THE UNIVERSITY of TENNESSEE KNOXVILLE

Lecture 03: Data, Python, and Python Ecosystem

Sergei V. Kalinin

Build the case for machine learning

- Explore the business/scientific problem
- Build workflow (operations, costs, latencies)
- Identify bottlenecks
- Chart the solution
- Prototype the solution
- Test and iterate
- Deploy
- Support
- Upgrade
- Sunset

Homework assignment 2

Identify the type of problem

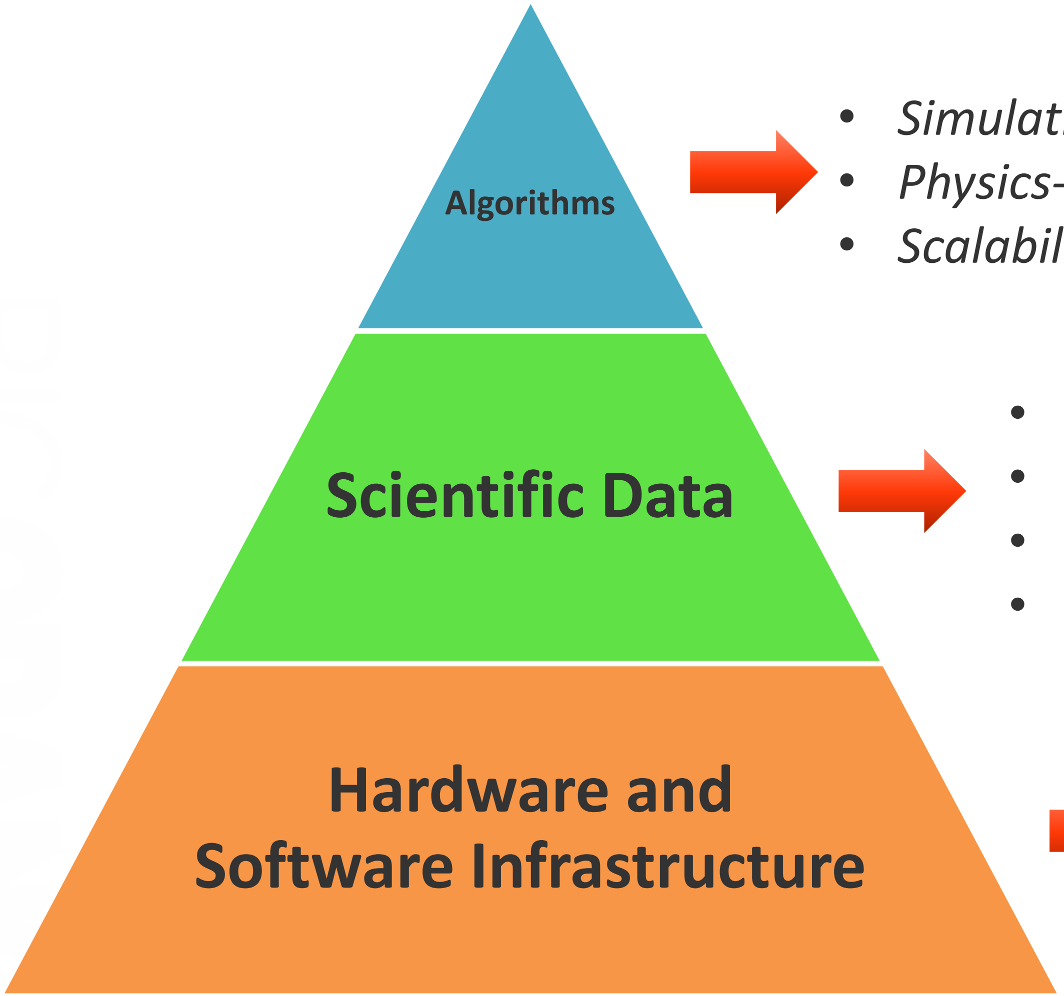
Supervised (inductive) learning
– Given: training data + desired outputs (labels)

- **Unsupervised learning**
– Given: training data (without desired outputs)

- **Semi-supervised learning**
– Given: training data + a few desired outputs

- **Reinforcement/active learning**
– Rewards from sequence of actions

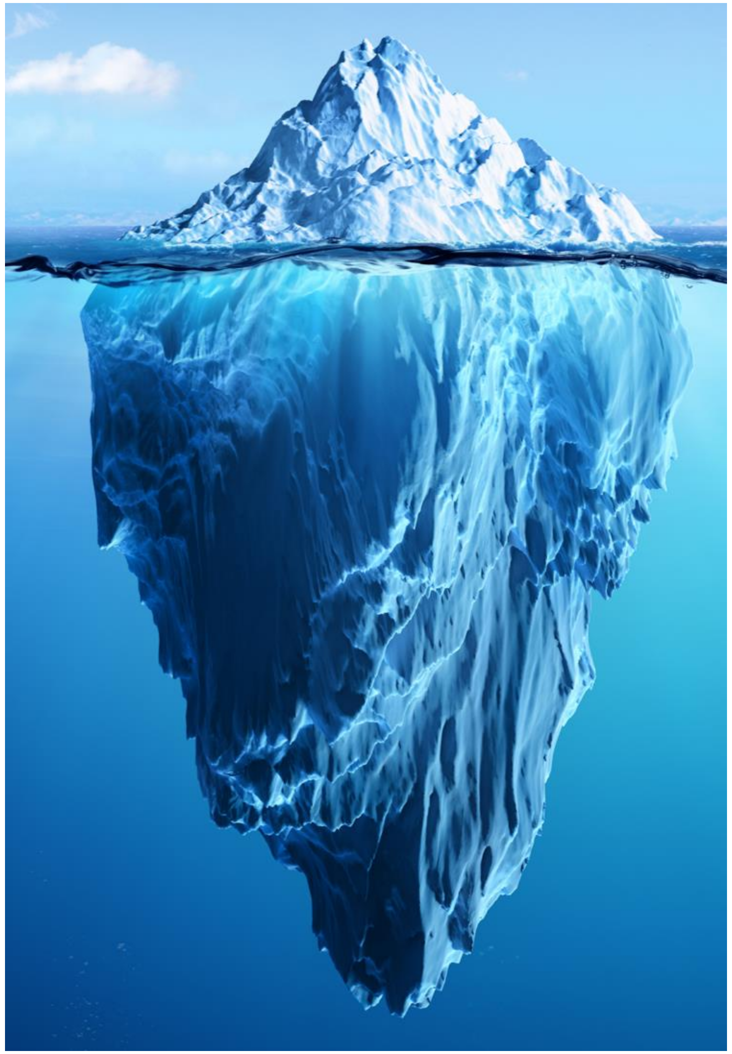
The pyramid of machine learning



- *Simulations*
- *Physics-informed methods*
- *Scalability*

- *Curated*
- *Searchable*
- *Standardized*
- *Best Practices*

- *Codes*
- *Databases*
- *Computing*



Slide courtesy of R. Vasudevan

Why bother with infrastructure?

- Almost all of machine learning relies extensively on having access to good quality data
- In most laboratories, this data is acquired via multiple instruments in different formats, and not findable or accessible, and often lacks necessary metadata for ML labeling
- As such, in many cases, ML in science is impossible especially in the experimental domains, without the necessary investments in data standardization and storage
- Similarly, reproducibility of workflows relies on strongly tested codebases, not one-off scripts.

