

Welcome to 2019 M&M Short Course X-15

Data Analysis in Materials Science with



Presented by (a selection of) the HyperSpy developers:

- Duncan Johnstone
- Katherine MacArthur
- Magnus Nord
- Eric Prestat
- Joshua Taillon

A decorative header featuring a repeating pattern of light blue and white hexagons, resembling a honeycomb or molecular structure, spanning the top of the slide.

Session 1: An Introduction to Python and HyperSpy:

The multi-dimensional data analysis toolbox

Josh Taillon

August 4, 2019

A decorative header featuring a pattern of overlapping hexagons in light blue and white, creating a honeycomb-like structure.

NIST 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.



What is this "Python" that I've heard of?

What is python[™] ?

- High-level, *general purpose* programming language with "batteries included"
- Simple enough for quick scripts; featured enough for complex projects
- Used extensively on the web, in applications, and throughout science
- Syntax emphasizes readability and explicitness


A decorative pattern of white hexagons with light blue shading, arranged in a honeycomb-like structure, spanning the top of the slide.

A "Hello, World!" comparison:

A decorative header featuring a repeating pattern of hexagons with a 3D effect, rendered in light blue and white tones.

Java:

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
  
}
```



C++:

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, World!" << endl;
    return 0;
}
```




Python:

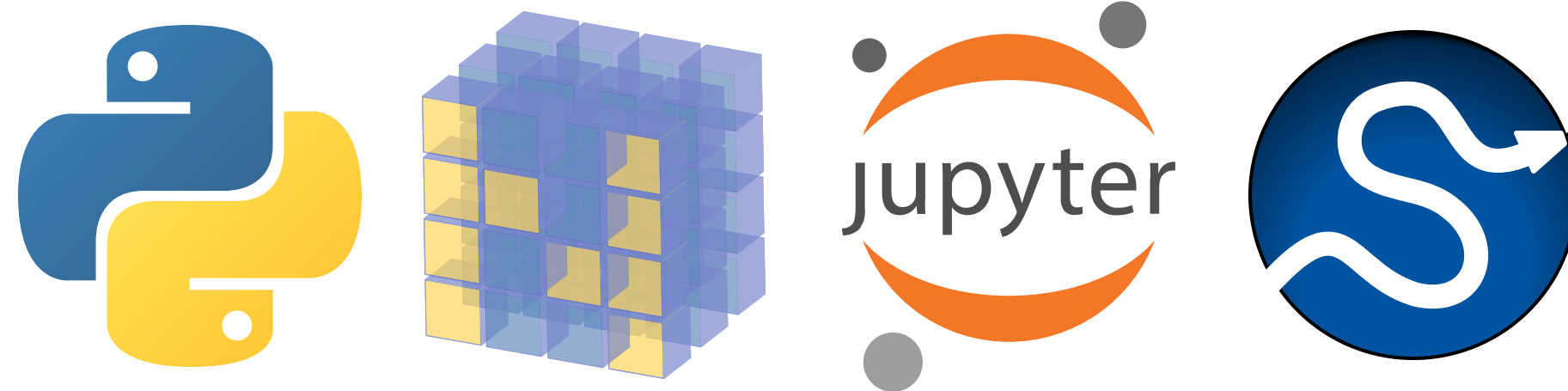
```
print("Hello, World!")
```

Matlab:

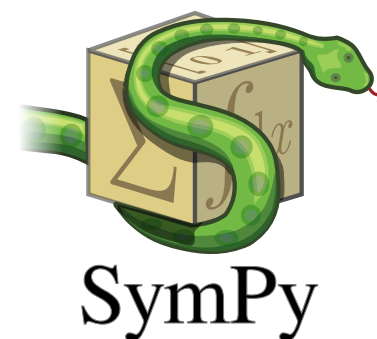
```
disp("Hello, World!")
```

Okay, but what does this have to do with science?

