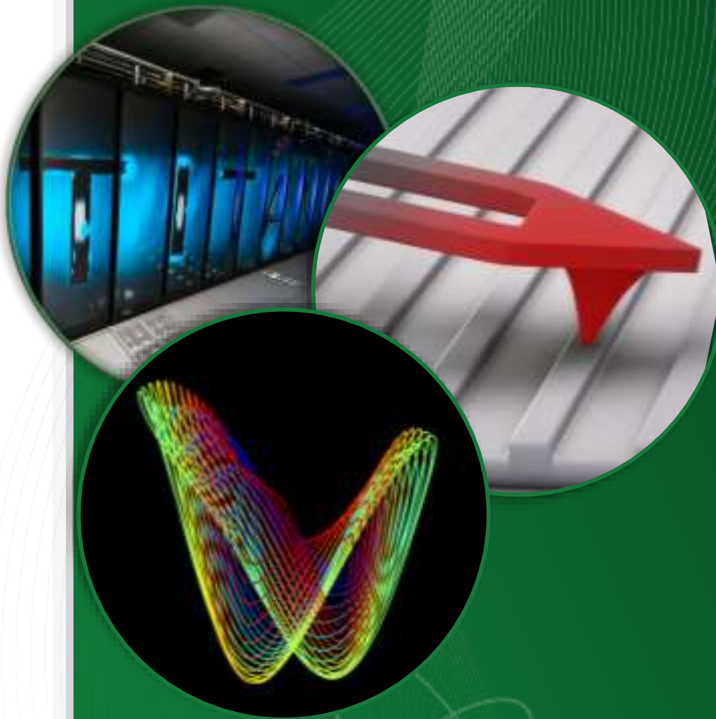
Pycroscopy –
A python package for storing and analyzing imaging and spectroscopy data



- <u>Suhas Somnath</u>
- Chris R. Smith
- 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

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

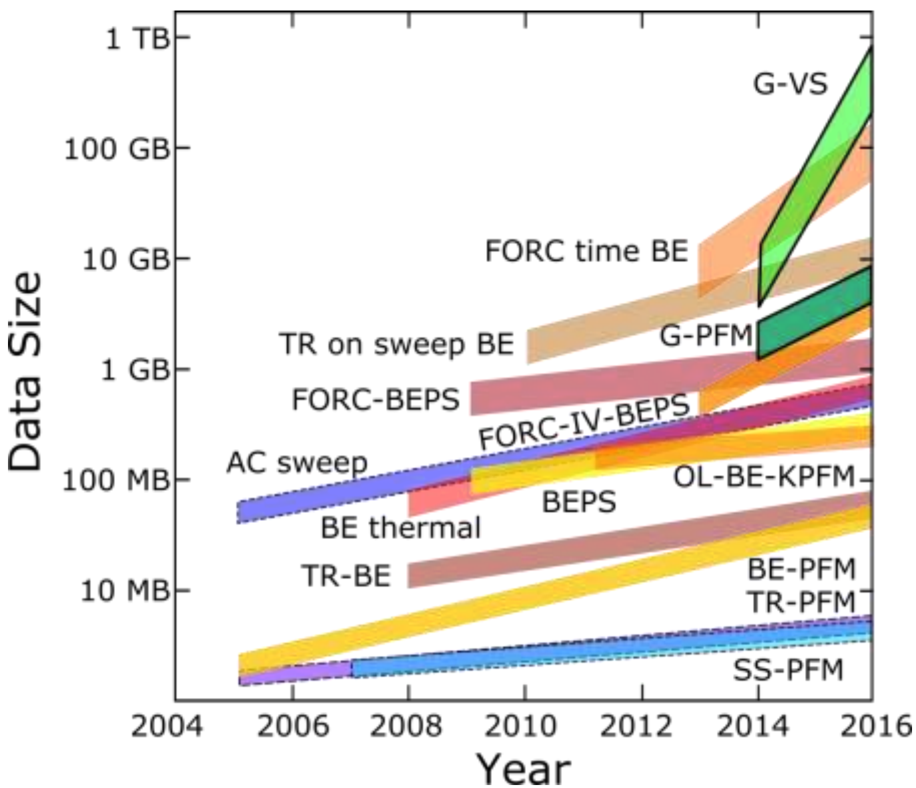
Instrument Tier

?

Interactive visualization, analysis, storage on supercomputers

Growing Data Sizes and Dimensionality

Evolution of Scanning Probe Microscopy Data



- 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

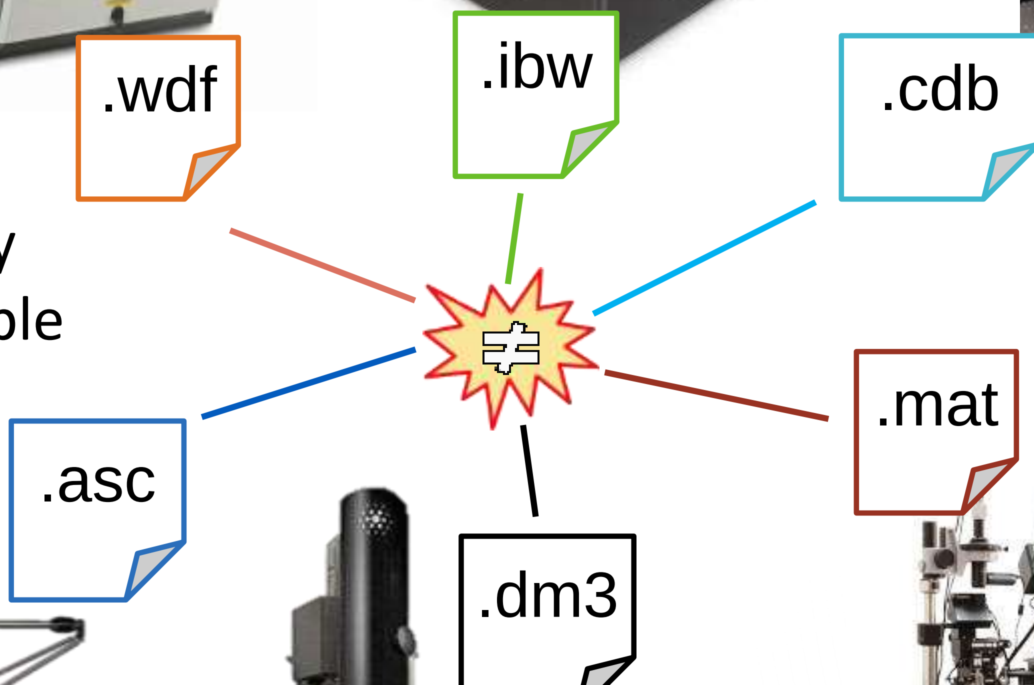
Kalinin et al., *ACS Nano*, 9068-9086, 2015

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

- Proprietary
- Incompatible



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

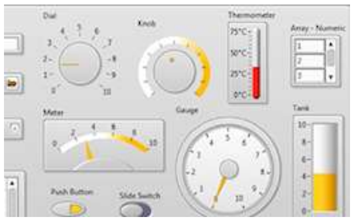
3. Disorganized and unorganized communities