Slide courtesy of R. Vasudevan

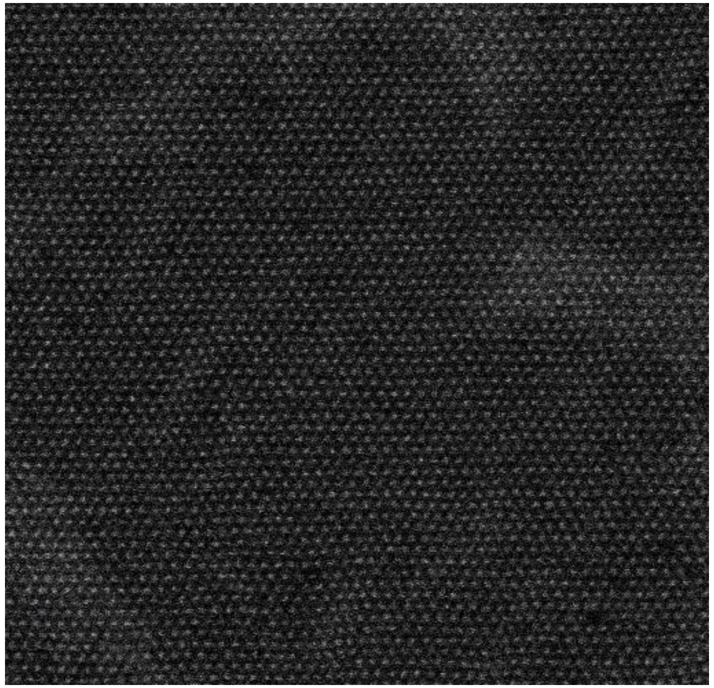
Soon to be a requirement!

- Data management plans in proposals
- Repositories of data and codes associated with publications
- Good to be ready!

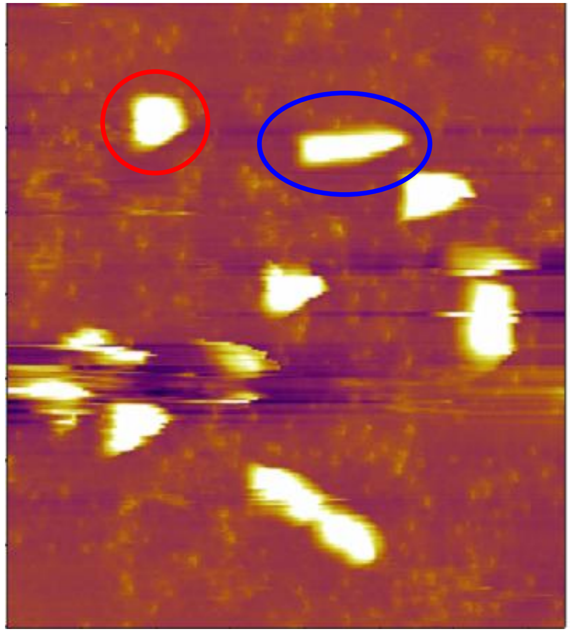
https://www.science.org/content/article/white-house-requires-immediate-public-access-all-u-s--funded-research-papers-2025

Get data – from scientific tools

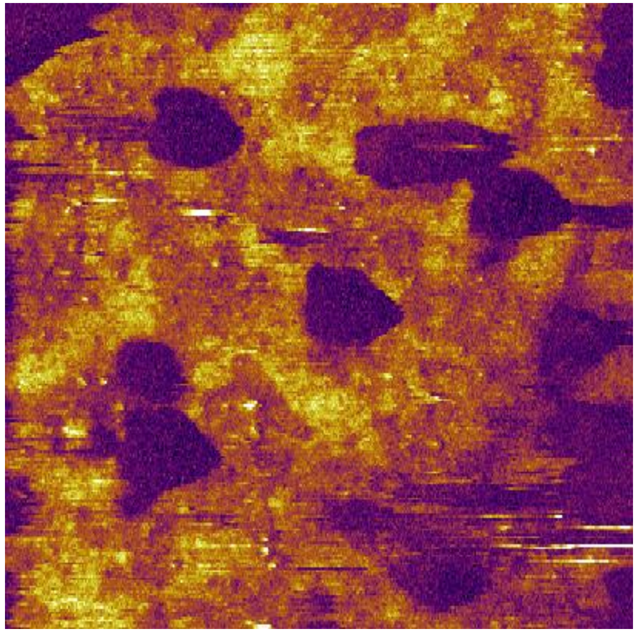
- Spectra
- (Multimodal) Images
- Hyperspectral images
- Videos
- Time traces
- …

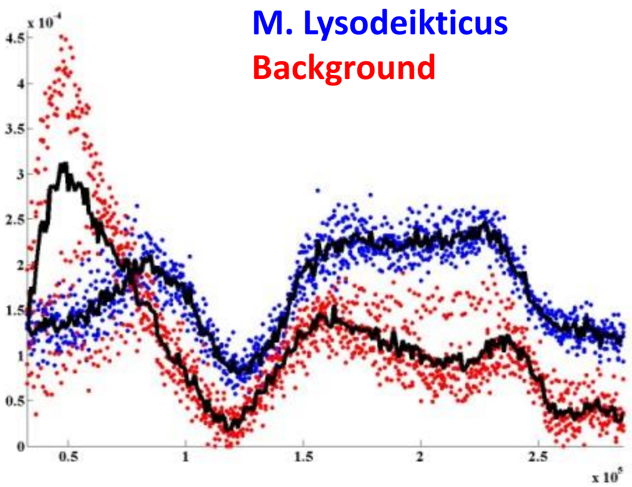


Topography



PFM Amplitude





As scientists, we rarely have to deal with the classical ELT (Extract-Load Transform, aka Data Wrangling) problems. But....

Standardization of Microscope Data

Micro Raman Microscope


Atomic Force Microscope (AFM)


AFM with Infrared spectroscopy (AFM-IR)

Scanning Tunneling Microscope (STM)


Scanning Transmission Electron Microscope (STEM)