The "scientific Python" ecosystem provides almost any functionality you may need:

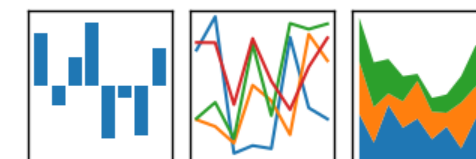


matplotlib



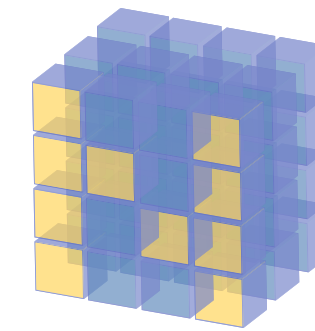
pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Linear algebra, optimization, machine learning...

- Numerical computing with `NumPy` (like what you would use Matlab for):




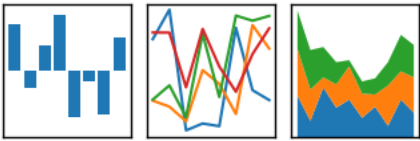

- Signal processing, numerical integration, optimization, etc. with `SciPy`:




- Machine learning with `scikit-learn` (and others):



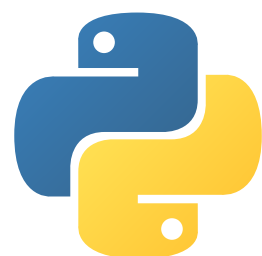
Visualization, statistics, computer algebra...

- Scientific visualization with `matplotlib`:  **matplotlib**
- Data series and statistical analysis with `pandas`:  **pandas**
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$
- Symbolic computation with `SymPy`: 
SymPy

A better (?) way to create and publish your work

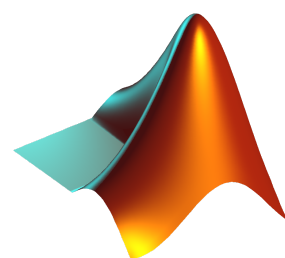
- Literate programming using a notebook interface with [Jupyter](#): 
- Creation of interactive computational analysis documents (not just scripts):
 - Facilitates reproducibility
 - Makes analyses more accessible (anyone can recreate your figure)
- Easy to contribute back to the "open source" scientific ecosystem

Can't I just use Matlab?



End To End development to execution	Slightly more code required than Matlab to do same procedures
Vast array of open source scientific packages	Quality of third-party packages varies quite a bit
Great for general programming and application development	Package management (dependencies) can be difficult (getting easier with tools like conda)
Can be a “glue” language to connect R, C++, and others	IDE is not “built-in” like Matlab
Free (as in speech and as in beer)!	By default, some operations are slower

VS.



Scripts tend to be short due to high level of integration	Can not execute code; must be compiled into another language at runtime
Easy visualization and GUI development	Expensive! Must be licensed per-user (\$\$)
Well-tested and supported (commercial product)	Does not integrate well with other languages large projects are <i>extremely</i> difficult to manage
Built-in IDE and powerful debugger	Much functionality locked away in extra licensed packages (\$)
Multi-threading easier than in Python	Cannot develop stand-alone applications (must distribute Matlab runtime)

Adapted from: <https://towardsdatascience.com/r-vs-python-vs-matlab-vs-octave-c28cd059aa69>

File Exchange is good, but...

MathWorks®

Products

Solutions

Academia

Support

Community

Events

Get MATLAB

File Exchange

Search File Exchange

File Exchange

MATLAB Central

Files

Authors

My File Exchange

Contribute

About

Filter by Source

☐ Community

35,621

☐ MathWorks

260

Filter by Category

Using MATLAB

Language Fundamentals

879

Data Import and Analysis

1,056

Mathematics

1,391

Graphics

1,857

Programming

366

App Building

409

Software Development Tools

144

External Language Interfaces

431

Environment and Settings

119

Installation, Licensing, and Activation

10

Parallel Computing

135

Application Deployment

60

Report Generation

63

Applications

Science and Industry

3,280

Image Processing and Computer Vision

2,450

Data Analytics and Machine Learning

1,492

Signal Processing and Wireless Communications

2,271

Mathematics and Optimization

930

Control Systems

891

Robotics and Autonomous Systems

240

Hardware Interfacing and IoT

605

Games

503

Using Simulink

Simulink Fundamentals

435

Most Recent

Benford's Law

A framework for Benford's Law conformity assessment.

2 Downloads

csearch

Fast C-MEX for Array Lookups and Masking

0 Downloads

.c81-File-Parser

Parser for .c81 airfoil table files.

