Machine Learning in Material Science: Science, Machine Learning, and Prediction

Bernardus Rendy[1,a)]

[1]*Engineering Physics Department, Faculty of Industrial Technology, Institut Teknologi Bandung, Indonesia*

[a)]bernardusrendy@students.itb.ac.id

Special Thanks to:

***It is strictly prohibited to share or copy this presentation slide as it contains sensitive informations**

Purpose and Outline:

To ***Introduce*** Machine Learning in Material Science and Engineering World

(Very Basic) Introduction to Material Science and Engineering

~~Linear Algebra and Statistics~~ (Skipped hehehe)

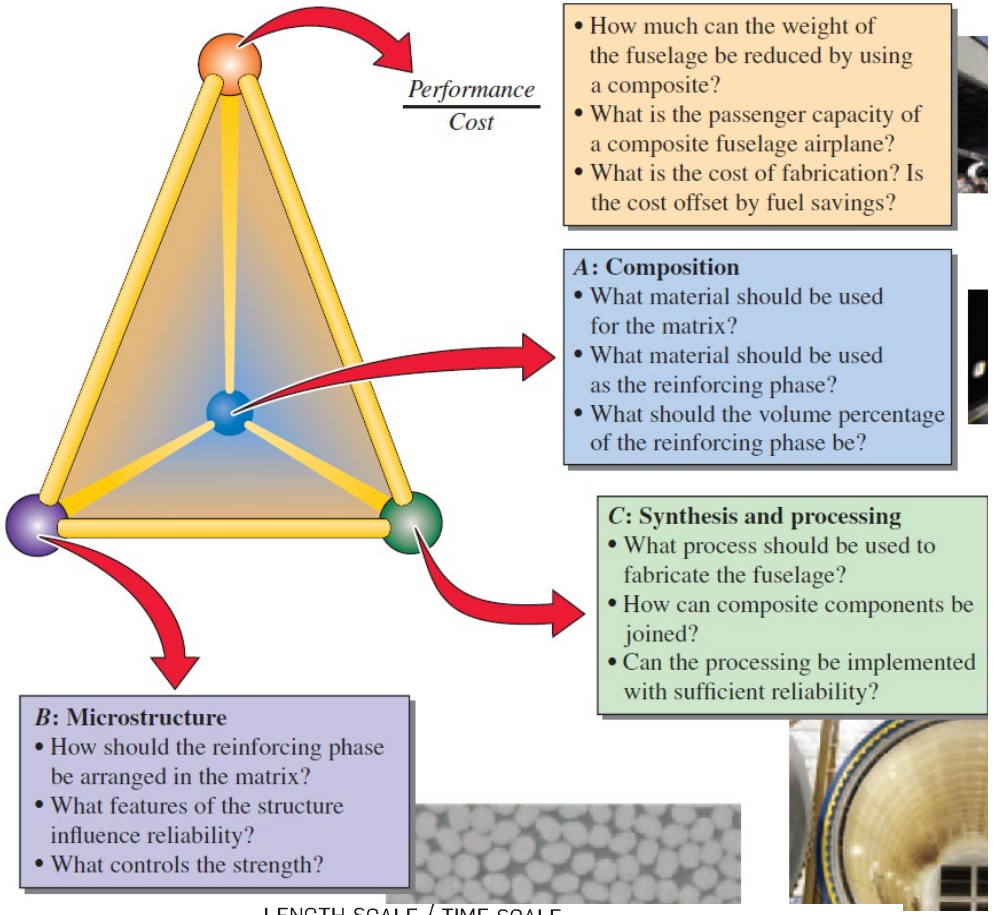
Machine Learning Basics: Data Pre-Processing and Transformation

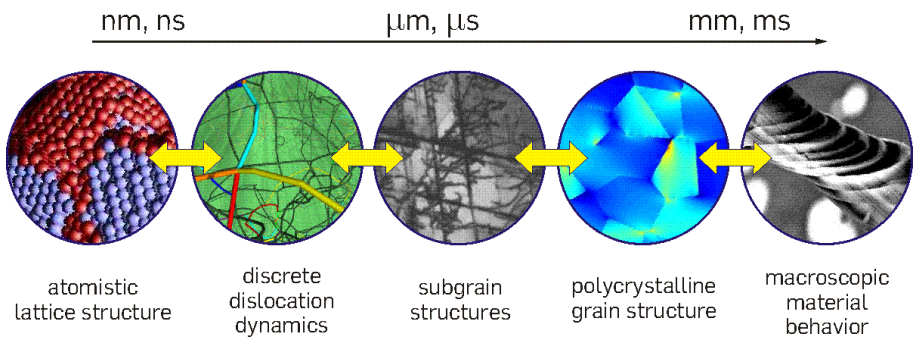
Machine Learning in Experiment and Characterization

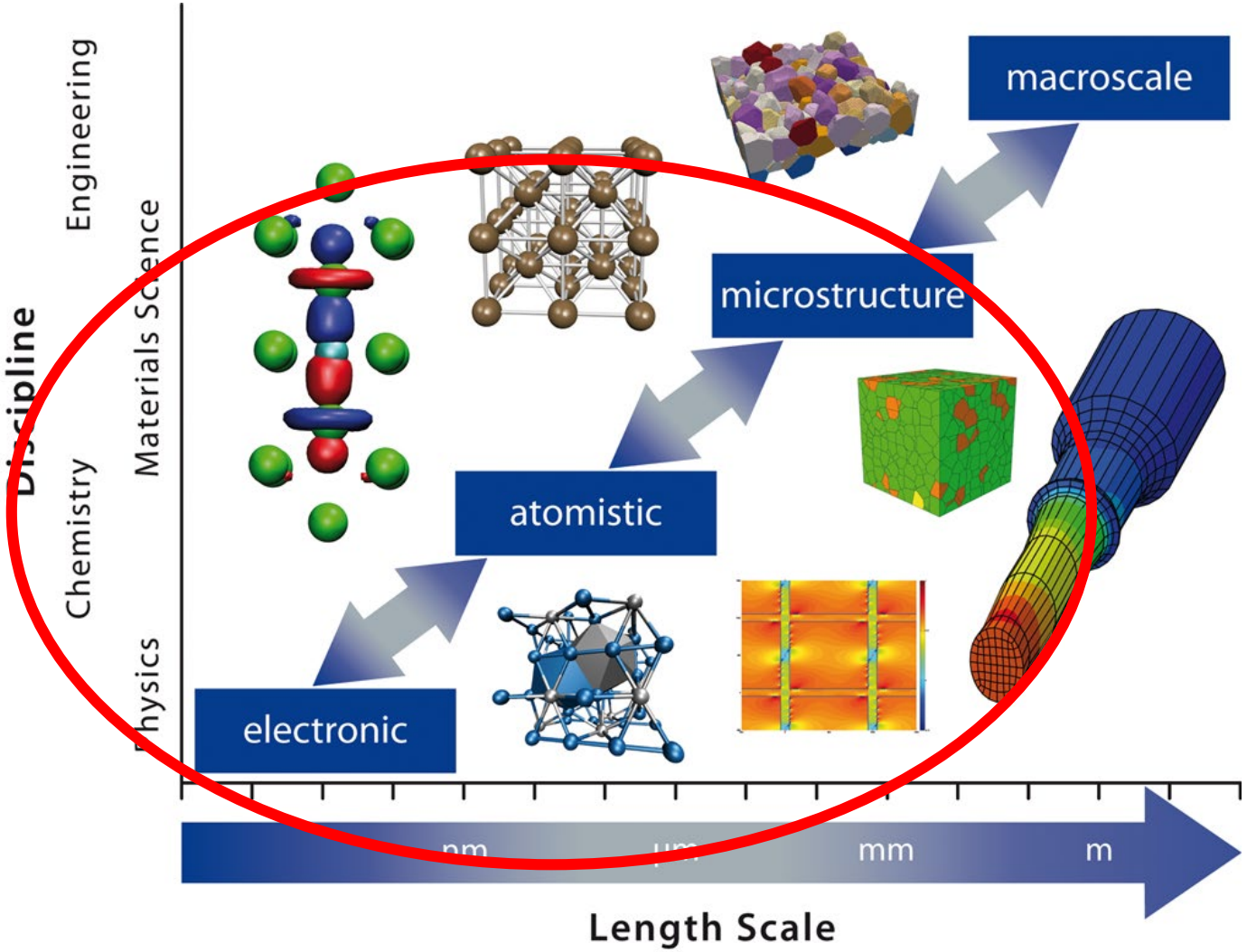
~~Machine Learning in Superconductivity (Next Time)~~

~~Material Informatics and Machine Learning (Next Time)~~

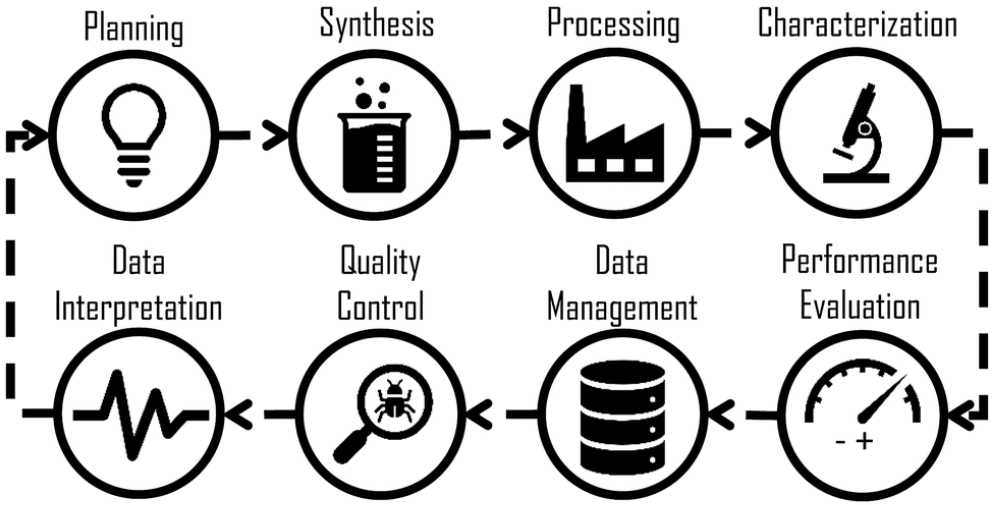
What is a Material?