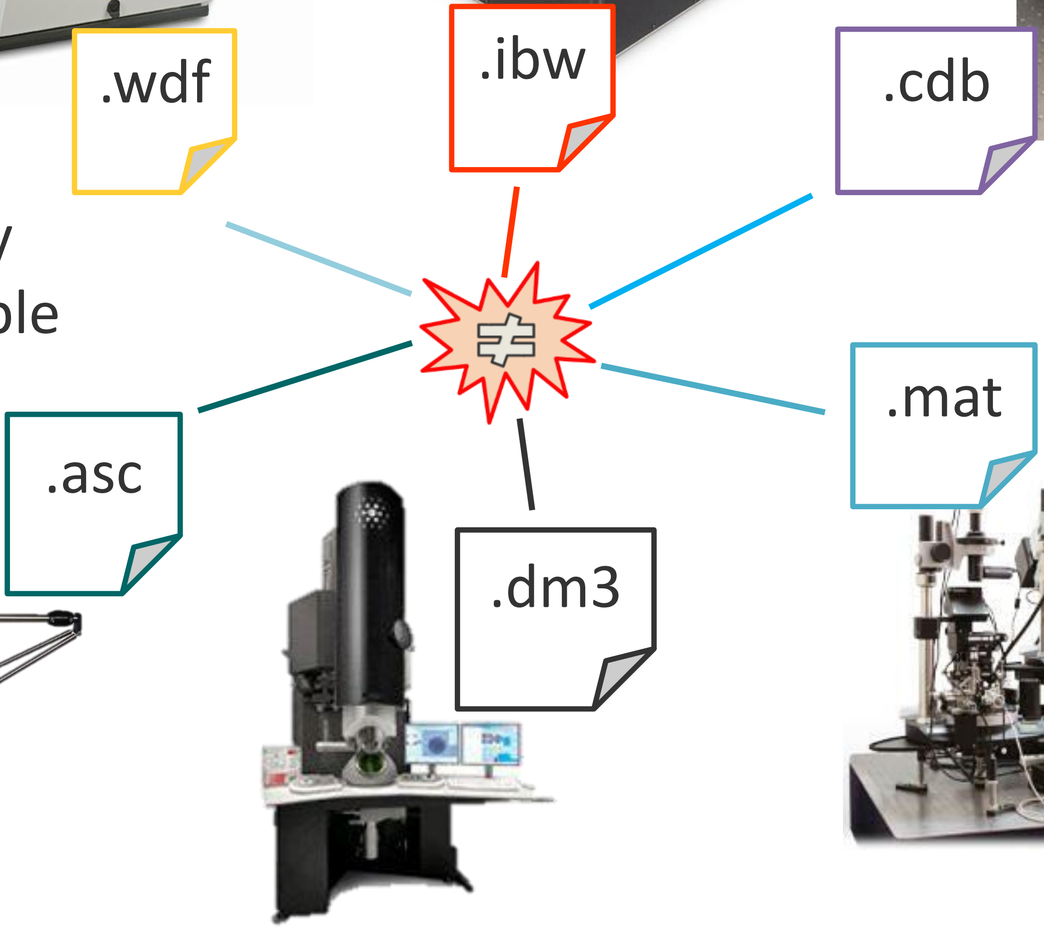
AFM with Raman spectroscopy


Slide by S. Somnath

Multitude of File Formats

Slide by S. Somnath

- Proprietary
- Incompatible



Slide by S. Somnath

Disjointed communities….

