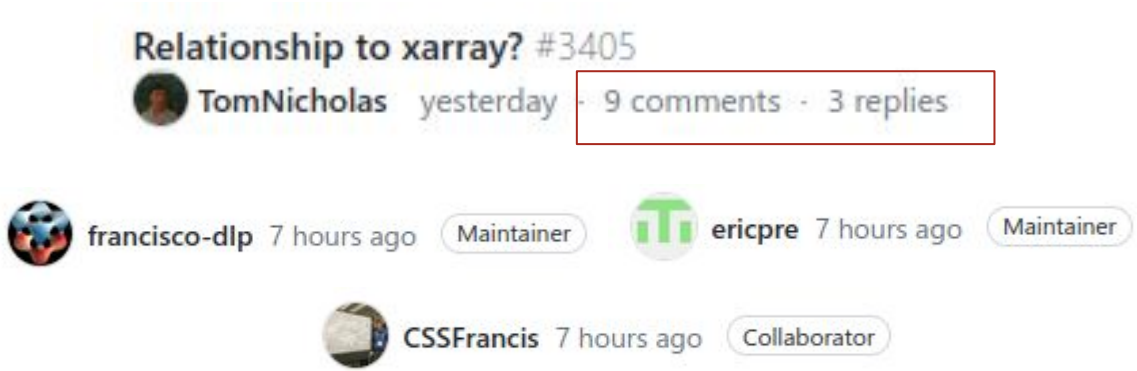
I am not intimately familiar with Xarray, and they may disagree with any/all of this

- HyperSpy and Xarray have “convergently evolved” from different domains:
  - HyperSpy from the materials characterization/EM community (first commit 2010)
  - Xarray from the climate/earth science community (first commit 2013)
- Both packages handle multidimensional data, with different approaches:
  - HyperSpy has more “batteries included” as an interactive multidimensional data analysis tool
  - Xarray is more narrowly focused on the data model and integration with other visualization packages (also “database-y” thing with `pandas` that HyperSpy doesn’t do at all)
- Potential for future collaboration between the projects?
  - https://github.com/hyperspy/hyperspy/discussions/3405

What about xarray?

- Potential for future collaboration between the projects?
  - https://github.com/hyperspy/hyperspy/discussions/3405



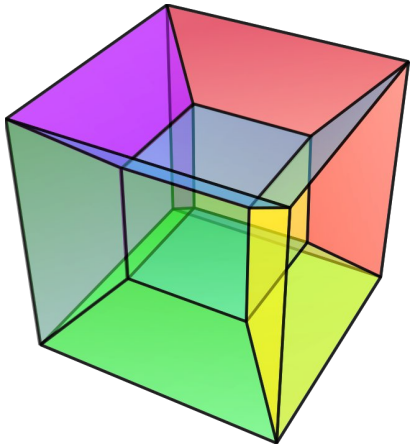


Parting thoughts

- HyperSpy has a mature codebase and well-established community
- The extension mechanism has given rise to
  a growing ecosystem of tools across scientific domains
- We'd love to have you!

Resources

- Best place to start is the HyperSpy website:
  - www.hyperspy.org
- Development happens on Github:
  - https://github.com/hyperspy
- Extensive user guide and documentation:
  - https://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