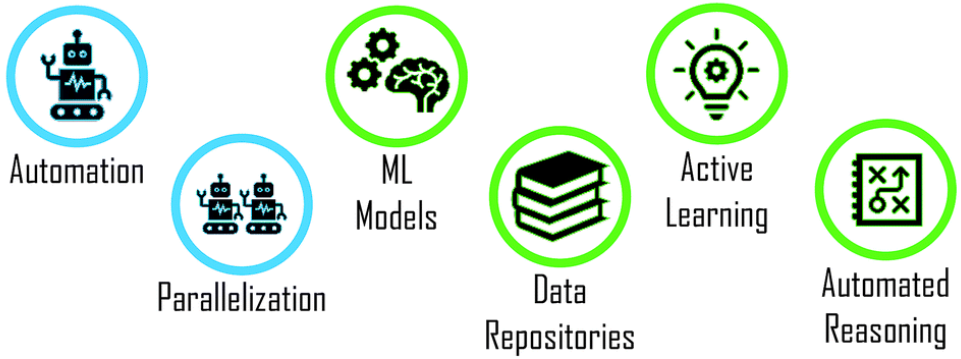




The Material Science Workflow: Experiment



b) Accelerators

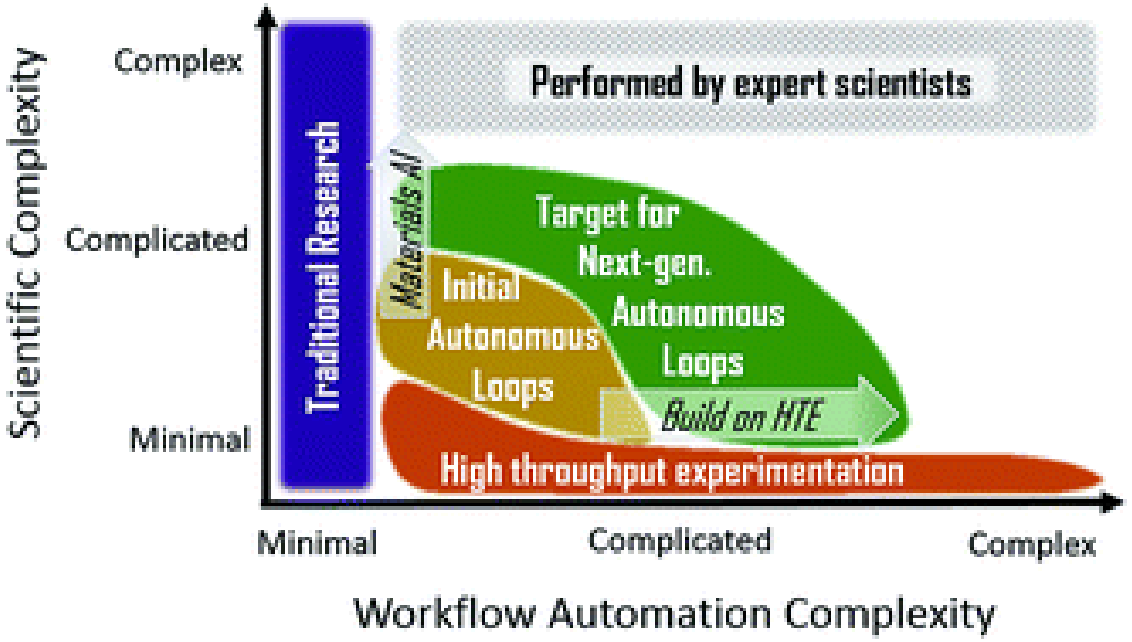


[Submitted on 11 Jun 2020]

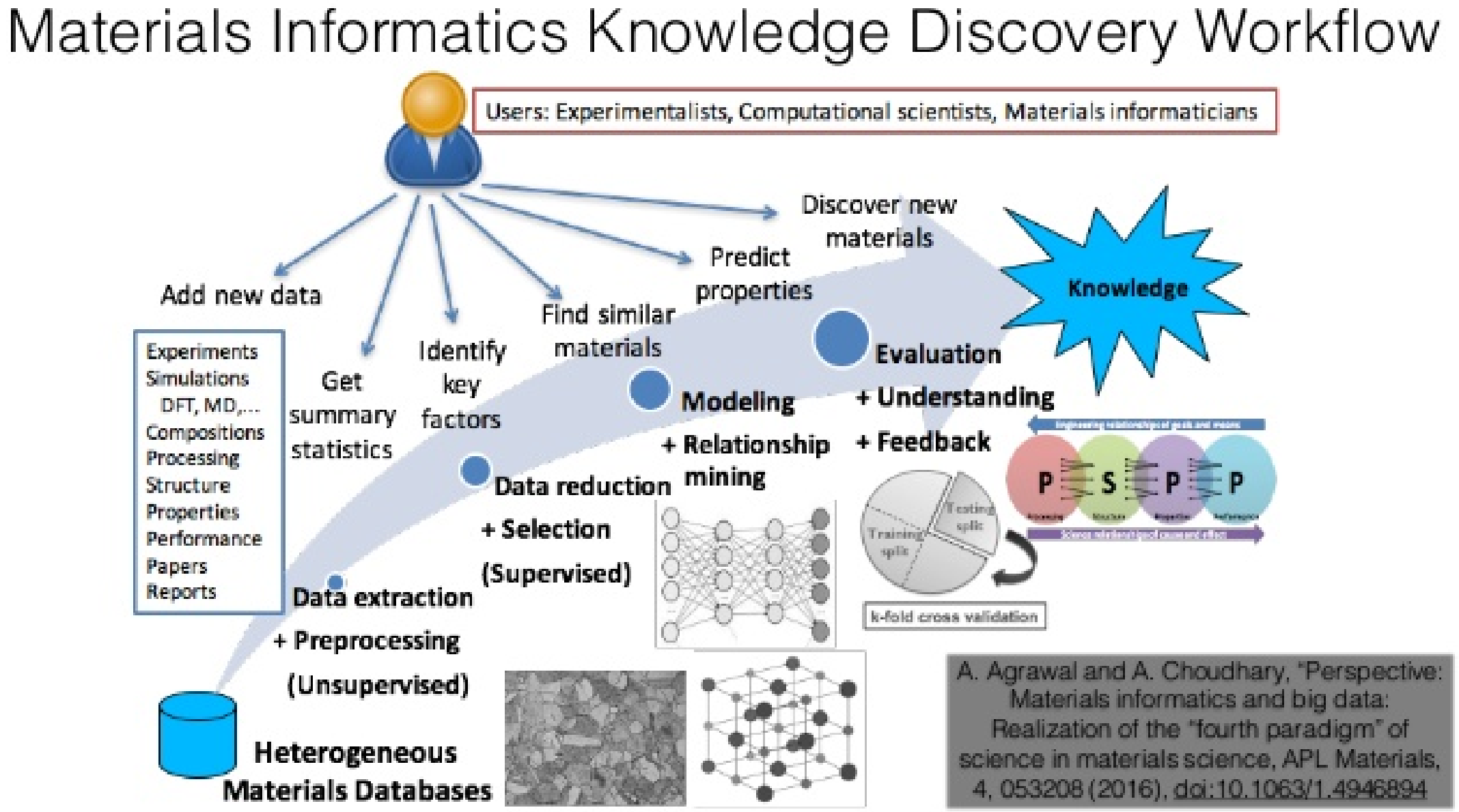
On-the-fly Closed-loop Autonomous Materials Discovery via Bayesian Active Learning

A. Gilad Kusne, Heshan Yu, Changming Wu, Huairuo Zhang, Jason Hattrick-Simpers, Brian DeCost, Suchismita Sarker, Corey Oses, Cormac Toher, Stefano Curtarolo, Albert V. Davydov, Ritesh Agarwal, Leonid A. Bendersky, Mo Li, Apurva Mehta, Ichiro Takeuchi