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 repository for data processing, analysis

Interactive visualization + analysis + storage on supercomputers

Expectation of Data Model / File Format

- 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)

Universal Data Model

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

<u>Example data types</u>:

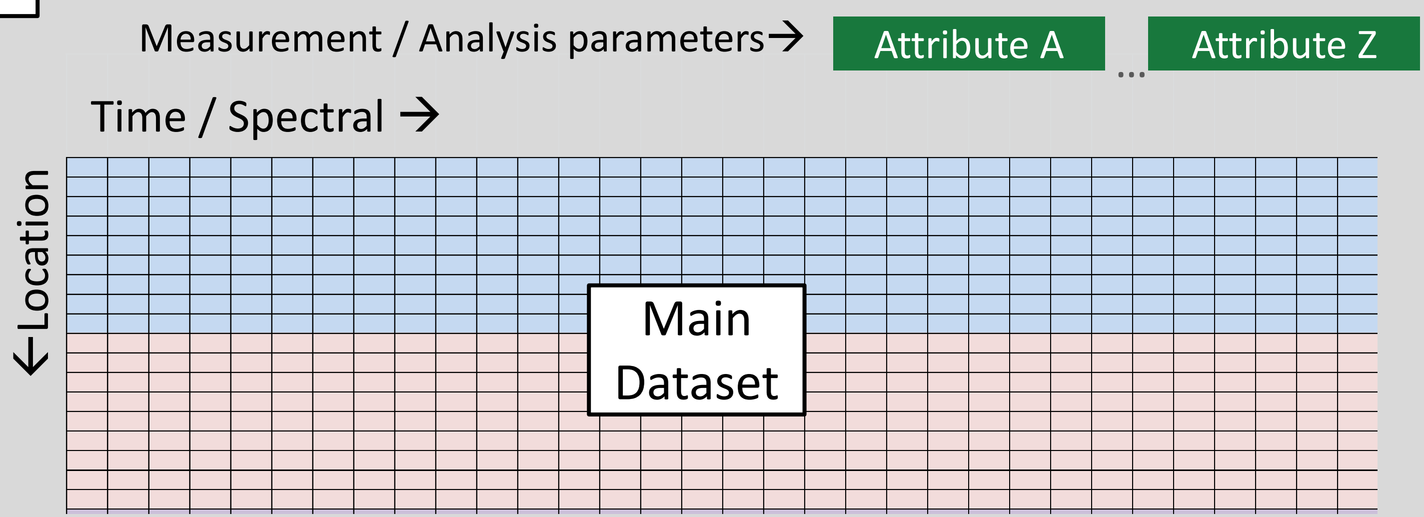
- 2D images = (N_{pixels} x 1)
- Single spectra = (1 x Z_{steps})
- Spectroscopy on grid = (N_{pixels}x V_{steps})

Position Values

x location value	y location value	z location value
0.5	0.5	0.5
1	0.5	0.5
1.5	0.5	0.5
0.5	1	0.5
1	1	0.5
1.5	1	0.5
0.5	1.5	0.5
1	1.5	0.5
1.5	1.5	0.5
0.5	0.5	1
1	0.5	1
1.5	0.5	1
0.5	1	1
1	1	1
1.5	1	1
0.5	1.5	1
1	1.5	1
1.5	1.5	1

Position Indices

x location index	y location index	z location index
1	1	1
2	1	1
3	1	1
1	2	1
2	2	1
3	2	1
1	3	1
2	3	1
3	3	1
1	1	2
2	1	2
3	1	2
1	2	2
2	2	2
3	2	2
1	3	2
2	3	2
3	3	2



Spectroscopic Indices

observable D index	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
observable C index	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
observable B index	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
observable A index	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4

observable D index	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
observable C index	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
observable B index	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
observable A index	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4

Spectroscopic Values

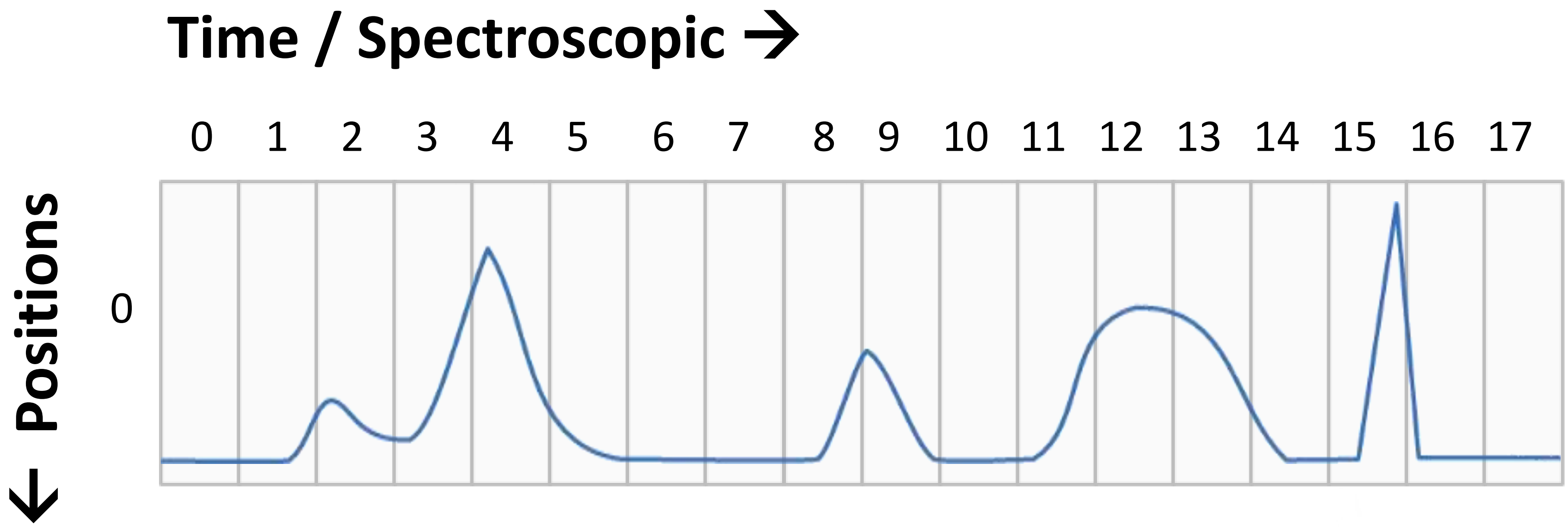
observable D value	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
observable C value	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
observable B value	0.125	0.125	0.125	0.125	0.25	0.25	0.25	0.25	0.375	0.375	0.375	0.375	0.5	0.5	0.5	0.5
observable A value	0.125	0.25	0.375	0.5	0.125	0.25	0.375	0.5	0.125	0.25	0.375	0.5	0.125	0.25	0.375	0.5

observable D value	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
observable C value	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
observable B value	0.125	0.125	0.125	0.125	0.25	0.25	0.25	0.25	0.375	0.375	0.375	0.375	0.5	0.5	0.5	0.5
observable A value	0.125	0.25	0.375	0.5	0.125	0.25	0.375	0.5	0.125	0.25	0.375	0.5	0.125	0.25	0.375	0.5

