

Introduction to pycroscopy

“SPM in the Cloud”

Rama Vasudevan
Center for Nanophase Materials Sciences,
Oak Ridge National Laboratory

October 4th, 2021

Virtual SPM ML School
Tutorial

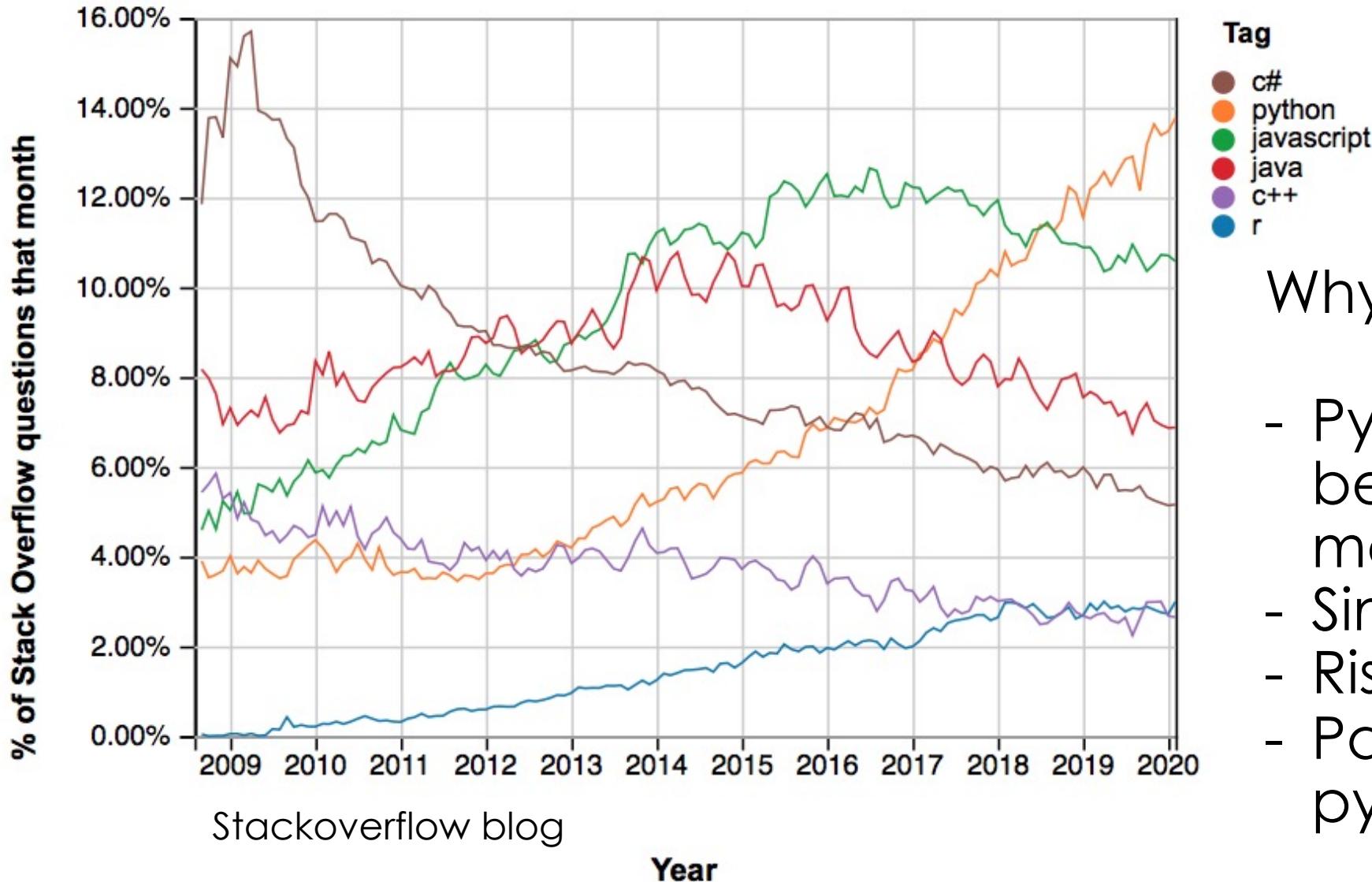
ORNL is managed by UT-Battelle, LLC for the US Department of Energy



Contributors

- Gerd Duscher
- Suhas Somnath
- Maxim Ziatdinov
- Mani Valetti
- Nicole Creange
- Many others...