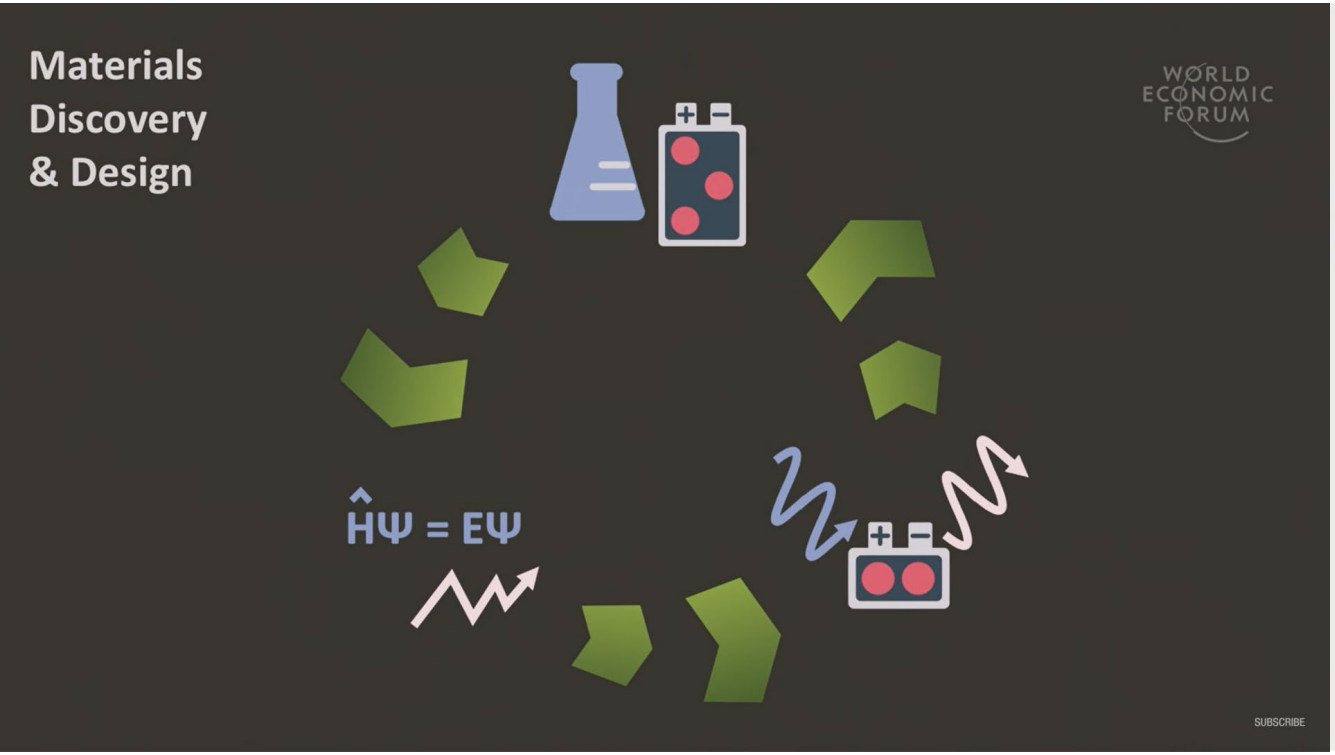
Active learning - the field of machine learning (ML) dedicated to optimal experiment design, has played a part in science as far back as the 18th century when Laplace used it to guide his discovery of celestial mechanics [1]. In this work we focus a closed-loop, active learning-driven autonomous system on another major challenge, the discovery of advanced materials against the exceedingly complex synthesis-processes-structure-property landscape. We demonstrate autonomous research methodology (i.e. autonomous hypothesis definition and evaluation) that can place complex, advanced materials in reach, allowing scientists to fail smarter, learn faster, and spend less resources in their studies, while simultaneously improving trust in scientific results and machine learning tools. Additionally, this robot science enables science-over-the-network, reducing the economic impact of scientists being physically separated from their labs. We used the real-time closed-loop, autonomous system for materials exploration and optimization (CAMEO) at the synchrotron beamline to accelerate the fundamentally interconnected tasks of rapid phase mapping and property optimization, with each cycle taking seconds to minutes, resulting in the discovery of a novel epitaxial nanocomposite phase-change memory material.

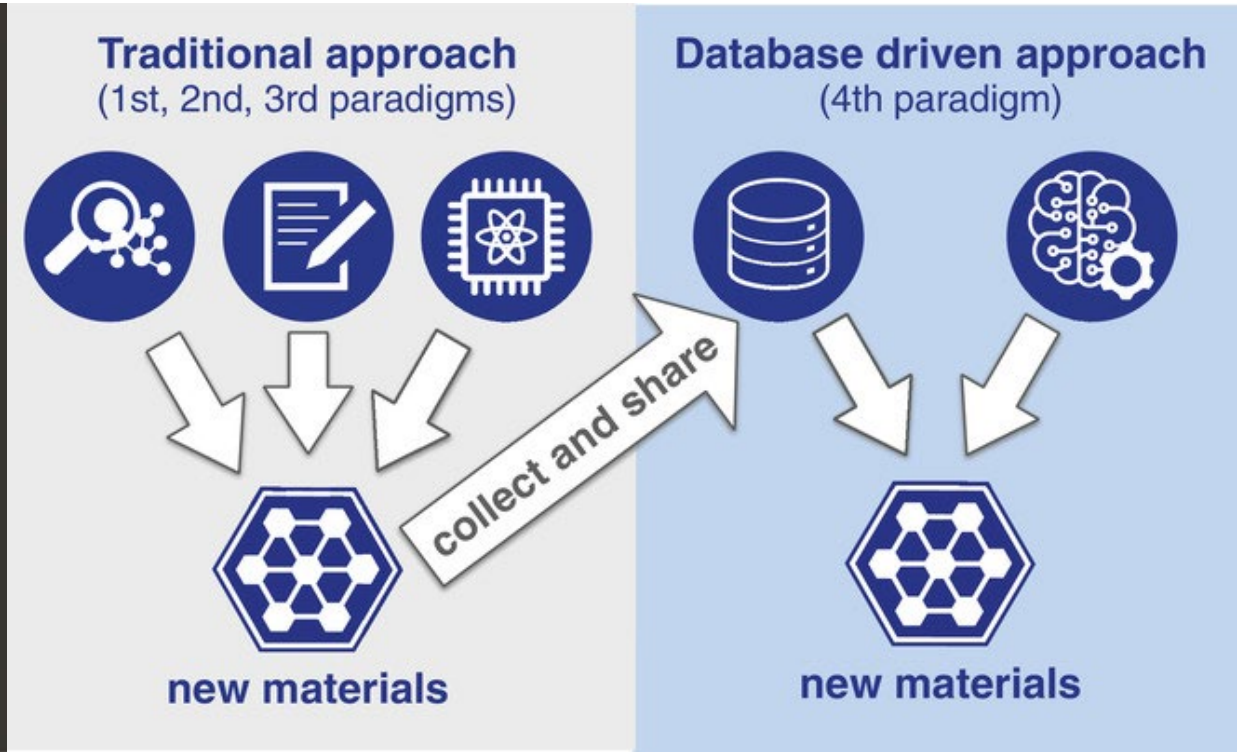


The Material Science Workflow: Material Informatics

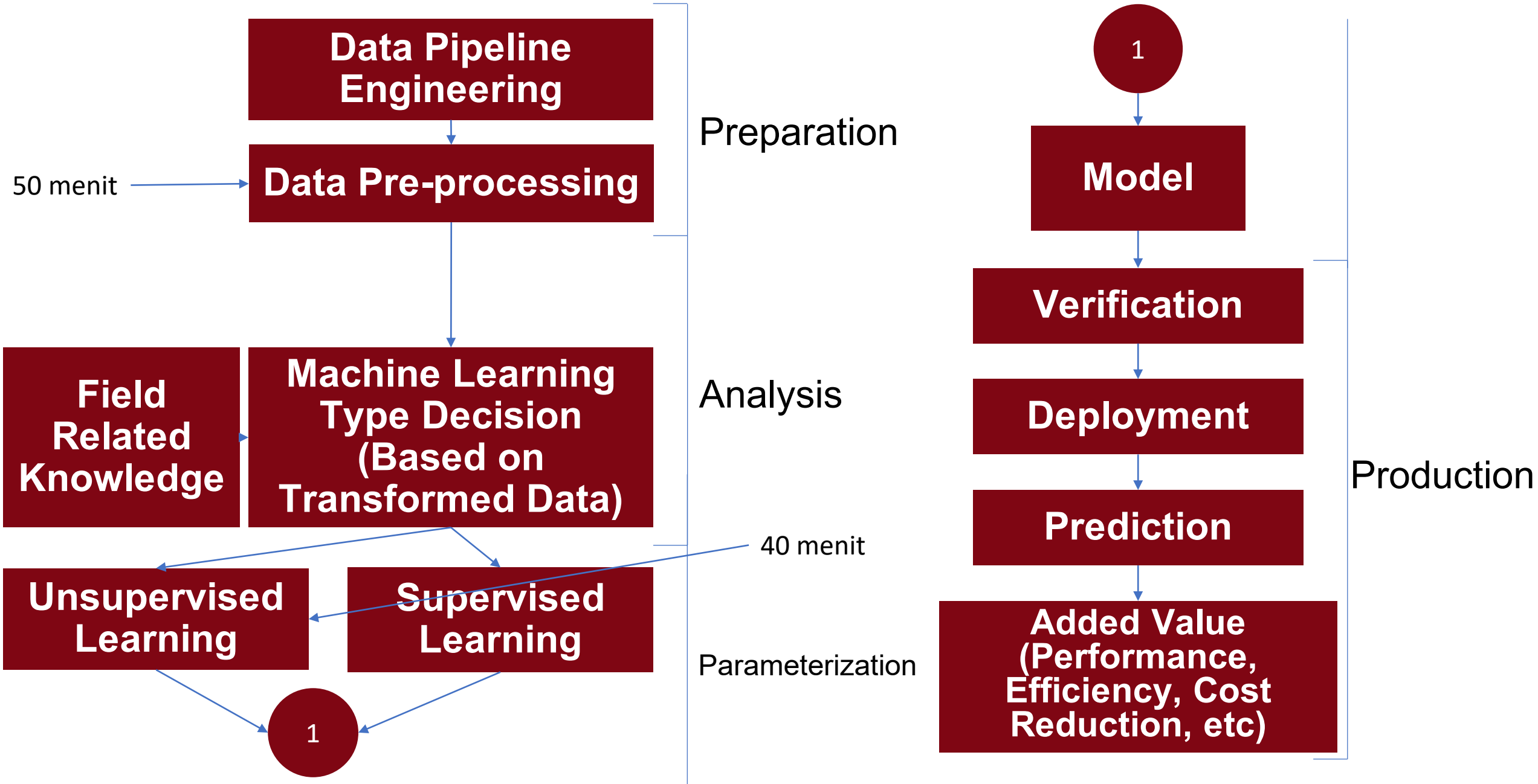


The Material Science Workflow: Both Worlds





Typical Machine Learning Workflow



Data Pre-processing

Isn't my data already perfect?

Real world data are generally:

Incomplete: missing values, lacking attributes of interest, or containing only aggregate data

Noisy: containing errors or outliers

Inconsistent: containing discrepancies in codes or names

Tasks in data preprocessing:

Data cleaning: fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies

Data integration: using multiple databases, data cubes, or files

Data transformation: normalization and aggregation

Data reduction: reducing the volume but producing the same or similar analytical results

Data discretization: part of data reduction, replacing numerical attributes with nominal ones.