0 Downloads

castor-matlab

Matlab class to connect to castor API

0 Downloads

Schwarzschild black hole simulation

Simulation of observers moving and emitting light in a Schwarzschild spacetime, in either Schwarzschild

5 Downloads

Structure fields to variables

Code writing tool for importing/exporting workspace variables to or from a struct.

26 Downloads

Show All

Community Toolboxes

GUI Layout Toolbox

Layout manager for MATLAB graphical user interfaces

892 Downloads

Simulink Onramp

Learn the basics of how to create, edit, and simulate Simulink models through an interactive tutorial.

829 Downloads

Numerical Computing with MATLAB

Toolbox containing files and app from Numerical Computing with MATLAB

505 Downloads

PIVlab - particle image velocimetry (PIV) tool

Easy to use, GUI based tool to analyze, validate, postprocess, visualize and simulate PIV data.

391 Downloads

Feature Selection Library

Feature Selection Library (MATLAB Toolbox)

391 Downloads

"JSON": MATLAB

A toolbox to encode/decode JSON and UBJSON files in MATLAB/Octave

219 Downloads

Show All 1,144

NIST

MATERIAL MEASUREMENT LABORATORY

Distributing code is simple in the Python community

(and integrated directly into Python tools)

-  **GitHub**: Community standard is to (at least) release code on GitHub (or similar service)

-  : Placing into the PyPI, the **P**ython **P**ackage **I**ndex enables installing with simple:

- `pip install my-awesome-package`

-  : Anaconda enables multiple environments and complex dependency management

- `conda install my-awesome-package`

What does it mean to be "open source"?

...something people can modify and share because its design is publicly accessible

— opensource.com

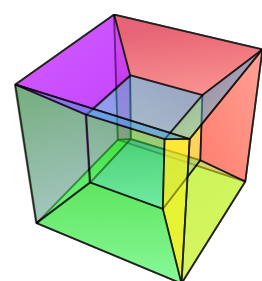
What about "open science"?

...science must be done in an open, and reproducible fashion where all components of research are open

— [Marcus Hanwell \(also opensource.com\)](#)

Open-source in the microscopy ecosystem:

General Purpose		Others	
HyperSpy	http://hyperspy.org/	PyQSTEM	https://github.com/jacobjma/PyQSTEM
Nion Swift	https://nionswift.readthedocs.io/en/stable/	HRTEMFringe Analyzer	https://github.com/ialxn/HRTEMFringeAnalyzer
pycroscopy	https://github.com/pycroscopy/pycroscopy	Atomap	https://atomap.org/
Pixelated STEM		Tomography	
pyXem	https://pyxem.github.io/pyxem/	tomopy	https://tomopy.readthedocs.io/en/latest/
pixStem	https://pixstem.org/	tomotools	https://github.com/AndrewHerzing/tomotools
LiberTEM	https://github.com/LiberTEM/LiberTEM	tomviz	https://tomviz.org/
fpd	https://gitlab.com/fpdpy/fpd/		

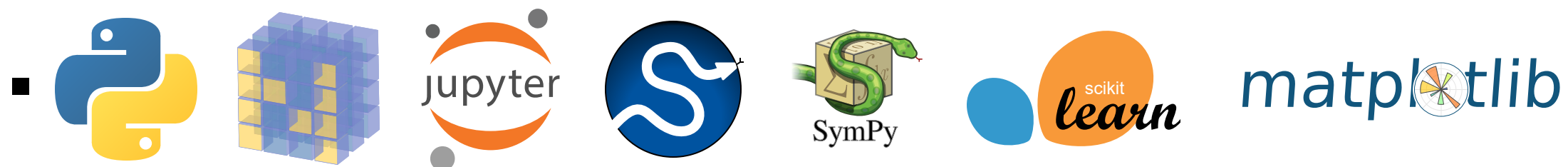


What is HyperSpy, anyway?

