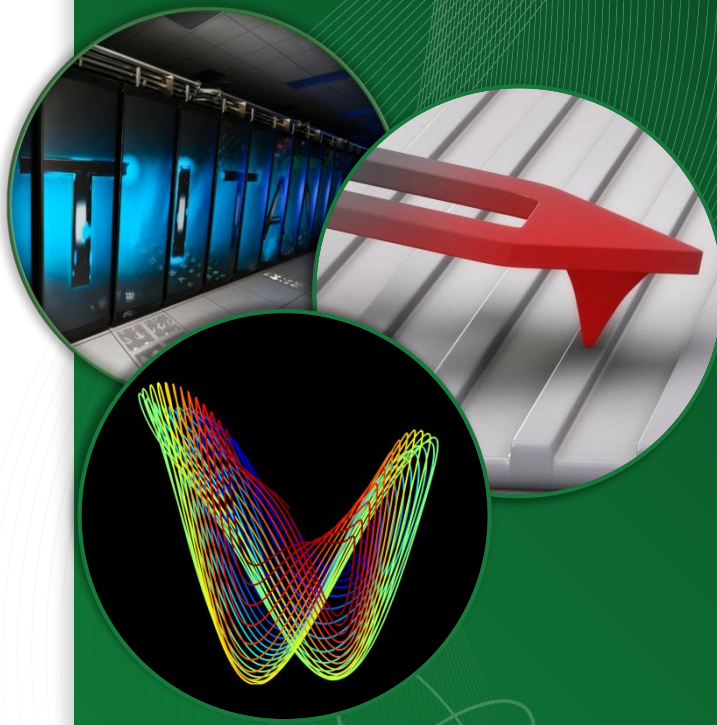
pyUSID & pycroscopy

Suhas Somnath, Chris R. Smith,
Nouamane Lanait, Stephen Jesse



OAK RIDGE National Laboratory | OAK RIDGE LEADERSHIP COMPUTING FACILITY

INSTITUTE FOR FUNCTIONAL IMAGING OF MATERIALS
OAK RIDGE NATIONAL LABORATORY | CENTER FOR NANOPHASE MATERIALS SCIENCES

ORNL is managed by UT-Battelle
for the US Department of Energy

Multitude of Instruments

Micro Raman Microscope

Atomic Force Microscope (AFM)

AFM with Infrared spectroscopy (AFM-IR)

Scanning Tunneling Microscope (STM)

Scanning Transmission Electron Microscope (STEM)

AFM with Raman spectroscopy

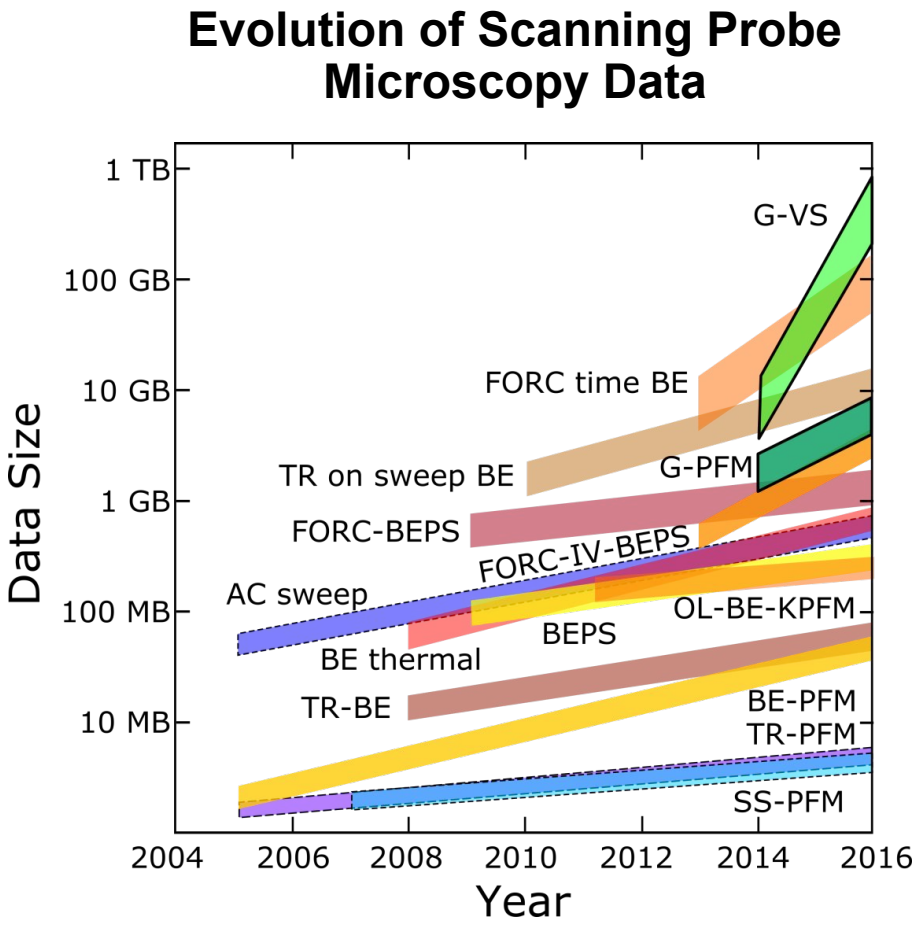
What we wanted


Instrument Tier

?

Interactive visualization, analysis on supercomputers

Growing Data Sizes and Dimensionality



- Data sizes have grown from ~ 10 MB to ~ 1 TB in 10 years!
- Dimensionality ranges from 1D spectra to 7D hyperspectral datasets
- Cannot use laptops to analyze data

Kalinin et al., *ACS Nano*, 9068-9086, 2015

Instrumentation Software Inadequate for Analysis

- Software provided for controlling instruments typically only comes with basic data analysis capabilities
- Integrating user-developed functionality often impossible

Multitude of File Formats

.wdf

.ibw

.cdb

- Proprietary
- Incompatible

≠

.asc

.dm3

.mat

Suhas Somnath, somnaths@ornl.gov

Disjoint & Unorganized Communities

- Clustering
- Fit spectra …

- Filter Image
- Register Image …

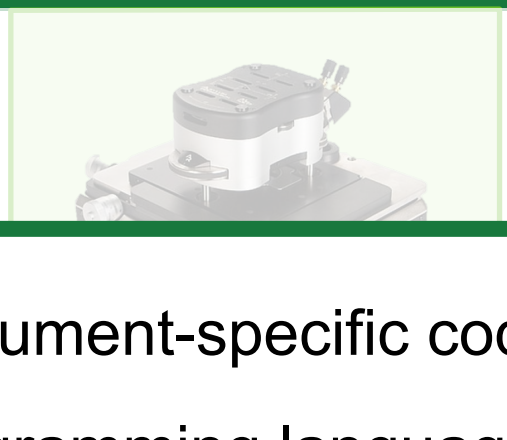
- Fit Spectra
- SVD Filtering …

- FFT Filtering
- SVD Filtering …

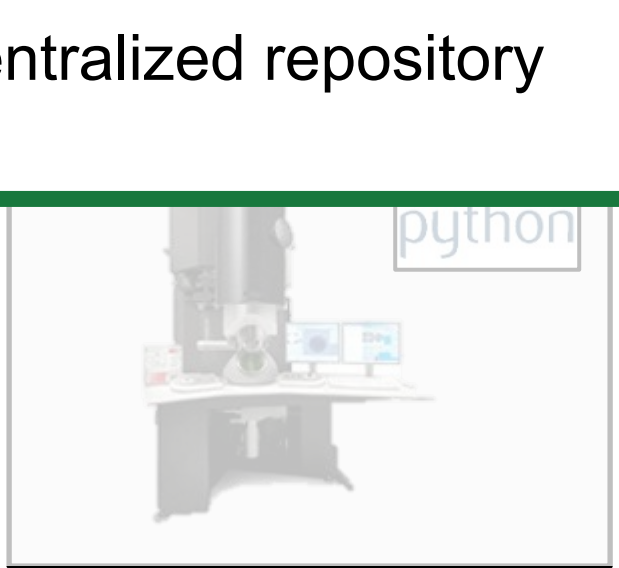
- FFT Filtering
- Classify Images …

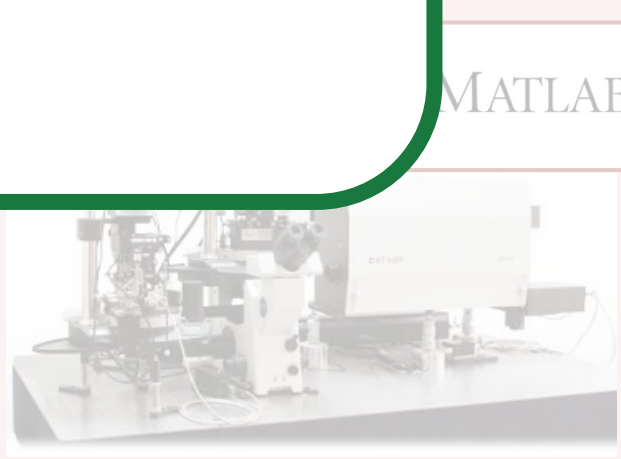
- Register Images
- Clustering

Cannot Share Code Efficiently



- HIGHLY instrument-specific code
- Different programming languages
- Often licensed / costly software like Matlab
- Most popular sharing method = email!
- No centralized repository

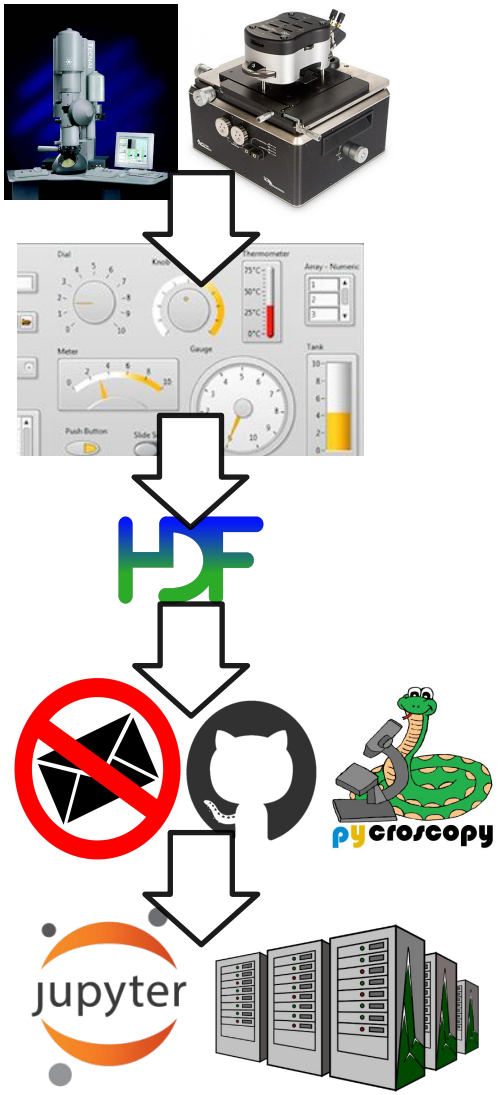




~~Problems~~ Opportunities in Imaging

1. Closed science
    a. No traceability for data analysis
    b. Results not (readily) reproducible
2. Multiple, incompatible, proprietary data formats
3. Disorganized and unorganized communities
4. No proper analysis software
5. Growing data volumes, variety, and dimensionality

