Welcome to the 2022 M&M Short Course X-16

Data Analysis in Materials Science with

Presented by (a selection of) the HyperSpy developers:

- Carter Francis
- Joshua Taillon

Session 3:
An Introduction to Machine Learning in Electron Microscopy

Electron Microscopy in the age of "Big Data"

Josh Taillon

July 31, 2022

NIST Disclaimer

NIST
MATERIAL MEASUREMENT LABORATORY

What types of machine learning are we talking about?

- For the most part, *unsupervised* learning:
    - *i.e.* no training or validation of models
    - Goal is to find previously unknown patterns in data set without pre-existing label
    - Many algorithms for this are built-in with HyperSpy

What types of machine learning are we talking about?

- For the most part, *unsupervised* learning:
    - *i.e.* no training or validation of models
    - Goal is to find previously unknown patterns in data set without pre-existing label
    - Many algorithms for this are built-in with HyperSpy

- Distinct from supervised learning, reinforcement learning, or deep learning
    - Not covered here — not included in HyperSpy directly
    - Using `scikit-learn`, TensorFlow, pyTorch, etc.

What can unsupervised learning do for you?

- The primary use for unsupervised learning is in *signal separation*
    - Sometimes called *hyperspectral unmixing* as well

What can unsupervised learning do for you?

- The primary use for unsupervised learning is in *signal separation*
    - Sometimes called *hyperspectral unmixing* as well

- Ideally, we can automatically determine what signals are present in the data, and where they are located
    - This is the basis of most vendors' *phase mapping* offerings

What can unsupervised learning do for you?

- The primary use for unsupervised learning is in *signal separation*
  - Sometimes called *hyperspectral unmixing* as well

- Ideally, we can automatically determine what signals are present in the data, and where they are located
  - This is the basis of most vendors' *phase mapping* offerings

- Can also be used for *dimensionality reduction* and denoising:
  - Picking out only the "interesting" features and discarding the rest of the noise

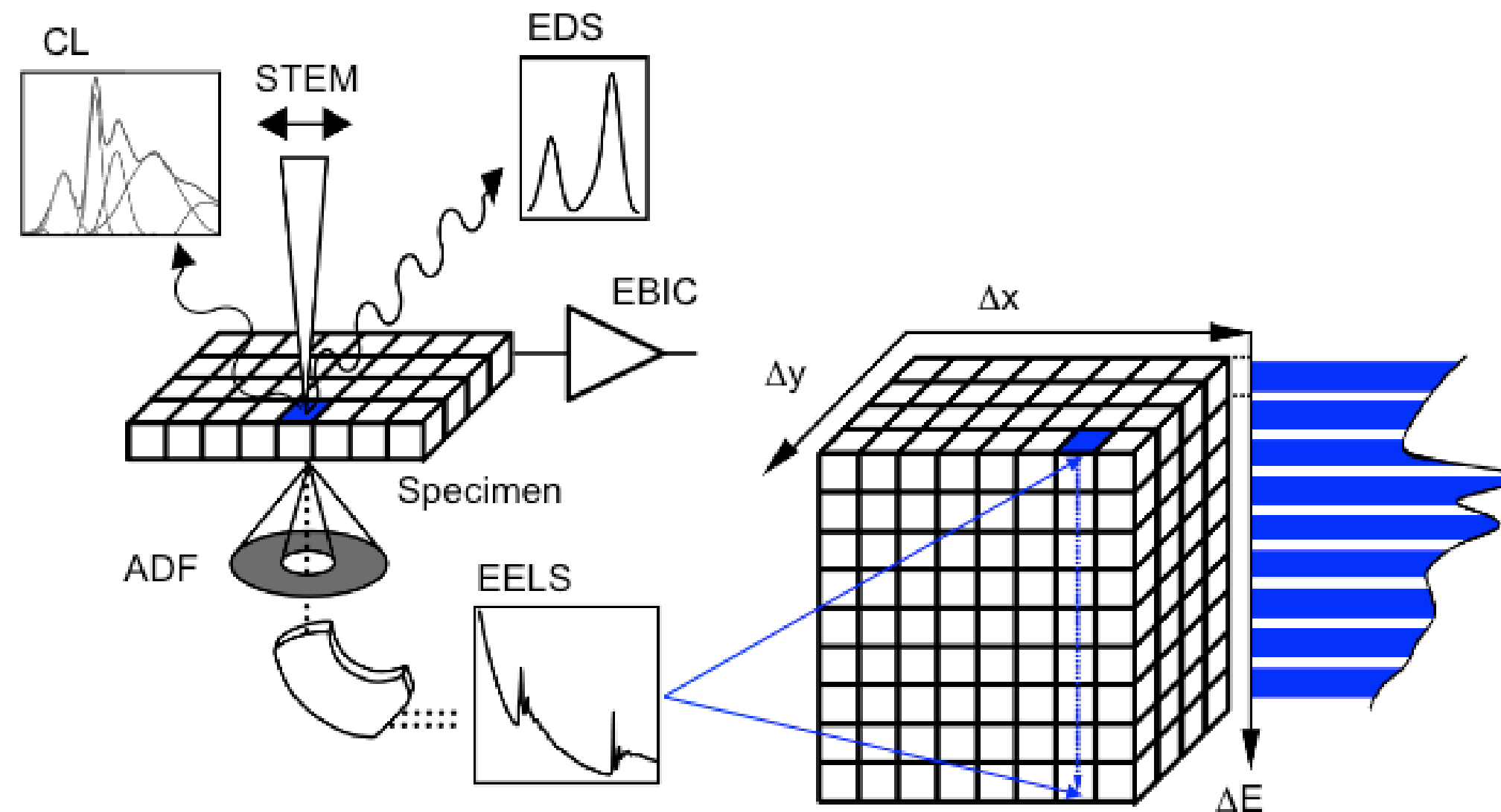# What techniques is this used on?