Additional information available at:
https://github.com/pycroscopy/pycroscopy/blob/master/docs/Pycroscopy_Data_Formatting.pdf

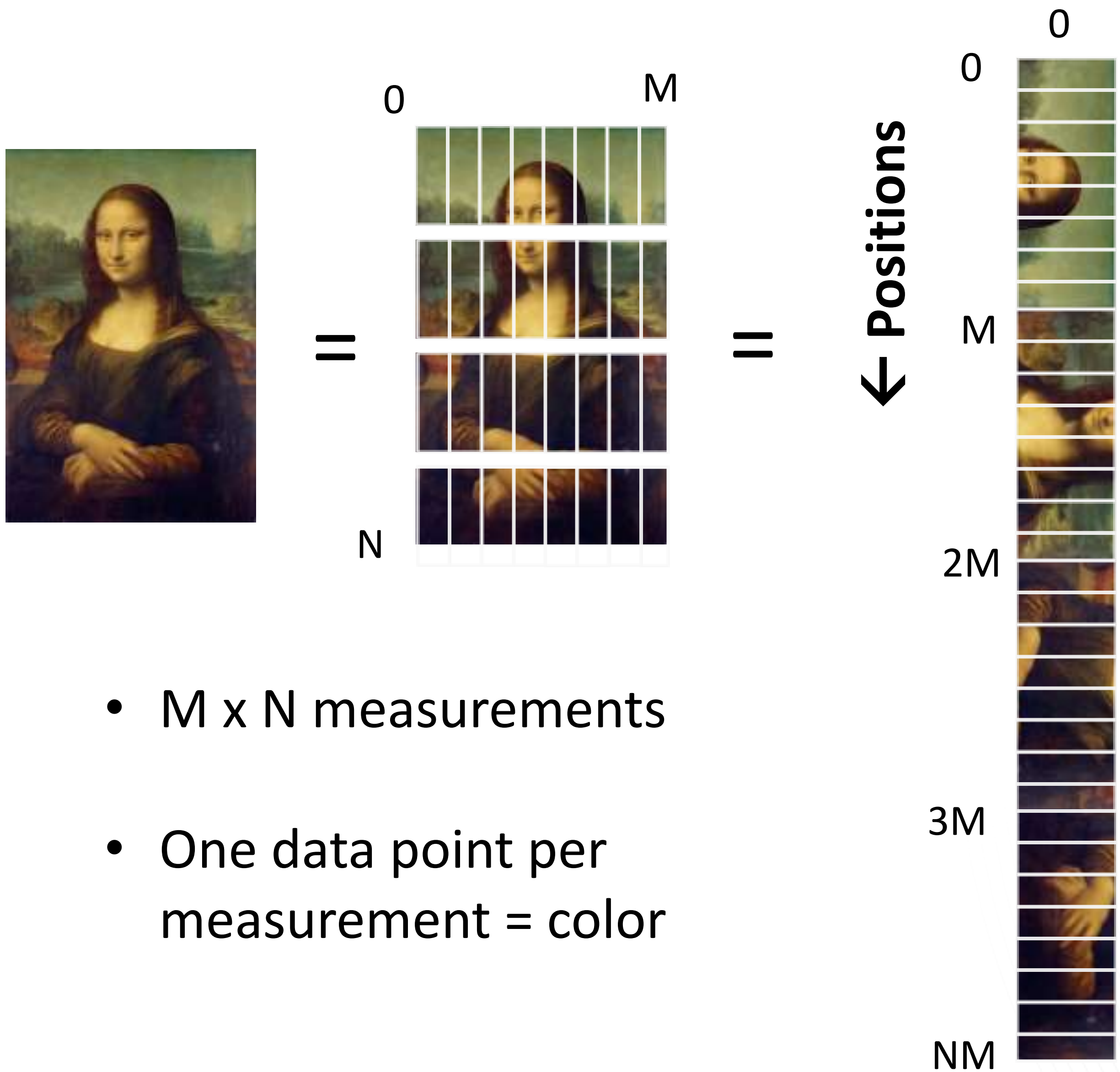
Suhas Somnath, somnaths@ornl.gov

Universal Data Model – 1D spectra



- Each measurement = 17 data points in time
- Single measurements = single entry in Positions axis