The Solution



Instrument Tier

Automated, standardized, modularized data acquisition

Instrument-agnostic, self-describing, model in HPC-friendly file format

Centralized, open repository for data processing

Interactive visualization + analysis on supercomputers

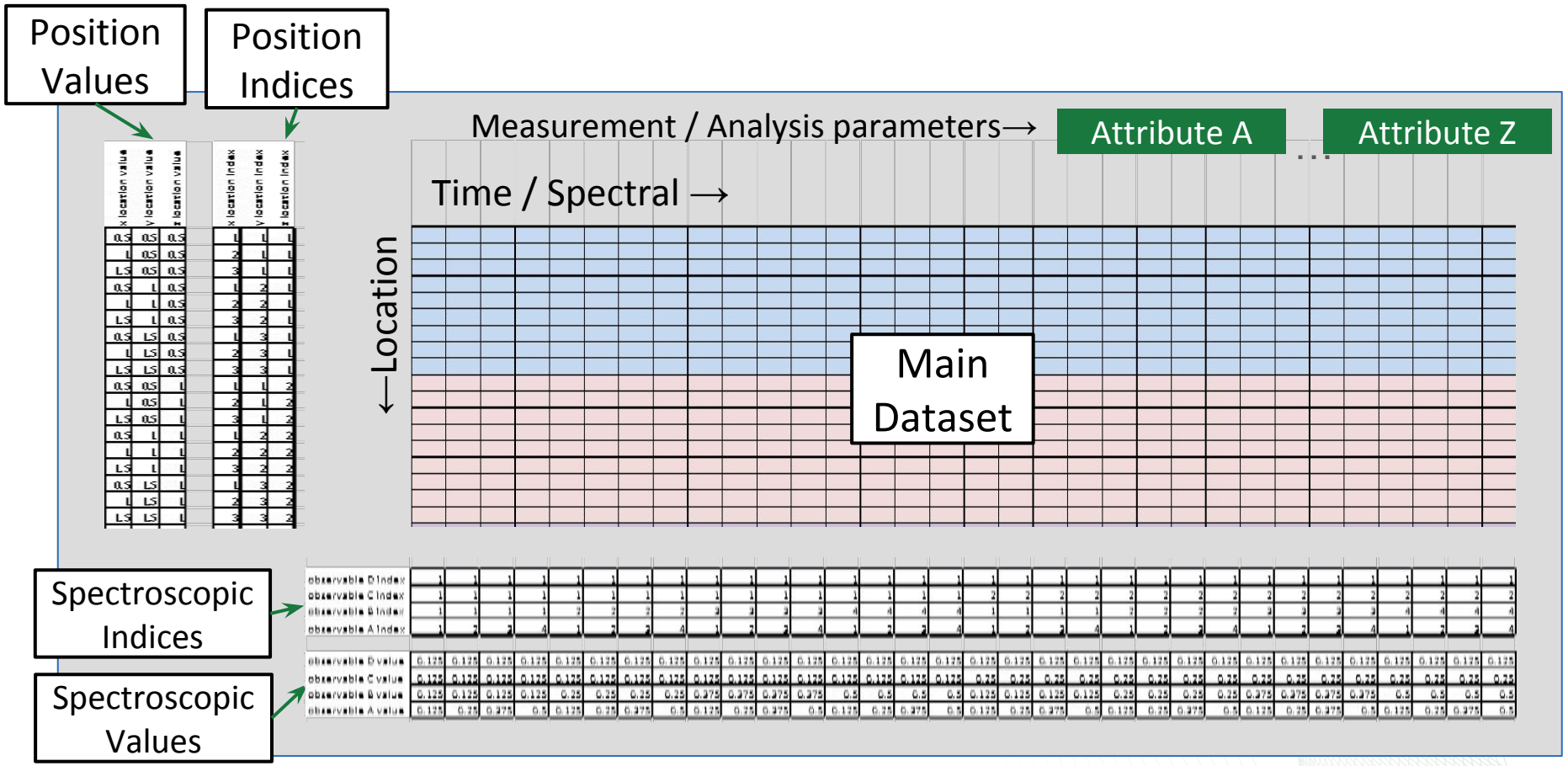
Suhas Somnath, somnaths@ornl.gov

Expectation of Data Model

- Accommodate data of different shapes, dimensionalities, precision and sizes.
- Accommodate data without N-dimensional form
  - Compressed sensing / sparse sampling
  - Not all combinations of spectroscopic variables
  - Incomplete experimental data

Universal Spectroscopic & Imaging Data (USID)

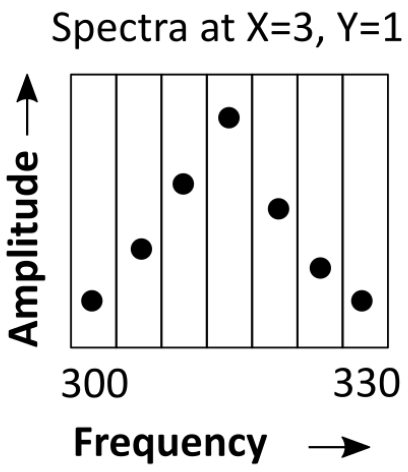
- Data stored as 2D matrix of (position x spectral values) regardless of dimensionality
- Ancillary datasets explain the data



Detailed information on pyUSID website

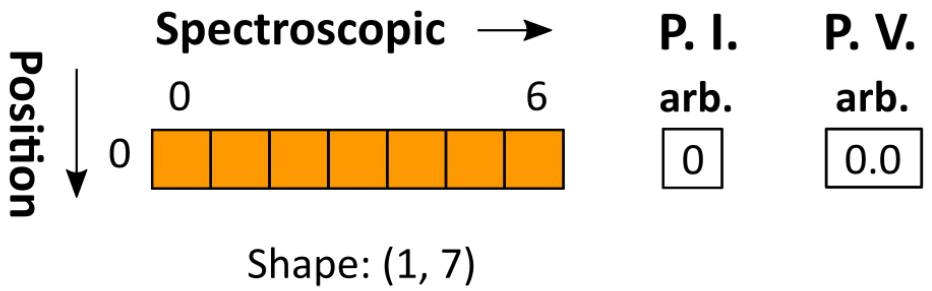
USID – 1D spectra

Original N-dimensional form



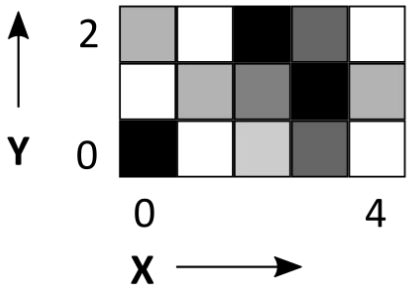
Shape: (7,)
Quantity: Amplitude
Units: V

USID 2-dimensional form



S. I.	Frequency	0	1	2	3	4	5	6
S. V.	Frequency	300	305	310	315	320	325	330

USID – 2D Image



Original N-D form

=

Spectroscopic →

Position ↓

P. I. X	P. I. Y
0	0
1	0
2	0
3	0
4	0
0	1
1	1
2	1
3	1
4	1
0	2
1	2
2	2
3	2
4	2

P. V. X	P. V. Y
-250	0
-125	0
0	0
125	0
250	0
-250	3.5
-125	3.5
0	3.5
125	3.5
250	3.5
-250	7
-125	7
0	7
125	7
250	7

Shape: (15, 1)

USID 2D form

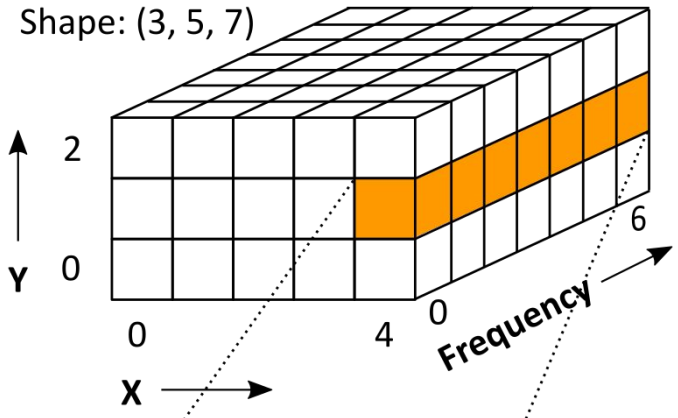
Quantity: Intensity
Units: arb. units

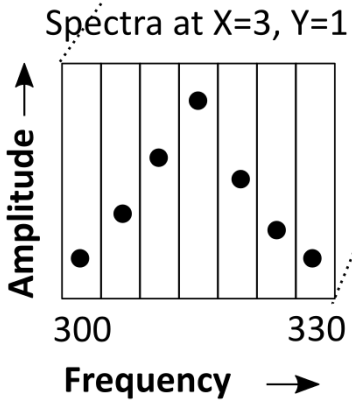
S. I. arb. 0

S. V. arb. 0.0

USID – Spectra on Grid (3D)