OAK RIDGE National Laboratory

Outline

- **Data Cleaning**
  - Background Subtraction
  - Noise
  - Interpolation/Smoothing
  - Inconsistencies – the importance of domain knowledge

- **Data Transformation**
  - Normalization/Whitening
  - Feature Detection
  - Attribute creation: Feature Extraction

Data cleaning – Background Subtraction

A robust background subtraction algorithm should be able to handle changes, repetition, and long-term changes.

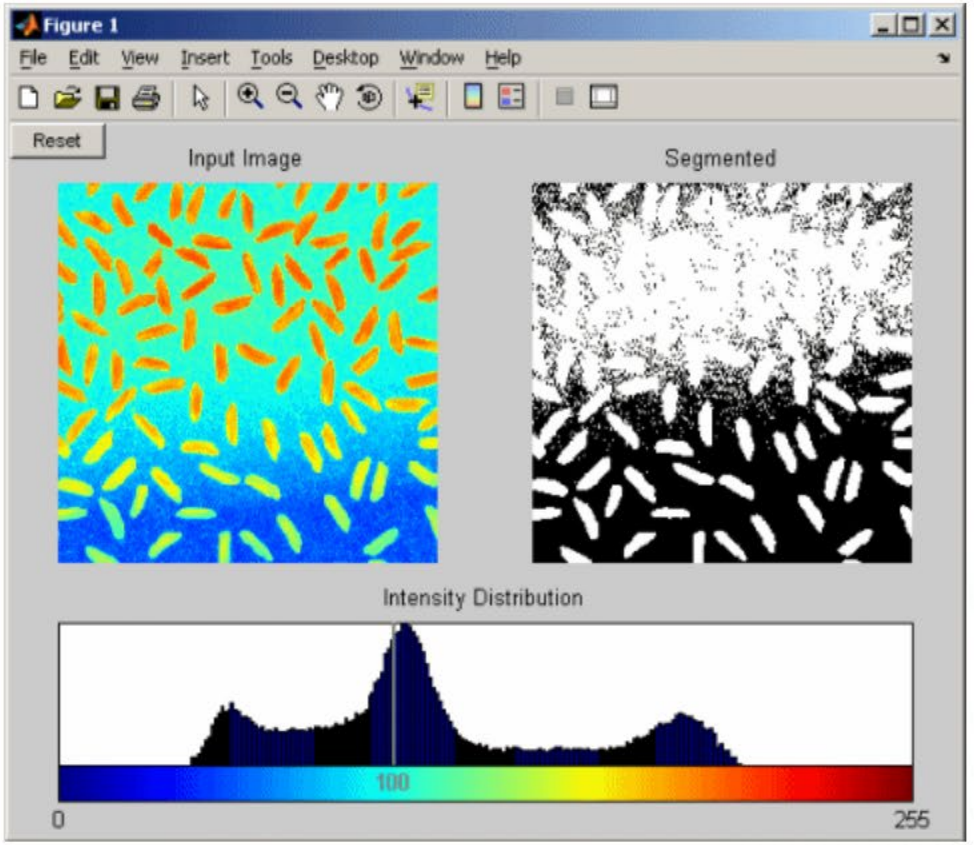
- **Thresholding and Frame Differencing (easy)**
  - *Thresholding is the simplest method of image segmentation that can be used to create binary images*
- **Mean/Median Filters (easy)**
  - *Create a filter based on averaging of preceding frames, or local neighbors*
- **Running Gaussian average (medium)**
  - *A filter used to classify pixels based on intensity confidence interval of its distribution's mean*
- **Background Mixture models (hard)**
  - *An approaches that models each pixel as a mixture of Gaussians and uses an on-line approximation to update the model*

OAK RIDGE National Laboratory

Thresholding for Background Extraction

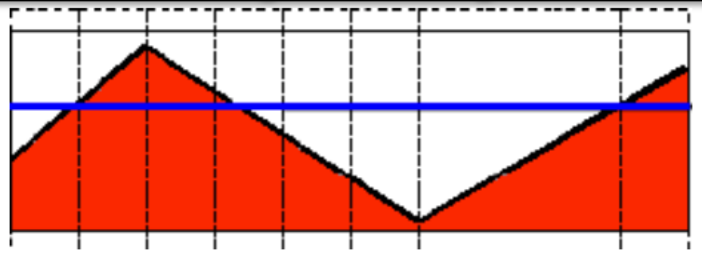
In an a given image, if the image intensity ***I*** is less than some fixed constant ***T*** , this pixel is replaced with a black pixel (0), or a white pixel (1)

$$I_{i,j} < T$$

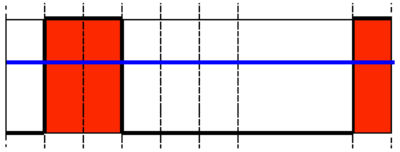


OAK RIDGE National Laboratory

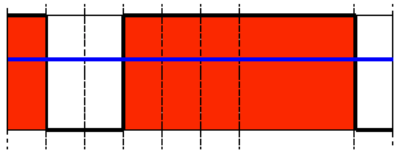
Global Thresholding for Background Extraction



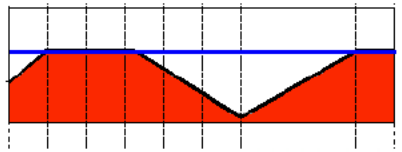
- Threshold Binary



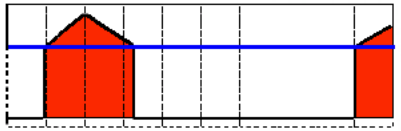
- Threshold Binary, Inverted



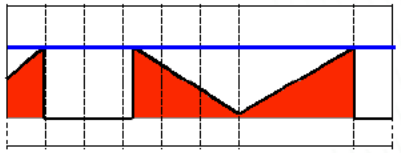
- Truncate



- Threshold to Zero



- Threshold to Zero, Inverted



OAK RIDGE National Laboratory

Thresholding - Otsu's method

Otsu's method, is used to automatically perform clustering-based image thresholding.

The algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their combined spread (intra-class variance) is minimal.

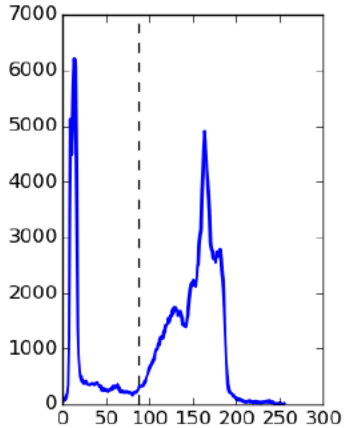
$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

1. Compute histogram and probabilities of each intensity level

2. Set up initial *class probabilities* and *class means*

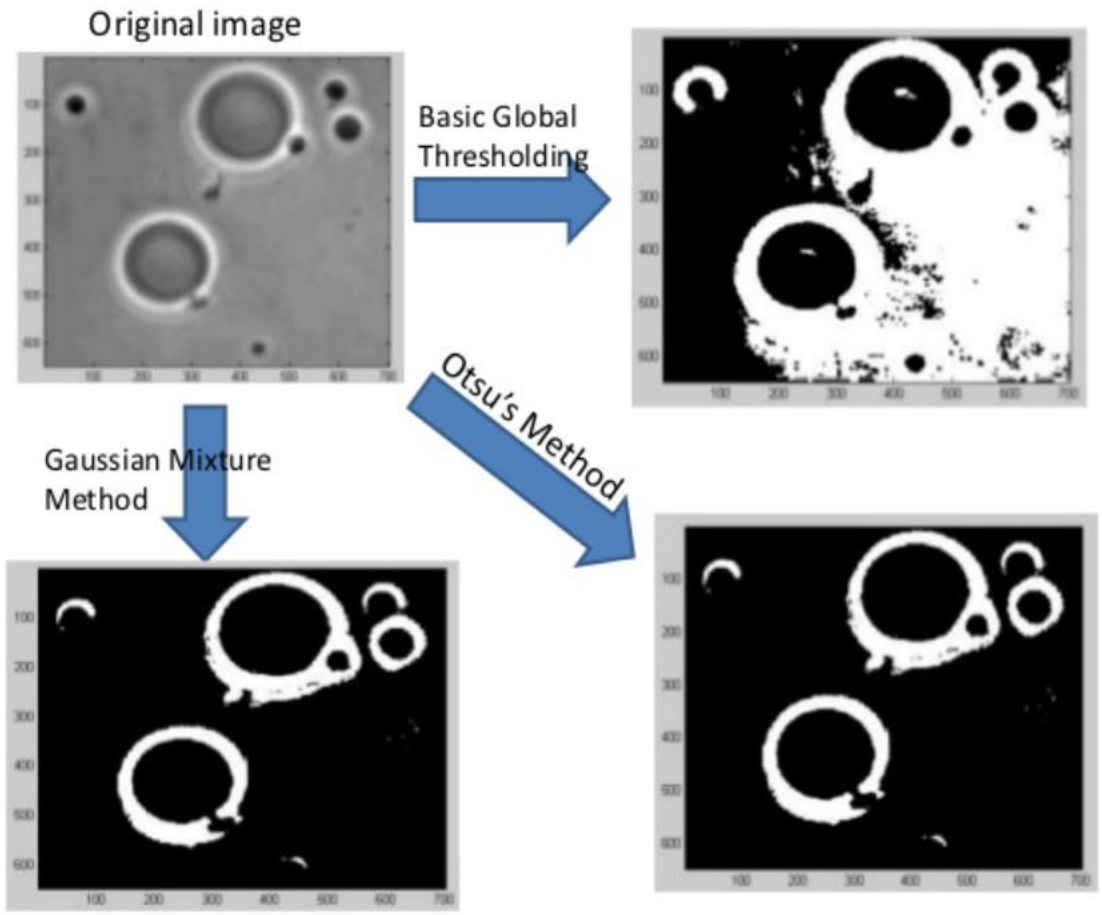
3. Step through all possible thresholds
   a. Update *class probabilities* and *class means*
   b. Compute *variance*

4. Desired threshold corresponds to the maximum *variance*



OAK RIDGE National Laboratory

Thresholding and Frame Differencing - Example



Thresholding and frame differencing methods are excellent in reducing your data set to only the features of interest. This can help to greatly improve performance.