Python is where it's at



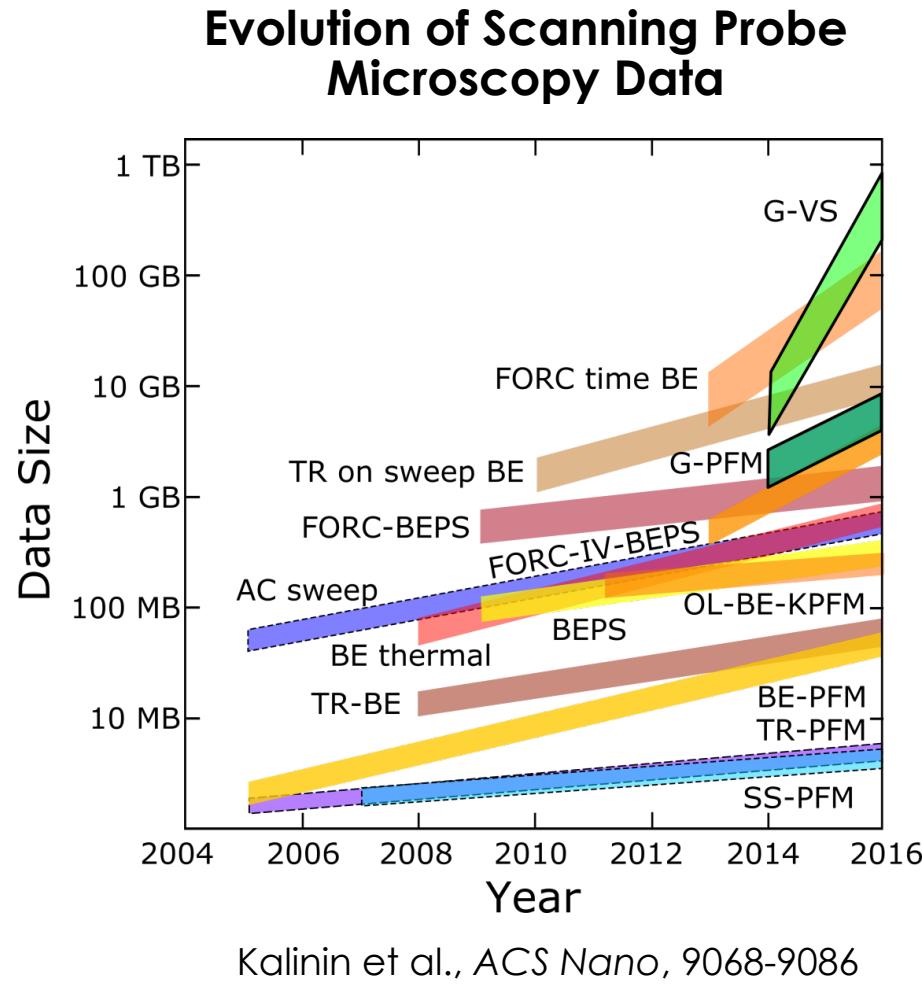
Tag

- c#
- python
- javascript
- java
- c++
- r

Why?

- Python is second best language for most things
- Simple/readable
- Rise of ML
- Pandas, flask, pyTorch etc.

These measurements lead to large datasets...



- 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

Disjoint & Unorganized Communities

The image consists of six panels arranged in a grid-like structure, each featuring a scientific instrument and a corresponding software logo.

- Top Left Panel:** Shows a RENISHAW inVia Raman Microscope. Below it is a large blue hexagonal icon containing a white letter 'R'.
- Top Middle Panel:** Shows an IGOR Pro software interface with a red waveform plot. Below it is a green hexagonal icon containing a blue hexagon with a white 'C++' symbol.
- Top Right Panel:** Shows a nanoIR spectrometer connected to a computer monitor displaying a spectroscopic image. Below it is an orange hexagonal icon containing a white letter 'C'.
- Bottom Left Panel:** Shows a scienta omicron X-ray microscopy system. Below it is a light blue hexagonal icon containing a black 'julia' logo.
- Bottom Middle Panel:** Shows a Python software interface with a yellow and blue Python logo. Below it is a white hexagonal icon containing a black 'python' logo.
- Bottom Right Panel:** Shows a Bruker NMR spectrometer. Below it is a grey hexagonal icon containing a MATLAB software interface with its characteristic 3D surface plot.