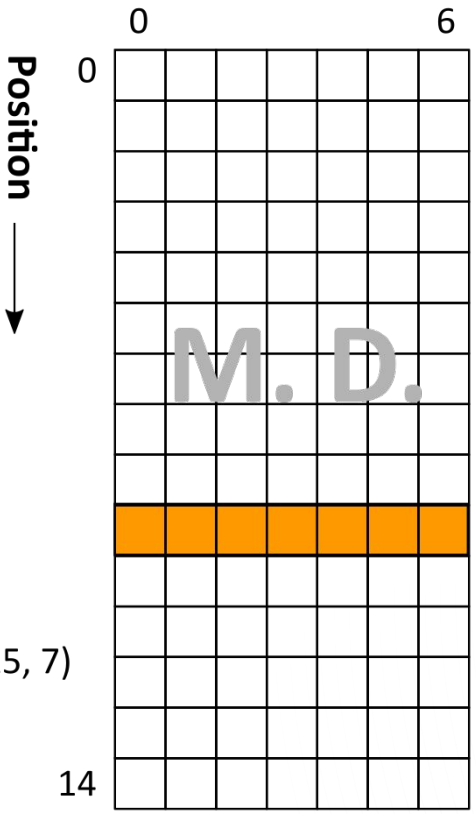
Original N-dimensional form





Quantity: Amplitude
Units: V

USID 2-dimensional Form



P. I.

X	Y
0	0
1	0
2	0
3	0
4	0
0	1
1	1
2	1
3	1
4	1
0	2
1	2
2	2
3	2
4	2

P. V.

X	Y
-250	0
-125	0
0	0
125	0
250	0
-250	3.5
-125	3.5
0	3.5
125	3.5
250	3.5
-250	7
-125	7
0	7
125	7
250	7

S. I. Frequency

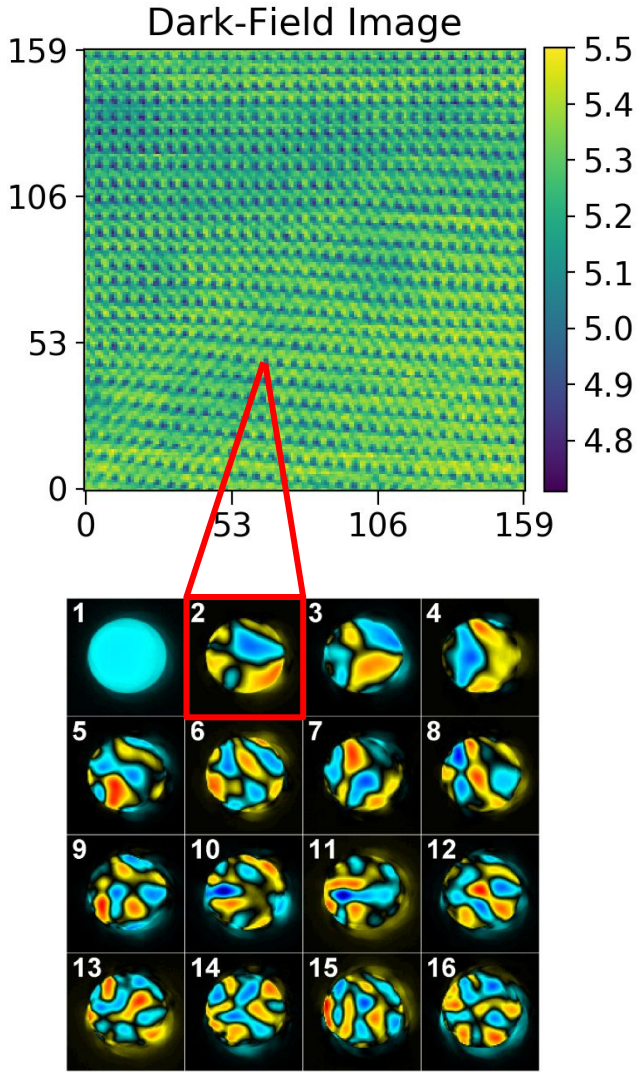
0	1	2	3	4	5	6

S. V. Frequency

300	305	310	315	320	325	330

Suhas Somnath, somnaths@ornl.gov

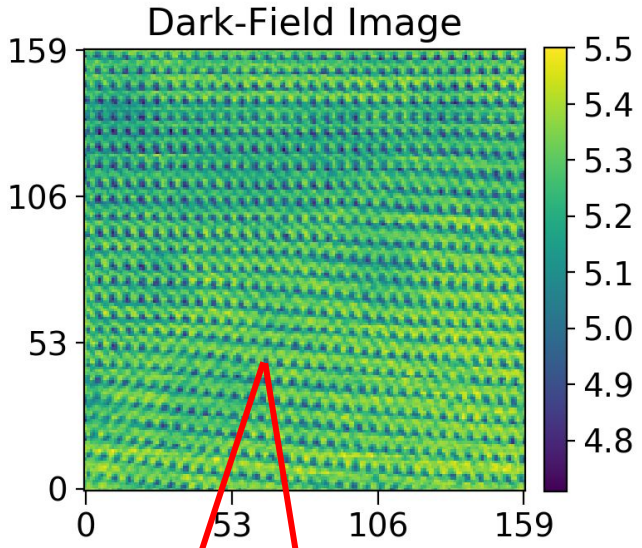
USID – Images on a grid (4D)

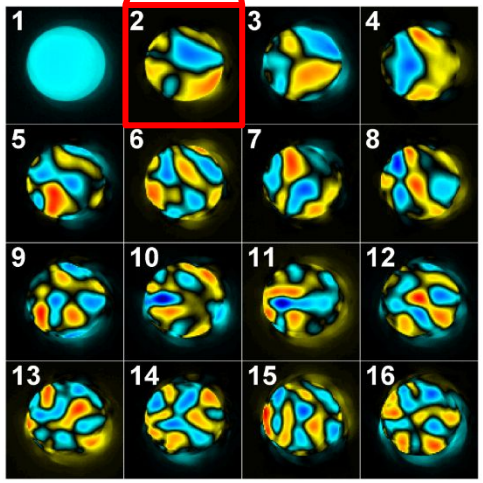


=

?

USID – Images on a grid (4D)





Flatten positions

Flatten images

Spectroscopic →

Position

0 6

0

14

M. D.

(5, 7)

P. I. X	P. I. Y
0	0
1	0
2	0
3	0
4	0
0	1
1	1
2	1
3	1
4	1
0	2
1	2
2	2
3	2
4	2

P. V. X	P. V. Y
-250	0
-125	0
0	0
125	0
250	0
-250	3.5
-125	3.5
0	3.5
125	3.5
250	3.5
-250	7
-125	7
0	7
125	7
250	7

S. V.

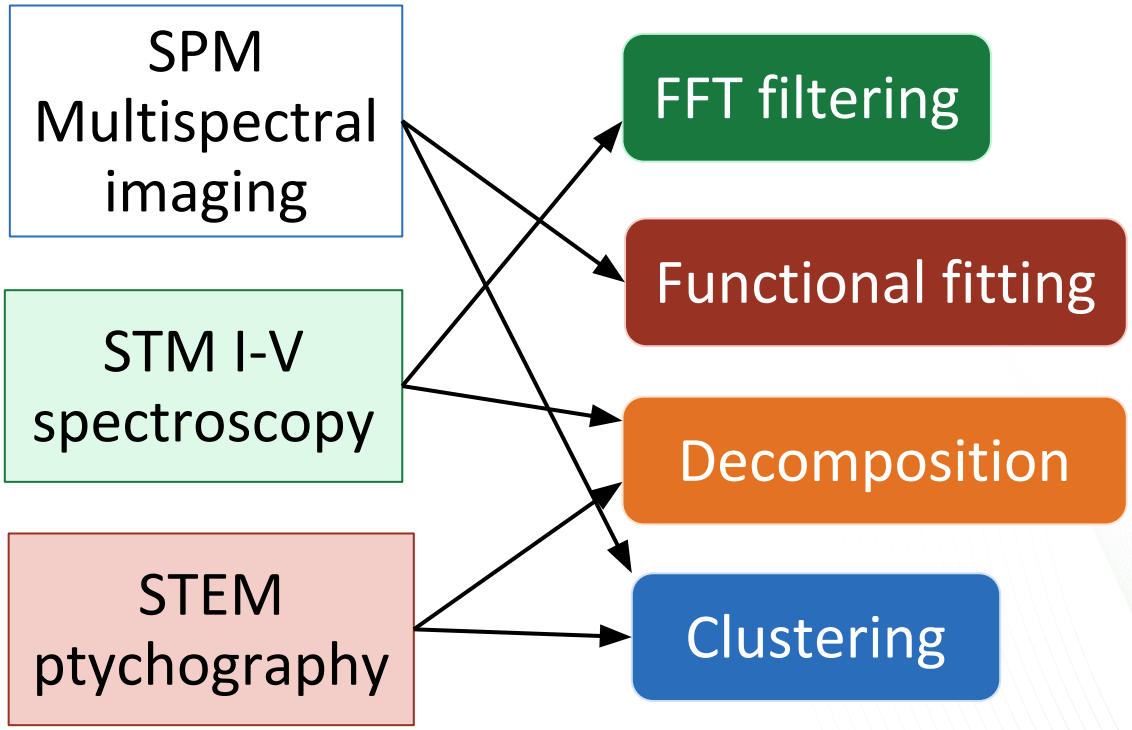
-250	-125	0	125	250	-250	-125	0	125	250	-250	-125	0	125	250
0	0	0	0	0	3.5	3.5	3.5	3.5	3.5	7	7	7	7	7

S. I.

0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
0	0	0	0	0	1	1	1	1	1	2	2	2	2	2

USID - Instrument Agnostic Code

- Instrument-agnostic data allows instrument-agnostic code
- Single version of analysis and processing routine
- Brings multiple scientific communities together

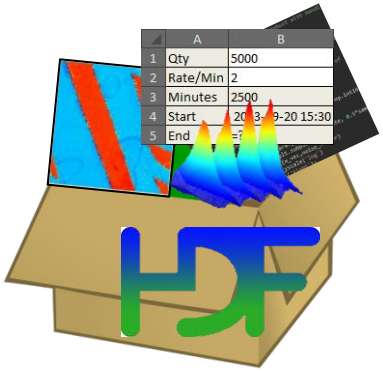


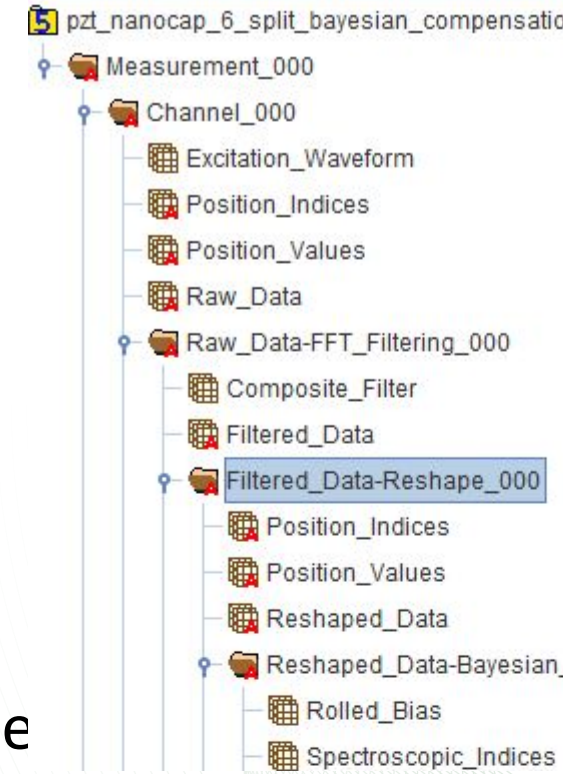
Expectation of File Format

- Established standard in scientific research
- Store multiple datasets of different shapes, dimensionalities, precision and sizes.
- Scale very efficiently from few kilobytes to several terabytes
- Able to read and write data using any programming language including Python, R, Matlab, C/C++, Java, Fortran, Igor Pro, etc.
  - (without requiring installation of modules that are hard to install)
- Store metadata - experimental or analysis parameters
- Highly flexible and poses minimal restrictions on how the data can and should be stored.
- Compatible with cloud and high-performance computing paradigms (support parallel read and write)

Hierarchical Data Format (HDF5)