OAK RIDGE National Laboratory

Basic Thresholding Example (Google Colab)

Mean Filters

The mean filter is a simple sliding-window spatial filter that replaces the center value in the window with the average (mean) of all the pixel values in the window. The window, or kernel, is usually square but can be any shape.

Mean

5	3	6
2	1	9
8	4	7

5 + 3 + 6 + 2 + 1 + 9 + 8 + 4 + 7 = 45

45 / 9 = 5

*	*	*
*	5	*
*	*	*

Mean filter would return 16:

Median filter removes “impulse noise” – good for edge preservation

Median

6	2	0
3	97	4
19	3	10

*	*	*
*	4	*
*	*	*

OAK RIDGE National Laboratory

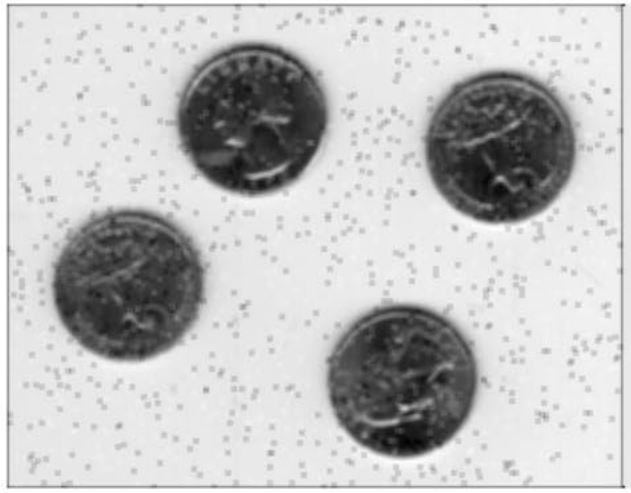
Mean/Median Filters

Original

Original

Median Filter

Mean Filter



Median Filter

OAK RIDGE National Laboratory

Mean-Median Filtering Example (Google Colab)

Adaptive Filters

Adaptive filters use a pixelwise adaptive methods based on statistics estimated from a local neighborhood of each pixel.

`wiener2` estimates the local mean and variance around each pixel.

$$\mu = \frac{1}{NM} \sum_{n_1,n_2 \in \eta} a(n_1, n_2)$$

and

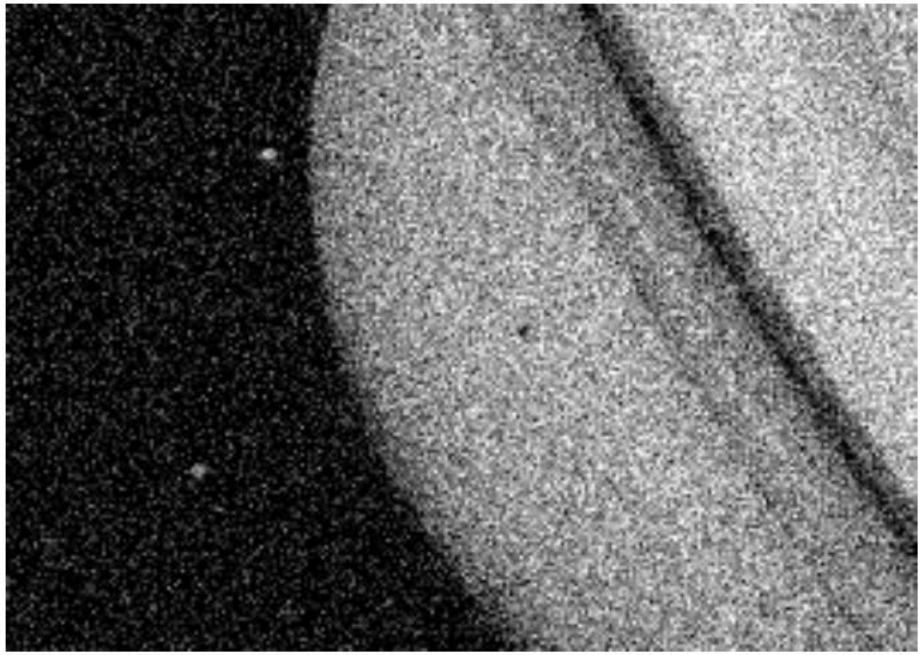
$$\sigma^2 = \frac{1}{NM} \sum_{n_1,n_2 \in \eta} a^2(n_1, n_2) - \mu^2,$$

where η is the *N*-by-*M* local neighborhood of each pixel in the image A. `wiener2` then creates a pixelwise Wiener filter using these estimates,

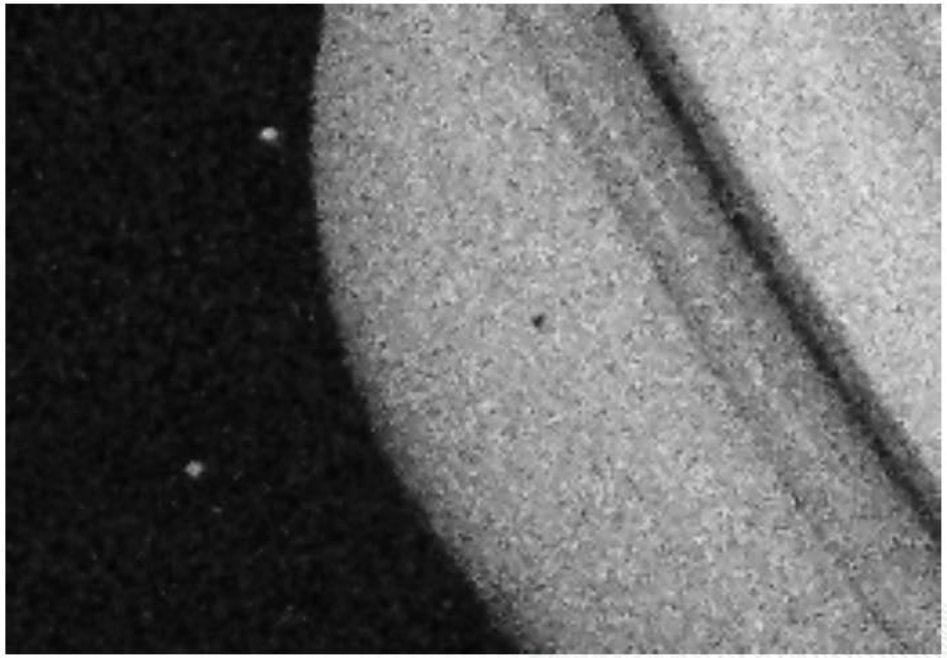
$$b(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} (a(n_1, n_2) - \mu),$$

where v^2 is the noise variance. If the noise variance is not given, `wiener2` uses the average of all the local estimated variances.

Before



After



Adaptive Filtering Example (Google Colab)

Data cleaning – Interpolation/Smoothing

*In numerical analysis, **interpolation** is a method of constructing new data points within the range of a discrete set of known data points.*

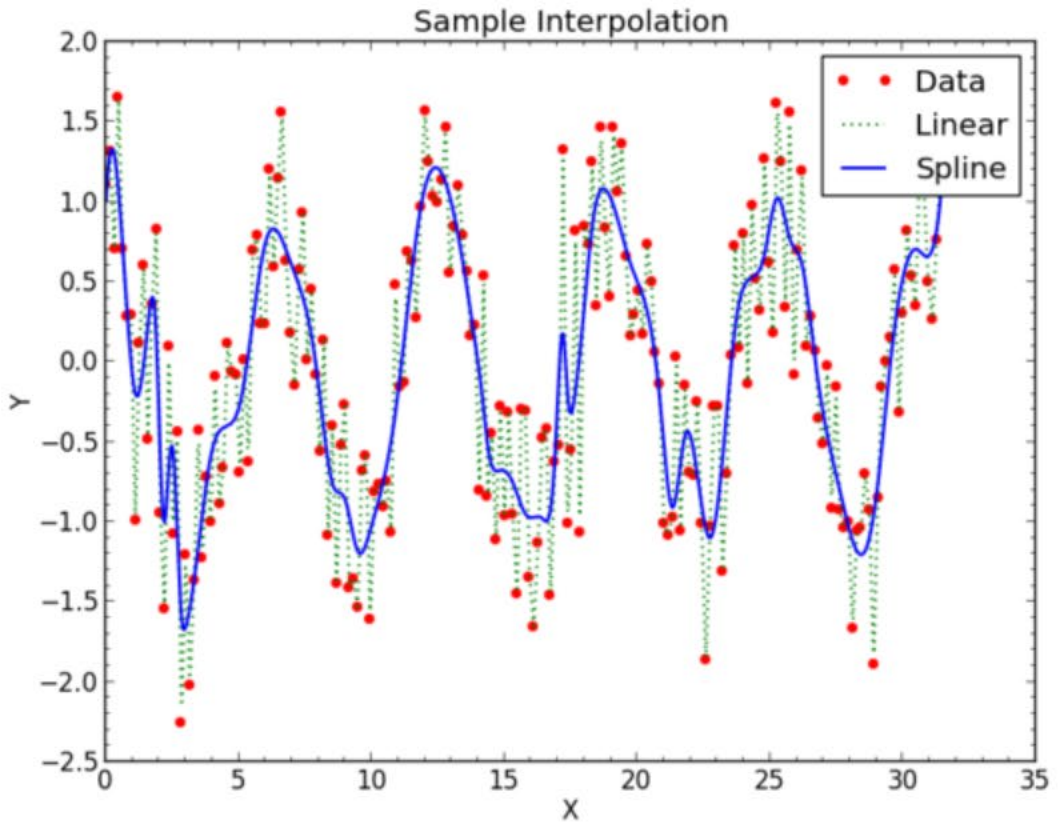
- **Linear Interpolation**
  - *Linear interpolation is a method of curve fitting using linear polynomials to construct new data points within the range of a discrete set of known data points. In linear interpolation **the error is proportional to the square of the distance between the data points**.*
- **Polynomial Interpolation**
  - *Polynomial interpolation is the interpolation of a given data set by a polynomial: given some points, find a polynomial which goes exactly through these points. Polynomial interpolation **can estimate local maxima and minima that are outside the range of the samples**, unlike linear interpolation.*
- **Spline Interpolation**
  - *Spline interpolation uses low-degree polynomials in each of the intervals, and chooses the polynomial pieces such that they fit smoothly together. The resulting function is called a spline.*