- Most typically in the EM fields, signal separation is performed on raster-based *hyperspectral mapping* methods:
    - STEM-EELS and STEM-EDS
    - SEM-EDS
    - (μ-)XRF mapping
    - FTIR mapping
    - Cathodoluminescence mapping

What techniques is this used on?

- Most typically in the EM fields, signal separation is performed on raster-based *hyperspectral mapping* methods:
    - STEM-EELS and STEM-EDS
    - SEM-EDS
    - (μ-)XRF mapping
    - FTIR mapping
    - Cathodoluminescence mapping

- Otherwise, these techniques can be used on any type of data:
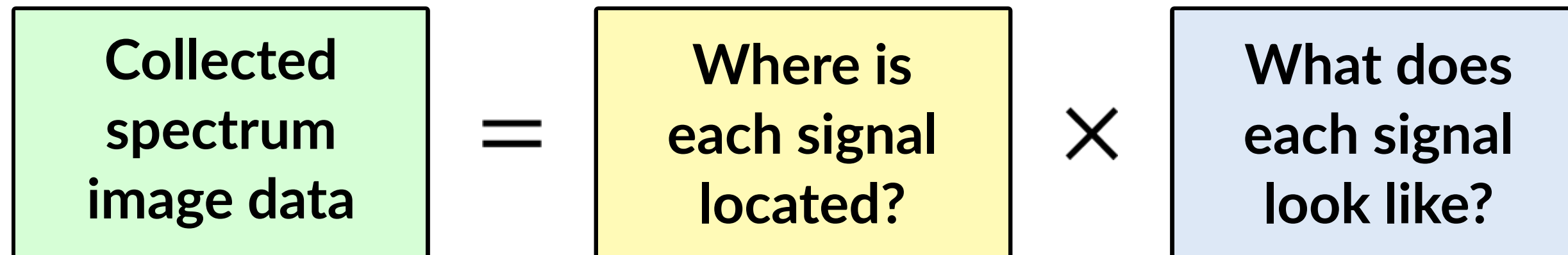   - Images, time series, etc.

