MANCHESTER 1824

The University of Manchester

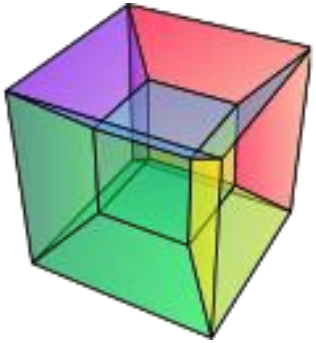
SuperSTEM DARESBURY

Sustainable open-source software development with HyperSpy

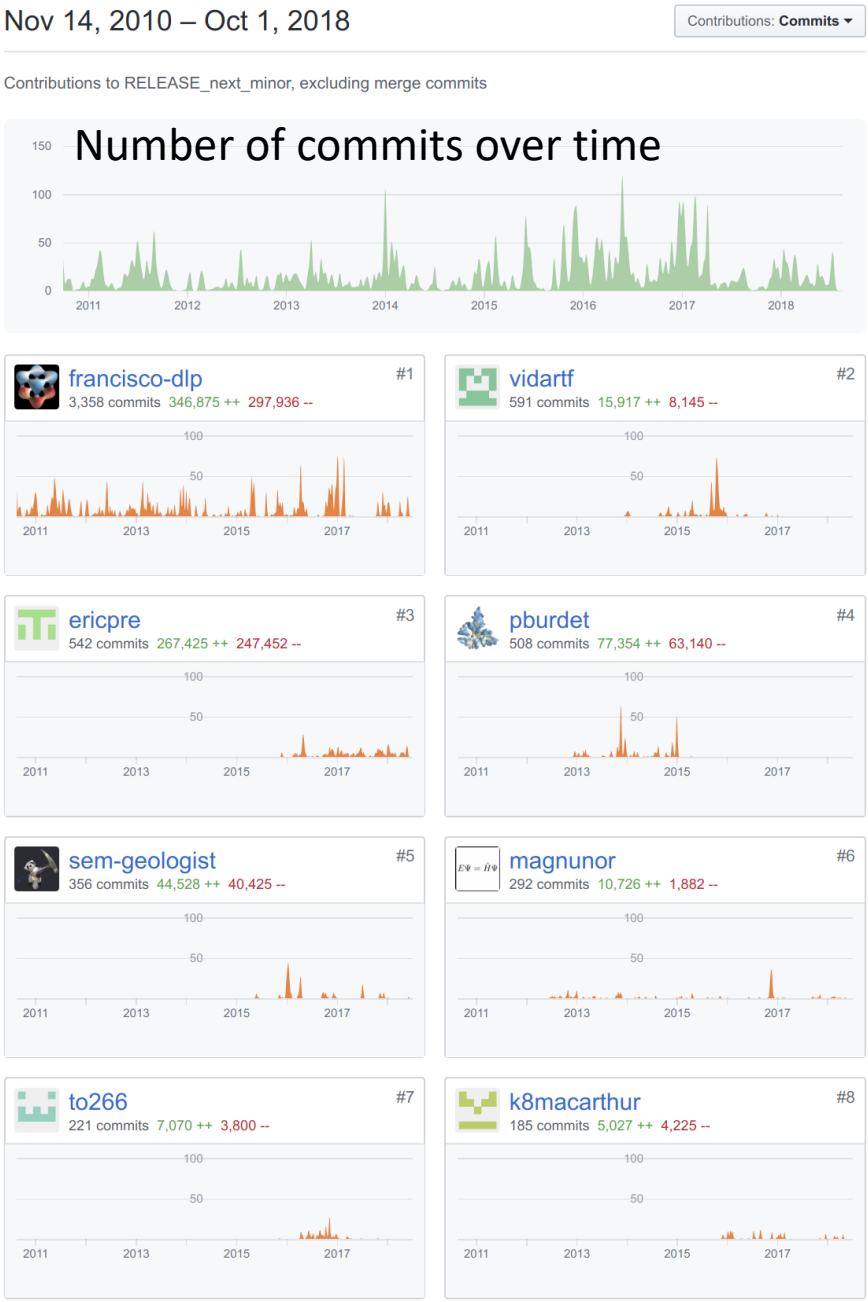
Eric Prestat[1,2]

[1]School of Materials, University of Manchester, Manchester, UK

[2]SuperSTEM Laboratory, SciTech Daresbury Campus, Daresbury, UK



- What is HyperSpy?
- What HyperSpy has to offer?
- Where shall we go as an open source scientific community?