Top Left Panel:

- Clustering
- Fit spectra ...

Top Middle Panel:

- Filter image
- Register Image
- ...

Top Right Panel:

- Fit Spectra
- SVD Filtering ...

Bottom Left Panel:

- FFT Filtering
- SVD Filtering ...

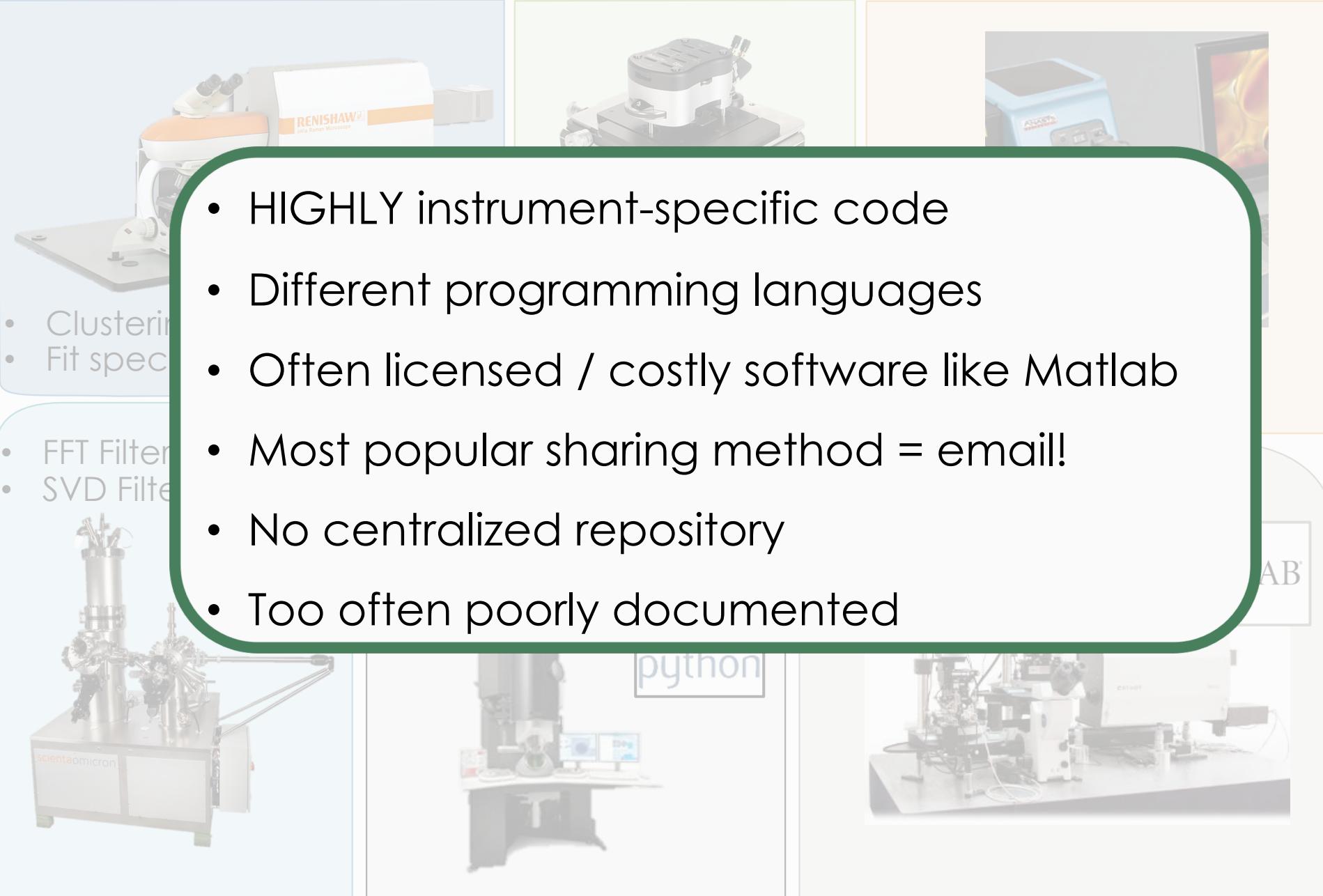
Bottom Middle Panel:

- FFT Filtering
- Classify Images ...