OAK RIDGE National Laboratory

Data cleaning – Interpolation/Smoothing

*In **interpolation** a Least Squares method is used to figure out the weights of each of the factors*

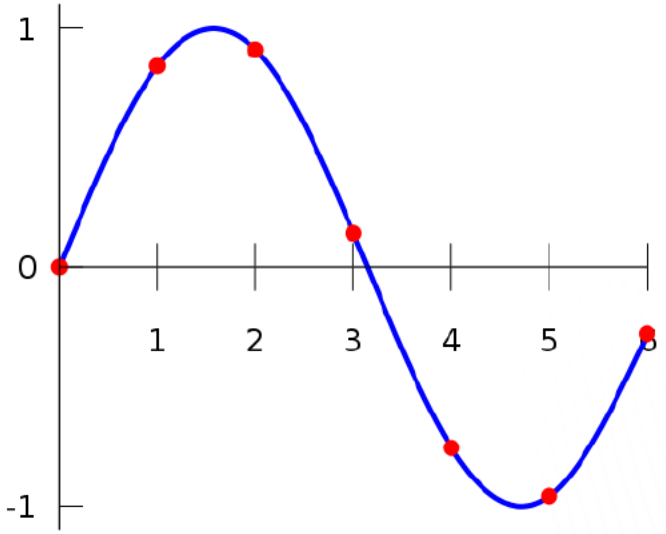
The following examples show idealized data, some of the more challenging examples will be shown by Dan



OAK RIDGE National Laboratory

Spline interpolation uses low-degree polynomials in each of the intervals, and chooses the polynomial pieces such that they fit smoothly together.

$$f(x) = \begin{cases} -0.1522x^3 + 0.9937x, & \text{if } x \in [0,1], \\ -0.01258x^3 - 0.4189x^2 + 1.4126x - 0.1396, & \text{if } x \in [1,2], \\ 0.1403x^3 - 1.3359x^2 + 3.2467x - 1.3623, & \text{if } x \in [2,3], \\ 0.1579x^3 - 1.4945x^2 + 3.7225x - 1.8381, & \text{if } x \in [3,4], \\ 0.05375x^3 - 0.2450x^2 - 1.2756x + 4.8259, & \text{if } x \in [4,5], \\ -0.1871x^3 + 3.3673x^2 - 19.3370x + 34.9282, & \text{if } x \in [5,6]. \end{cases}$$



Like polynomial interpolation, spline interpolation incurs a smaller error than linear interpolation, and the interpolant is easier to evaluate than the high-degree polynomials used in polynomial interpolation.

OAK RIDGE National Laboratory

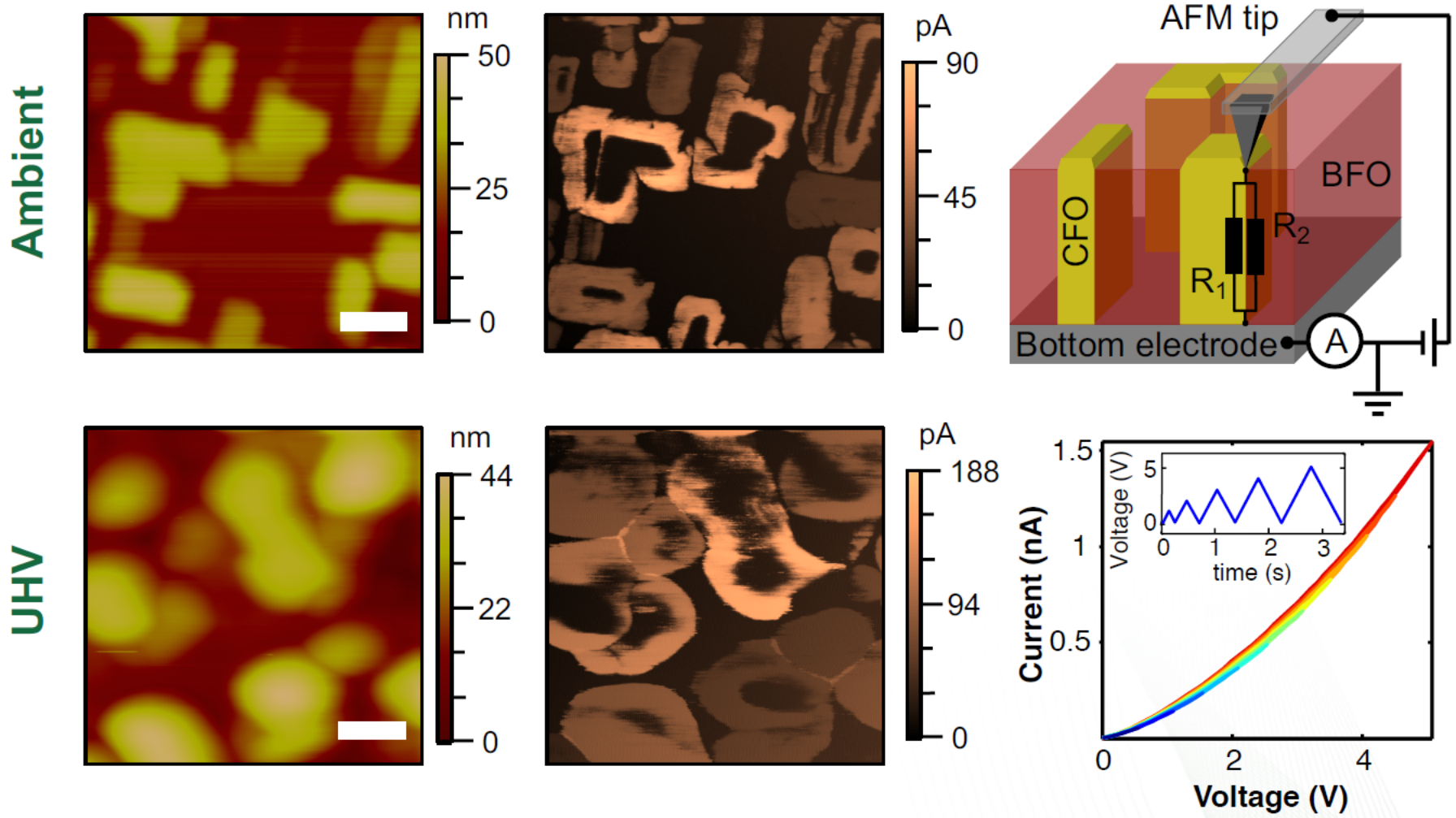
Interpolation Example (Google Colab)

Inconsistencies – the importance of domain knowledge

- **True Zeros**
  - A value that reads $1e^{-9}$, $1e^{-12}$, or another small value that's due to a double/float conversion is not a true zero.
- **Negative Values/Imaginary Numbers**
  - Always check your data types. This can mean a difference between an algorithm that works and a nonsensical answer.
- **Check your range**
  - What is the precision of the values you're measuring? Are these captured by your recording methods? Is there an up/down conversion? What is the range between the smallest and largest values?