Francisco de la Peña[1,2,3,4], Tomas Ostasevicius[1], Vidar Tonaas Fauske[5], Pierre Burdet[1], **Eric Prestat[6,7]**, Petras Jokubauskas[8] , Magnus Nord[5,9,10], Mike Sarahan[7], Katherine E. MacArthur[11], Duncan N. Johnstone[1], Joshua Taillon[12], Alberto Eljarrat[13,14], Vadim Migunov[11], Jan Caron[11], Tom Furnival[1], Stefano Mazzucco[2], Thomas Aarholt[15,16], Michael Walls[2], Florian Winkler[11], Gaël Donval[4,18], Robert McLeod[19], Thomas Slater[6]

[1] Department of Materials Science and Metallurgy, University of Cambridge, Cambridge, U.K.
[2] Laboratoire de Physique des Solides, Paris-Sud University, Orsay, France
[3] Unité Matériaux et Transformations, University Lille1, Lille, France
[4] CEA Grenoble, Grenoble, France
[5] Department of Physics, NTNU, Trondheim, Norway
[6] School of Materials, The University of Manchester, Manchester, U.K.
[7] SuperSTEM, SciTech Daresbury Campus, Warrington, U.K.
[8] Institute of Geochemistry, Mineralogy and Petrology, University of Warsaw, Poland
[9] School of Physics and Astronomy, University of Glasgow, Glasgow, U.K.
[10] EMAT, University of Antwerp, Antwerp, Belgium
[11] Forschungszentrum Jülich GmbH, Jülich, Germany
[12] Material Measurement Laboratory, NIST, Gaithersburg, MD, U.S.A.
[13] Laboratory of Electron NanoScopies, Universitat de Barcelona, Barcelona, Spain
[14] Department of Physics, Humboldt University of Berlin, Newtonstr. 15, 12489 Berlin, Germany
[15] Department of Materials, University of Oxford, Oxford, United Kingdom
[16] Department of Physics, University of Oslo, Oslo, Norway
[18] Institut des Matériaux Jean Rouxel (IMN), Université de Nantes, Nantes, France
[19] Entropy Reduction Algorithmics, Victoria, Canada

SuperSTEM DARESBURY

+ many more: 35 contributors (~10 is a one off)

The University of Manchester

'Enhanced Data Generated by Electrons'

Following the very successful workshops at Lake Tahoe, Leukerbad, Port Ludlow, Guadeloupe, Grundlsee and Banff, the next international workshop on electron energy loss spectroscopy and imaging will be held in

Sainte Maxime, France,

- Francisco's talk on HyperSpy
  - Mostly about open source and scientific python community
  - NOT about the cool stuff that HyperSpy could already do at the time

SuperSTEM DARESBURY

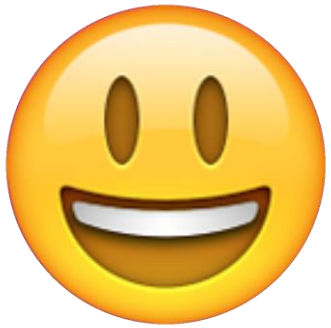
The University of Manchester

'Enhanced Data Generated by Electrons'

Following the very successful workshops at Lake Tahoe, Leukerbad, Port Ludlow, Guadeloupe, Grundlsee and Banff, the next international workshop on electron energy loss spectroscopy and imaging will be held in

Sainte Maxime, France,

- Francisco's talk on HyperSpy
  - Mostly about open source and scientific python community
  - NOT about the cool stuff that HyperSpy could already do at the time

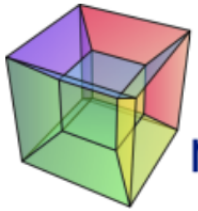


What I did not understand at the time:

- Open source is not only about opening the code but also about community
- *Small* paradigm shift required

SuperSTEM DARESBURY

The University of Manchester



HyperSpy
multi-dimensional data analysis

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.

Originally develop by electron microscopists but designed to analyse any kind of multi-dimensional data

SuperSTEM DARESBURY

http://hyperspy.org

HyperSpy's core concept

The University of Manchester

- Signal object
  - Signal dimension
  - Navigation dimension
  - Slicing
  - Generic:
    - Signal1D, Signal2D
  - Specialised:
    - EDS, EELS, etc.
- Syntax
  - Consistent API
  - Try to be not too verbose for the user

```
data = (np.arange(5*10*20*50).reshape(5, 10, 20, 50))
s = hs.signals.Signal1D(data)
s
```

```
<Signal1D, title: , dimensions: (20, 10, 5|50)>
```

```
s.T
```