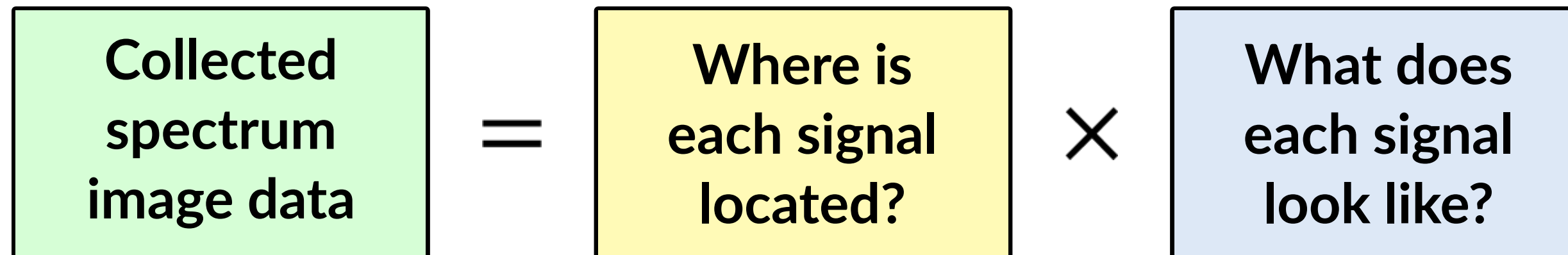# What *is* hyperspectral unmixing?

- Start with some hyperspectral data:

Image courtesy of Gatan, Inc.

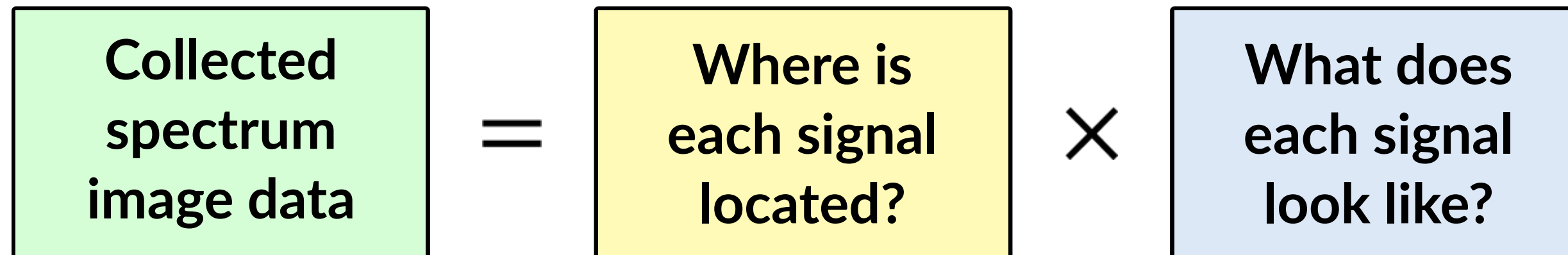# What *is* hyperspectral unmixing?

| Collected spectrum image data | = | Where is each signal located? | × | What does each signal look like? |

What *is* hyperspectral unmixing?

Collected spectrum image data	=	Where is each signal located?	×	What does each signal look like?

$$\mathbf{D}_{(x,y),E} = \mathbf{W}_{(x,y),n} \times \mathbf{S}_{n,E} + \epsilon_{(x,y),E}$$

What *is* hyperspectral unmixing?

Collected spectrum image data	=	Where is each signal located?	×	What does each signal look like?

$$\mathbf{D}_{(x,y),E} = \mathbf{W}_{(x,y),n} \times \mathbf{S}_{n,E} + \epsilon_{(x,y),E}$$