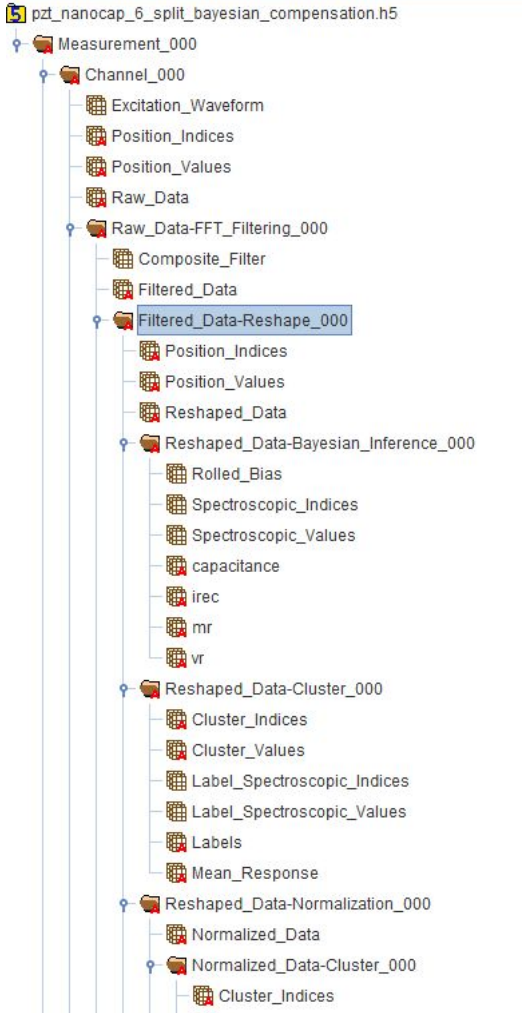
- A HDF5 file is a smart container
  - Capable of storing multidimensional datasets, Images, text, measurement parameters, etc.
  - Contents organized like traditional folders and files
    - **Groups** - Analogous to file folders
    - **Dataset** – 1 to N dimensional data
      - Integer, floating point, complex numbers etc
    - **Attributes** – {Key : value} pairs useful for describing data and experimental parameters, etc.
- Easily accessible – C, C++, python, Java….
- Tree structure + nomenclature +attributes are **records of workflow** applied to dataset
- Parallel read / write, HPC & cloud compatible





Traceability and Reproducibility

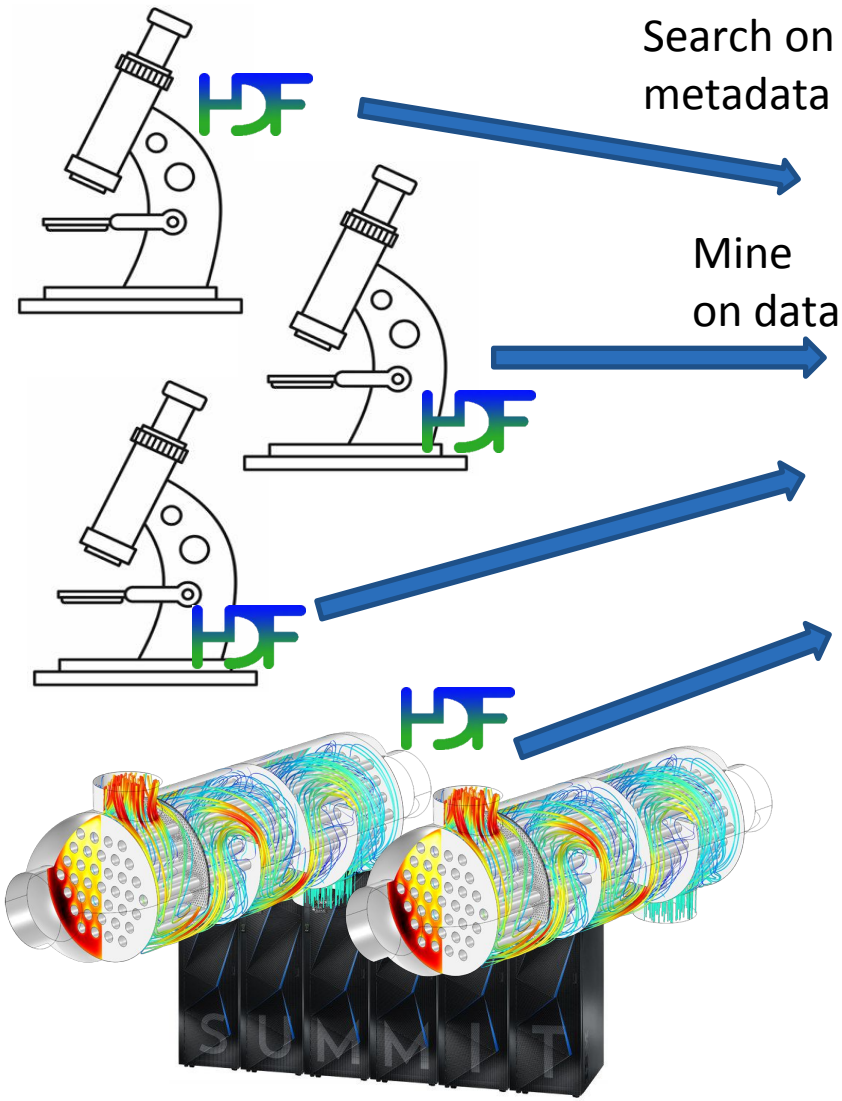
Raw, intermediate, final results stored in same file, leveraging hierarchy

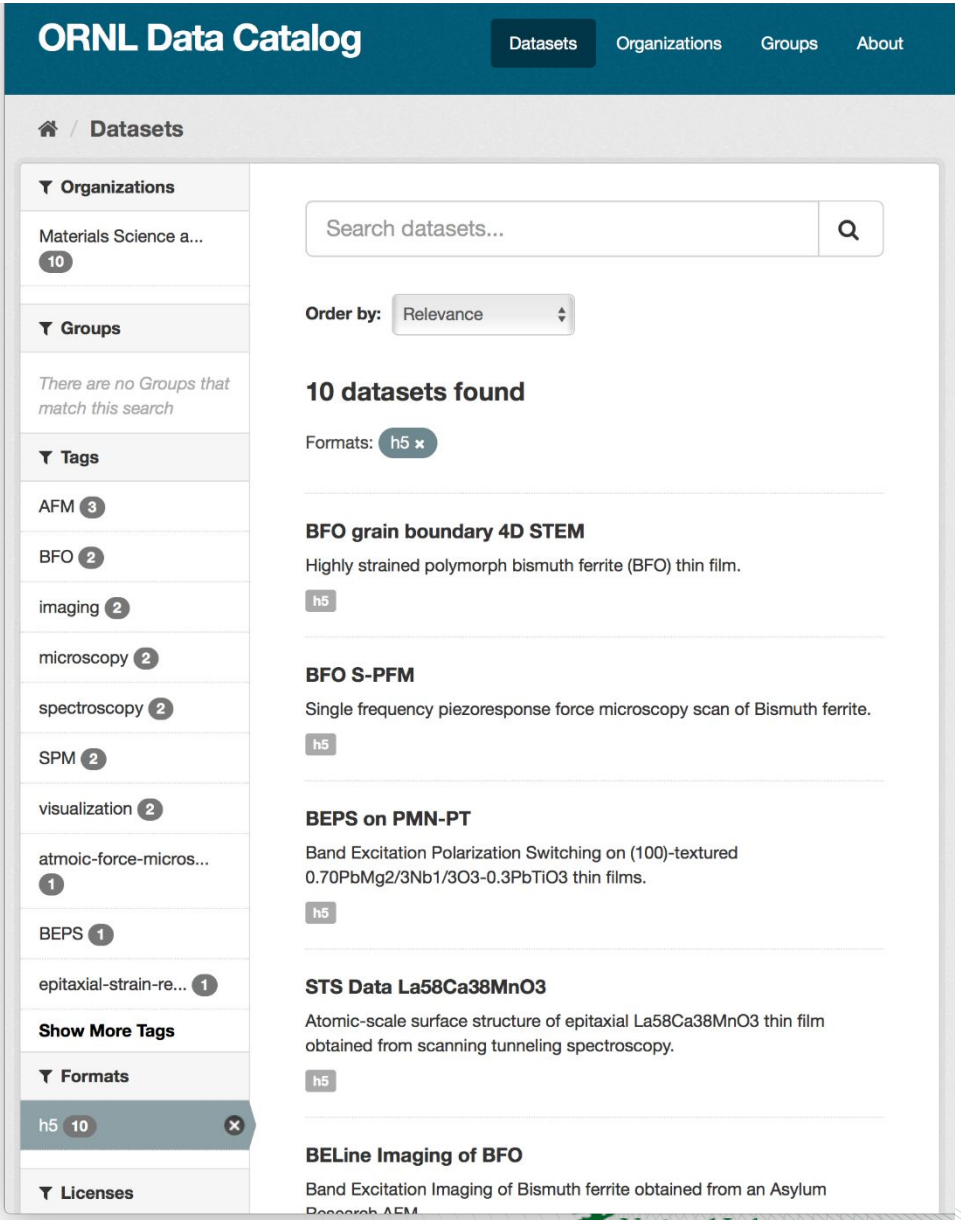


All measurement and analyses parameters stored for repeatability

```
Measurement_000 (2088, 4)
    Group size = 1
    Number of attributes = 43
        BE_actual_scan_time_[s] = 0.004
        BE_amplitude_[V] = 4
        BE_auto_smoothing = auto smoothing on
        BE_band_edge_trim = 0.14614
        BE_band_smoothing_[Hz] = 2560.5
        BE_band_width_[Hz] = 60000
        BE_bins_per_band = 0
        BE_center_frequency_[Hz] = 370000
        BE_phase_variation = 1
        BE_points_per_BE_wave = 0
        BE_repeats = 8
        BE_signal_type = chirp-sinc hybrid
        BE_time/pixel_[s] = 0.004
        File_date_and_time = 06-Feb-2015 11:46:13
        File_file_name = BELine
        File_file_path = C:\Users\Administrator\Documents\Asylum Research Data\150205\
        File_file_suffix = 9
        IO_AO_amplifier = 1
        IO_AO_range_[V] = +/- 10
        IO_Analog_Input_1 = +/- 1V, FFT
        IO_Analog_Input_2 = +/- 10V, mean
        IO_Analog_Input_3 = off
        IO_Analog_Input_4 = off
        IO_DAQ_platform = NI 5412/5122
        IO_rate_[Hz] = 4000000
        data_type = BELineData
        grid_contact_set_point_[V] = 1
        grid_current_row = 1
        grid_cycle_time_[s] = 0.05
        grid_lift_height_[m] = 5.0E-8
        grid_nap_mode = nap mode off
        grid_num_cols = 128
        grid_num_rows = 128
        grid_scan_time_/_line_[s] = 1
        grid_time_remaining_[h;m;s] = 10
        grid_total_time_[h;m;s] = 10
        machine_id = mac109728.ornl.gov
        num_bins = 44
        num_pix = 16384
        num_udvs_steps = 1
        platform = Darwin-17.4.0-x86_64-i386-64bit
        pycroscopy_version = 0.60.0rc1
        timestamp = 2018_05_03-15_23_37
```