Bottom Right Panel:

- Register Images
- Clustering

Cannot Share Code Efficiently

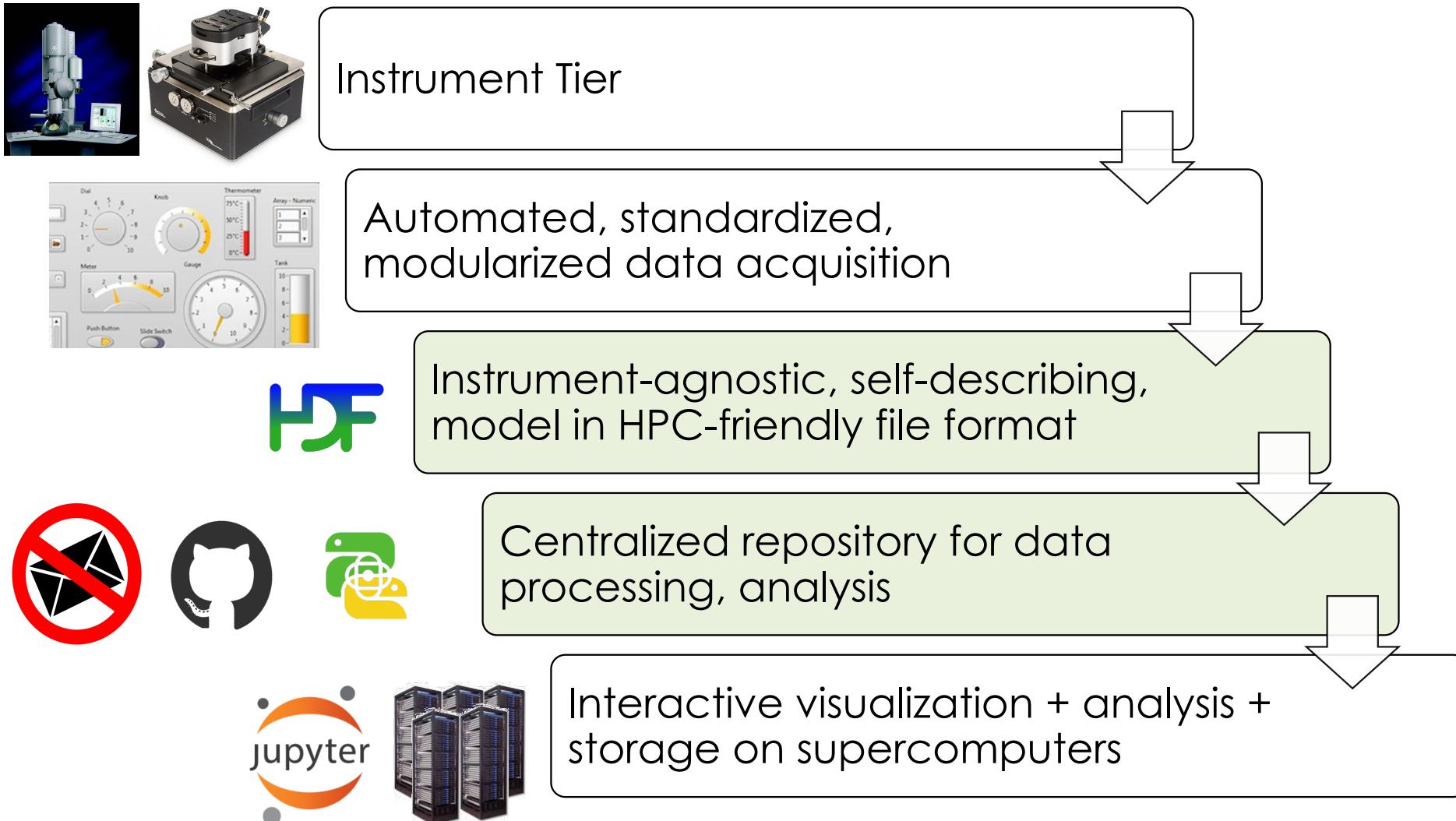


- HIGHLY instrument-specific code
- Different programming languages
- Often licensed / costly software like Matlab
- Most popular sharing method = email!
- No centralized repository
- Too often poorly documented