- Clustering
- Fit spectra …

- Filter Image
- Register Image …

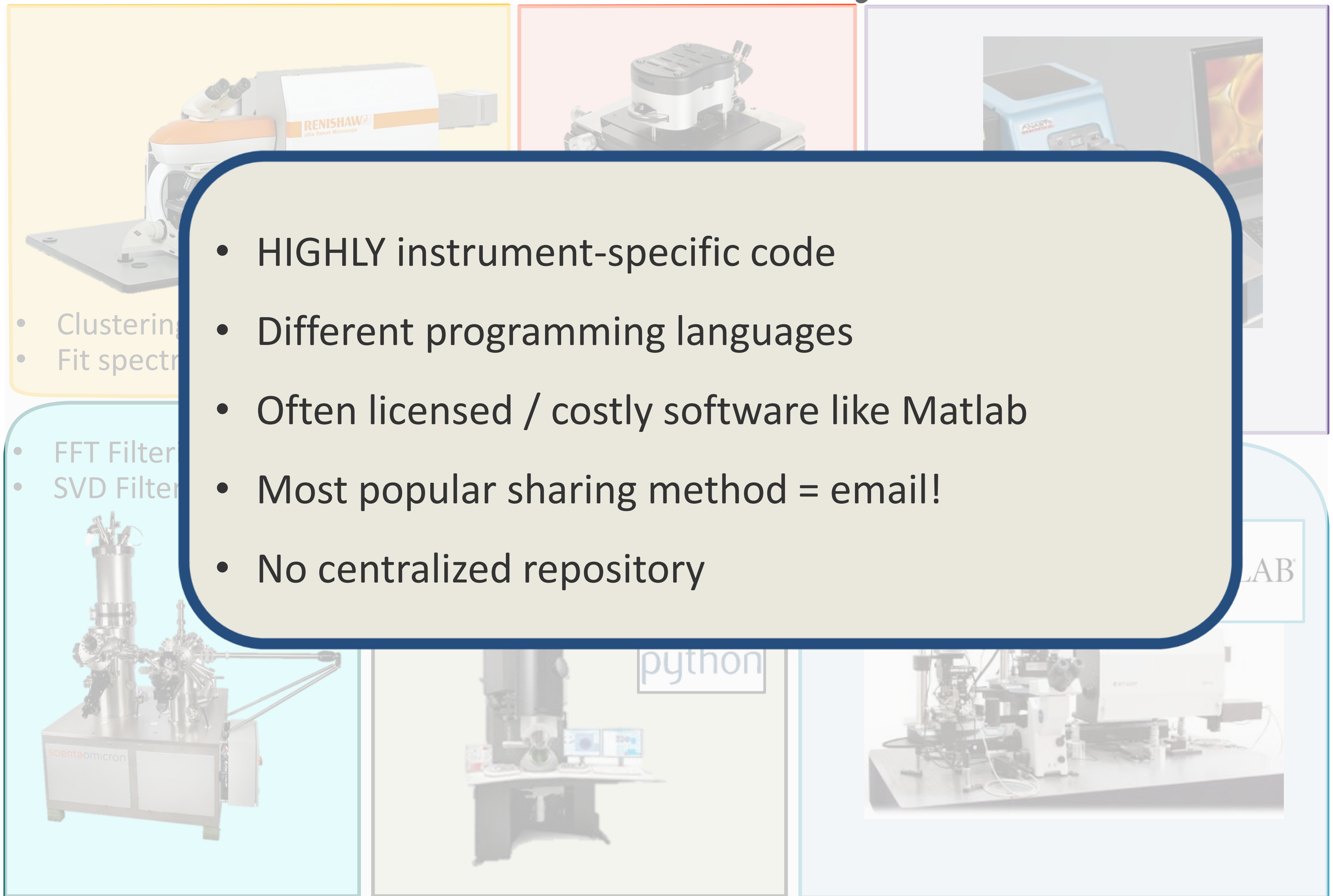
- Fit Spectra
- SVD Filtering …

- FFT Filtering
- SVD Filtering …

- FFT Filtering
- Classify Images …

- Register Images
- Clustering

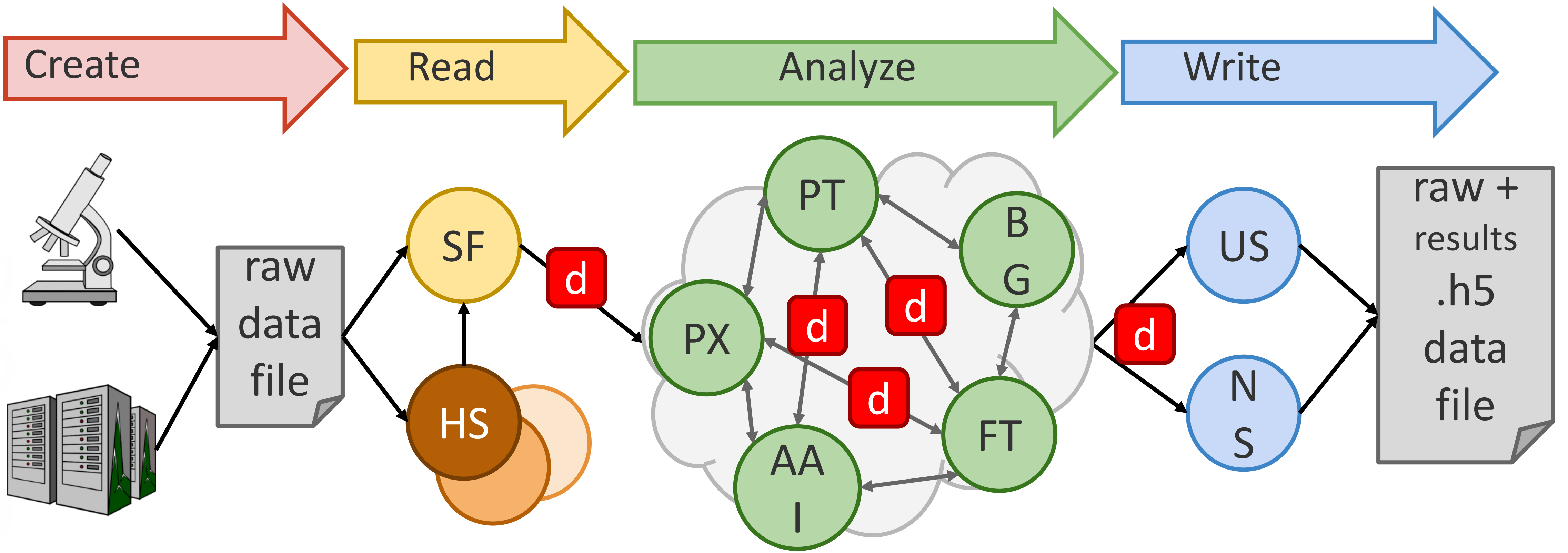
… cannot share code efficiently



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

Slide by S. Somnath

Solutions: Integrated Ecosystems

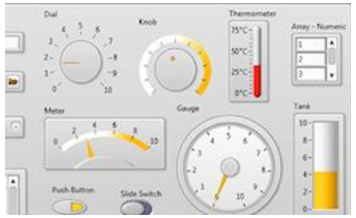


Data from measurements or simulations are read into **`sidpy.Dataset`** (d) objects directly by **`SciFiReaders`** (SF). Data are processed using multiple science packages in the Pycroscopy ecosystem that interoperate via **`Dataset`** objects. **`Dataset`** objects are written to HDF5 files via **`pyUSID`**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

Slide courtesy of R. Vasudevan

Solutions: Integrated Ecosystems

Instrument Tier



Automated, standardized, modularized data acquisition

Instrument-agnostic, self-describing, model in an open friendly file format

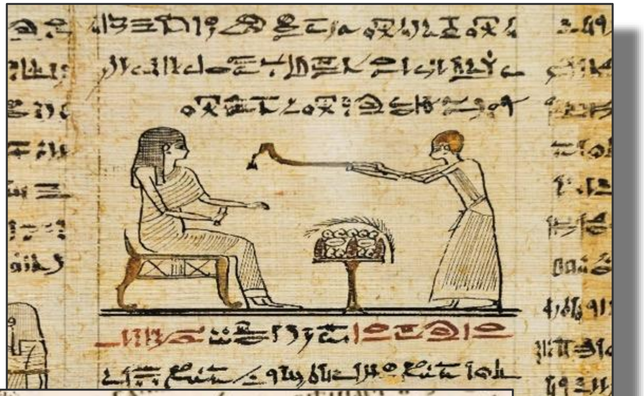
Centralized repository for data processing, analysis

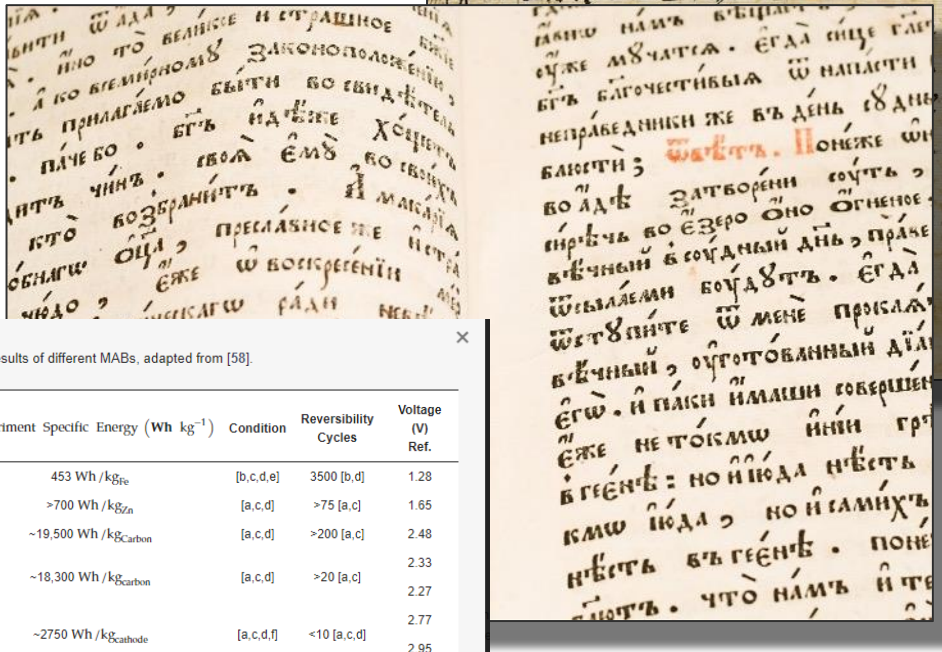
Interactive visualization + analysis + storage on the cloud

Slide courtesy of R. Vasudevan

Get data – from publications

- Printed matter
- Pdf files with figures and tables
- Deposited data files
- (Rarely) workflows





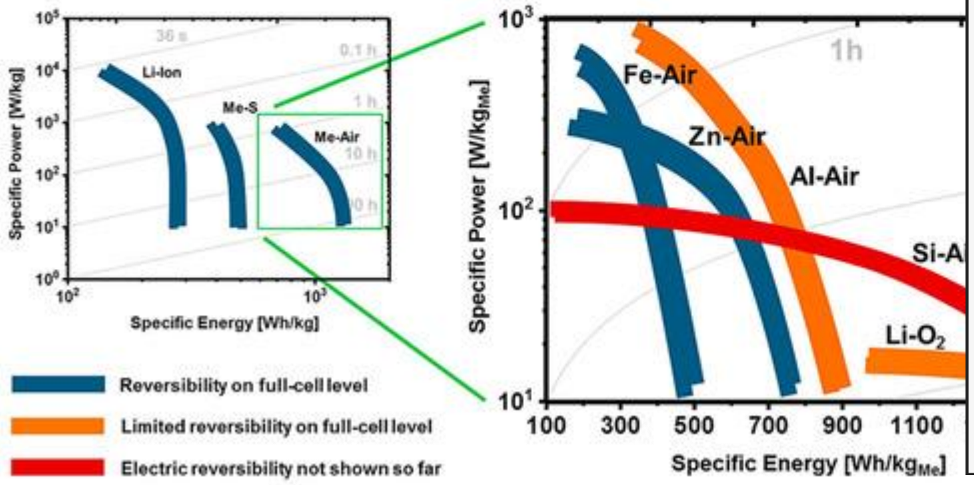
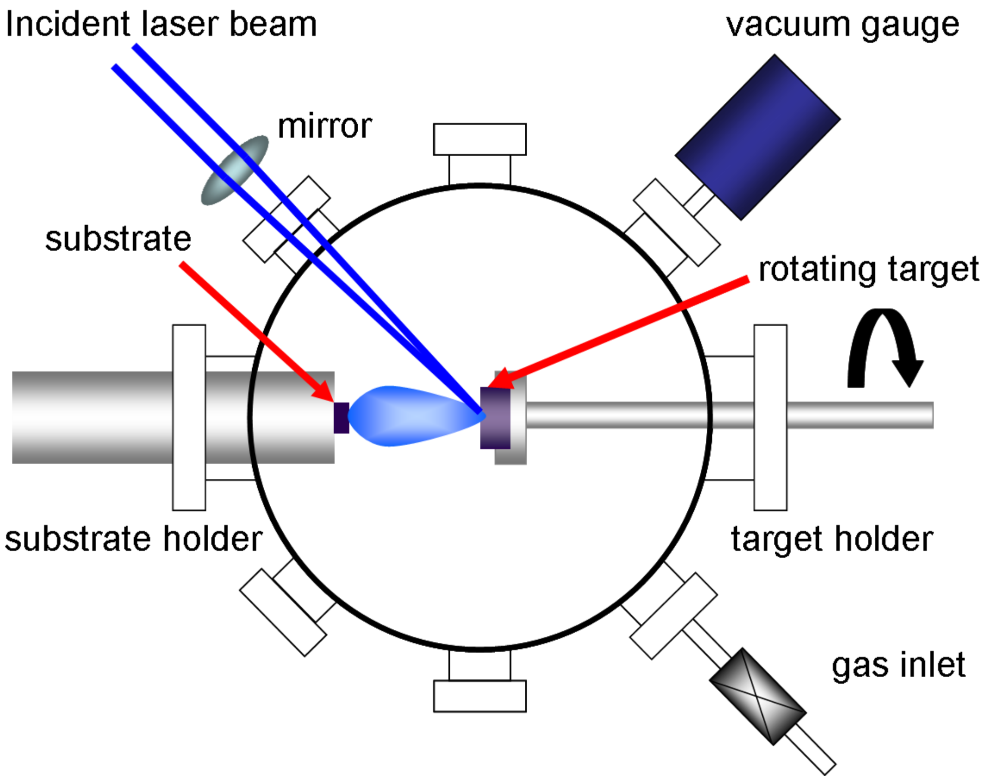