$$\mathbf{D}_{i,E} = \mathbf{W}_{i,n} \times \mathbf{S}_{n,E} + \epsilon_{i,E}$$
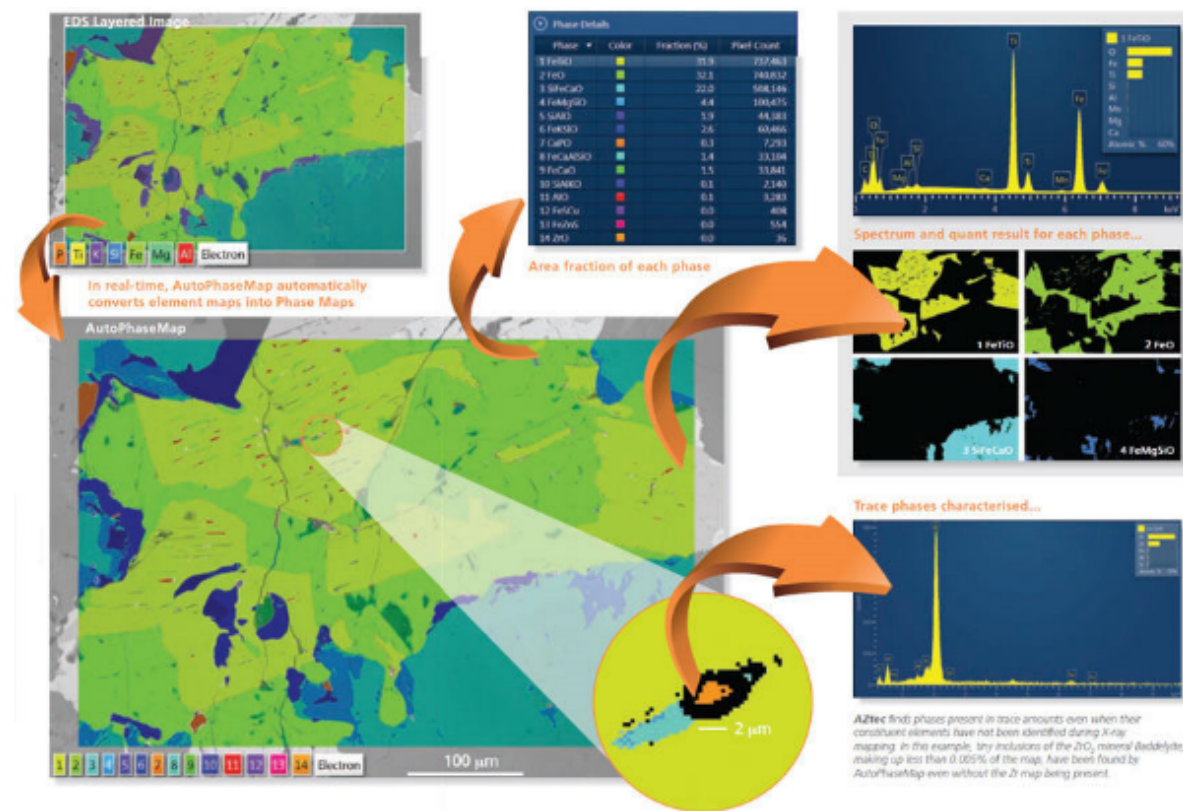
What *is* hyperspectral unmixing?

A linear algebra problem!

What do the vendors offer?

- If you've used a modern EDS software package, you probably have done hyperspectral unmixing (they usually call it *phase mapping*)…