Universal Data Model – 2D Image

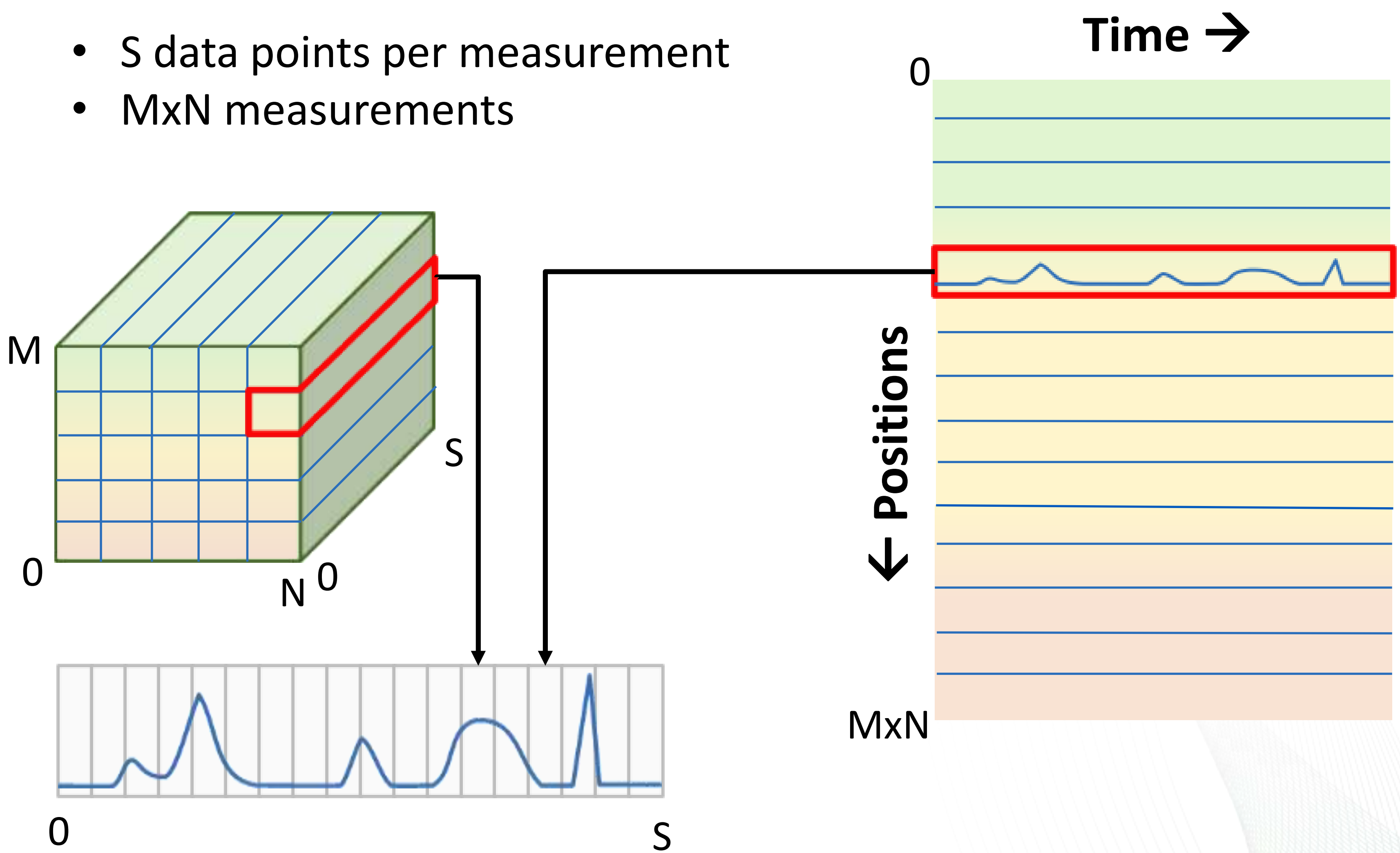


- M x N measurements flattened to 1D array

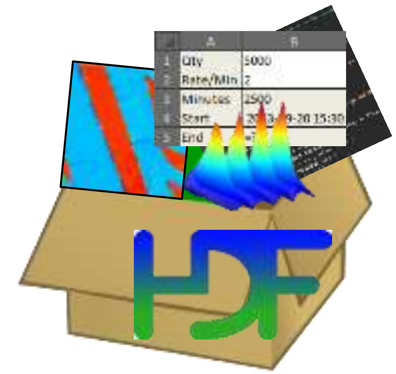
- M x N measurements
- One data point per measurement = color

Universal Data Model – Spectra on Grid

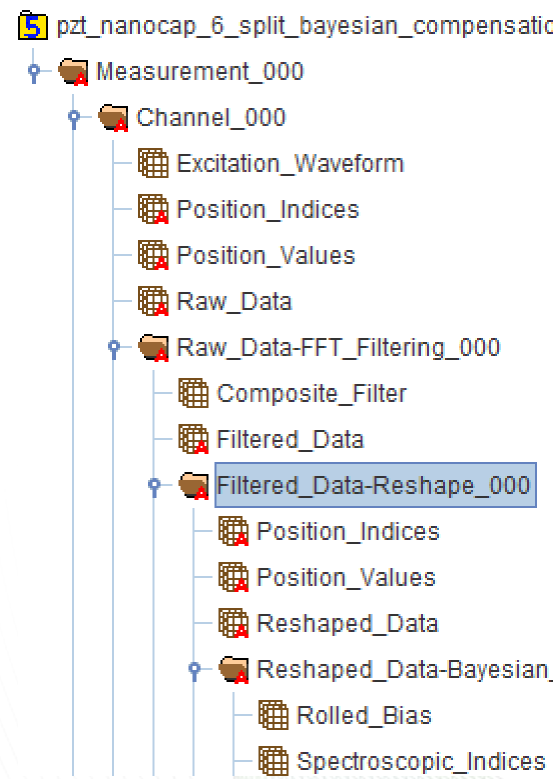
- S data points per measurement
- MxN measurements



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 compatible



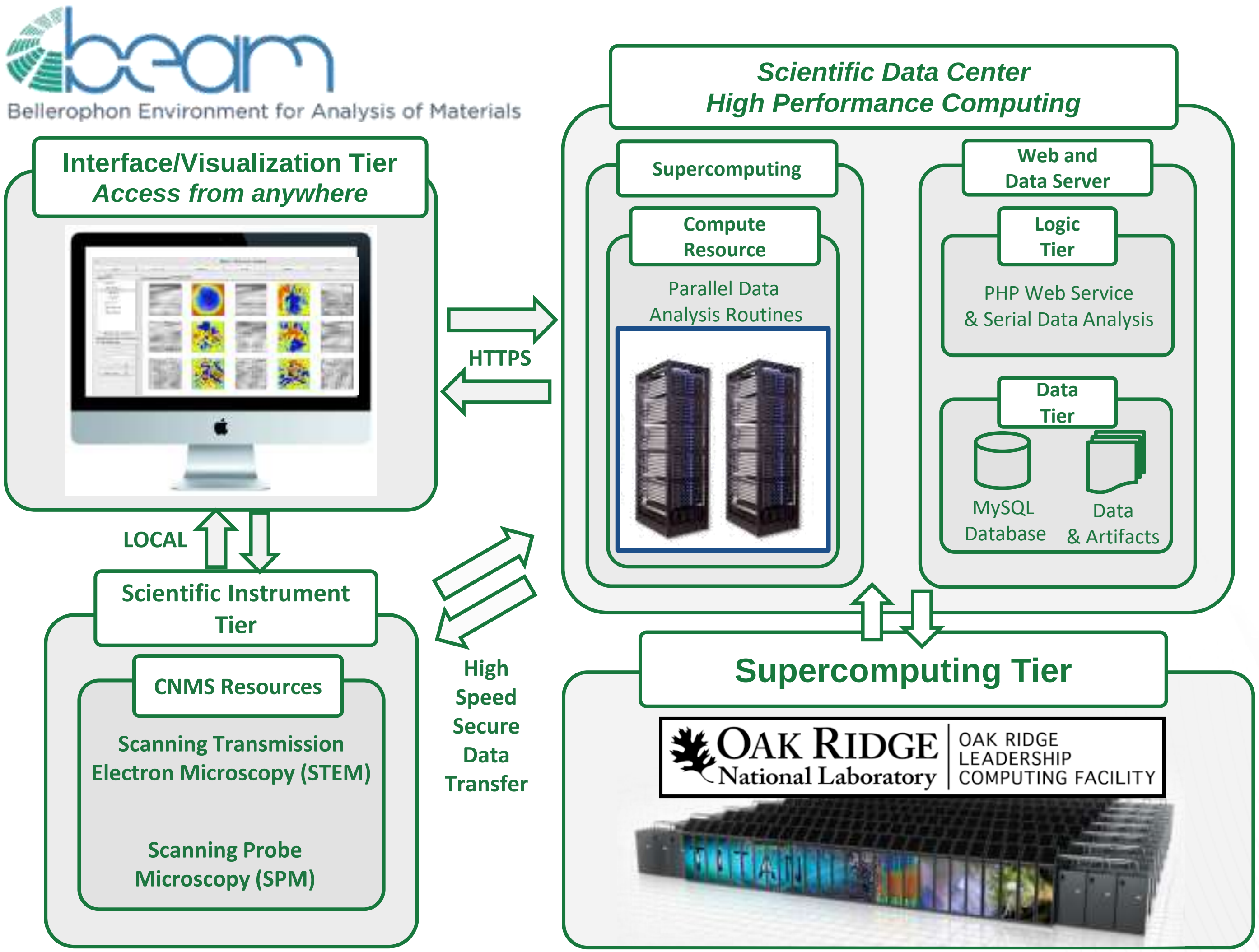
Suhas Somnath, somnaths@ornl.gov

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
- Free
- Scalable to multiple CPU cores and HPC

(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).