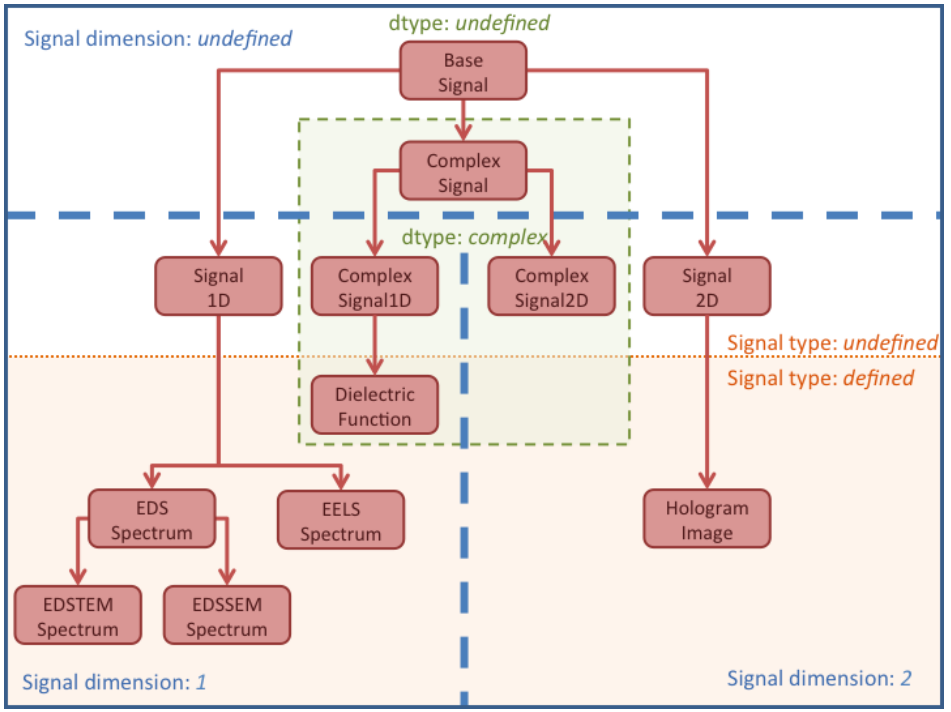
```
<BaseSignal, title: , dimensions: (50|20, 10, 5)>
```

```
s.inav[1:15, ...]
```

```
<Signal1D, title: , dimensions: (14, 10, 5|50)>
```

```
s.isig[:25]
```

```
<Signal1D, title: , dimensions: (20, 10, 5|25)>
```



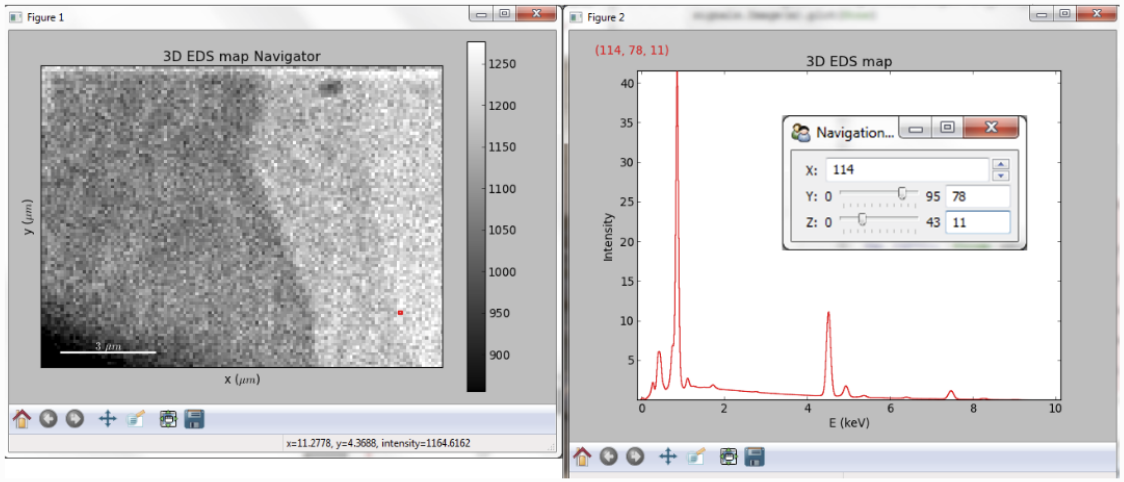
SuperSTEM DARESBURY

HyperSpy main features

The University of Manchester

- Input/Output support almost all format in electron microscopy
- Data visualisation and interactivity rely on matplotlib

```
>>> s = hs.load('Ni_superalloy_0*.rpl', stack=True).as_signal1D(0)
>>> s.plot()
```



Visualisation of a 3D spectrum image with sliders.

Supported file formats

Format	Read	Write	lazy
Gatan's dm3	Yes	No	Yes
Gatan's dm4	Yes	No	Yes
FEI's emi and ser	Yes	No	Yes
HDF5	Yes	Yes	Yes
Image: jpg	Yes	Yes	Yes
TIFF	Yes	Yes	Yes
MRC	Yes	No	Yes
MRCZ	Yes	Yes	Yes
EMSA/MSA	Yes	Yes	No
NetCDF	Yes	No	No
Ripple	Yes	Yes	Yes
SEMPER unf	Yes	Yes	Yes
Blockfile	Yes	Yes	Yes
DENS heater log	Yes	No	No
Bruker's bcf	Yes	No	Yes
Bruker's spx	Yes	No	No
EMD (NCEM)	Yes	Yes	Yes
EMD (FEI)	Yes	No	Yes
Protochips log	Yes	No	No
EDAX .spc and .spd	Yes	No	Yes