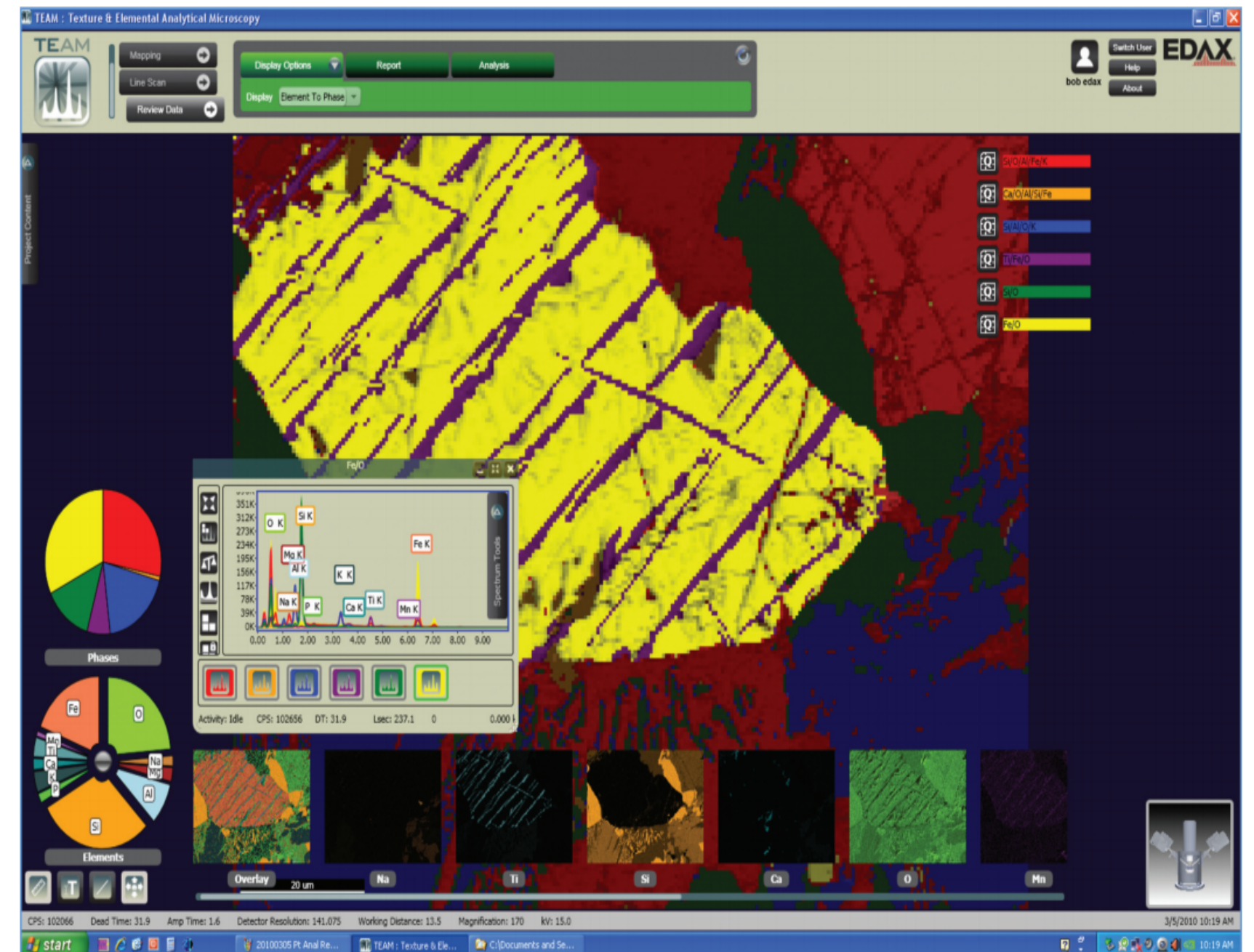Oxford — *AutoPhaseMap*
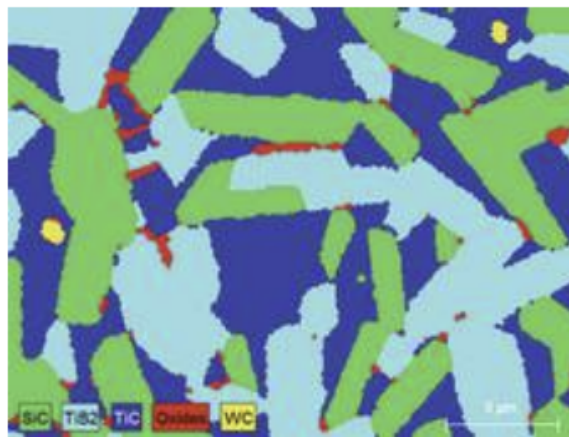

EDAX — *Smart Phase Mapping*

What do the vendors offer?