~~Problems~~ Opportunities in SPM

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

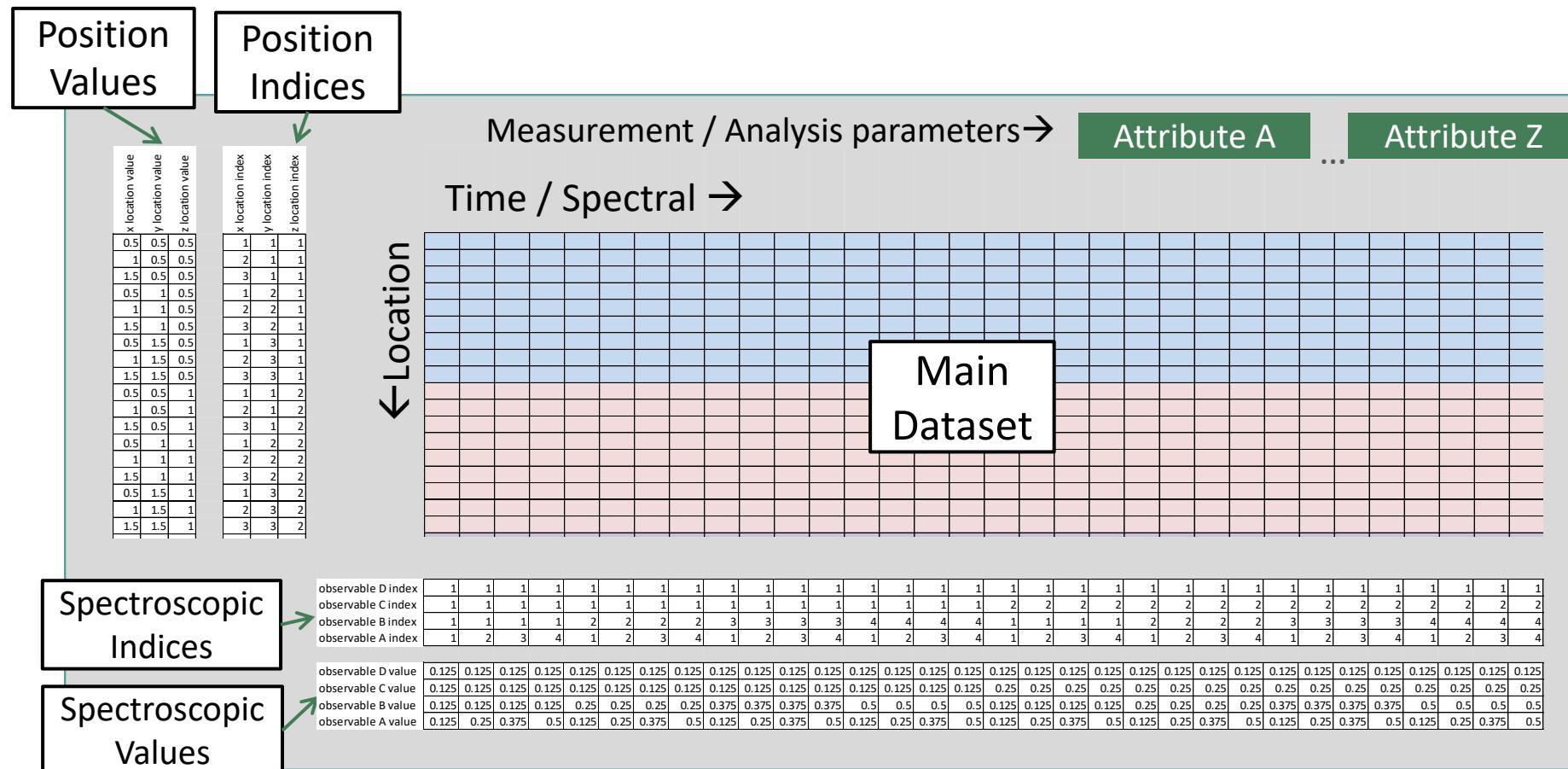


Expectation of Data Model

Feature	sidpy	USID
Accommodate N-dimensional datasets of different shapes, sizes	Yes	Yes
Accommodates data without N-dimensional representation	No	Yes
Stores metadata	Yes	Yes
Stores very large files	Yes	Yes
Enables Parallel processing	Yes (dask)	Yes
Traceable	Medium	High
Ease of use	High	Medium

Universal Spectroscopic and Imaging Data (USID)