Python for Scientific Research

Very easy to learn + code

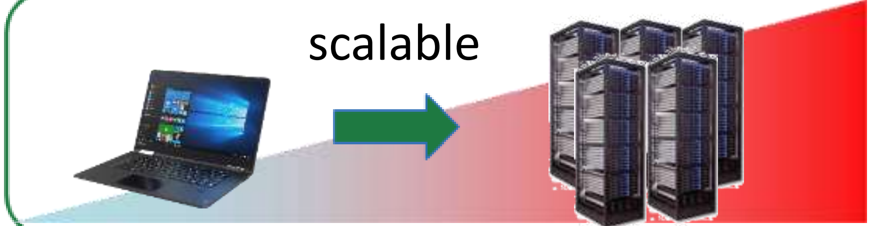
Numerous, **powerful** libraries for science

scikit learn

NumPy

SciPy

scalable



Strong user community

stackoverflow

open source

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

Cross-platform

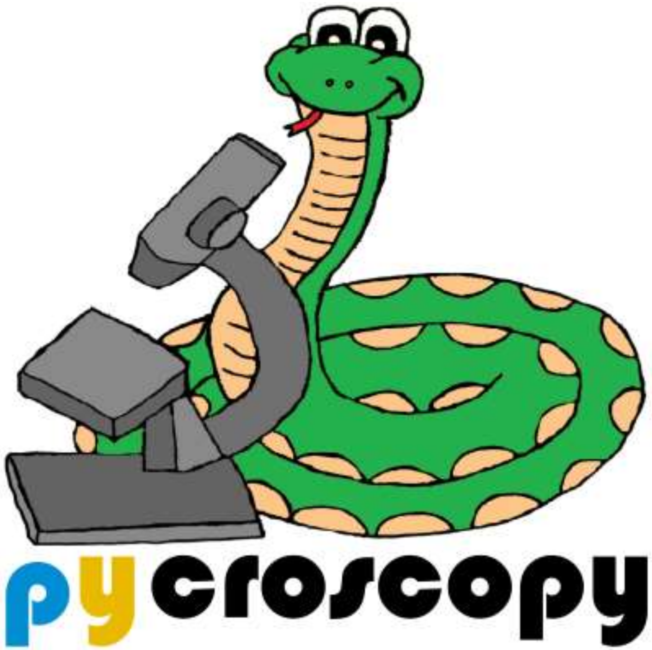
MacOS

Established standard for:

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

All for a princely sum of …. **$0!**

pycroscopy



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

Pycroscopy - Organization

pycroscopy

I/O
- Data translators (proprietary formats to HDF5)

Analysis
- Physical model specific
- Fitting to model, etc.
- Physics based Regression

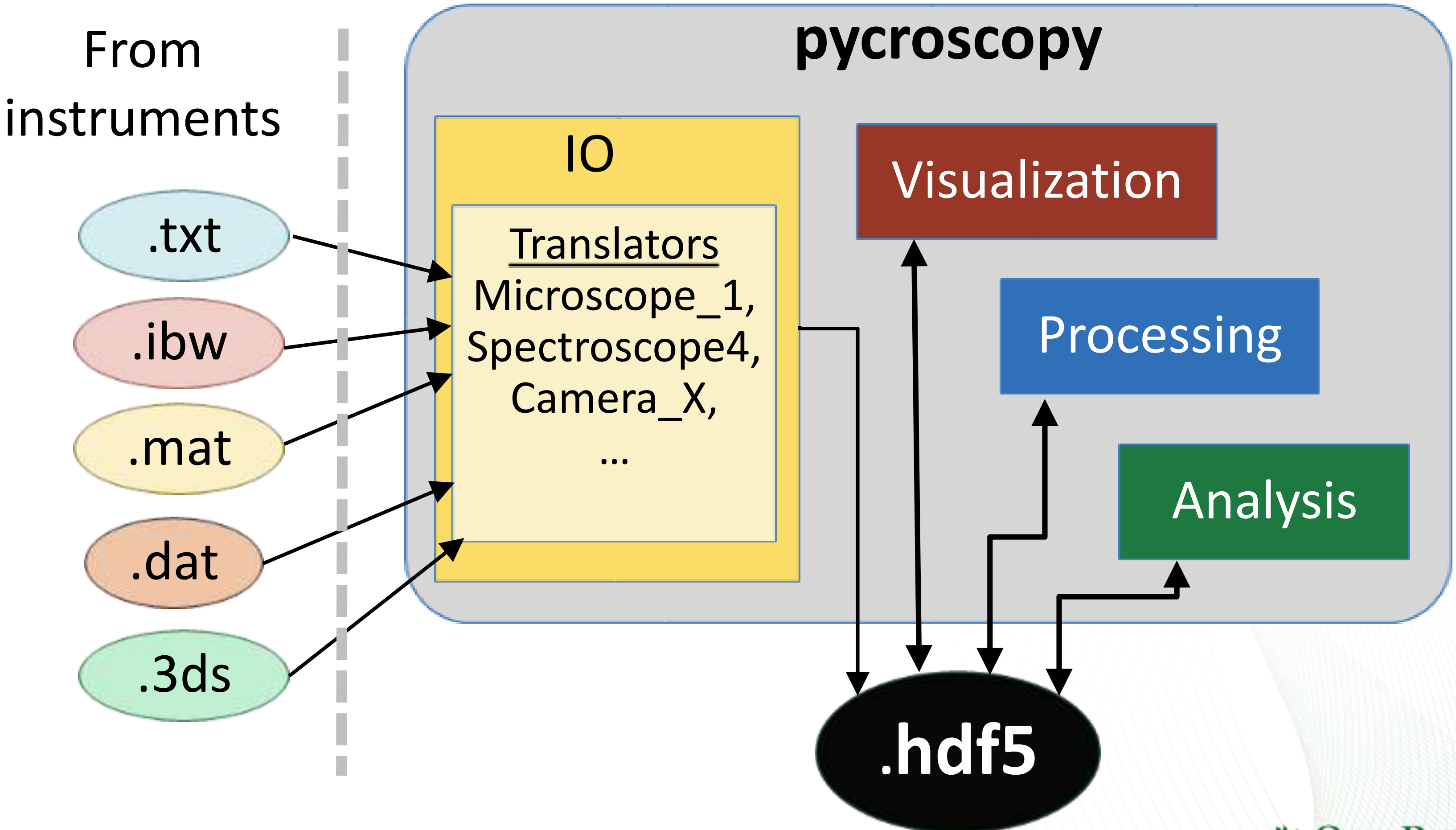
core
- HDF5 file i/o operations
- Base classes, visualization…

Processing
- Physical model agnostic
- Image filtering, registration, etc.