- If you've used a modern EDS software package, you probably have done hyperspectral unmixing (they usually call it *phase mapping*)…
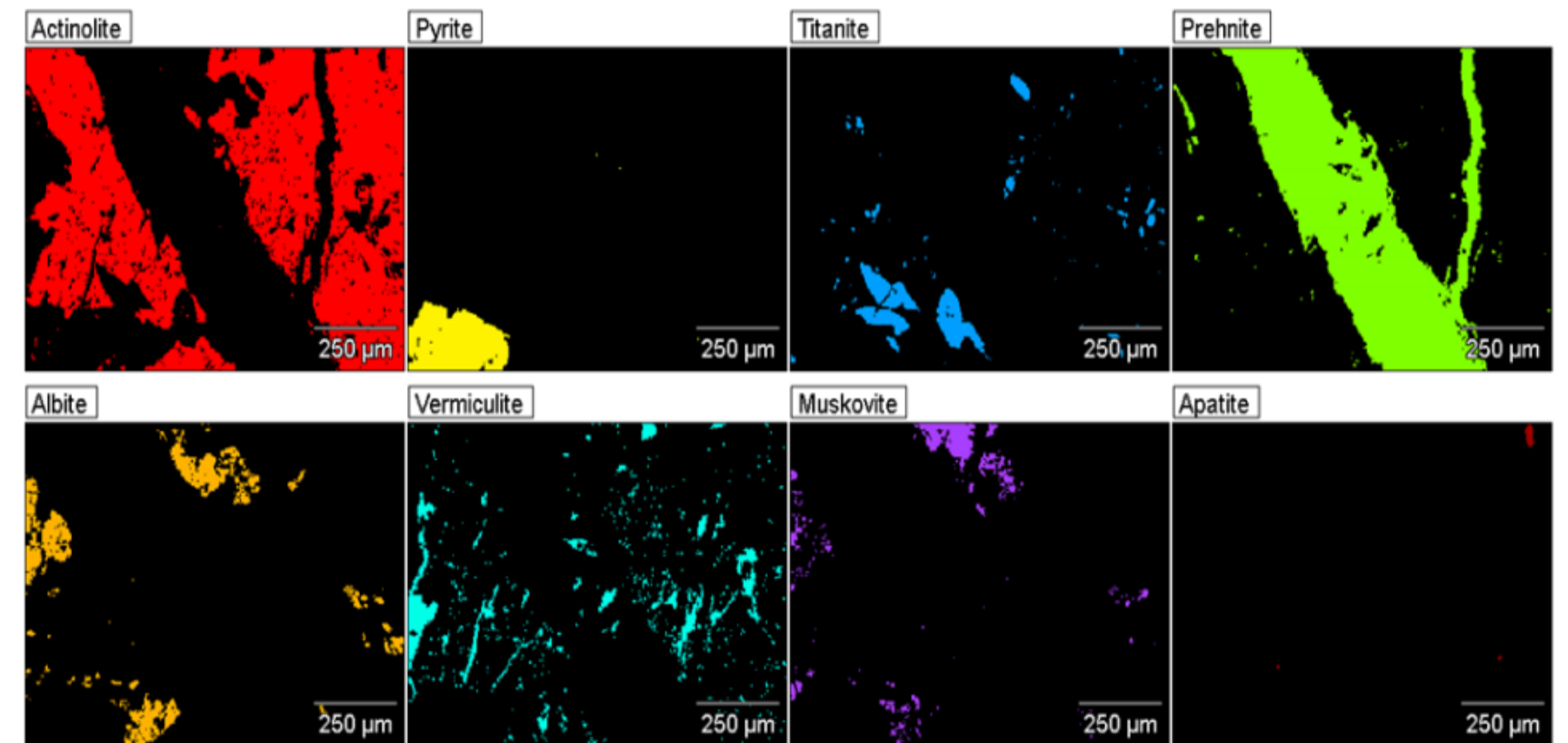

Bruker — *AutoPhase*

Thermo Fisher — *COMPASS*

Strengths/challengs with *vendor* options?

The Good

- **Simple point-and-click operation**

- **Tight integration**

  - Collection, visualization, reporting, etc.

- **Usually runs in real-time**

- **Integration with other data sources (*e.g.* EBSD)**

- **Generally "just works"**

Strengths/challengs with *vendor* options?

The Good

- **Simple point-and-click operation**

- **Tight integration**

    - Collection, visualization, reporting, etc.

- **Usually runs in real-time**

- **Integration with other data sources (*e.g.* EBSD)**

- **Generally "just works"**

The Not So Good

- **Extremely "black box"**

- **Reproducibility (!)**

    - Configurable options with little understanding of why

- **What are the uncertainties?**

- **Tied to software ($)**

- **Choice of vendor should not change the scientific result**

Strengths/challengs with *open-source* options?