SuperSTEM DARESBURY

HyperSpy main features : machine learning

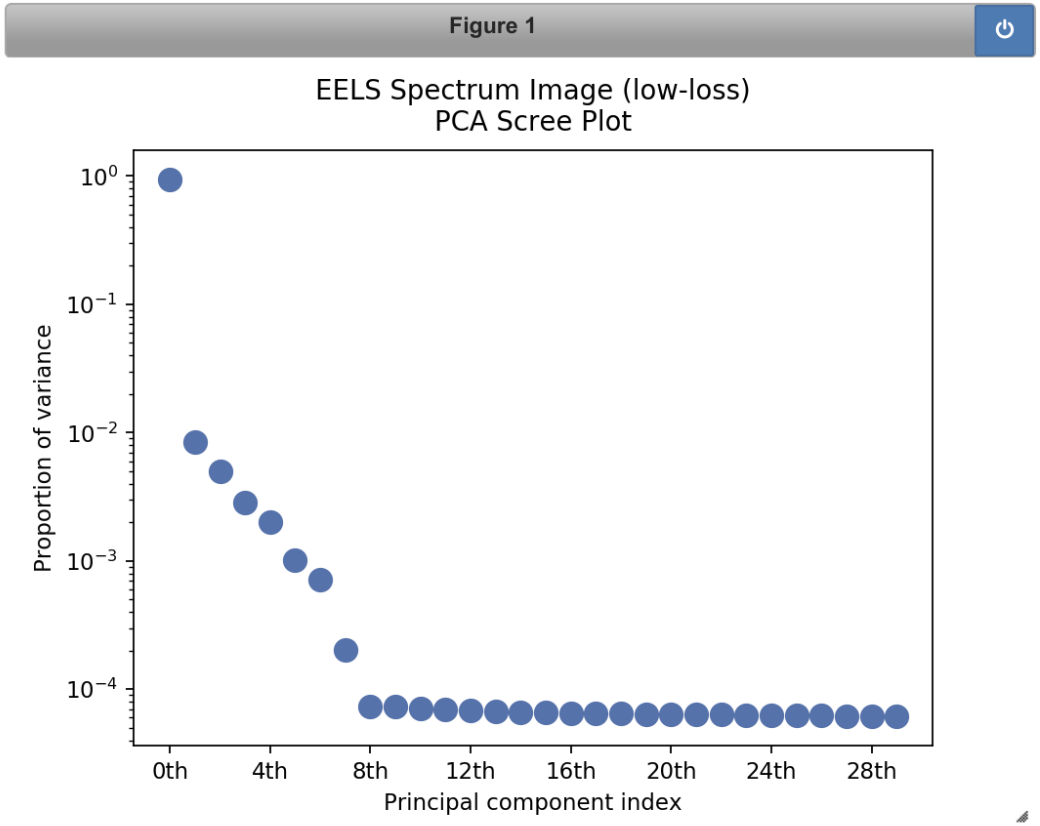
The University of Manchester

- SVD, BSS, NMF
  - from scikit-learn

- Robust PCA to deal with outliers
- “online” algorithm for big data

```
s1.decomposition(True)
```

```
s1.plot_explained_variance_ratio()
```




```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb26d832ef0>
```

```
s1_denoised = s1.get_decomposition_model(9)
```

SuperSTEM DARESBURY

HyperSpy main features: multidimensional curve-fitting

The University of Manchester

- Any model can be build (1D, 2D)
- Model interactivity

```
m = s.create_model()

m.append(hs.model.components.Gaussian())
m.append(hs.model.components.Polynomial(5))
m.append(hs.model.components.EELSCLEdge('C_K'))
m.append(hs.model.components.ScalableFixedPattern(s.deepcopy(

m.notebook_interaction()
```