What is HyperSpy ?

multi-dimensional data analysis

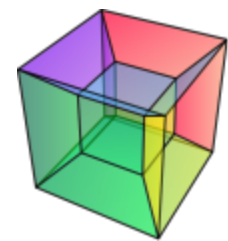
- Open-source Python library for interactive data analysis of multi-dimensional datasets
- Leverages the Scientific Python ecosystem for much of its functionality:



- Accessed like any other Python library:

```
import numpy as np      ----->      import hyperspy.api as hs
```

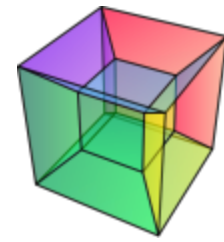
Why would you use



HyperSpy
multi-dimensional data analysis ?

- Makes it easy to operate on multi-dimensional arrays as you would a single spectrum (or image)
- Easy access to cutting-edge signal processing tools
- Modular structure makes it easy to add custom features
- Use of Jupyter notebooks encourages reproducible and sharable analyses (FAIR data)

Why would you use

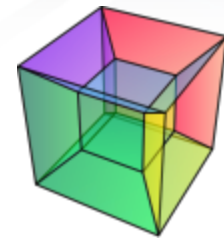


HyperSpy
multi-dimensional data analysis

?

- Beyond generic signal processing, provides many tools specifically for electron microscopy:
 - Provides facilities for easy access to proprietary software formats
 - EDS:
 - Background removal
 - Net intensity line map extraction
 - Quantification (k-factor, ζ -factors, ionization cross sections)

Why would you use



HyperSpy
multi-dimensional data analysis

?

- Beyond generic signal processing, provides many tools specifically for electron microscopy:
 - EELS:
 - Background removal
 - Curve fitting for quantification (including ELNES)
 - All of Egerton's famous methods
 - "Advanced" methods:
 - Multivariate statistical analysis
 - General curve fitting
 - Dimensionality reduction/signal separation

Why python[™] ?

- Quickly becoming the *de facto* standard of scientific computing
- Free (as in speech and as in beer)
 - No pesky licenses to checkout
- Vast array of scientific libraries available:
 - `pip install <just about anything>`
- Thanks to `numpy` and other libraries, similar (or often better) performance than MATLAB

History of HyperSpy

- Developed by [Francisco de la Peña](#) in 2007 – 2012 as part of his Ph.D. Thesis at Université Paris-Sud in Orsay, France
- See [Chapter 5](#) of the dissertation for a historical introduction
- Originally called EELSLab:

Talk EELSLab: a Python toolbox for (hyper)spectroscopy data analysis

Presented by Francisco de la Peña in Poster and Demo Session 2011

Abstract

EELSLab: a Python toolbox for (hyper)spectroscopy data analysis

F. de la Peña^{1, 3, *}, M. Sarahan², S. Mazzucco^{4, 5}, L-F. Zagonel^{1, **}, M. Walls¹

1) Laboratoire de Physique des Solides, Bât. 510, Université Paris-Sud. 91405 Orsay Cedex

2) SuperSTEM, STFC Daresbury Laboratories, Keckwick Lane, Warrington WA4 4AD - UK

3) CEA-LETI, MINATEC 17, avenue des Martyrs, 38054 GRENOBLE Cedex 9 – France

4) Center for Nanoscale Science and Technology, National Institute of Science and Technology, 100 Bureau Drive, Gaithersburg, MD 20899-6203, USA

5) Institute for Research in Electronics and Applied Physics (Bldg. 223), Paint Branch Drive, University of Maryland, College Park, MD 20742-3511, USA

* Current address: department of Materials Science and Metallurgy, University of Cambridge, Pembroke Street, Cambridge, CB2 3QZ – UK

** Current address: Associação Brasileira de Tecnologia de Luz Síncrotron, Laboratório de Microscopia Eletrônica.13083-970 - Campinas, SP - Brasil

Modern scientific instruments from several disciplines now yield multidimensional spectroscopic data. As an example, a modern transmission electron microscope (TEM) can acquire spectral data from sub-atomic volumes that ultimately could reveal the position and nature of each atom of a material [1]. However, such datasets are usually quite large and difficult to work with.

EELS_{Lab} [2] has been developed as a tool to facilitate hyperspectral data analysis. Originally it was intended for TEM data analysis, but it has been successfully used in other domains. Specifically, it provides easy access to multidimensional curve fitting, peak analysis and machine learning algorithms, as well as a viewing framework for navigating data and reading and writing capabilities for several popular hyperspectral formats.