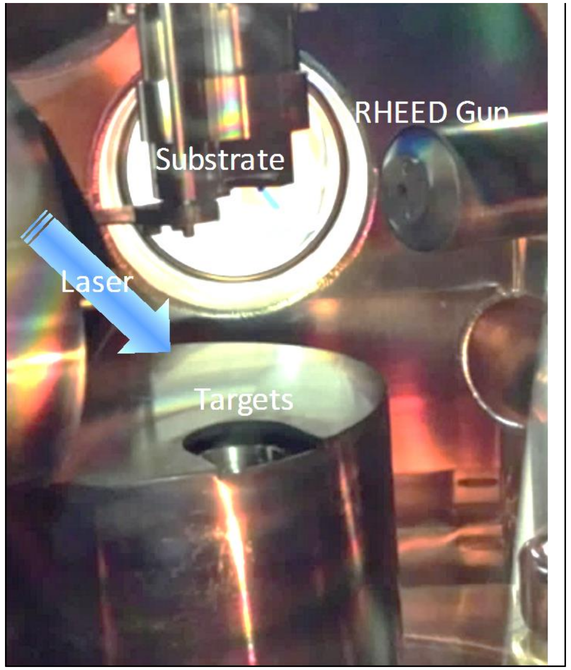
Table 5. The recent experimental results of different MABs, adapted from [58].

MAB	Discharge Product	Experiment Specific Energy (Wh kg^{-1})	Condition	Reversibility Cycles	Voltage (V) Ref.
Fe/O_2	$Fe(OH)_2$	453 Wh /kg$_{Fe}$	[b,c,d,e]	3500 [b,d]	1.28
Zn/O_2	ZnO	>700 Wh /kg$_{Zn}$	[a,c,d]	>75 [a,c]	1.65
K/O_2	KO_2	~19,500 Wh /kg$_{Carbon}$	[a,c,d]	>200 [a,c]	2.48
Na/O_2	Na_2O_2 NaO_2	~18,300 Wh /kg$_{carbon}$	[a,c,d]	>20 [a,c]	2.33 2.27
Mg/O_2	$Mg(OH)_2$ MgO	~2750 Wh /kg$_{cathode}$	[a,c,d,f]	<10 [a,c,d]	2.77 2.95
Si/O_2	$Si(OH)_4$ SiO_2	~1600 Wh /kg$_{Si}$	[a,c,d]	Not yet	2.09 2.21
Al/O_2	$Al(OH)_3$ Al_2O_3	~2300 Wh /kg$_{Al}$	[a,c,d]	Limited	2.71 2.1
Li/O_2	Li_2O_2 Li_2O	>11,050 Wh /kg$_{carbon}$	[a,c,d]	>250 [a,c]	2.96 2.91

Conditions: a is anode sheet/foil, b is porous/particulate anode, c is full-cell measurements, d is 100% deep discharge, e is repeated charge/discharge, and f is elevated temperature.

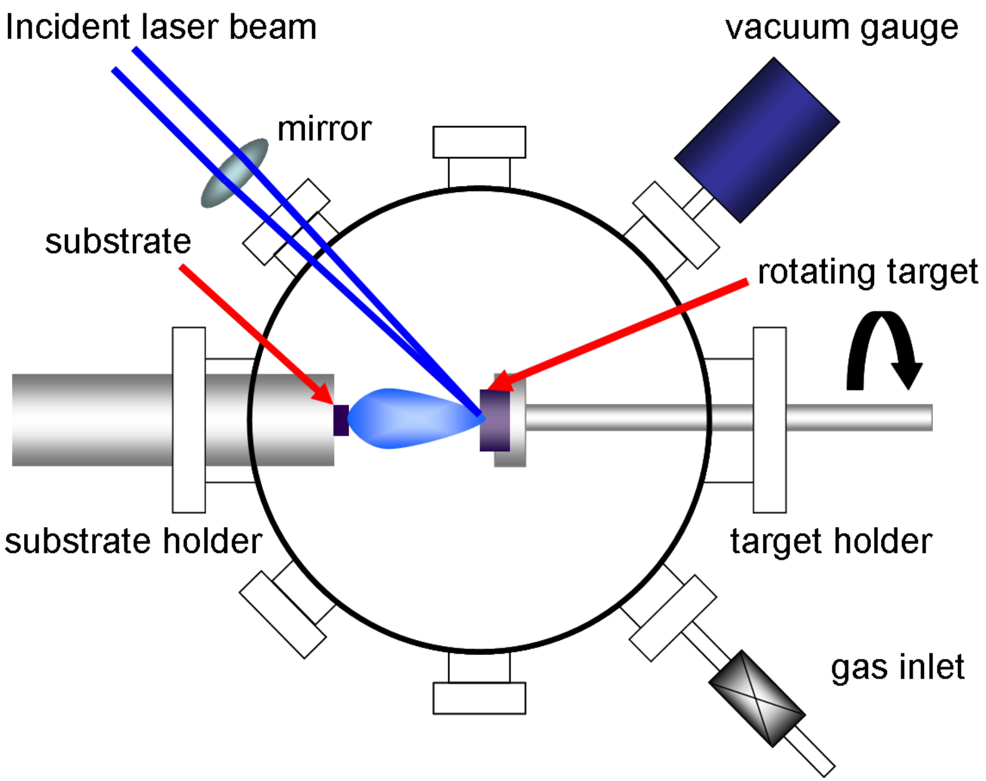
Pulsed Laser Deposition





- PLD benefits: great films, ease of setup, wide variety, unit-cell precision in growth
- Downside: difficult to correlate growth parameter with film properties, largely trial and error approach