background

Gaussian

Polynomial

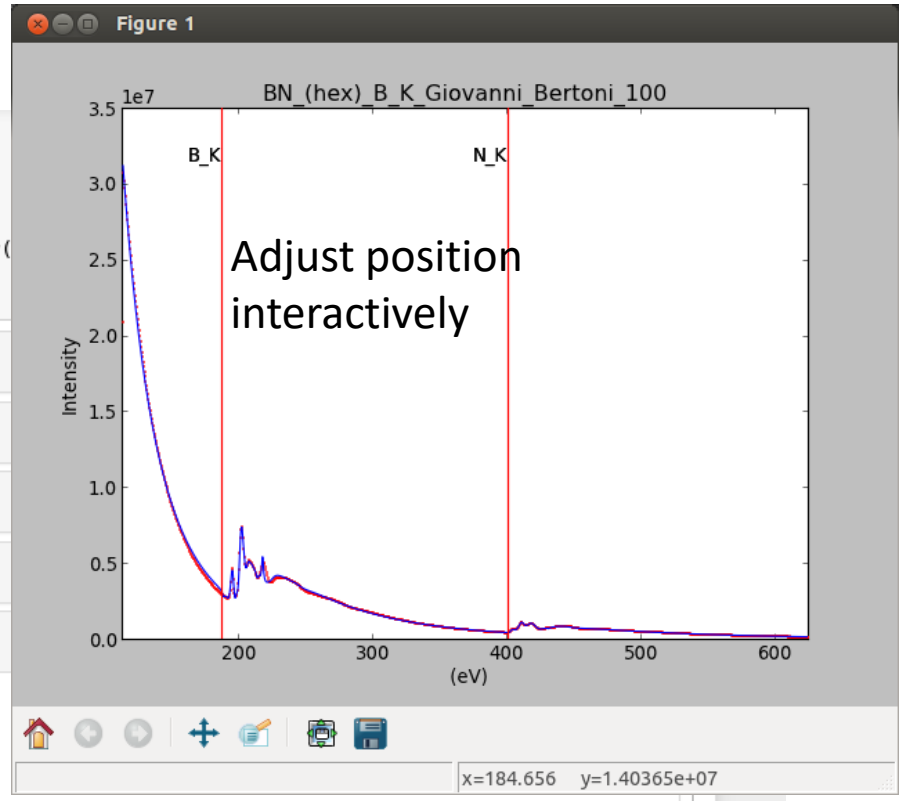
C_K

ScalableFixedPattern

active

interpolate

min	-9	yscale	1.00	max	11
min	-9	xscale	1.00	max	11
min	0	shift	0.00	max	499



SuperSTEM DARESBURY

HyperSpy main features: multidimensional curve-fitting

The University of Manchester

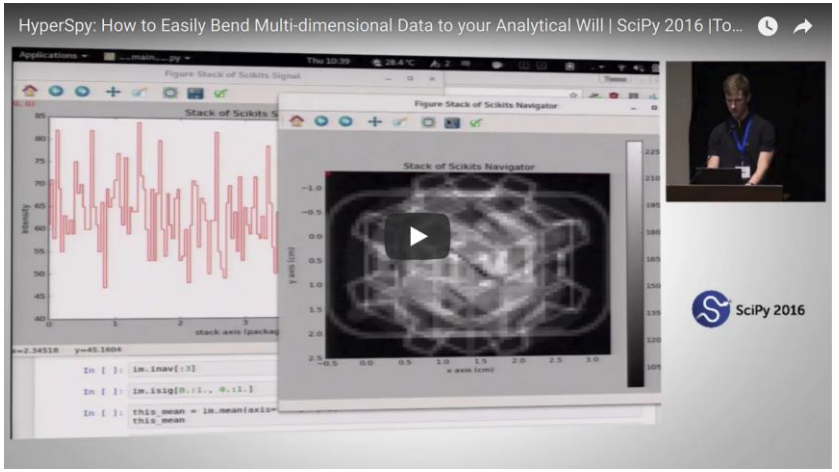
- Any model can be build (1D, 2D)
- Model interactivity
- Write your own expression

Create a 2D component from a string, which is turned into a function with Sympy

```
>>> g = hs.model.components2D.Expression(
... "k * exp(-((x-x0)**2 / (2 * sx ** 2) + (y-y0)**2 / (2 * sy ** 2)))",
... "Gaussian2d", add_rotation=True, position=("x0", "y0"),
... module="numpy", )
```

- Smart Adaptive Multi-dimensional Fitting (SAMFire)

https://doi.org/10.1002/9783527808465.EMC2016.6233



https://youtu.be/kVlf3bMZcsc

SuperSTEM DARESBURY

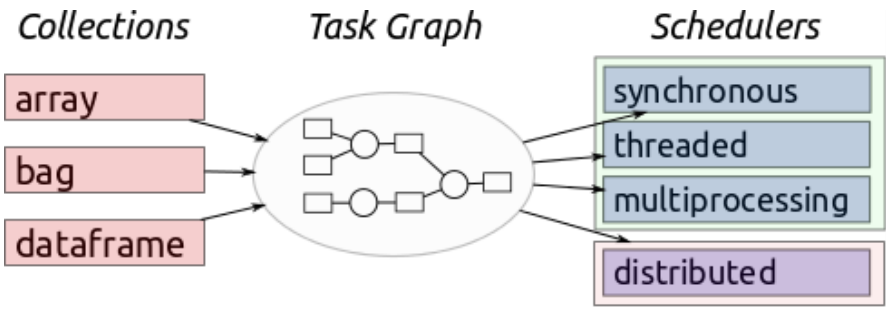
HyperSpy main features: out-of-core computation

The University of Manchester

- Perform operation “lazily”
- Delay the execution until the result is requested and then perform them in a blocked fashion