This talk will discuss how Python has been used to implement these features, with demonstrations of applications to both spatially-resolved spectroscopic data (so-called spectrum images) [3] and to structural analysis of atomic resolution image stacks [4]. The blend of intuitiveness, power, and availability of high-quality scientific libraries that Python offers has allowed the creation of a simple, natural tool that scientists from many disciplines can both use and easily extend into new scientific domains.

References

[1] Sandra Van Aert et al., « Three-dimensional atomic imaging of crystalline nanoparticles », *Nature* 470, n°. 7334 (17th of February, 2011): 374-377.

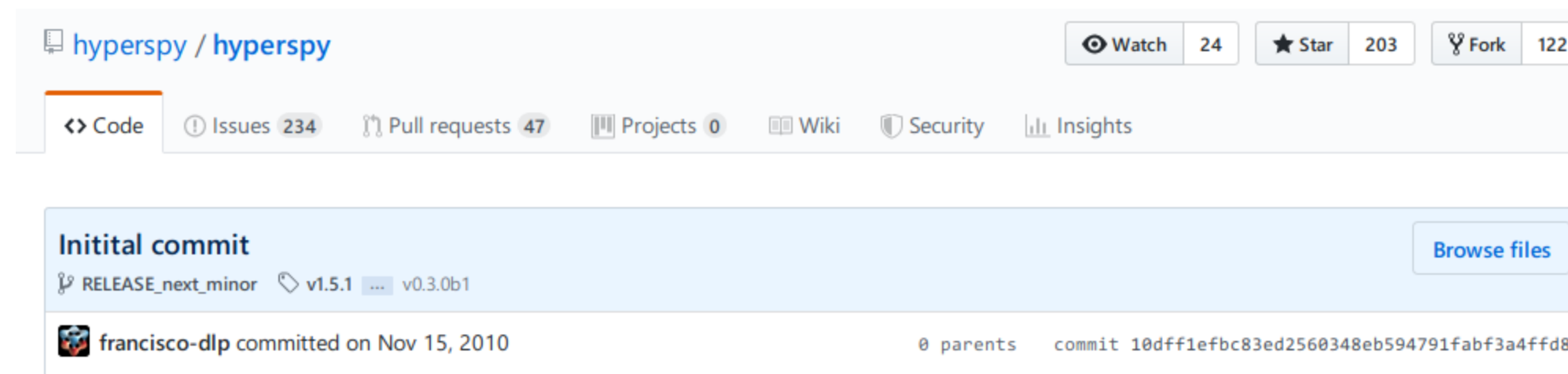
[2] <http://www.eelslab.org>

[3] R. Arenal et al., « Extending the analysis of EELS spectrum-imaging data, from elemental to bond mapping in complex nanostructures », *Ultramicroscopy* 109, n° 1 (December 2008): 32-38.

[4] Michael C. Sarahan et al., « Point defect characterization in HAADF-STEM images using multivariate statistical analysis », *Ultramicroscopy* 111, n°. 3 (February 2011): 251-257.

History of HyperSpy

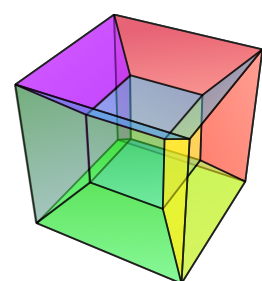
- Open-sourced (on [Github](#)) in 2010:



- Renamed to HyperSpy in 2011
- Now... over 350 citations, 37 releases, 38 contributors, 100K lines of Python code, and rapidly growing!

Design philosophy of HyperSpy

- HyperSpy is a Python library, rather than standalone program
 - Part of the greater scientific Python ecosystem
- Data storage is in an open hierarchical format (HDF5)
 - Saves all metadata by default (including most processing steps)
- Analysis typically done via reproducible notebooks
- Feature development is completely open-source ([GPLv3](#))



Welcome to the community!


A decorative header featuring a repeating pattern of light blue and white hexagons, some of which are slightly offset to create a 3D effect.