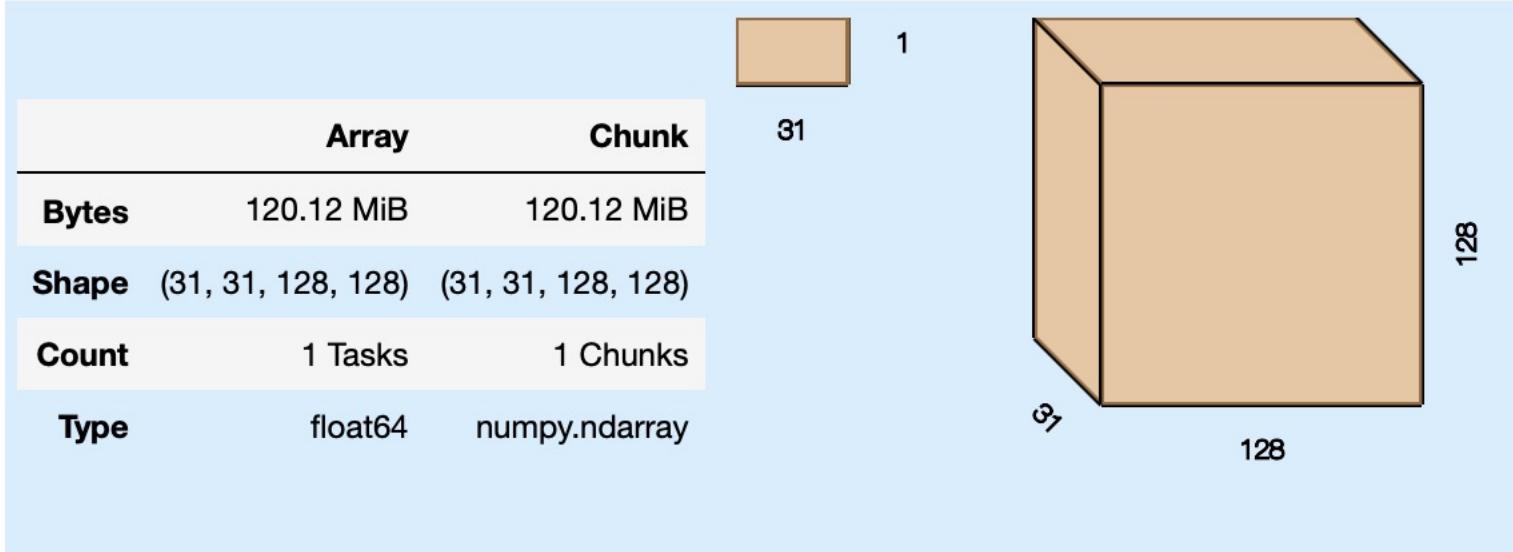
- Data stored as 2D matrix of (position x spectral values) regardless of dimensionality
- Ancillary datasets explain the data



Detailed information on [pyUSID website](#)

Sidpy Data Model (sidpy.Dataset)

Dataset Object built on top of dask arrays



Benefits of the model:

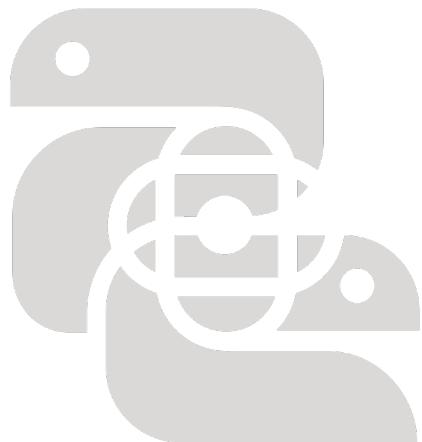
- Easy to understand
- Sidpy takes care of plotting (dataset.plot())
- Can easily perform parallel computations
- Easy to push to file including metadata
- Useful for data pipelining

- Maintain N-dimensional form
- All of the advantages of dask (large sizes, parallel compute)
- Additional data given for each dimension of dataset
- Metadata stored in dictionary
- Can be readily pushed to hdf5 files



pycroscopy

An ecosystem for microscopy data ingestion, analytics and visualization



pycroscopy

A general-purpose package for microscopy imaging and spectroscopy data analytics, including registration, image cleaning, unmixing, ect.



scifireaders

For ingesting a variety of microscopy files for output to sidpy dataset objects

pyusid

Python package for reading and visualizing our universal spectral imaging dataset format

pynsid

Python package for reading writing and visualizing our N-dimensional spectral imaging dataset format

sidpy

Python utilities for storing, visualizing, and Spectroscopic and Imaging Data (SID)

sidpy.hdf.hdf_utils.get_attr

`sidpy.hdf.hdf_utils.get_attr(h5_object, attr_name)`
[source]

Returns the attribute from the h5py object

- Parameters:
- `h5_object` (`h5py.Dataset`, `h5py.Group` or `h5py.File`) – object whose attribute is desired
 - `attr_name` (`str`) – Name of the attribute of interest

Returns: `att_val` – value of attribute, in certain cases (byte strings or list of byte strings) reformatted to readily usable forms

Return type: `object`

© Copyright 2020, Suhas Somnath, Gerd Duscher, and contributors.