Pulsed Laser Deposition



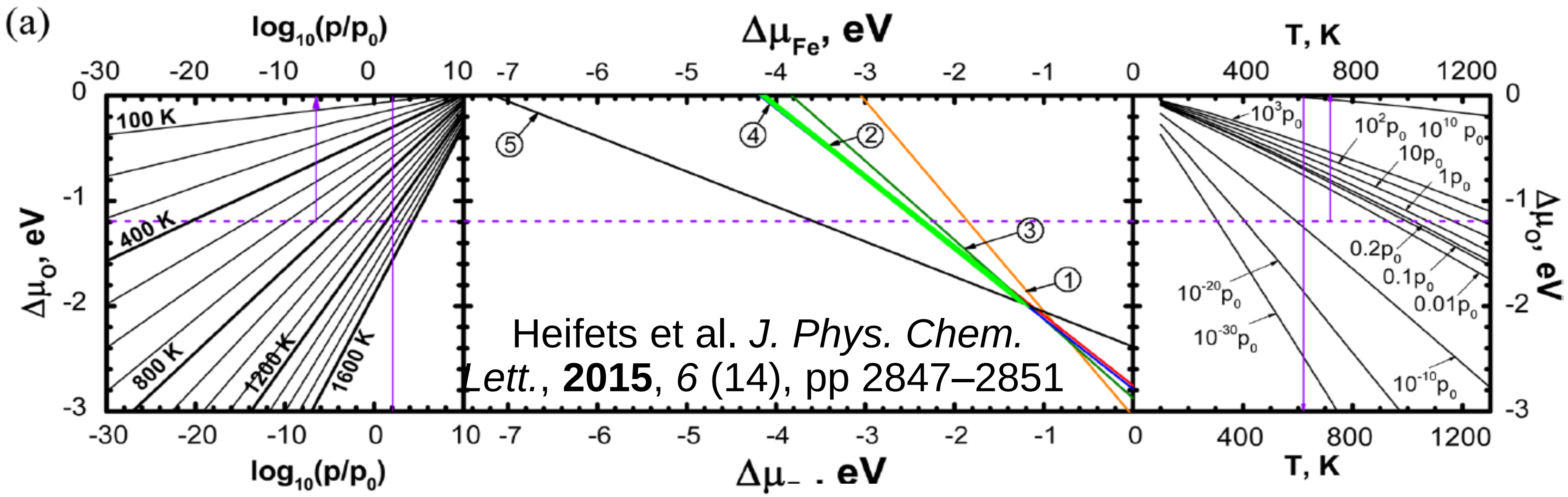
Parameters to vary:

1. Substrate (compound, orientation)
2. Film (compound)
3. Growth temperature
4. Growth environment (e.g., pO_2)
5. Laser Fluence and repetition rate
6. Target-substrate Distance
7. Heterostructures (film electrodes, buffer layers, etc.)
8. Post-annealing
9. Cooling Rate

- Researchers will grow films by varying parameters, achieving different **functional property** results. These could include film roughness, stoichiometry, and specific properties, such as capacitance, remnant polarization, Curie temperature, etc.

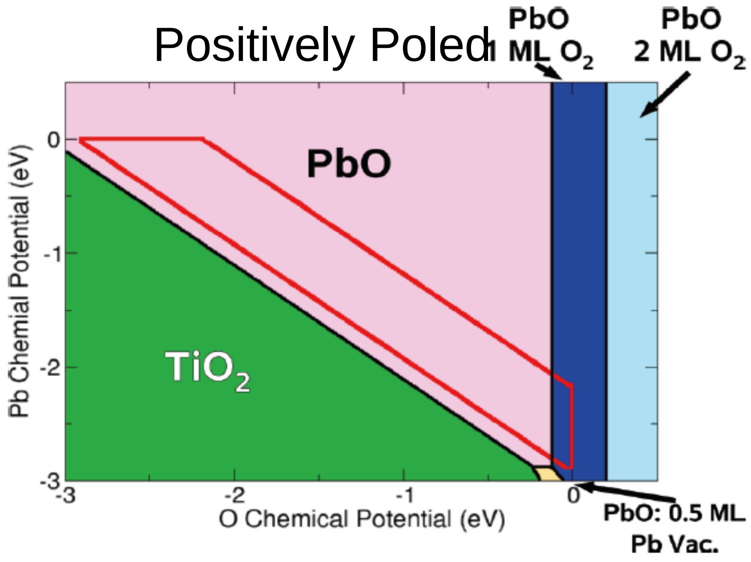
- Iterative procedure, since PLD conditions are generally not transferable.

Can theory help?

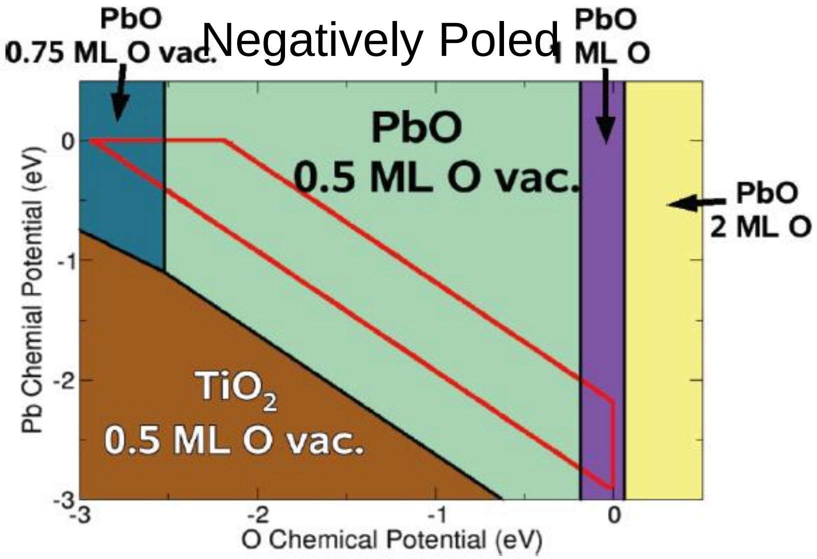


Heifets et al. *J. Phys. Chem. Lett.*, **2015**, *6* (14), pp 2847–2851

Positively Poled



Negatively Poled



Garrity et al. PRB 88, 045401 (2013)

Practical Considerations

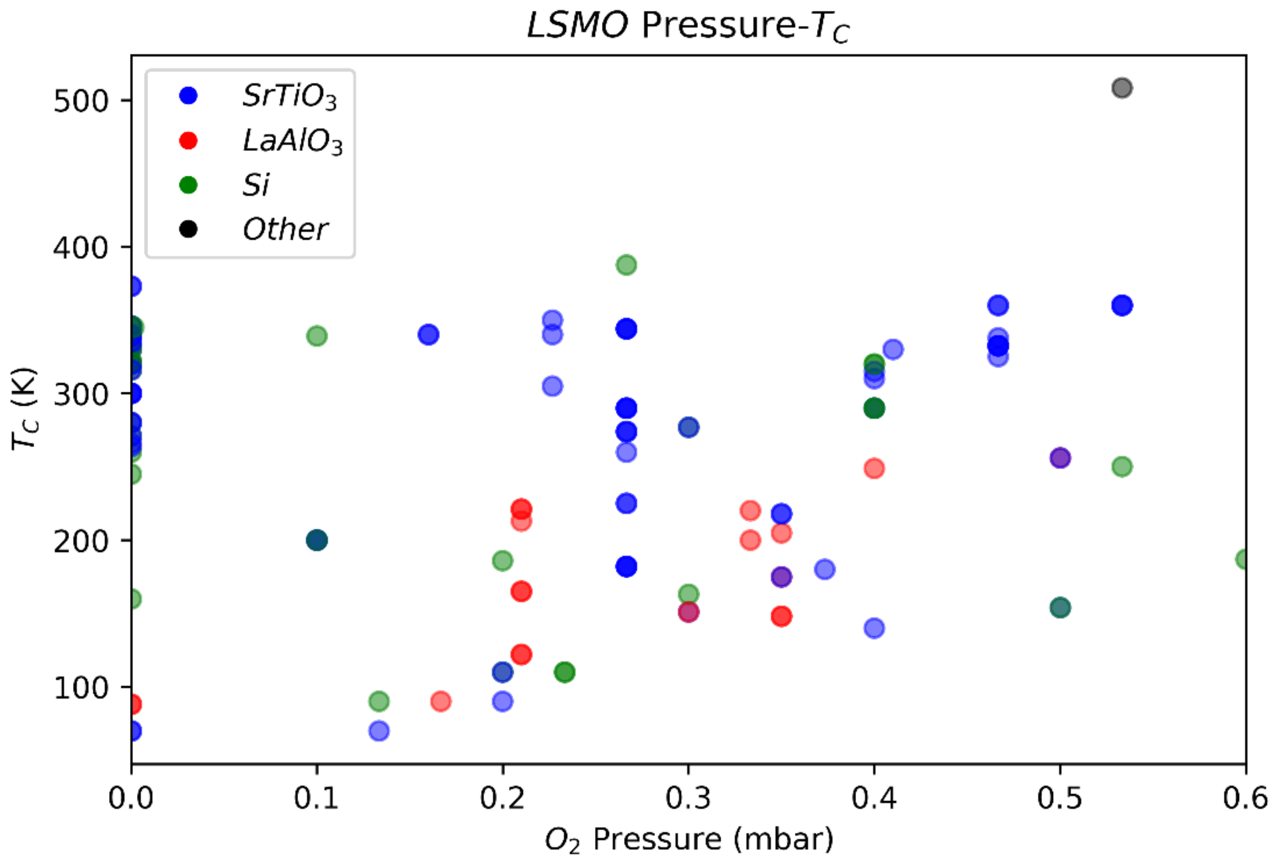
1. DFT is not always accurate, often doesn't account for defects.
2. Kinetic factors during growth.
3. Ideal stoichiometry not always correlated to best properties.