The power of community

- One of the *best* parts of HyperSpy is the community that surrounds it (personal opinion)
- This software is made by researchers, for researchers
- HyperSpy is built from collaboration:
 - By collaborating, we advance faster and avoid reinventing the wheel

How to get help?

- HyperSpy website (www.hyperspy.org):



HyperSpy multi-dimensional data analysis

[Home](#) · [Download](#) · [Documentation](#) · [Demos](#) · [News](#) · [Support](#) · [Citing](#) · [Credits](#)

HyperSpy: multi-dimensional data analysis toolbox

HyperSpy is an open source Python library which provides tools to facilitate the interactive data analysis of multi-dimensional datasets that can be described as multi-dimensional arrays of a given signal (e.g. a 2D array of spectra a.k.a spectrum image).

HyperSpy aims at making it easy and natural to apply analytical procedures that operate on an individual signal to multi-dimensional arrays, as well as providing easy access to analytical tools that exploit the multi-dimensionality of the dataset.

Its modular structure makes it easy to add features to analyze different kinds of signals.

Highlights

- Two families of named and scaled axes: *signal* and *navigation*.
- Visualization tools for multi-dimensional spectra and images.
- Easy access multi-dimensional curve fitting and blind source separation.
- Built on top of NumPy, SciPy, matplotlib and scikit-learn.
- Modular design for easy extensibility.

VERSIONS

Stable
1.5.1
[Documentation](#)
[Demos](#)
[Known issues](#)

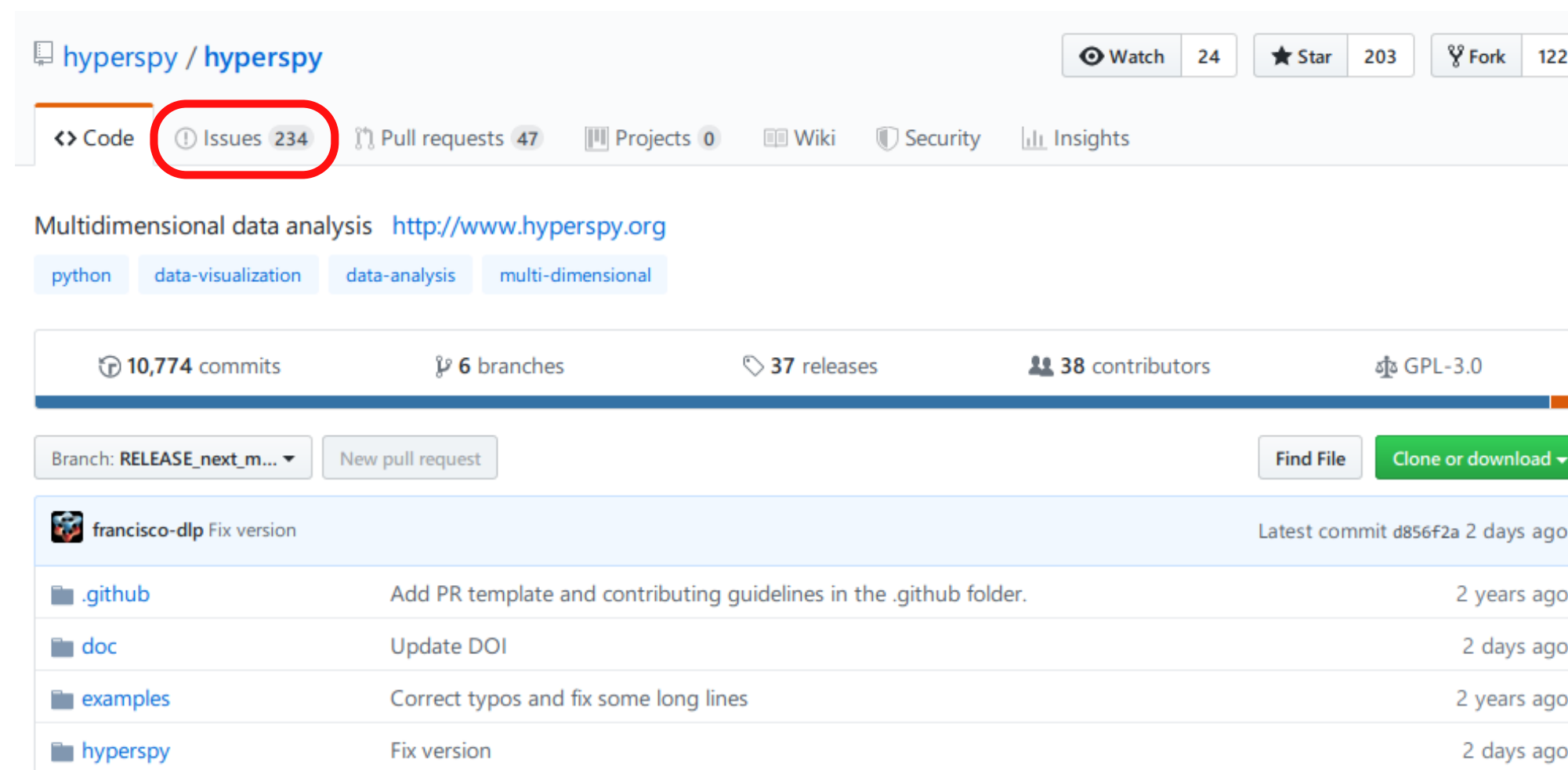
Development
[View on Github](#)
[Documentation](#)