Inconsistencies – the importance of domain knowledge

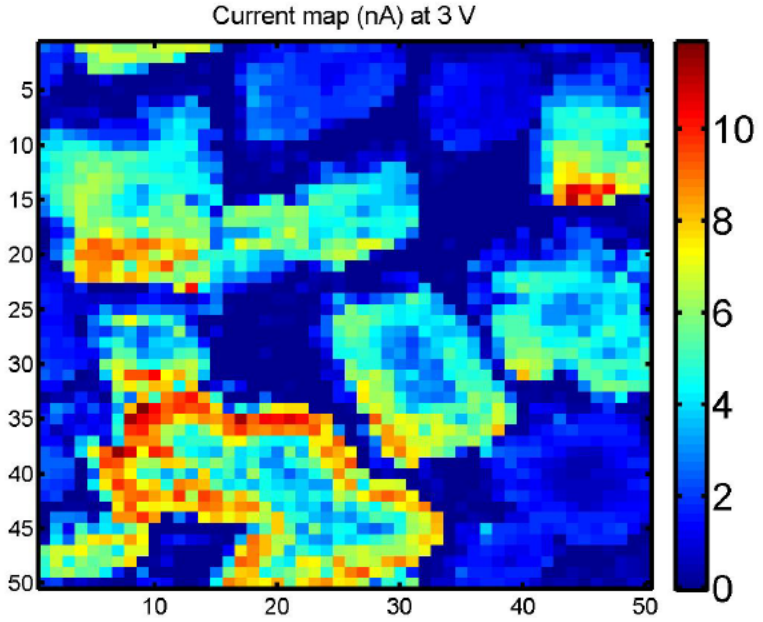
$BiFeO_3$-$CoFe_2O_4$ FORC-IV

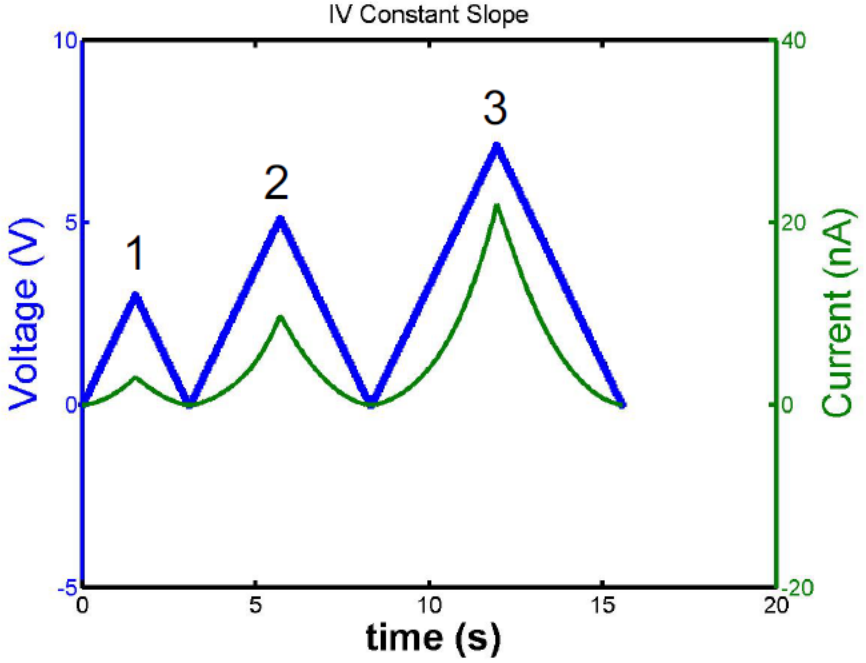


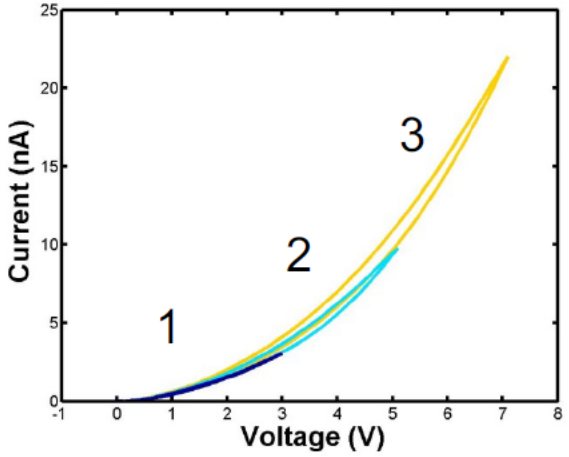
E. Strelcov et. al. ACS Nano 8 (6) 6449-6457 (2014)
E. Strelcov et. al. Nano Letters 15 (4), 2343-2349 (2015)

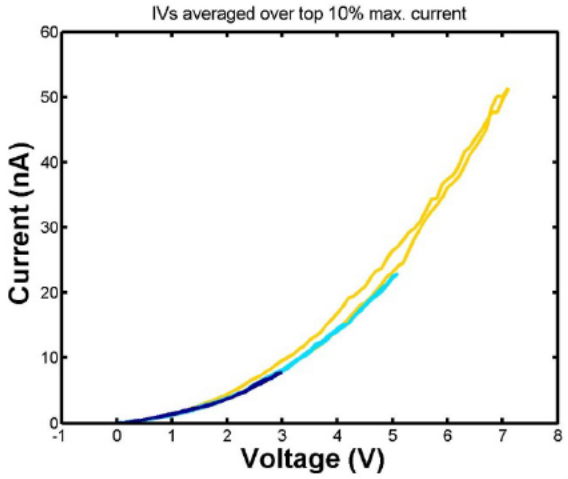
OAK RIDGE National Laboratory

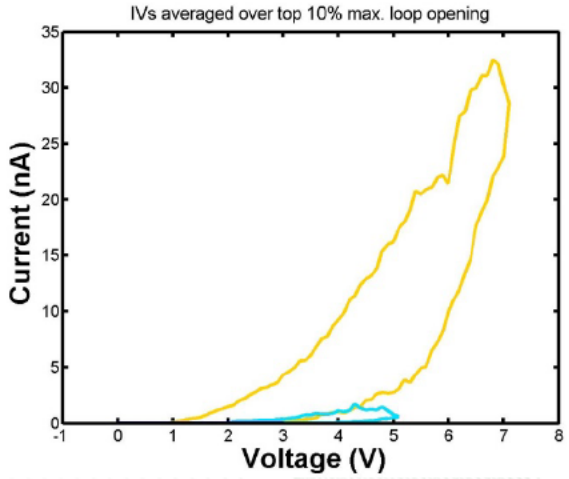
FORC-IV Measurement





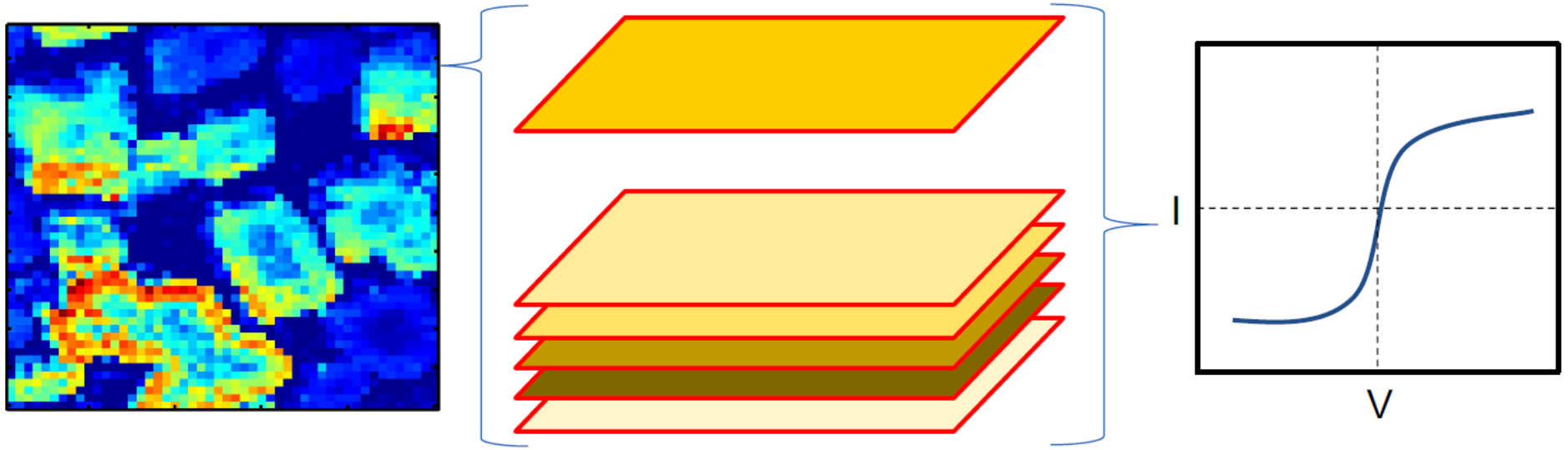






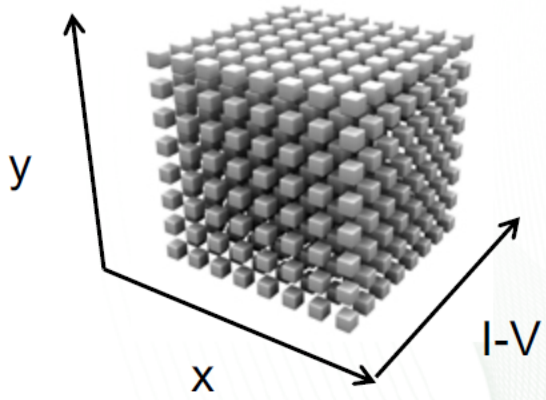
OAK RIDGE National Laboratory

3D Data Architecture of a FORC-IV Measurement

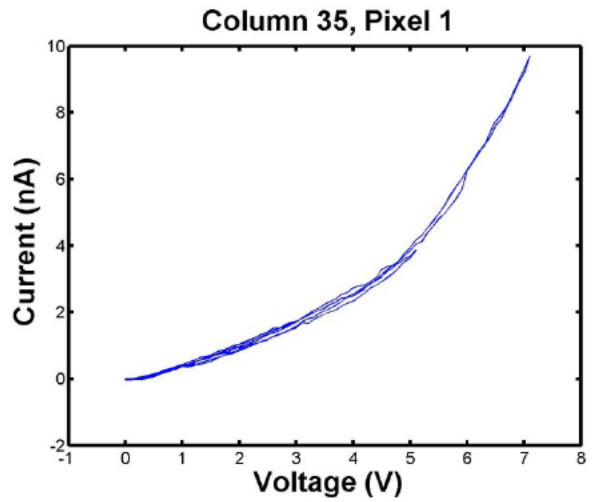


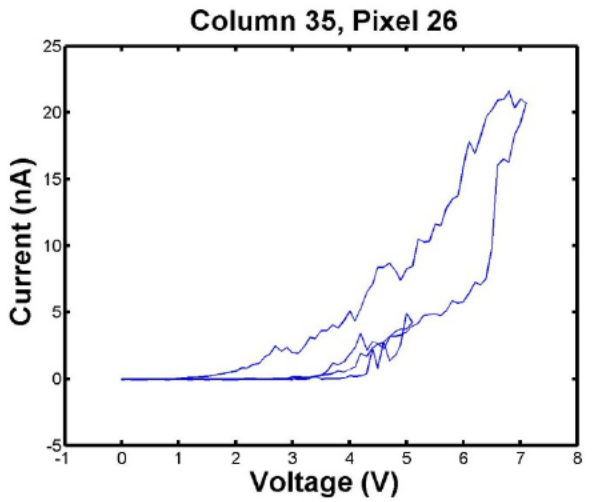
The IV response is measured at each spatial pixel as a function of a triangular excitation wave peaking at an arbitrary pre-set maximal value

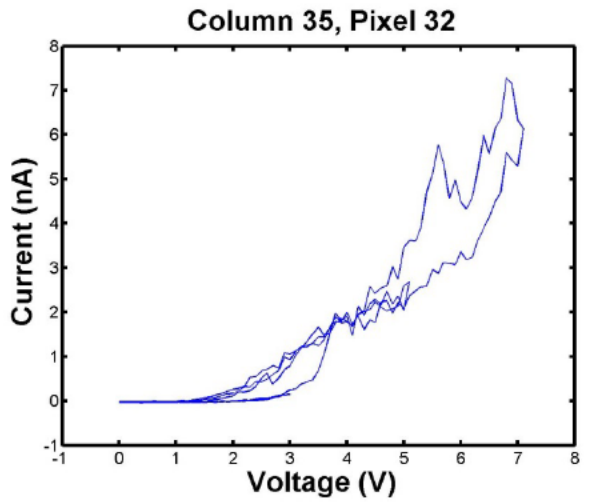
Multiple range IV curves are collected as a result – forming a 3D hyperspectral dataset; where the IV behavior vector is supported on a 2D grid