Facilitating Data Mining

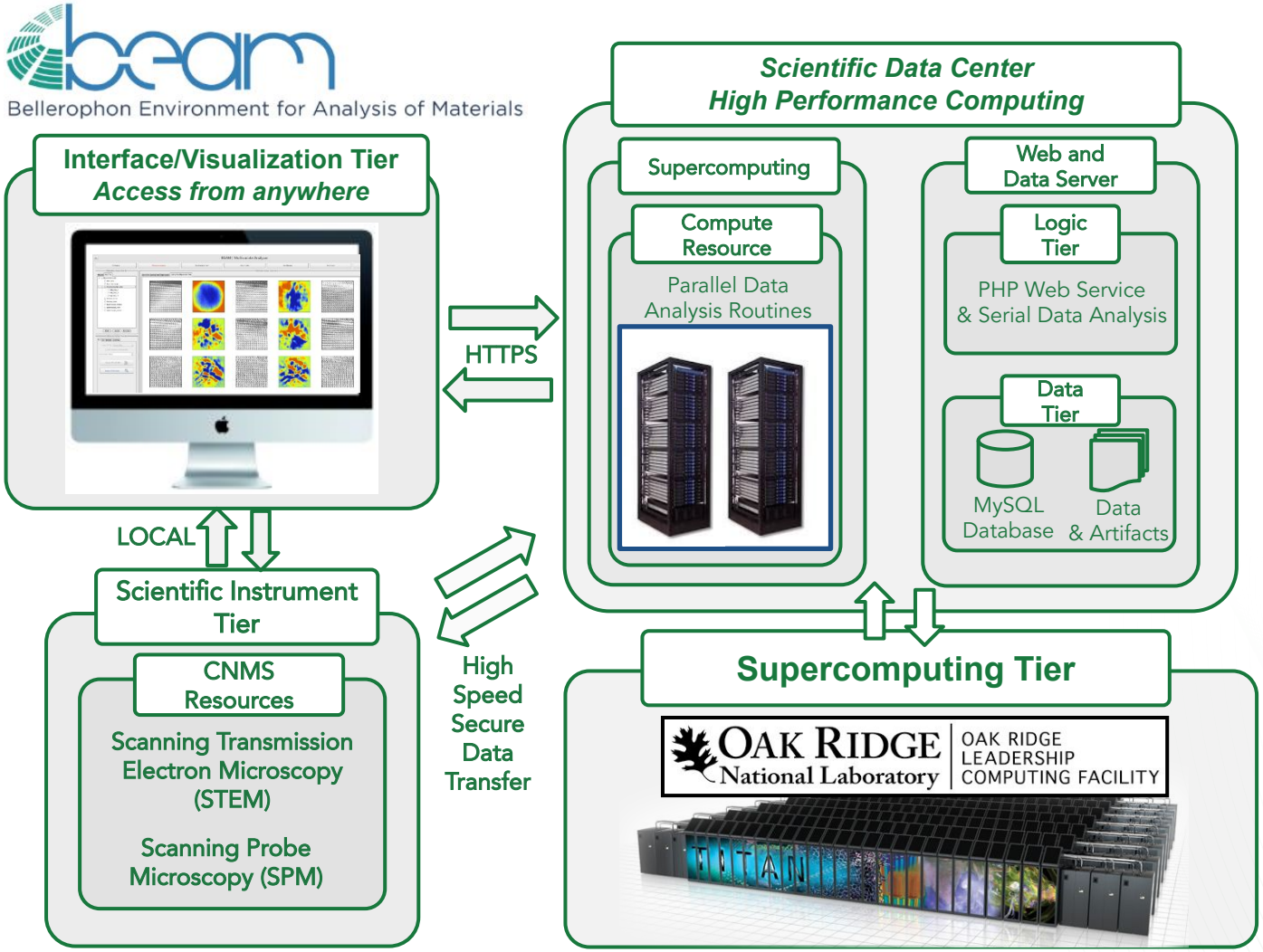




Suhas Somnath, somnaths@ornl.gov

(Purely) Programmer-Driven Solution

Software connecting scientific instruments to supercomputers



- **Successes**:
  - Easy to use – Point-click
  - Fast – on super-computers
- **Shortcomings**:
  - Very long development cycle
  - Very expensive
  - Brittle (points of failure)
  - Scientists had no control!!

Lingerfelt et al., *Procedia Computer Science* **80**, 2276-2280 (2016).

Expectation from Software

- Easy to learn and understand
- Strong support-base
- Established community standard
- Straightforward to implement and maintain
- Optimized libraries for scientific and numeric algorithms
- Access to existing imaging related packages
- Affordable
- Scalable to multiple CPU cores + distributed computing

Two-prong (inverse) Software Strategy

Mode	Exploratory / Development	Production
Compute Resource	Instrument-loca l Workstation  Cloud compute 	High-performance computing cluster 
Who?	Domain scientists	Collaboration with computational facility
Algorithms	Simpler but approximate	Complex but accurate

Python for Scientific Research

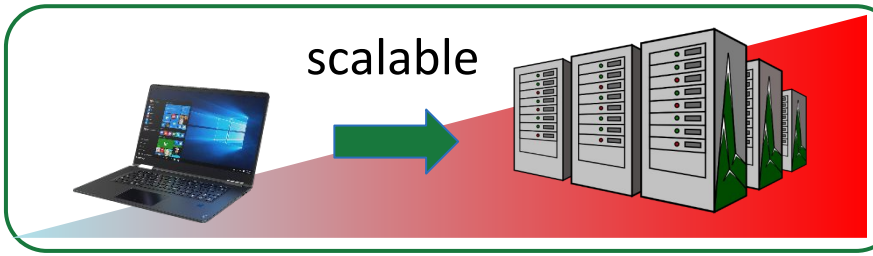
Very easy to learn + code

Numerous, **powerful** libraries for science

scikit learn

NumPy

SciPy



Strong user community

stackoverflow

open source

- Facilitates innovation
- More robust code
- Improved adoption of new methods / standards
- Accelerates scientific progress

Cross-platform

MacOS

Established standard for:

- Microscopy
- Microbiology
- Deep learning
- Data science
- Neutron science
- More…!

All for a princely sum of …. **$0!**

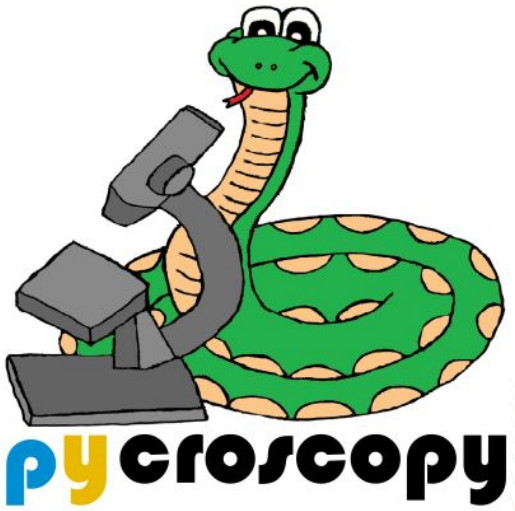
See Jake Vanderplas' Pycon 2017 Keynote talk:
http://www.youtube.com/watch?v=ZyjCqQEUa8o&t=19m20s

Two Software Packages

- Written in Python
- Open source & free
- Written by scientists
- Data centric
- Instrument-independent data model in HDF5
- Instrument-independent analysis algorithms
  - Reusable across scientific domains

pyUSID

2,800+ downloads



121,000+ downloads

pyUSID – Software Organization

pyUSID

io

- Tools to read and write to USID HDF5 files
- *USIDataset* – slicing, visualization, reduction, reshaping of 2D USIDsets
- Data Translators
  - Images & numpy arrays to HDF5

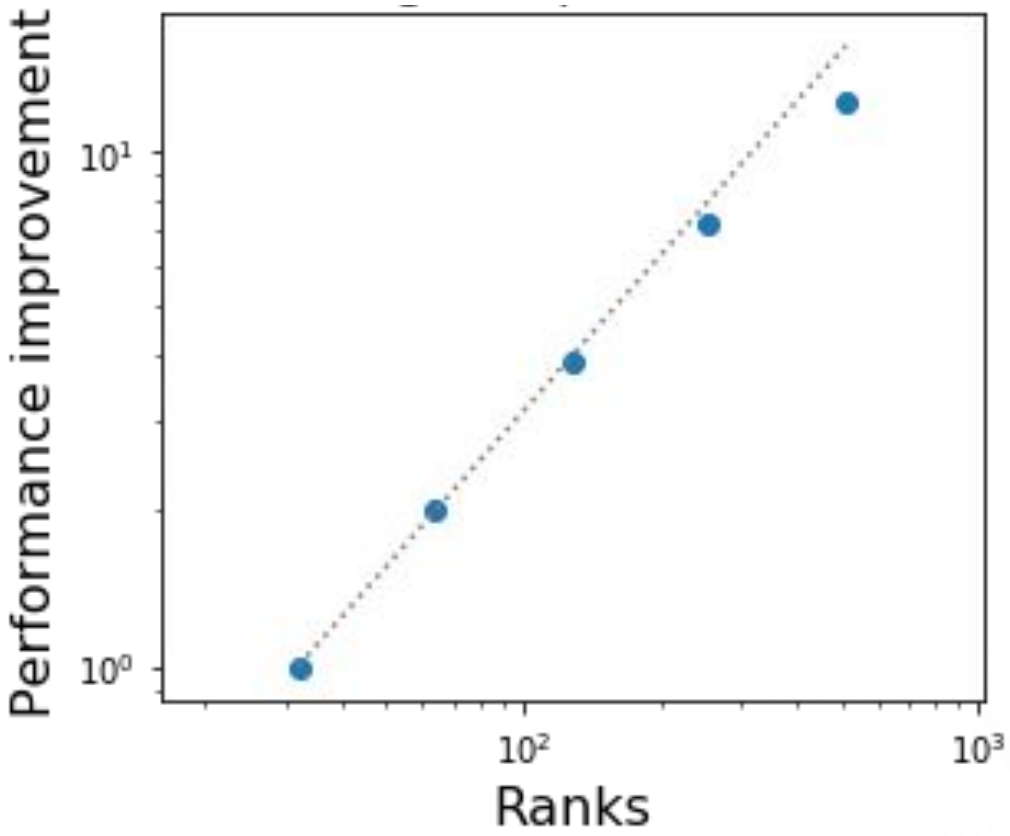
Processing

- *Process* –
  - Frame science as compute problem
  - Parallel & scalable computation
    - Multi-core (joblib)
    - Multi-node (MPI)
    - Piecewise, resume
    - Manages memory

Visualization

- Plotting utilities
- Interactive jupyter widgets for N-dim datasets

Scaling on HPC



* Preliminary results on ORNL's CADES SHPC Condo

** Lower than ideal trend because compute job was too small to use many nodes

Pycroscopy - Software Organization

pycroscopy

io
- Data translators (proprietary formats to HDF5)