SUPPORT

[Issue tracker](#)
[Mailing list](#)
[Gitter chat](#)

How to get help?

- HyperSpy repository (<https://github.com/hyperspy/hyperspy>):



hyperspy / hyperspy

Watch 24 Star 203 Fork 122

Code Issues 234 Pull requests 47 Projects 0 Wiki Security Insights

Multidimensional data analysis <http://www.hyperspy.org>

python data-visualization data-analysis multi-dimensional

10,774 commits 6 branches 37 releases 38 contributors GPL-3.0

Branch: RELEASE_next_m... New pull request Find File Clone or download

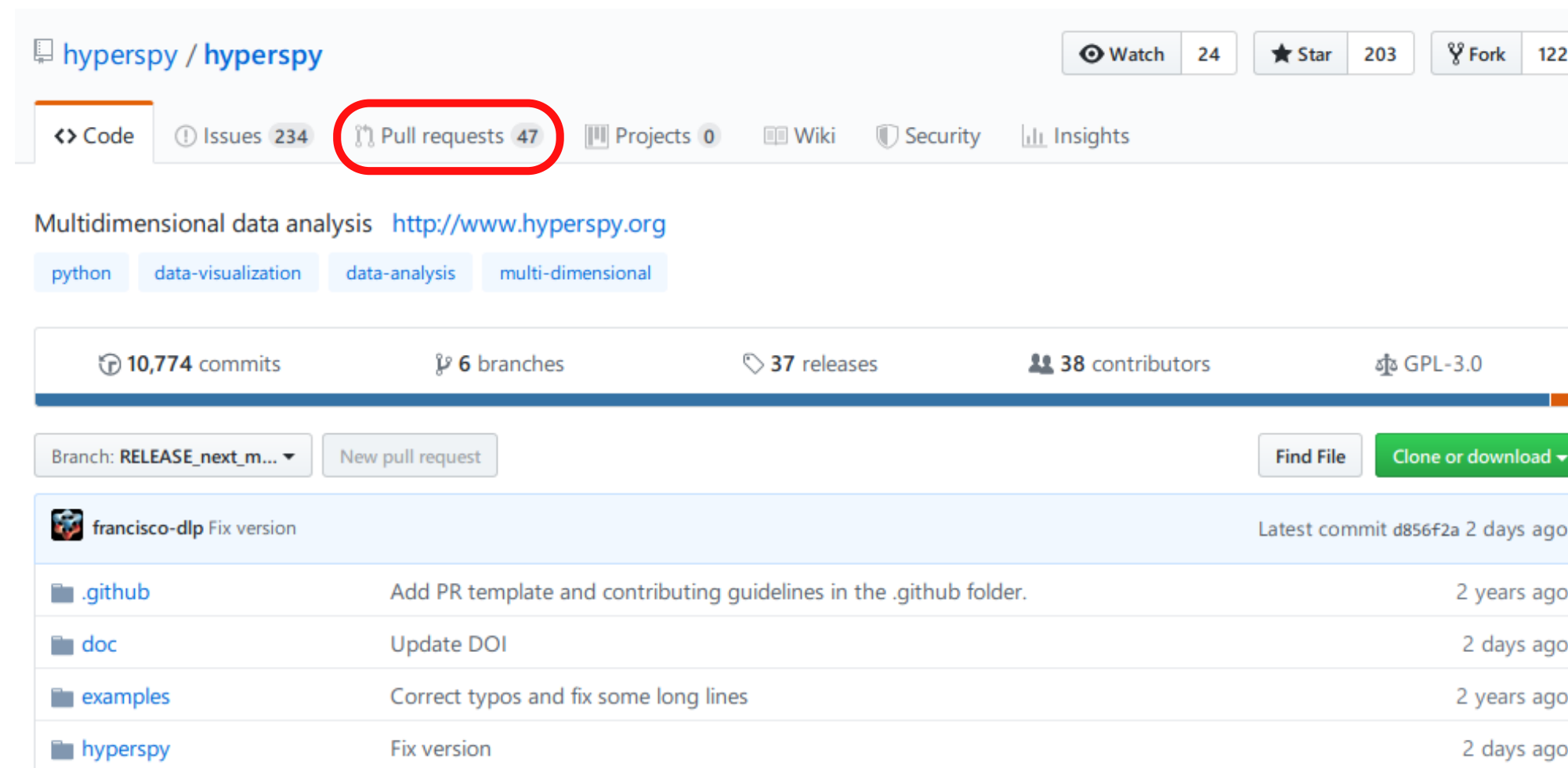
francisco-dlp	Fix version	Latest commit d856f2a 2 days ago
.github	Add PR template and contributing guidelines in the .github folder.	2 years ago
doc	Update DOI	2 days ago
examples	Correct typos and fix some long lines	2 years ago
hyperspy	Fix version	2 days ago