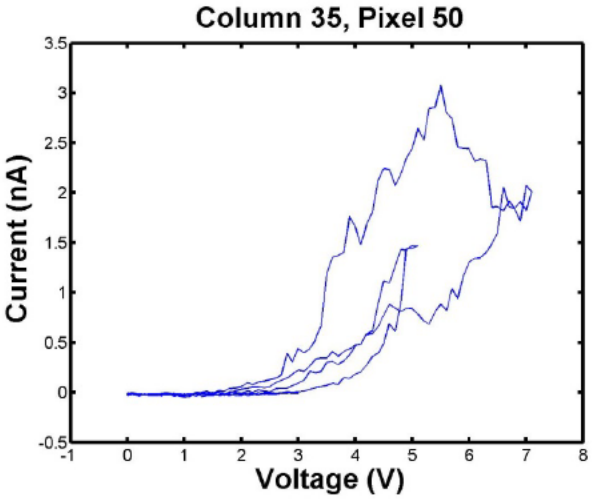


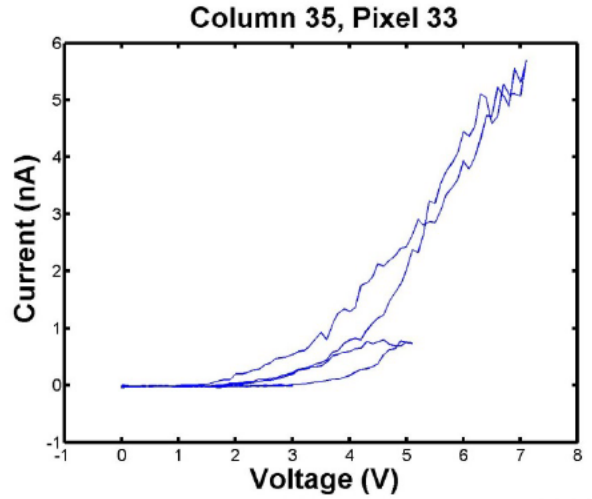
3D Data Architecture of a FORC-IV Measurement

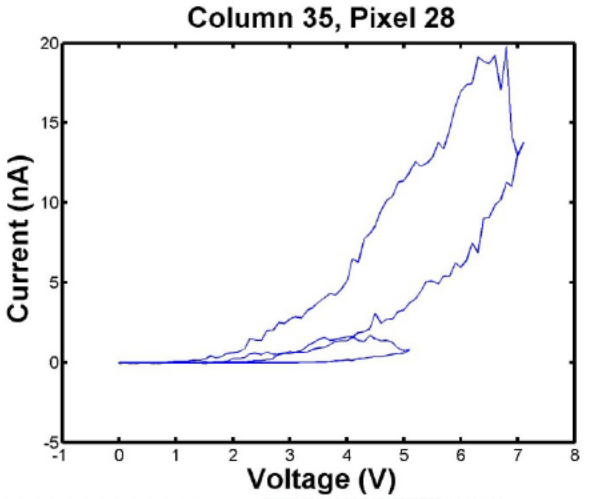






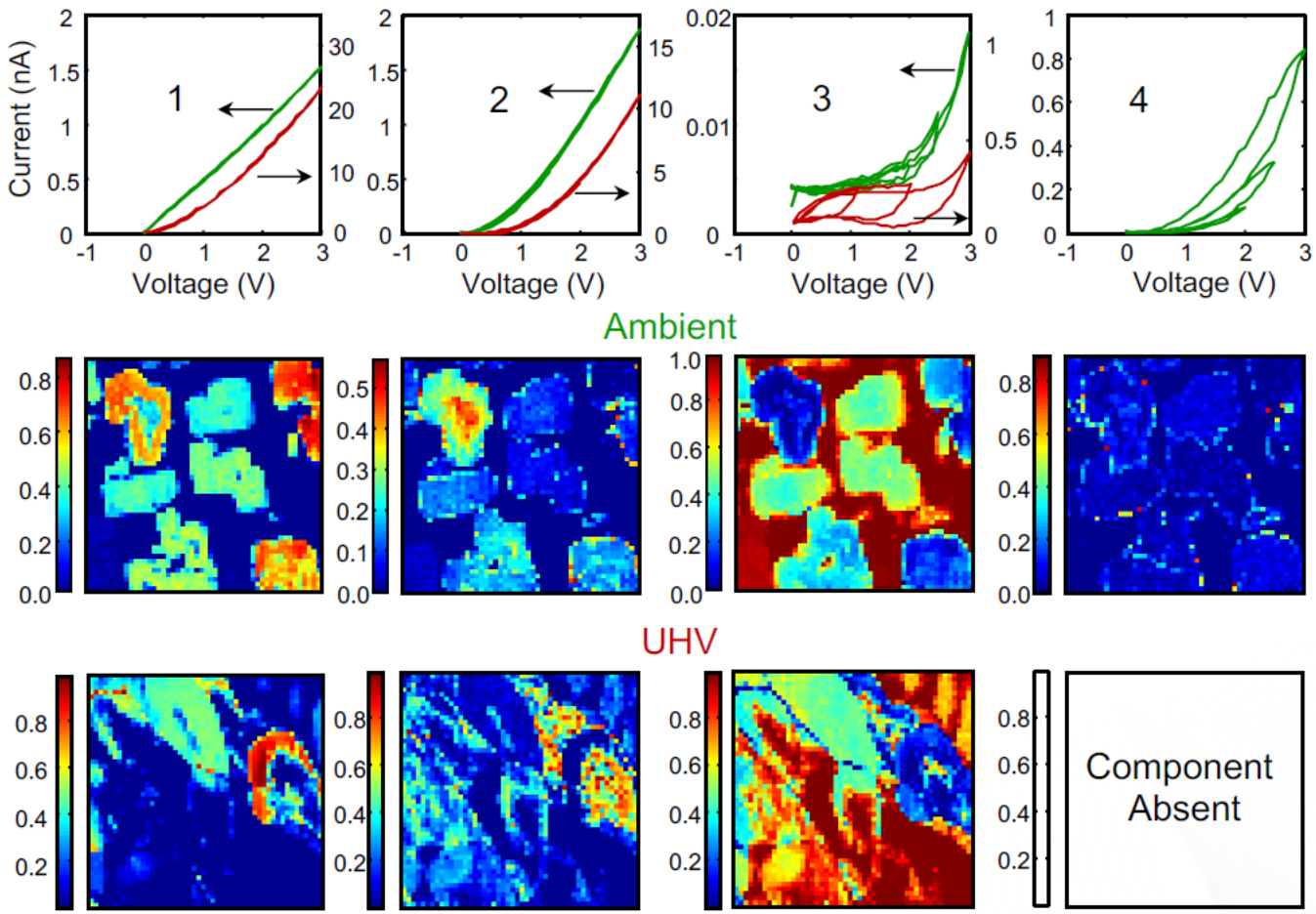






OAK RIDGE National Laboratory

Bayesian Example: BiFeO3-CoFe2O4 FORC-IV, Vmax = 3V



OAK RIDGE National Laboratory

Inconsistencies

- **What am I sensing the signal with? What is the dynamic range of this tool?**
  - What is the smallest/largest value? How quickly can it sample the signal
  - Is $10^{-25} = 0$?
- **What is the natively captured type and precision of the signal? Am I up- or down-sampling?**
  - Run of the mill analytical chemistry detectors are 8 bit
  - Higher end detectors are 14-16 bit
  - High precision ($$$) 22 bit
  - Basic OS, 32 bit; is it upconverting the data? Remember your Sig Figs!
  - Single, double, unsigned, string … Oh My!
- **Artificial data poisoning – negative/imaginary values**
  - Bits can flip, and you may induce these with prior preprocessing
  - Imaginary values can be your best friend, or your worst enemy – you decide

Hyperspectral Pytchography (Google Colab)

Data Cleaning Conclusions

- No measurement is perfect, every device, detector, data collector has a limitation and a spec
- Consider data flows, analog to digital conversion, and precision
- Know how outliers are treated and represented
- Use interpolation to your advantage, matching the number of data points between measurements unlocks multivariate algorithms capabilities
- Do not overfit your data, use the scientific insight
- Know the capabilities and limitations of the filters you are using
- Play! Try out things to see if they work
- People with insight of how the data was collected, amassed, and pre-processes almost always area at advantage

OAK RIDGE National Laboratory

Unsupervised Learning

Summary of Links

1. Thresholding

https://colab.research.google.com/drive/1DyXRj9fLAK74hrkpZnUno0kTm_cnheVQ?usp=sharing

2. Filtering

https://colab.research.google.com/drive/1iG5XapW7DPS-PC_2v0yNMAEdB2DzZ5t5?usp=sharing

3. Adaptive Thresholding

https://colab.research.google.com/drive/1dtgf5AcYFJh_DjDaRCyHERlUi9gfyRYZ?usp=sharing

4. Interpolation

https://colab.research.google.com/drive/1Qi7eQk2VW5H7a9wTqiJvk0AwdUprYNek?usp=sharing

5. Hyperspectral Ptychography

https://colab.research.google.com/drive/1QPxfQtZXA6mHfL3kyowPjPrc_O1vTf7f?usp=sharing

6. Latent Variable Analysis

https://colab.research.google.com/drive/1MUdb5TqWJqX1HXK_eU9ukwdUAEE3icMW?usp=sharing

7. Dissimilarity Analysis

https://colab.research.google.com/drive/1H85mpY4Oz52lcCTE1xcxOPi_JAsCjYjy?usp=sharing

8. Clustering

https://colab.research.google.com/drive/1F7GmEQcUNoN8KkW-CmWJCu4xHjqvNKBH?usp=sharing