Built with [Sphinx](#) using a theme provided by [Read the](#)

bglib

Utilities to analyze, fit and visualize Band-excitation and G-mode imaging and spectroscopy data primarily for SNMS SPM users

atomai

Deep learning toolkit for analysis of atomically resolved imaging and spectroscopy datasets

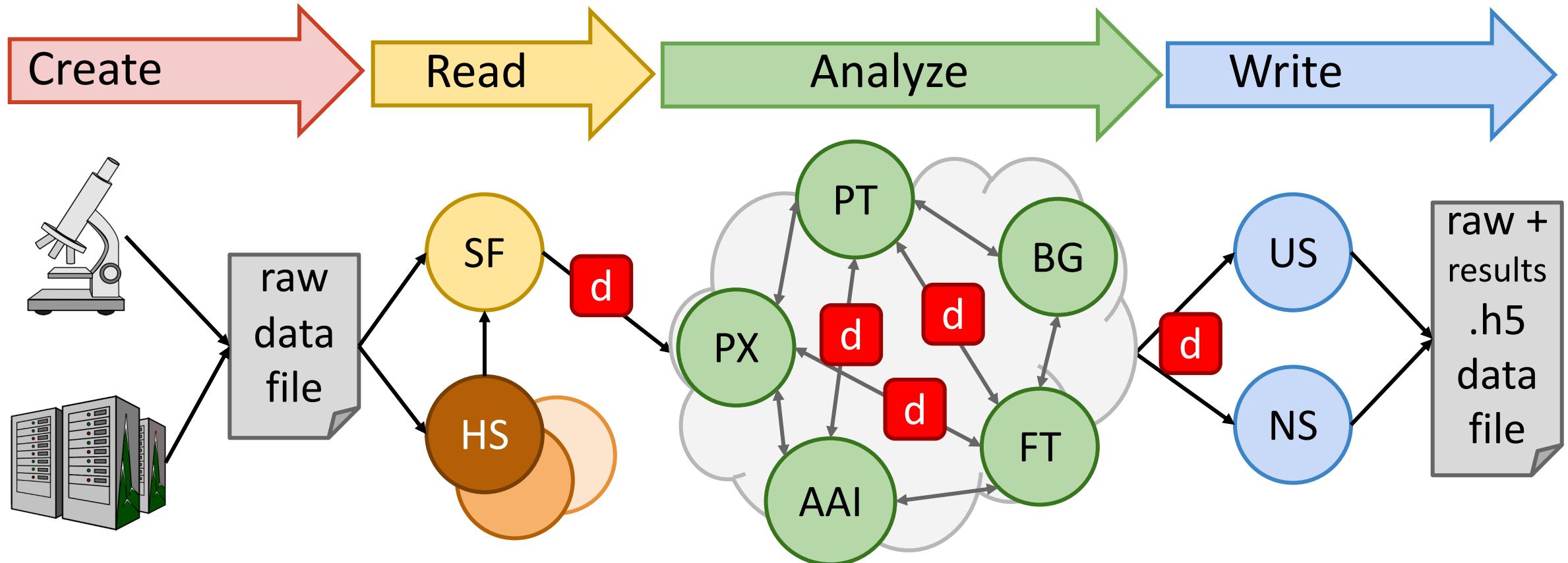
stemtools

Python based codes for analysis of 4D-STEM and aberration corrected vanilla STEM datasets

pytemlib

Python tools for simulation, registration, analysis and visualization of TEM datasets

Pycroscopy philosophy



Data from measurements or simulations are read into `sidpy.Dataset` (d) objects directly by `SciFiReaders` (SF), or through external packages like `HyperSpy` (HS). Data are processed using multiple science packages in the Pycroscopy ecosystem that interoperate via `Dataset` objects. `Dataset` objects are written to HDF5 files via `pyUSID` (US) or `pyNSID` (NS).

Pycroscopy Example Notebook