Visualization
- Plotting utilities
- Jupyter widgets

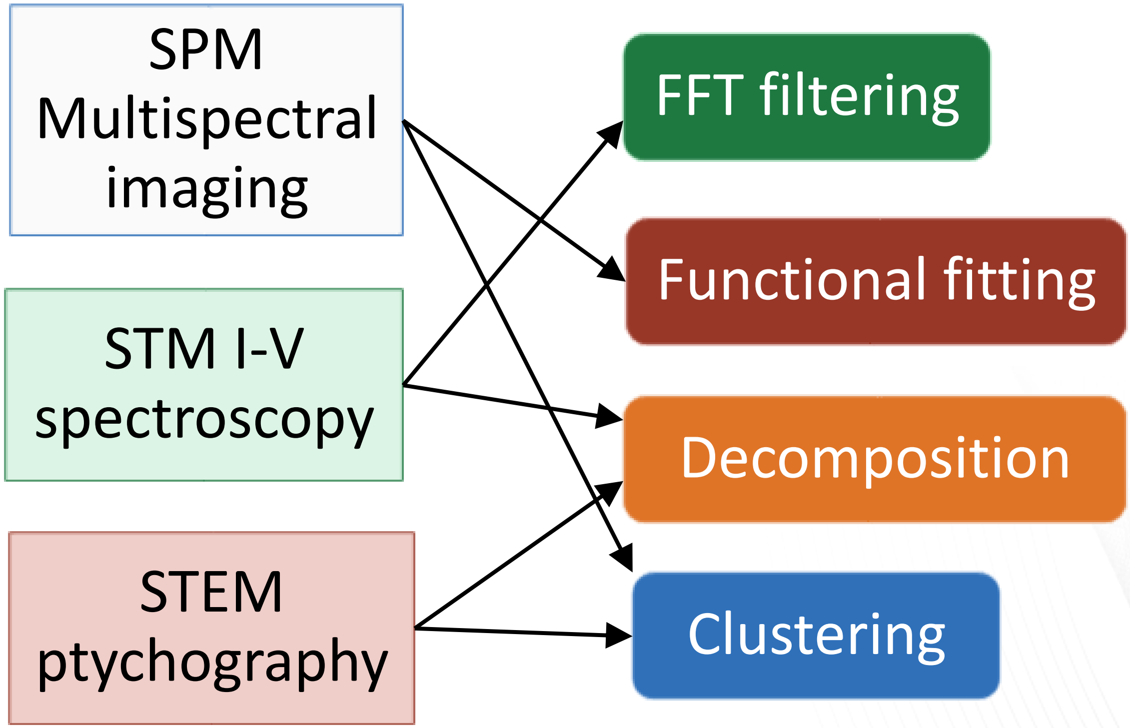
Entering the Pycroscopy Ecosystem

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



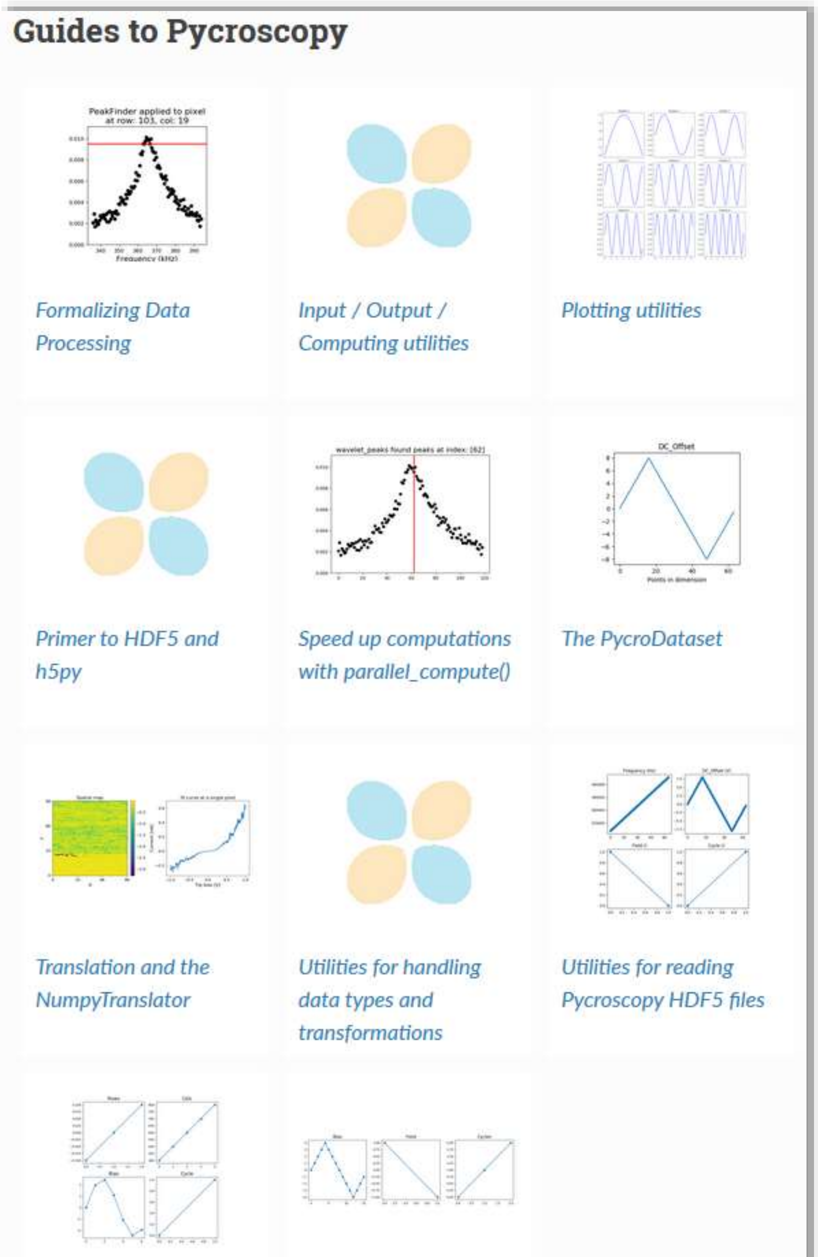
Pycroscopy – Instrument Agnostic Code

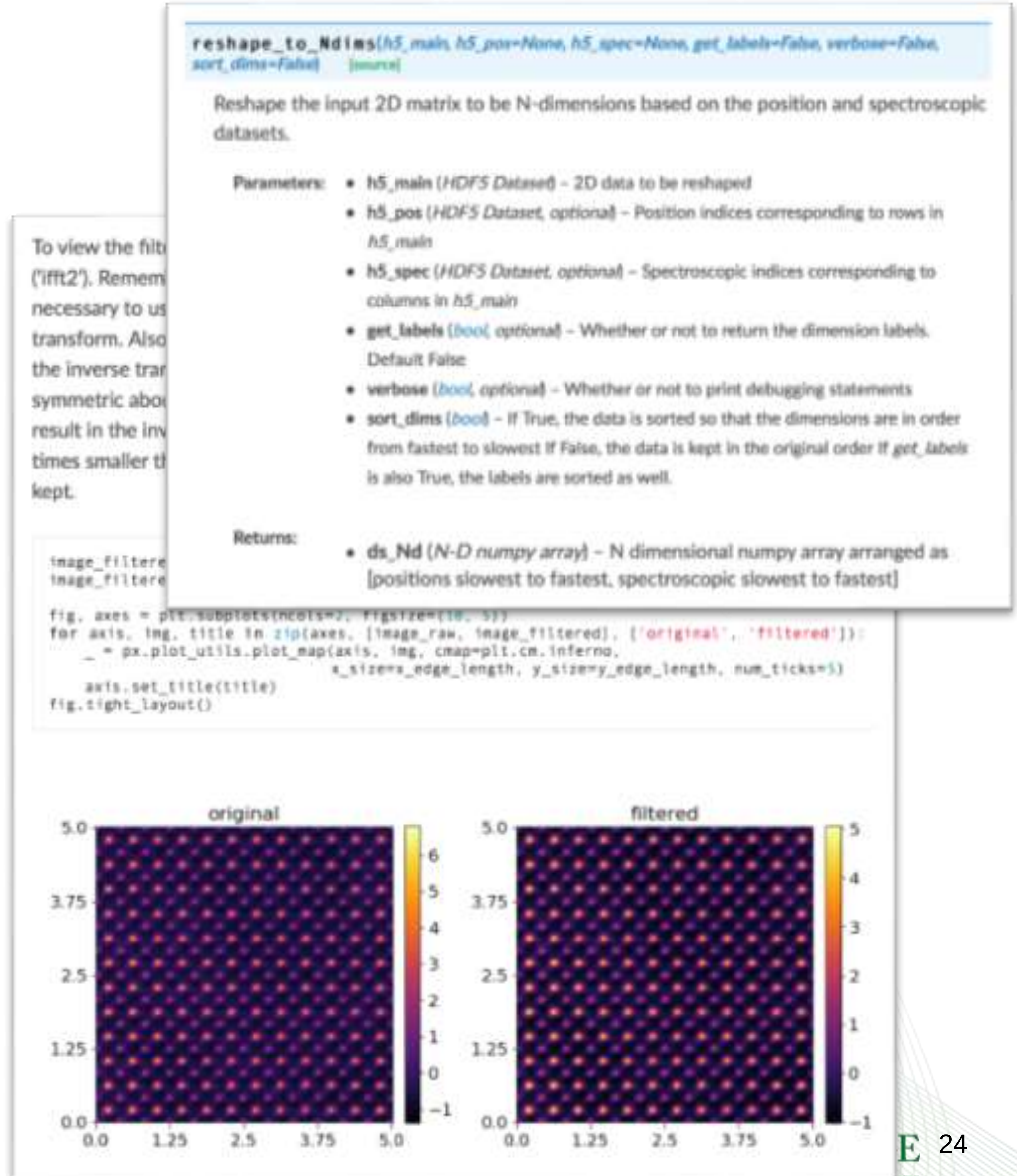
- Instrument-agnostic data allows instrument-agnostic code
- Single version of analysis and processing routine
- Brings multiple scientific communities together



Suhas Somnath, somnaths@ornl.gov

Pycroscopy - Well documented