Visualization
- Science-specific plotting & interactive jupyter widgets

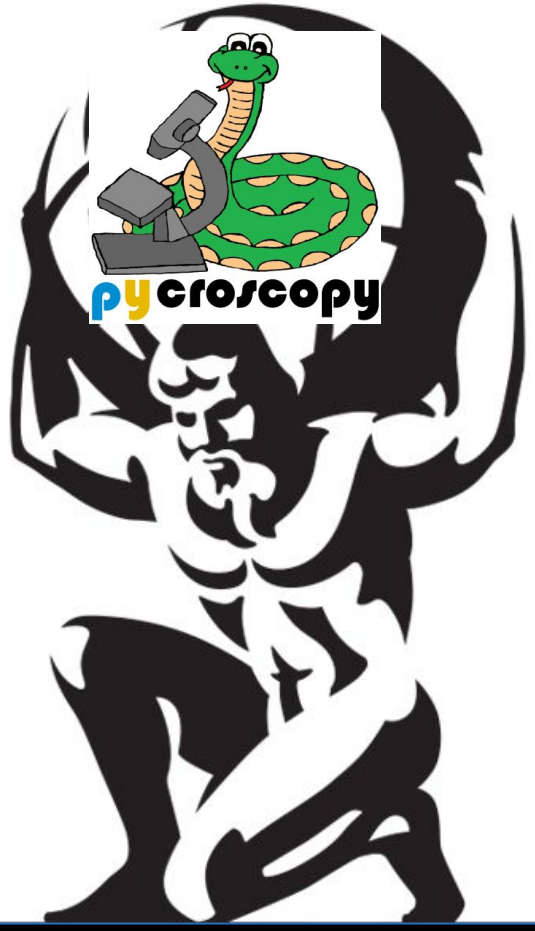
Simulation
- AFM Force-distance …

Analysis
- Physical model specific
- Fitting to model, etc.

Processing
- Physical model agnostic
- Image filtering, registration,
- Multivariate analysis

Software Organization

Science / data analytics applications



File & computing tools

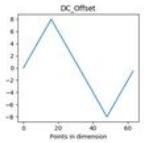
pyUSID

Well documented

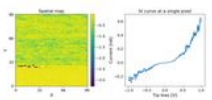
Beginner topics

To learn **how to use pyUSID**, Please go through the following documents in the recommended order:

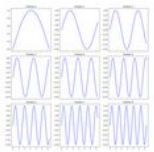
01. Primer to HDF5 and h5py



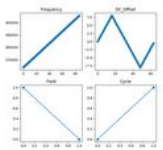
02. The USIDataset



03. Translation and the NumpyTranslator



04. Plotting utilities

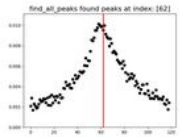


05. Utilities for reading h5USID files

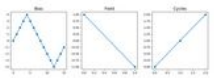
Intermediate topics

To learn **how to write to h5USID files, write data processing classes, or adding functionality to pyUSID**, go through these additional documents in the recommended order: Those interested in contributing to pyUSID are encouraged to read our guidelines for contributing code

06. Utilities for handling



07. Speed up



08. Utilities that assist

reshape_to_Ndims(*h5_main, h5_pos=None, h5_spec=None, get_labels=False, verbose=False, sort_dims=False*) [source]

Reshape the input 2D matrix to be N-dimensions based on the position and spectroscopic datasets.

Parameters:

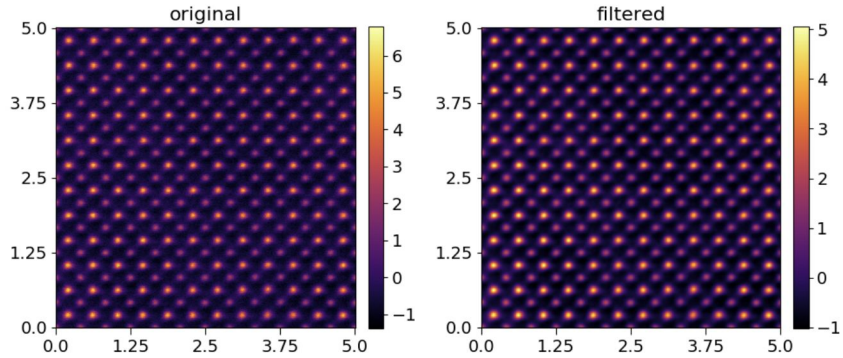
- **h5_main** (*HDF5 Dataset*) – 2D data to be reshaped
- **h5_pos** (*HDF5 Dataset, optional*) – Position indices corresponding to rows in *h5_main*
- **h5_spec** (*HDF5 Dataset, optional*) – Spectroscopic indices corresponding to columns in *h5_main*
- **get_labels** (*bool, optional*) – Whether or not to return the dimension labels. Default False
- **verbose** (*bool, optional*) – Whether or not to print debugging statements
- **sort_dims** (*bool*) – If True, the data is sorted so that the dimensions are in order from fastest to slowest If False, the data is kept in the original order If *get_labels* is also True, the labels are sorted as well.

Returns:

- **ds_Nd** (*N-D numpy array*) – N dimensional numpy array arranged as [positions slowest to fastest, spectroscopic slowest to fastest]

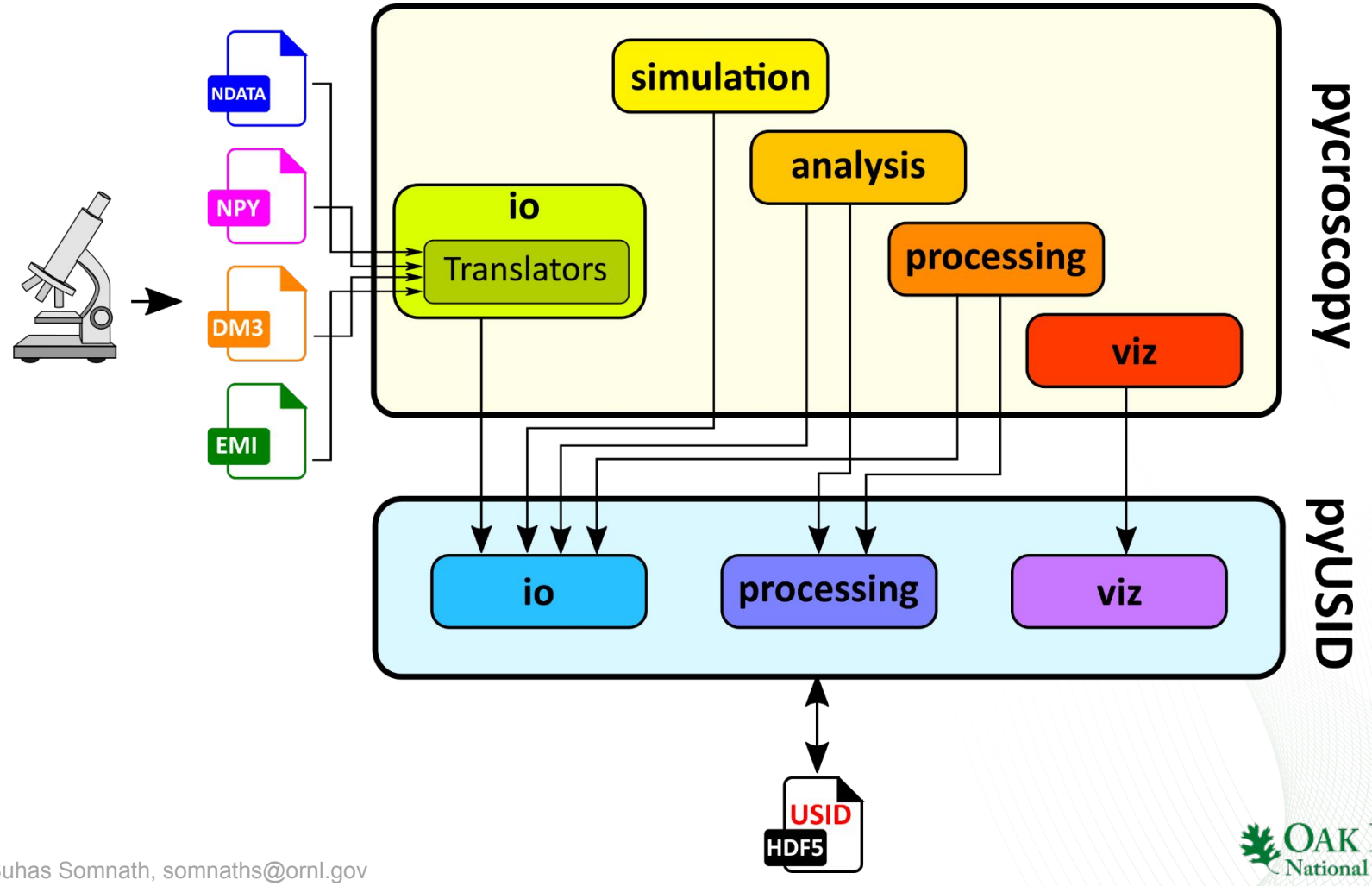
To view the fil
('ifft2'). Remem
necessary to u
transform. Als
the inverse tra
symmetric abo
result in the in
times smaller
kept.

```
image_filter
image_filter

fig, axes =
for axis, img, title in zip(axes, [image_raw, image_filtered], ['original', 'filtered']):
    _ = px.plot_utils.plot_map(axis, img, cmap=plt.cm.inferno,
                               x_size=x_edge_length, y_size=y_edge_length, num_ticks=5)

    axis.set_title(title)
fig.tight_layout()
```



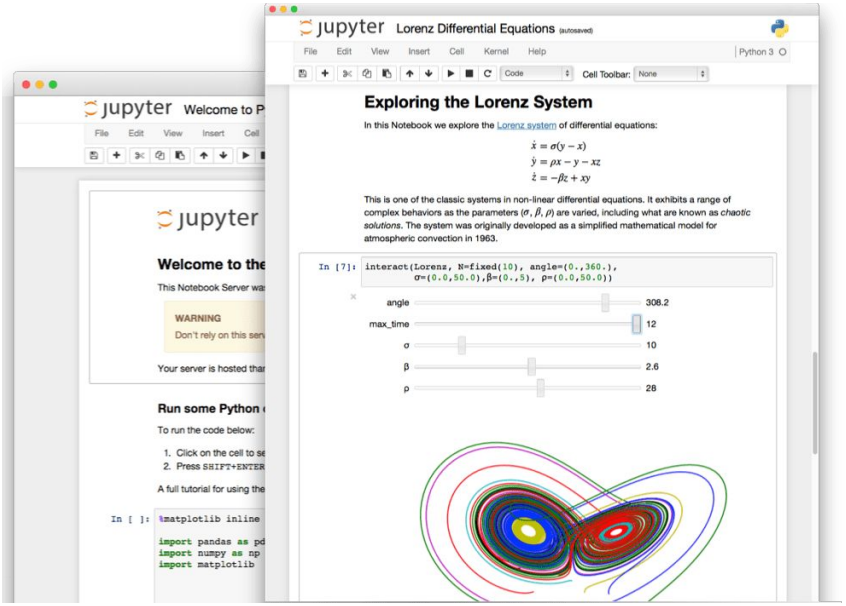
Entering the USID Ecosystem

- hdf5 file is the hub for all operations
- Analysis, processing, visualization available after translation to .hdf5