So, we need to know what conditions to use practically…

Slide courtesy of R. Vasudevan

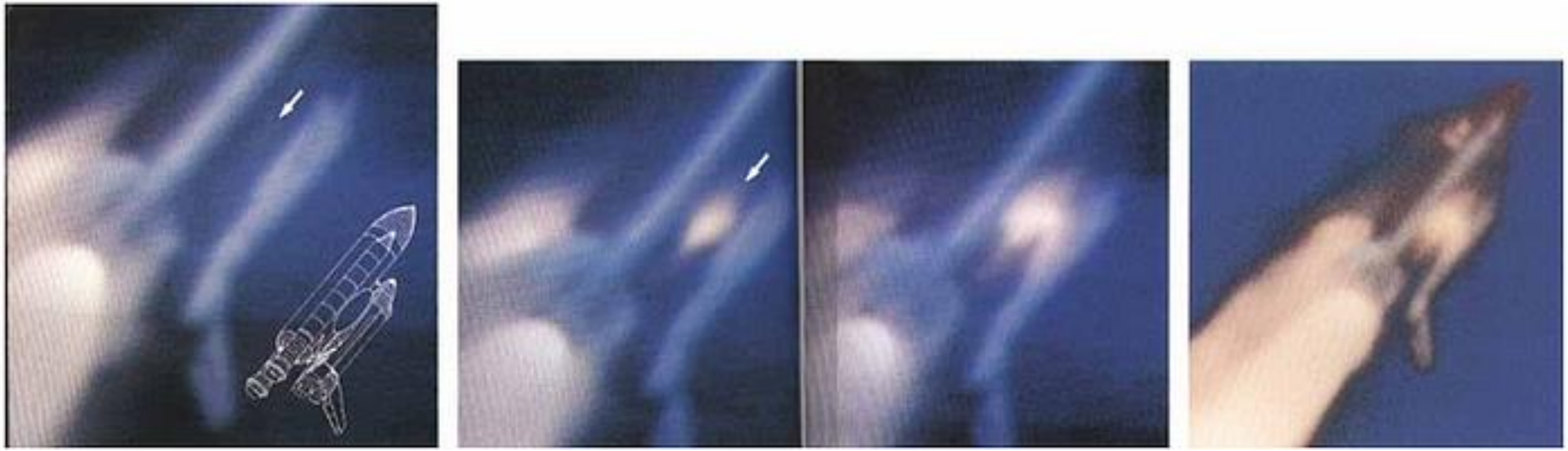
We can mine publications for data. But…

- We get coercive temperature dependent on growth temperature and partial pressure.
- What's next?



Slide courtesy of R. Vasudevan

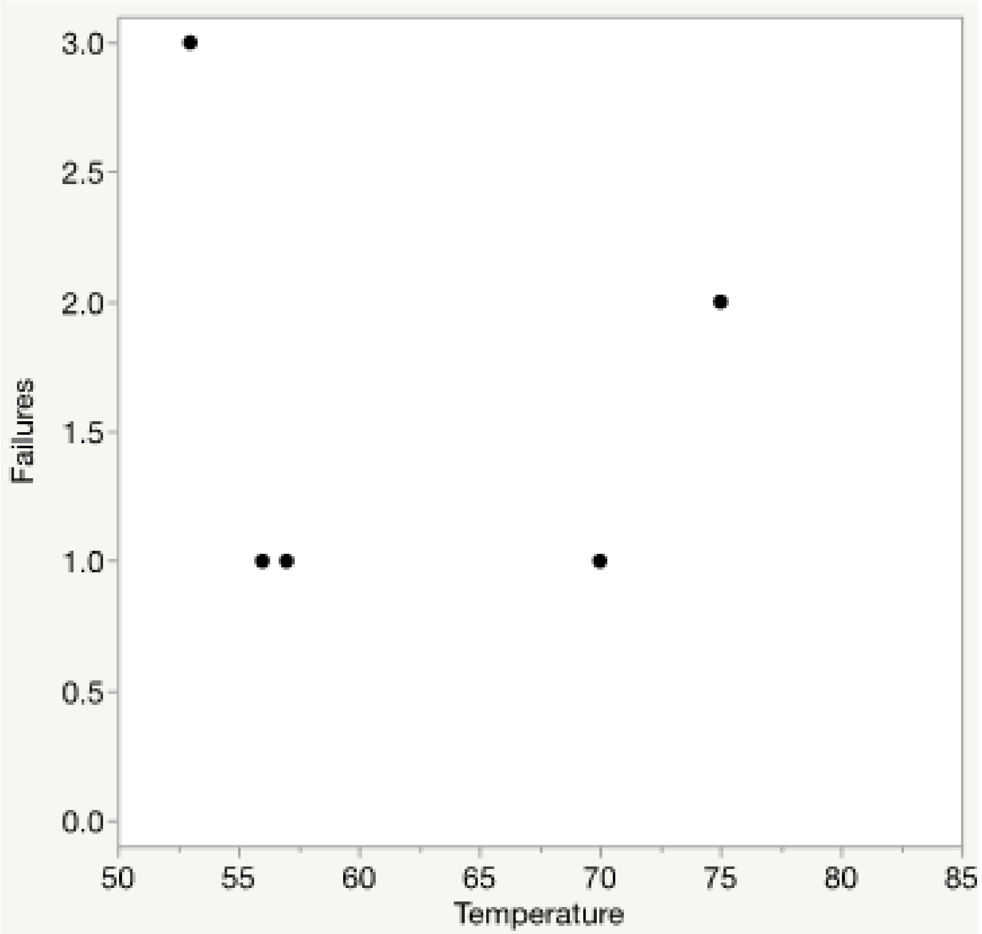
Challenger disaster



https://medium.com/habits-for-success/the-challenger-disaster-a-lesson-in-the-power-of-data-analysis-and-visualization-398d2ac8f59b