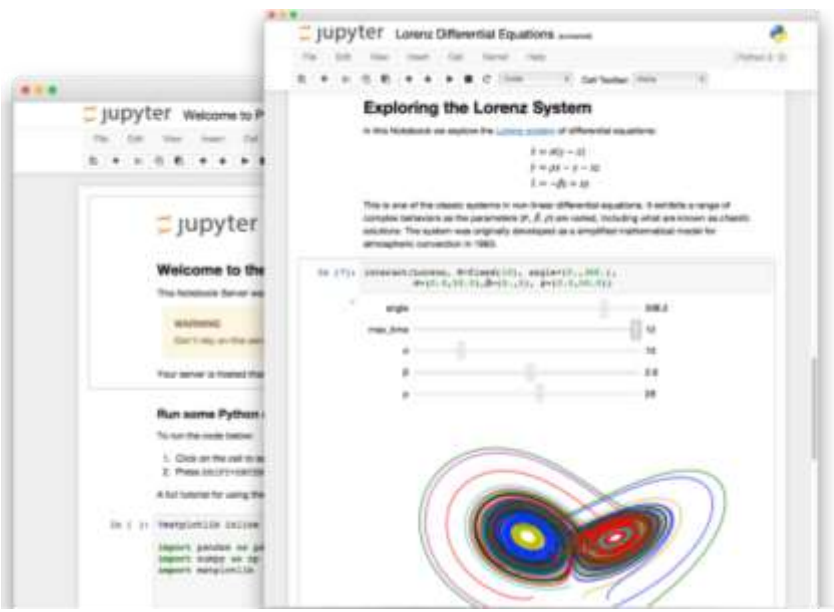




Jupyter Notebooks

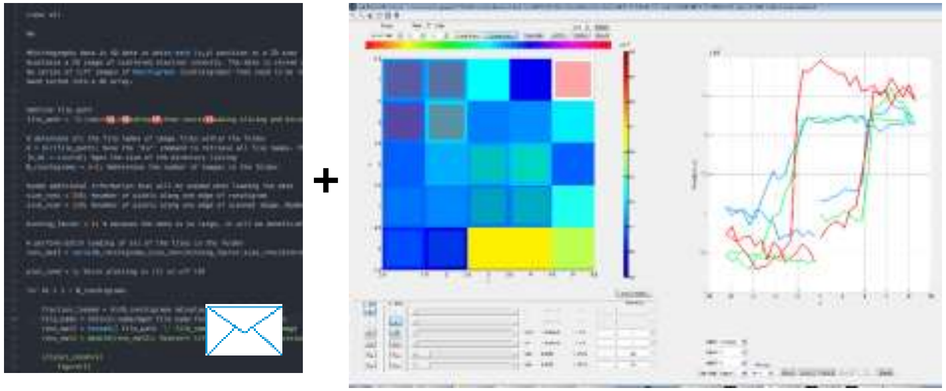
jupyter

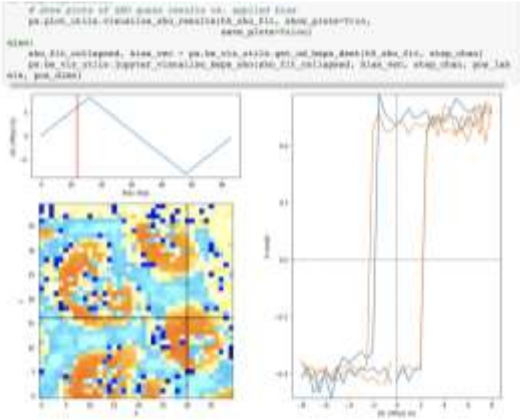
Jupyter Notebook



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

Pycroscopy - Supporting User Research





Before 2016	Since 2016
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

Truly Achieving Open Science, Reproducibility

Aim – ALL scientific journal papers accompanied with:

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

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.