Jupyter Notebooks

Jupyter Notebook



- Interactive documents
- Exploratory programming
- Code
- Text
- Images
- Interactive – slice through data, pan, move, rotate …

Towards Open & Reproducible Science

Aim – ALL scientific journal papers accompanied with:

- Jupyter notebook that shows all analysis (raw data → figures).
- Data with DOI number
- Containers – software environment

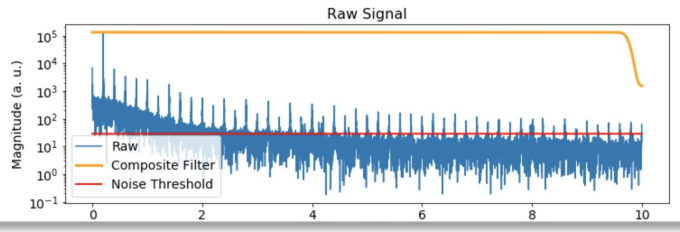
Jupyter notebook associated with paper

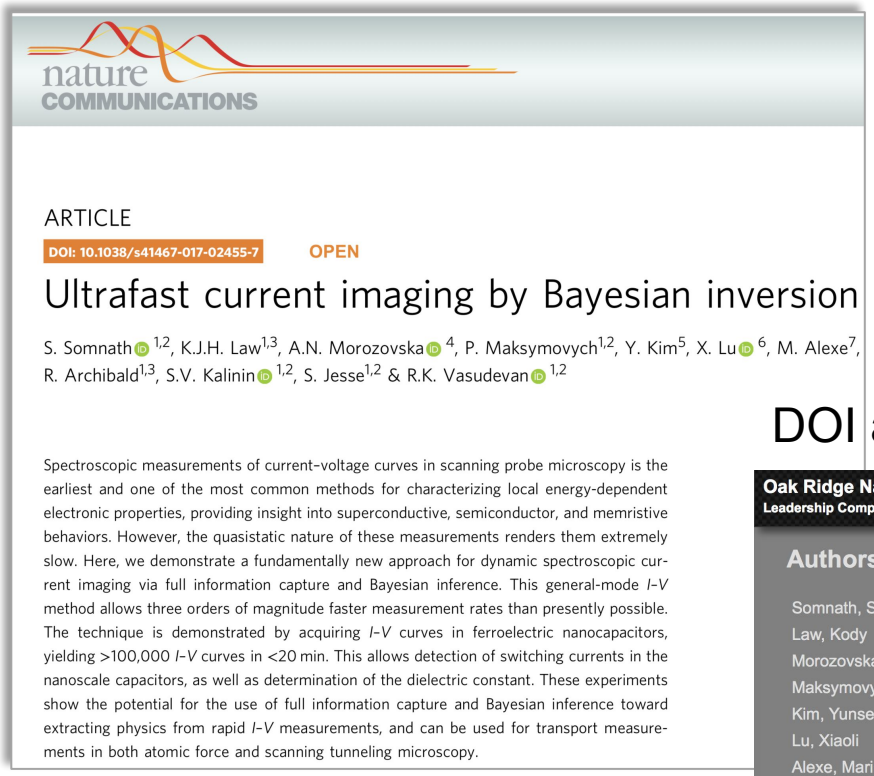
Visualizing Filtering Results

Run the next cell to see what the IV curves look like for the chosen filter parameters. If the current set of filter parameters do not work as well, change the parameters in the previous cell and try again.

Once the results look good, proceed to filter the entire dataset

```
In [7]: # Test filter on a single line:
row_ind = 50
filt_line, fig_filt, axes_filt = px.processing.gmode_utils.test_filter(h5_main[row_ind], filte
r_parms, samp_rate,
                                                                       show_plots=True, use_rai
nbow_plots=False)
fig_filt.savefig(os.path.join(other_figures_folder, 'FFT_filter_on_line_{}.png'.format(row_ind
)), format='png', dpi=300)

raw_row = np.reshape(h5_main[row_ind], (-1, pts_per_cycle))
filt_row = filt_line.reshape(-1, pts_per_cycle)

fig, axes = px.plot_utils.plot_loops(single_AO, [raw_row, filt_row], dataset_names=['Raw', 'Fi
ltered'],
                                     line_colors=['r', 'b'], x_label='Bias (V)', title='FFT Fi
ltering',
                                     plots_on_side=3, y_label='Current (nA)',
                                     subtitles='Row: ' + str(row_ind) + ' Col:')
fig.savefig(os.path.join(other_figures_folder, 'Example_filtered_loops_from_line_{}.png'.forma
t(row_ind)), format='png', dpi=300)
```





nature COMMUNICATIONS

ARTICLE

DOI: 10.1038/s41467-017-02455-7 OPEN

Ultrafast current imaging by Bayesian inversion

S. Somnath[1,2], K.J.H. Law[1,3], A.N. Morozovska[4], P. Maksymovych[1,2], Y. Kim[5], X. Lu[6], M. Alexe[7], R. Archibald[1,3], S.V. Kalinin[1,2], S. Jesse[1,2] & R.K. Vasudevan[1,2]

Spectroscopic measurements of current–voltage curves in scanning probe microscopy is the earliest and one of the most common methods for characterizing local energy-dependent electronic properties, providing insight into superconductive, semiconductor, and memristive behaviors. However, the quasistatic nature of these measurements renders them extremely slow. Here, we demonstrate a fundamentally new approach for dynamic spectroscopic current imaging via full information capture and Bayesian inference. This general-mode *I*–*V* method allows three orders of magnitude faster measurement rates than presently possible. The technique is demonstrated by acquiring *I*–*V* curves in ferroelectric nanocapacitors, yielding >100,000 *I*–*V* curves in <20 min. This allows detection of switching currents in the nanoscale capacitors, as well as determination of the dielectric constant. These experiments show the potential for the use of full information capture and Bayesian inference toward extracting physics from rapid *I*–*V* measurements, and can be used for transport measurements in both atomic force and scanning tunneling microscopy.

DOI associated with data (raw → paper figures)

Oak Ridge National Laboratory
Leadership Computing Facility

10.13139/OLCF/1410993

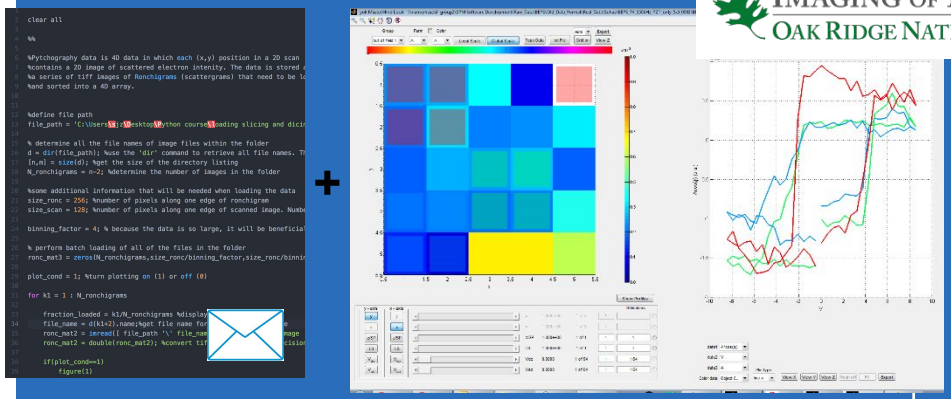
Download

Authors

Somnath, Suhas	somnaths@ornl.gov
Law, Kody	lawkj@ornl.gov
Morozovska, Anna	anna.n.morozovska@gmail.com
Maksymovych, Petro	maksymovychp@ornl.gov
Kim, Yunseok	ykim943@gmail.com
Lu, Xiaoli	xllu@xidian.edu.cn
Alexe, Marin	M.Alexe@warwick.ac.uk

Pycroscopy - Supporting User Research

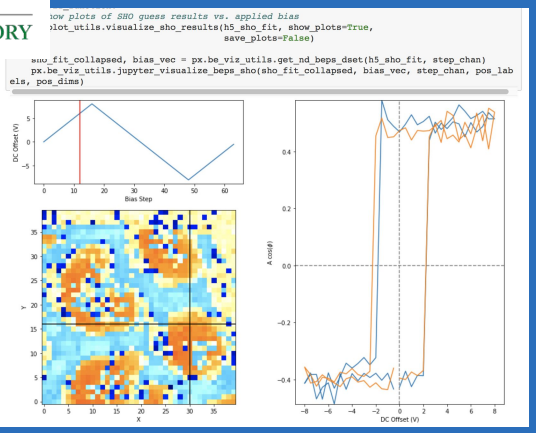
Before 2016



+

Since 2016

INSTITUTE FOR FUNCTIONAL IMAGING OF MATERIALS
OAK RIDGE NATIONAL LABORATORY



Scripts + complicated, Matlab GUI	Set of simple Jupyter notebooks
Witten by dedicated software engineer	Written by material scientists
Not customizable	Completely customizable.
2-3 hours of training before use	Notebooks include instructions. NO training required!
Deployed only on two offline workstations due to licensing restrictions = queue	Each user gets VMs with jupyter notebook server
Will remain on off-line desktops	In the process of switching to computations on clusters

Outreach - workshops

MM_2018_Workshop

Tutorial materials for the Microscopy and Microanalysis Workshop

Prerequisites:

Please follow instructions here to install Anaconda and pycroscopy

More information on the contents:

Please see our website. You should be able to find all information ranging tuto
enabled by pycroscopy, reading different microscopy file formats, tutorials on

Getting in touch

Time	Details
8:00-9:00 AM	Setup and installation
9:00-9:30 AM	Introduction to machine learning in python
9:30-10:15 AM	Linear unmixing methods for spectral analysis
10:15-10:30AM	Coffee Break
10:30-12:00 PM	Nonlinear unmixing: overview and applications
12:00-1:00 PM	Lunch on your own
1:00-2:00 PM	Traditional classification methods
2:00-2:45 PM	Clustering methods applied to 4D STEM data
2:45-3:00 PM	Break
3:00-4:30 PM	Neural networks and deep learning for microscopy images
4:30-5:00 PM	Wrap up/Discussion/Summary

Imaging and Spectral Data Analysis in Python

We present a set of jupyter notebooks and presentations that go over the basics of:

- Image and spectral data analysis in Python
- USID, pyUSID, and pycroscopy as tools for analyzing large and multidimensional imaging / spectroscopy datasets

This material was developed as part of the 2018 CNMS User Meeting

See our list of tutorials for more information

Getting started

(Recomn

1. (one ti
   i. Fo
   ii. Al
2. (one ti
   i. cli
   ii. Cl
   iii. Do
   iv. Ur

CNMS_ML_in_MS_Workshop_2018

CNMS User Meeting Workshop 2018: Machine learning in Materials Science with Python

Welcome! This workshop will go over machine learning methods for materials science with python. The full agenda is published below. For remote participants, please click https://bluejeans.com/1766282540

The workshop will be conducted on August 15, 2018 at ORNL. The livestream begins at 9AM EDT.