https://www.youtube.com/watch?v=hOEt1MOuYX4

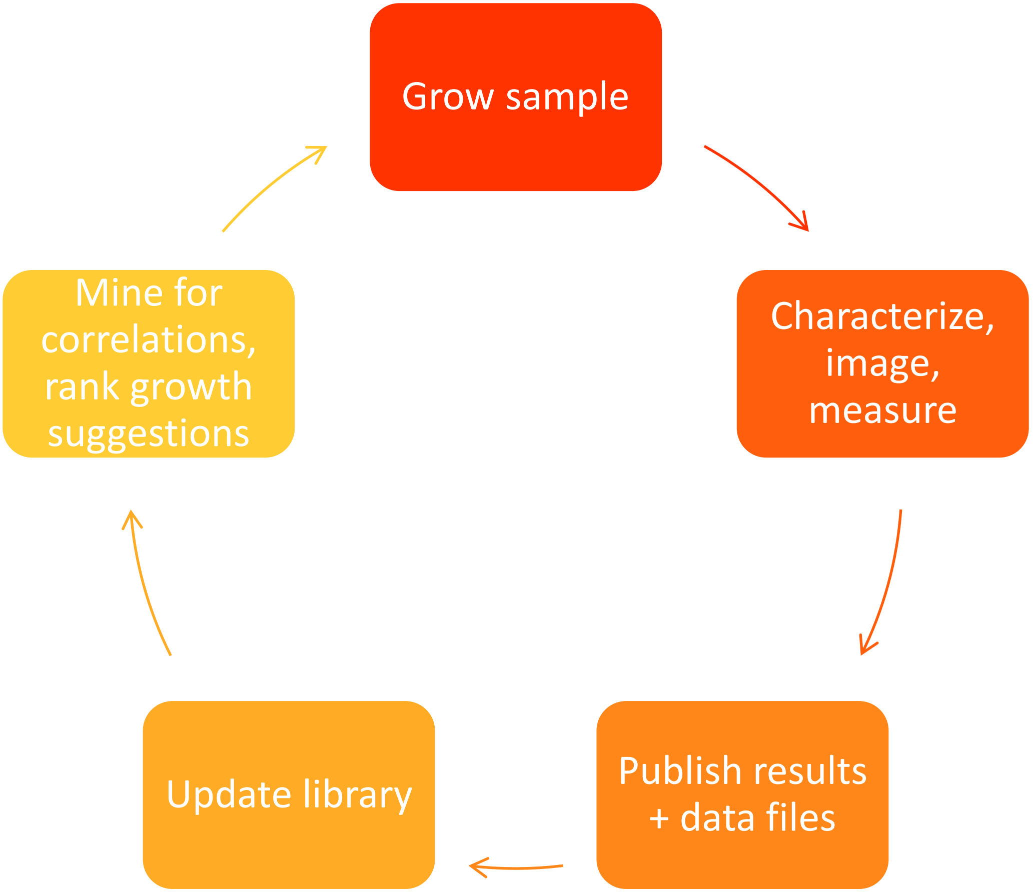
Challenger disaster

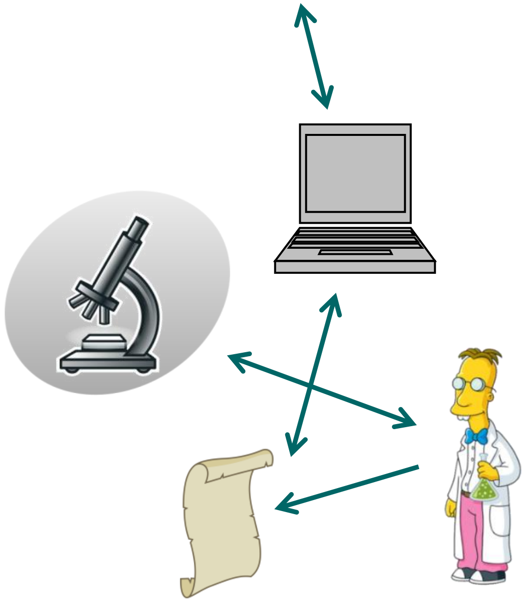


https://www.researchgate.net/figure/Challenger-Data-Number-of-O-Ring-Failures-Launch-Versus-TemperatureLeft-Panel-Five_fig2_344257416

Conclusion: requires community involvement

- Relevant sample preparation conditions
- Unique sample identifier
- Measured properties in appropriate format





Slide courtesy of R. Vasudevan

Get data – data bases!

- **On-line data bases**
  - Materials Project, https://next-gen.materialsproject.org/
  - Materials Data Facility: https://materialsdatafacility.org/
  - Papers with code: https://paperswithcode.com/datasets
  - Standard data sets for molecular properties, e.g. http://quantum-machine.org/datasets/
  - Integrators, e.g.: https://github.com/sedaoturak/data-resources-for-materials-science

- **Enterprise data bases**

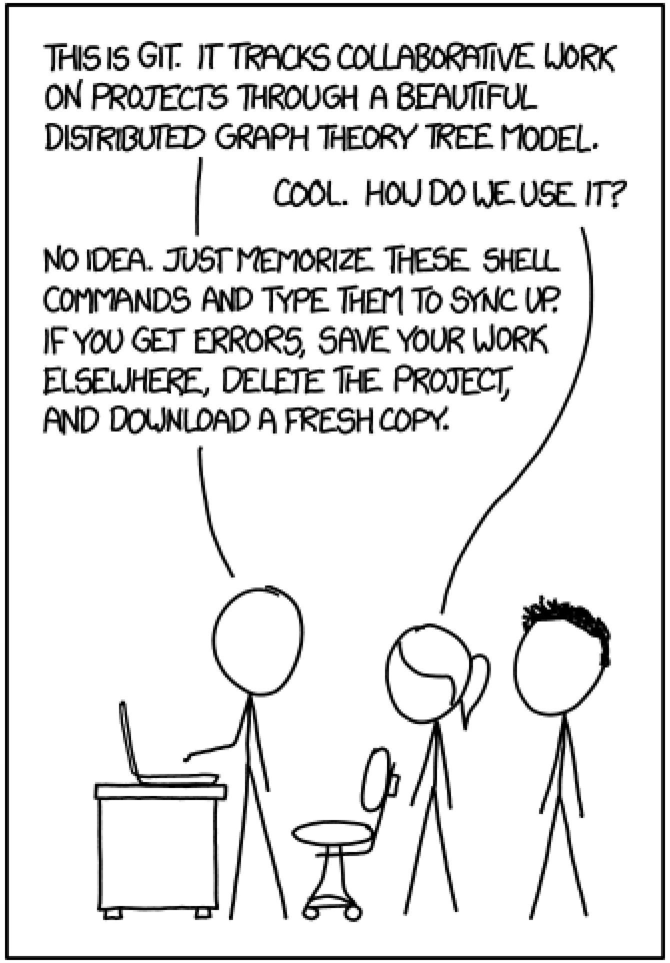
How we can run code:

- Google Colabs
- AWS SageMaker notebooks
- IDE: Spyder, PyCharm, etc.
- Command line interface

Code Repositories and Version Control

- Sharing scripts between users can be workable for immediate or short-term needs, but is not scalable nor lasting
- For reproducibility, it is better to have codes that reside in packages that are documented and well tested
- Most of you are familiar with python packages; but many are probably new to version control
- Version control systems such as git enable multiple people to work on a single software project at the same time to speed up development and ensure consistency
- Git is an open-source distributed version control system. It maintains a history of changes that have occurred in the project and allows for updates as well as reversions to older 'commits'.

How we can share code:



Git would take a significant amount of time to explain in detail. However, there are plenty of online tutorials, e.g.

https://www.atlassian.com/git/tutorials

We will test some of the basic git functionality as demo.

1. Make sure you have a GitHub account

2. Example: https://github.com/ramav87/ML_Summer_Course

3. Head on over to https://github.com/SergeiVKalinin/MSE_2023_Git

We will practice forking a repository, making a change to the code and submitting a pull request

What language do we use:

Main Python libraries we will use:

1. NumPy
2. MatPlotlib
3. Scikit-learn
4. Keras

Other libraries we may use:

1. Seaborn
2. GPax
3. SciPy

We will learn these as we need them!