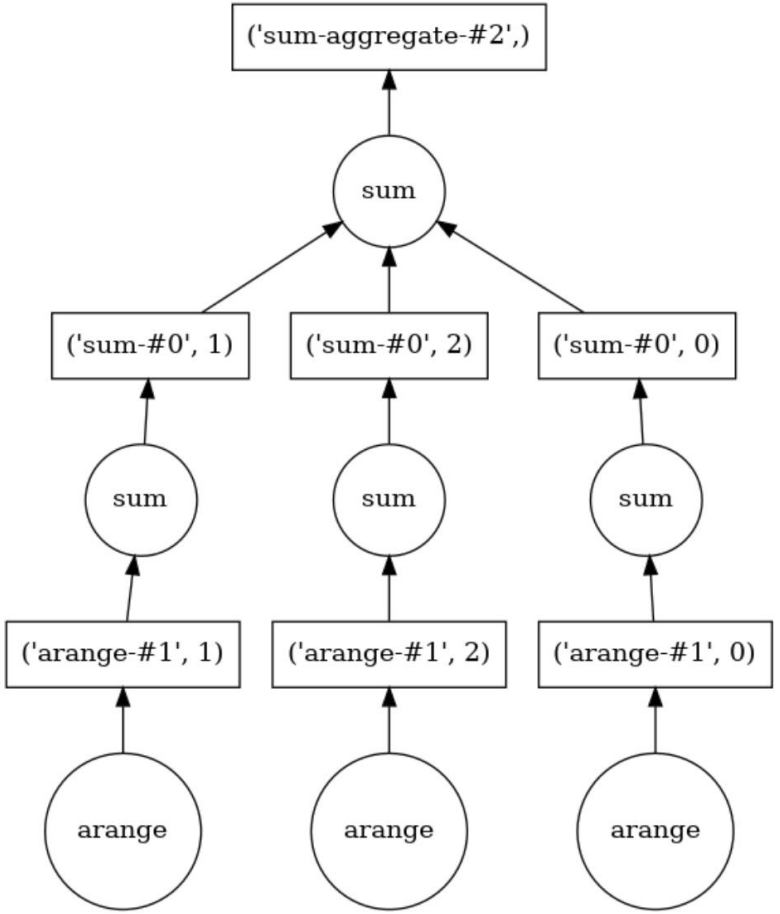
DASK

```
In [1]: import dask.array as da

In [2]: a = da.arange(30, chunks=(10,)).sum()

In [3]: a.visualize()
```



http://dask.pydata.org

Out[3]:



SuperSTEM DARESBURY

The University of Manchester

Performance optimisation

- Numba
  - JIT compilation


```python
@jit_ifnumba()
def _fill_array_with_stream_sum_frames(spectrum_image, stream,
                                       first_frame, last_frame, rebin_energy=1):
    # jit speeds up this function by a factor of ~ 30
    navigation_index = 0
    frame_number = 0
    shape = spectrum_image.shape
    for count_channel in np.nditer(stream):
        # when we reach the end of the frame, reset the navigation index to 0
        if navigation_index == (shape[0] * shape[1]):
            navigation_index = 0
            frame_number += 1
            # break the for loop when we reach the last frame we want to read
            if frame_number == last_frame:
                break