Jupyter notebook associated with paper

Visualizing Filtering Results

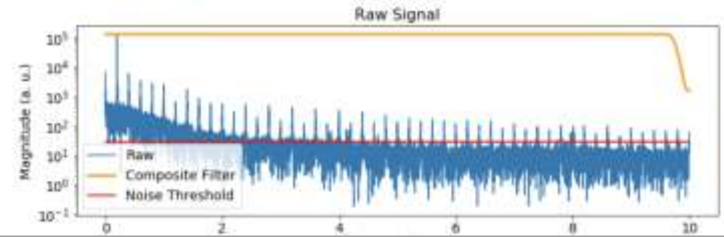
Run the next cell to see what the IV curves look like for the chosen filter parameters. If the current set of filter parameters do not work as well, change the parameters in the previous cell and try again.

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)
```



DOI associated with data (raw → paper figures)

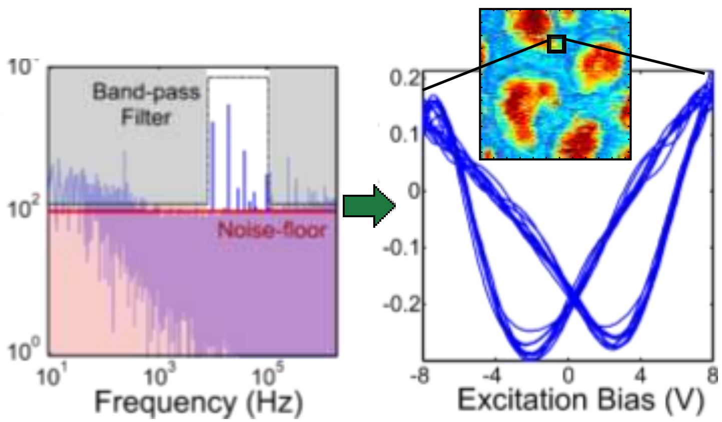
Oak Ridge National Laboratory
Leadership Computing Facility
10.13139/OLCF/1410993
Download

Authors

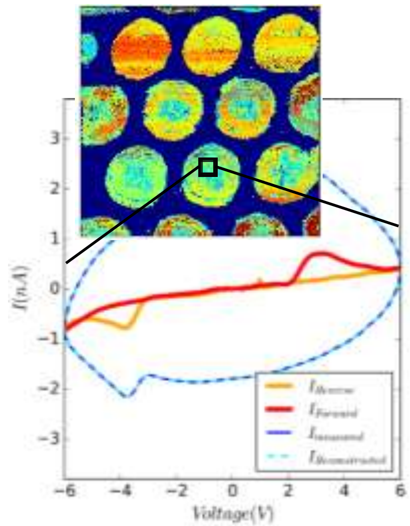
Somnath, Suhas	somnaths@ornl.gov
Law, Kody	lawkj@ornl.gov
Morozovska, Anna	anna.n.morozovska@gmail.com
Maksymovych, Petro	maksymovychp@ornl.gov
Kim, Yunseok	ykim943@gmail.com
Lu, Xiaoli	xllu@xidian.edu.cn
Alexe, Marin	M.Alexe@warwick.ac.uk

Pycroscopy - Scientific Advancements

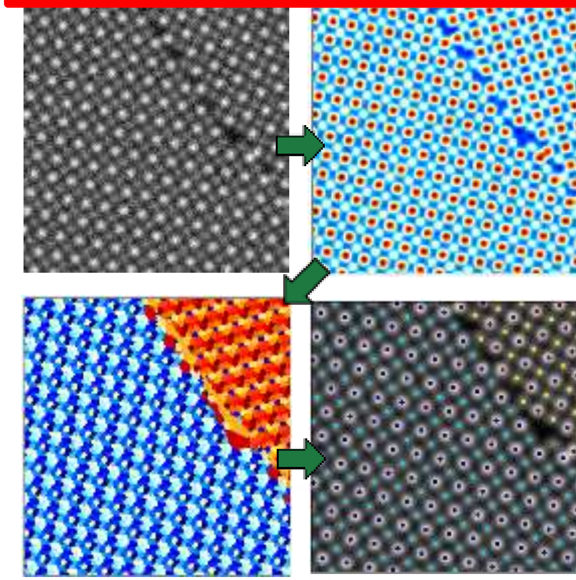
3,500x faster imaging via adaptive signal filtering, linear unmixing of signals



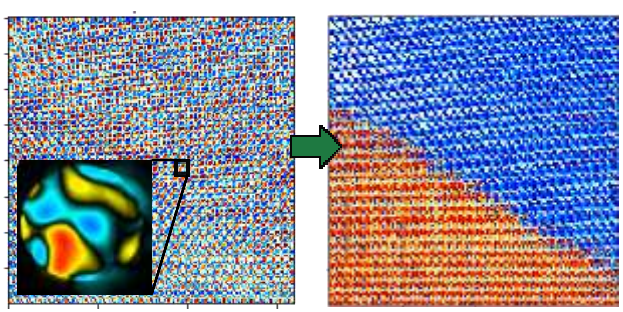
200x faster spectroscopy via Bayesian inference



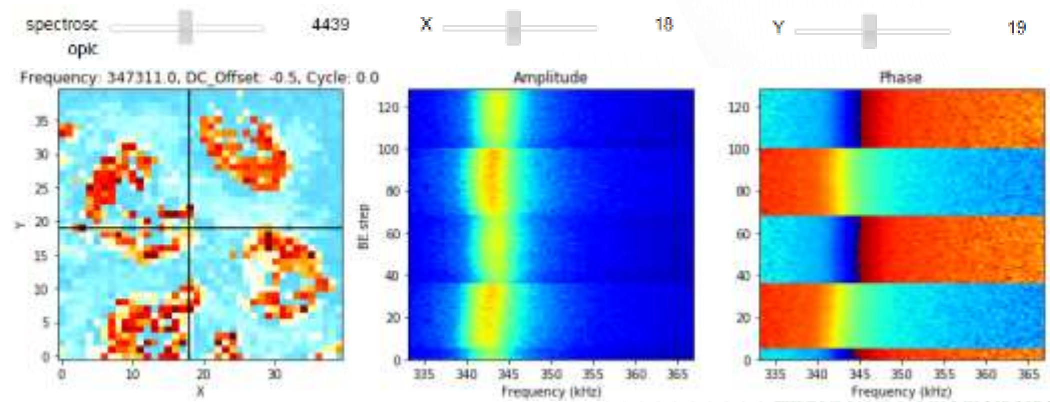
Separating uncorrelated data from correlated data to clean images



Identifying invisible patterns using multivariate analysis



Simplified navigation multidimensional data - users



Suhas Somnath, somnaths@ornl.gov

Pycroscopy Progression

Scaling up Computing:

2016	2017	2018	2019
Single core	Multi-core Single CPU	Multi-core Multi CPU	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 + Bio-Chemistry	+ Neutron Science

Thank you

Questions?

Suhas Somnath, somnaths@ornl.gov