How to get (more) help?

- Sessions like this (good job!)
- Expansive user guide and documentation:
http://hyperspy.org/hyperspy-doc/current/user_guide/index.html
- Tutorials and demos: <https://github.com/hyperspy/hyperspy-demos>
- User group list: hyperspy-users@googlegroups.com
- Gitter chat: <https://gitter.im/hyperspy/hyperspy>
- Developer guide (if you're into that sort of thing):
http://hyperspy.org/hyperspy-doc/current/dev_guide/index.html#dev-guide-label

Anyone can make HyperSpy better!

- HyperSpy repository (<https://github.com/hyperspy/hyperspy>):



hyperspy / hyperspy

Watch 24 Star 203 Fork 122

Code Issues 234 Pull requests 47 Projects 0 Wiki Security Insights

Multidimensional data analysis <http://www.hyperspy.org>

python data-visualization data-analysis multi-dimensional

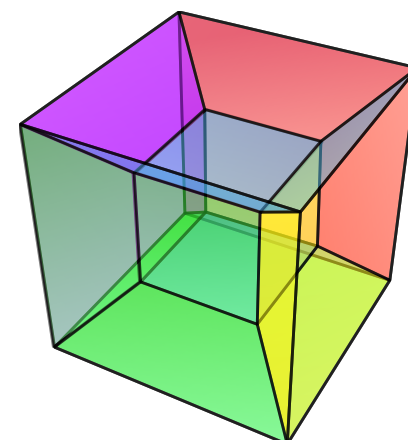
10,774 commits 6 branches 37 releases 38 contributors GPL-3.0

Branch: RELEASE_next_m... New pull request Find File Clone or download

francisco-dlp Fix version	Latest commit d856f2a 2 days ago
.github	Add PR template and contributing guidelines in the .github folder. 2 years ago
doc	Update DOI 2 days ago
examples	Correct typos and fix some long lines 2 years ago
hyperspy	Fix version 2 days ago

What about general Python questions?

- There are vast numbers of resources about Python online (and in print):
 - Official documentation: <https://docs.python.org>
 - Getting started with scientific Python: <https://scipy-lectures.org>
 - NumPy for Matlab users:
<https://docs.scipy.org/doc/numpy/user/numpy-for-matlab-users.html>
- StackOverflow for general programming/Python questions: <https://stackoverflow.com/>



Thank you!

Any questions?

joshua.taillon@nist.gov