        # if different of '65535', add a count to the corresponding channel
        if count_channel != 65535:
            if first_frame <= frame_number:
                spectrum_image[navigation_index // shape[1],
                               navigation_index % shape[1],
                               count_channel // rebin_energy] += 1
        else:
            navigation_index += 1
```

Particularly interesting when vectorisation is not possible

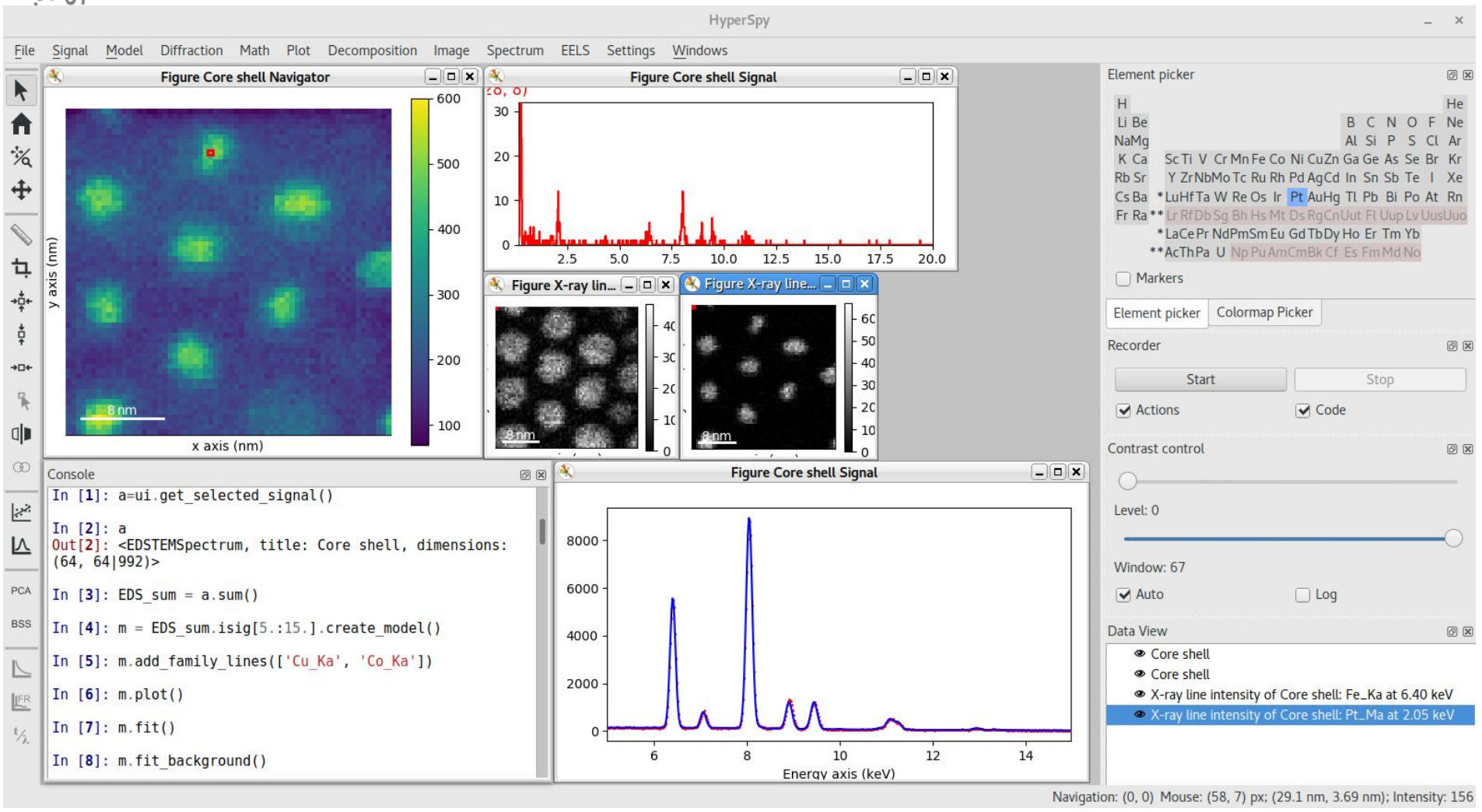
- Numexpr (CPU cache optimisation for numpy)

```python
# algebraic expression:
z = 2*y + 4*x

# transcendental expression:
z = np.sin(x)**2 + np.cos(y)**2
```

```
NumPy version: 1.15.1
Time for an algebraic expression:     0.343 s / 3.263 GB/s
Time for a transcendental expression: 0.597 s / 1.873 GB/s
Numexpr version: 2.6.8. Using MKL: True
Time for an algebraic expression:     0.061 s / 18.457 GB/s
Time for a transcendental expression: 0.078 s / 14.399 GB/s
```