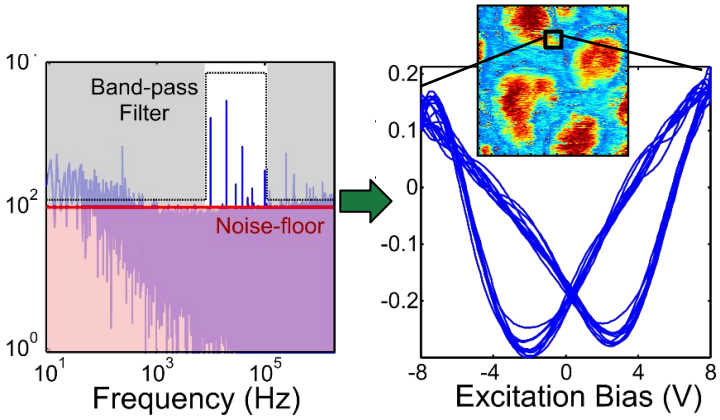
To get the most out of this workshop, you may want to work along with us on your own laptop. If you do, we highly recommend that you:

1. Download and install the following:
- Anaconda5.2 for python 3.6 - https://www.anaconda.com/download/
- Pycroscopy- https://pycroscopy.github.io/pycroscopy/install.html
- HDFview- https://support.hdfgroup.org/products/java/release/download.html

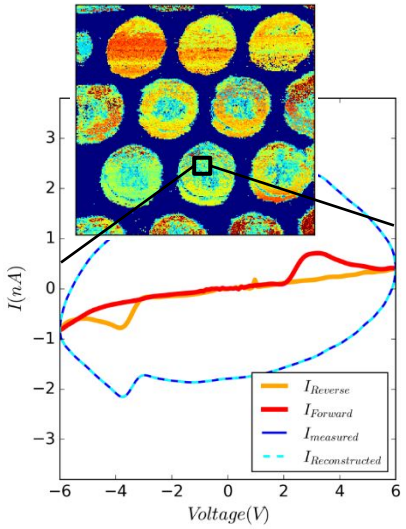
2. Familiarize yourself with basics concepts.
Links to quick and helpful tutorials are available here - https://pycroscopy.github.io/pyUSID/external_guides.html.
Topics include:
- Basic python usage
- Jupyter notebooks for running code
- Numpy for numeric operations

Pycroscopy - Scientific Advancements

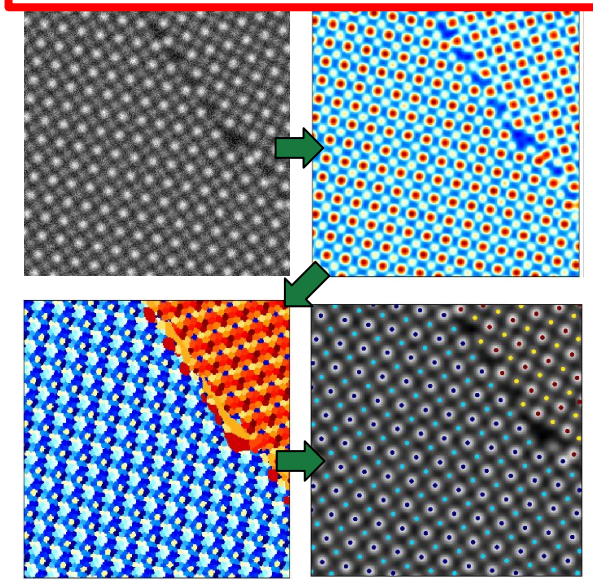
3,500x faster imaging via adaptive signal filtering, linear unmixing of signals



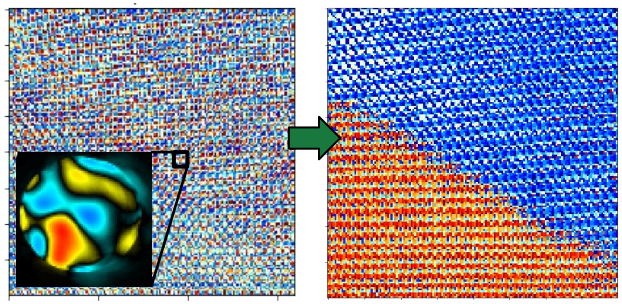
200x faster spectroscopy via Bayesian inference



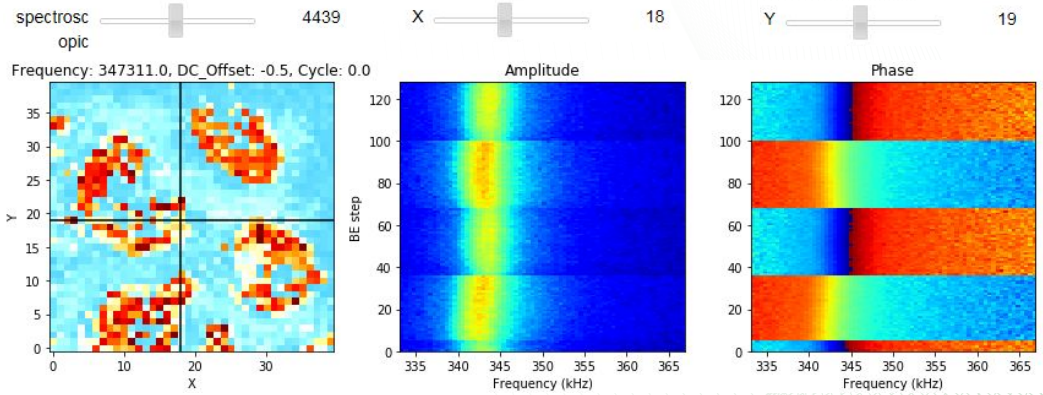
Separating uncorrelated data from correlated data to clean images



Identifying invisible patterns using multivariate analysis



Simplified navigation multidimensional data - users



Suhas Somnath, somnaths@ornl.gov

Software Progression

Scaling up Computing:

2016	2017	2018	2019
Single core	Multi-core Single CPU	Multi-core Multi CPU	Web-service via JupyterHub On HPC

Emphasis always on ease-of-development instead of raw performance

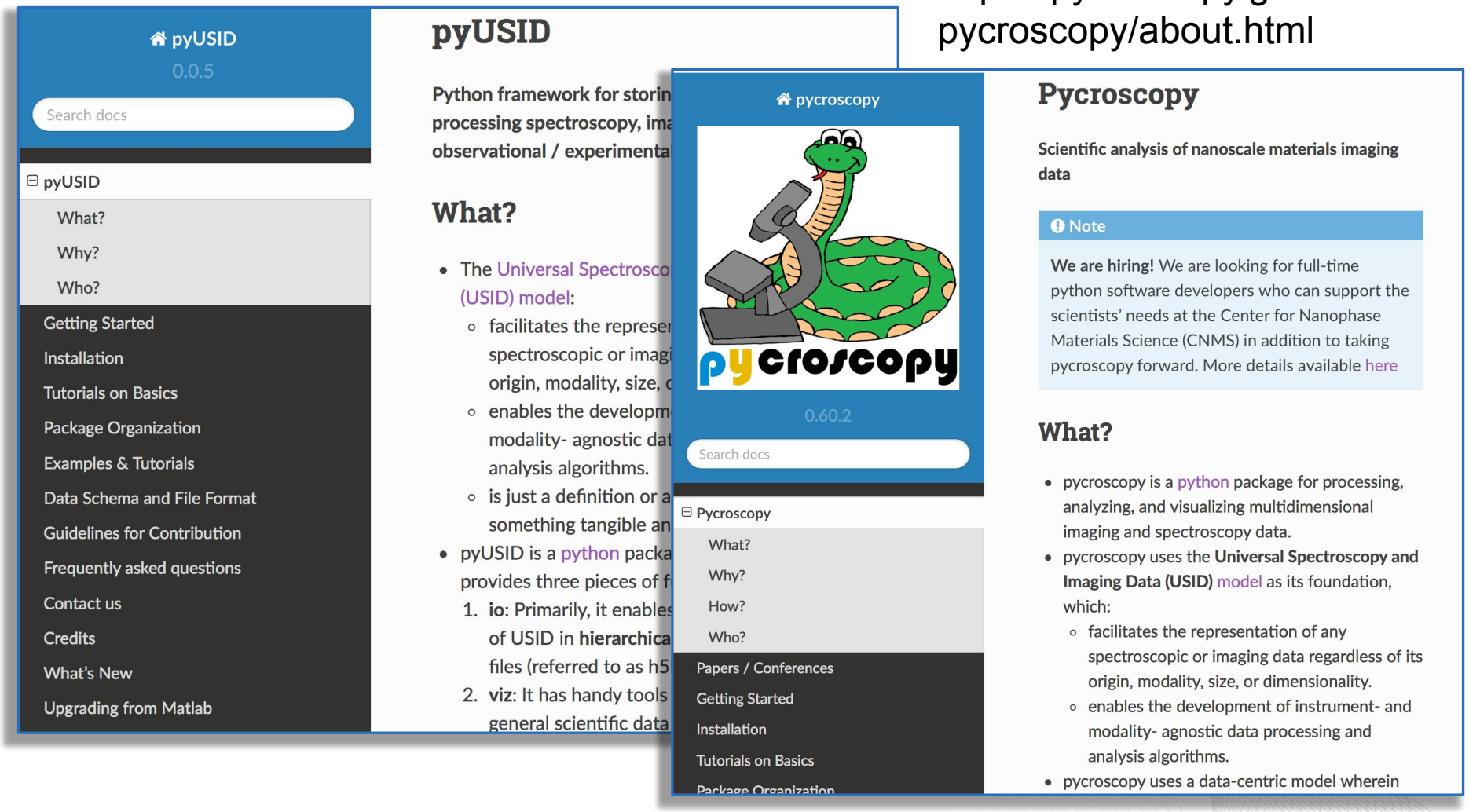
New Scientific Domains:

2016	2017	2018	2019
Atomic Force Microscopy	+ Scanning Transmission Electron Microscopy	+	Mass Spectrometry + Synchrotron research

Lots more information available online

https://pycroscopy.github.io/pyUSID/about.html

https://pycroscopy.github.io/pycroscopy/about.html



Thank you

Questions?

Science, data analysis, reading proprietary instrument data …. - https://groups.google.com/forum/#!forum/pycroscopy

File / data tools related- https://groups.google.com/forum/#!forum/pyusid

This presentation - https://docs.google.com/presentation/d/1K9SCtQU6KSSAewlkiwWz77FDfKSzZJqZPnTy2z9OujU/edit?usp=sharing