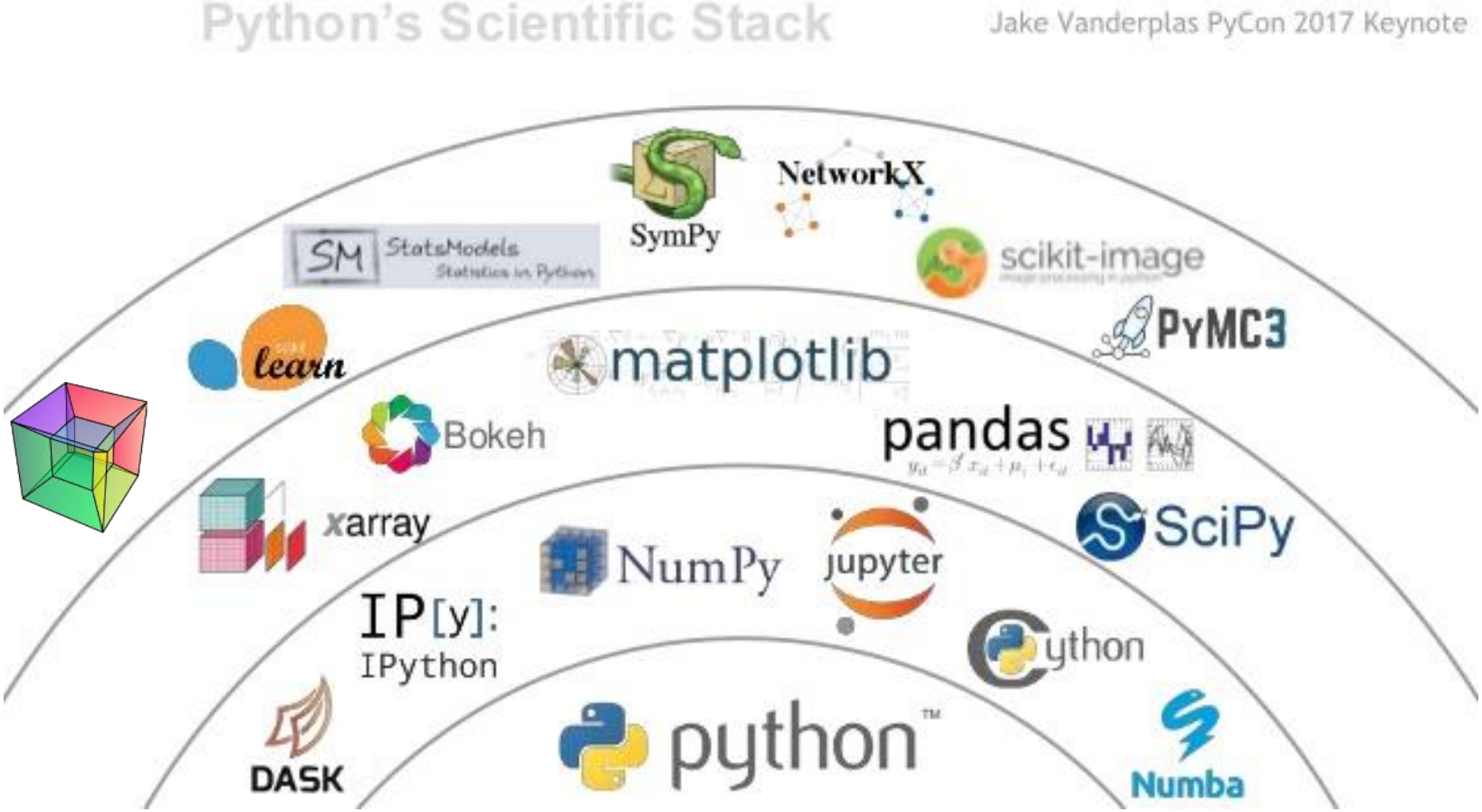
SuperSTEM DARESBURY

User interface: HyperSpyUI



SuperSTEM DARESBURY

Python scientific stack: each library find its natural position



- Strength of the major libraries for the user:
  - work nicely with each other
  - Similarities in the syntax

Current HyperSpy ecosystem

The University of Manchester

HyperSpyUI

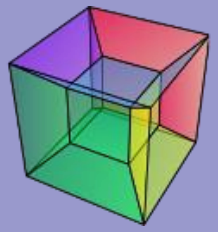
HyperSpy-ipywidget

HyperSpy-traitsui

EELS

holography

Big data



EDS

Multi-dimensional

IO

Visualisation and interactivity

Machine learning

Curve-fitting

Strain mapping

Orientation mapping

pyXem

Electron cristallography

Scanning Electron diffraction

4D STEM

Magnetic measurement

pixSTEM

Differential contrast imaging

Atomic position fitting

Atomap

Quantitative STEM

Strain mapping

SuperSTEM DARESBURY

The University of Manchester

Future HyperSpy ecosystem and other relevant libraries

EELS

IO

EDS

Holography

HyperSpyUI

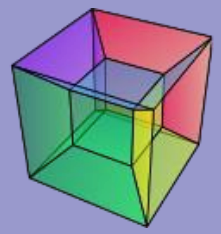
HyperSpy-ipywidget

HyperSpy-traitsui

Multi-dimensional

Curve-fitting

Visualisation and interactivity



Big data

Machine learning

Nion Swift

GMS 3

LiberTEM

tomviz

pycroscopy

Strain mapping

Orientation mapping

pyXem

Electron cristallography

Scanning Electron diffraction

4D STEM

Magnetic measurement

pixSTEM

Differential contrast imaging

Atomic position fitting

Atomap

Quantitative STEM

Strain mapping

Other extension

SuperSTEM DARESBURY

- HyperSpy managed to built a *distributed* community of users/contributor
  - Led by its own contributors
  - Does not depend of any academic group/institution
  - Decision based on contributor consensus
- Motivate (at least not discourage) users
  - Pay attention to users feedback
  - From user to contributor: make the learning curve easier
- What are contributors doing?
  - Contribution to user guide, tutorials and online discussion
  - Code writing and/or code review

This is one way to make a library sustainable

As individual is it worth contributing to HyperSpy?

The University of Manchester

- HyperSpy acknowledgement through zenodo DOI
  - One DOI for each release (important for reproducibility)
  - New contributors will get acknowledged
  - Google scholar not *yet* fully compatible with zenodo DOI (work in progress)
- Contribution to HyperSpy (or any other library) can be useful for career development
  - Github profil can be used as linkedin, etc.
  - Recognition by the microscopy community
- As PI/group leader
  - PI are not acknowledged but they can benefit a lot from the expertise gained through HyperSpy
  - Fairly useful “training” for post-doc/student

- Bundle installer for windows
  - Based on winpython (fairly well tested by the community)
  - Build on appveyor (one click on github to make a release)
  - Independent from HyperSpy release
  - Easy installer (NSIS 2)
  - Can package any python library easily (with C code)
- Bundle installer for mac and linux
  - Use the same tool that build anaconda and miniconda
  - One current issue to fix… coming anytime soon?
- Offer a toolbox
  - Generic multidimensional processing
  - Fitting, big data, IO, etc
  - `map` function (apply any processing on the signal dimension)

Concluding remarks: what would be ideal for the user?

The University of Manchester

- Good intercompatibility between softwares/libraries; integrated workflow
  - Similar syntax, data format, etc.
  - Users should be able to switch easily from one library to another
- Run the same code in different environment
  - Jupyter notebook or qtconsole
  - HyperSpyUI, Nion Swift, GMS, etc.
  - Supercomputer (High-performance computing)
- Code implemented at the right position of the python stack
  - Usually implementation improve significantly during review
  - Code exposed to larger users base -> better feedback

Thanks for your attention

SuperSTEM DARESBURY