The Not So Good

- Usually not point-and-click

- (Can be) difficult to access raw data from the vendor software

- Generally only post-processing

- Learning curve can be substantial

- Can take a lot more fiddling

Strengths/challengs with *open-source* options?

The Not So Good

- Usually not point-and-click

- (Can be) difficult to access raw data from the vendor software

- Generally only post-processing

- Learning curve can be substantial

- Can take a lot more fiddling

The Good

- You know what's happening

- Reproducibility (!)

  - Anyone can recreate your analysis (including you!)

- Uncertainty can be understood

- Usually free

- Results do not depend on vendor

Open-source "phase mapping"

- Many algorithms exist to solve: $\mathbf{D}_{(x,y),E} = \mathbf{W}_{(x,y)} \times \mathbf{S}_E$
    - This is a highly underdetermined problem, with an infinite number of solutions
    - Assumptions implicit in each algorithm affect their suitability for EDS, EELS, etc.

Open-source "phase mapping"

- Many algorithms exist to solve: $\mathbf{D}_{(x,y),E} = \mathbf{W}_{(x,y)} \times \mathbf{S}_E$

  - This is a highly underdetermined problem, with an infinite number of solutions
  - Assumptions implicit in each algorithm affect their suitability for EDS, EELS, etc.

- Primary methods (built into ⬚ HyperSpy):

  - Principal component analysis (PCA) — finds non-physical spectra that describe the most variance in the datacube
  - Independent component analysis (ICA) — maximizes independence between spectral results
  - Non-negative matrix factorization (NMF) — enforces positivity in spectral components and weights

Principle component analysis + blind source separation

- PCA will find the spectra that describe the maximum amount of variance within the data set
    - These principle components (PCs) will contain the most "information" — but not information relevant to a microscopist
    - The results will look like vaguely spectral, but almost guaranteed to be non-physical

Principle component analysis + blind source separation

- PCA will find the spectra that describe the maximum amount of variance within the data set
    - These principle components (PCs) will contain the most "information" — but not information relevant to a microscopist
    - The results will look like vaguely spectral, but almost guaranteed to be non-physical

- Blind source separation often gives more interpretable results:
    - Independent component analysis (ICA):
        - Operates on the PCs to maximize independence between spectral results
    - Non-negative matrix factorization (NMF):
        - Decompose the matrix with constraints that match physical expectations

A simple example:

- Signal separation enabling EDS tomography

- Atom probe specimen fabricated from NIST SRM 2135c
    - Ni/Cr thin film depth profile standard (on Si substrate)
    - Layer thicknesses are approximately 56 nm
    - Data collected by Andrew Herzing (NIST)

- Data collected from 0 to 360 degrees tilt in increments of 5 degrees (73 spectrum images)
    - Dataset is $(x, y, \theta, E) : (165, 124, 73, 900)$– 12.2GB

- HAADF and XEDS SI data collected simultaneously

A simple example:

HAADF Signal

XEDS Signal
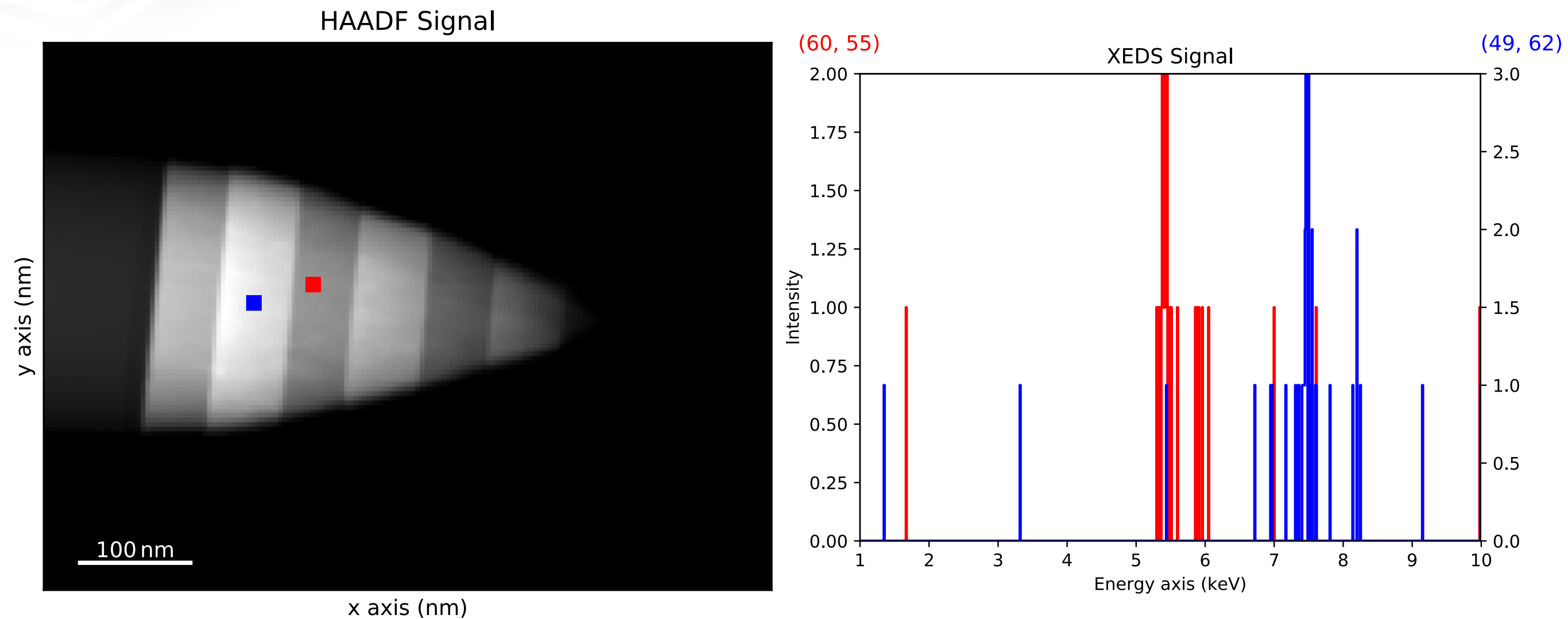
Single pixel counts in the single digits
Cr and Ni visible, but noisy

A simple example:

Ni-K$_\alpha$ intensity   Si-K$_\alpha$ intensity   Cr-K$_\alpha$ intensity
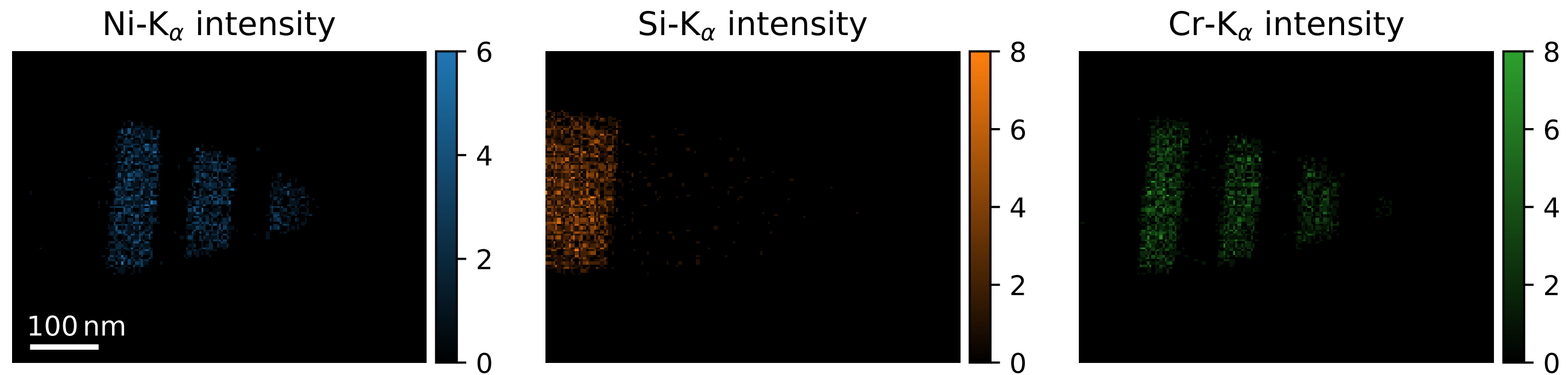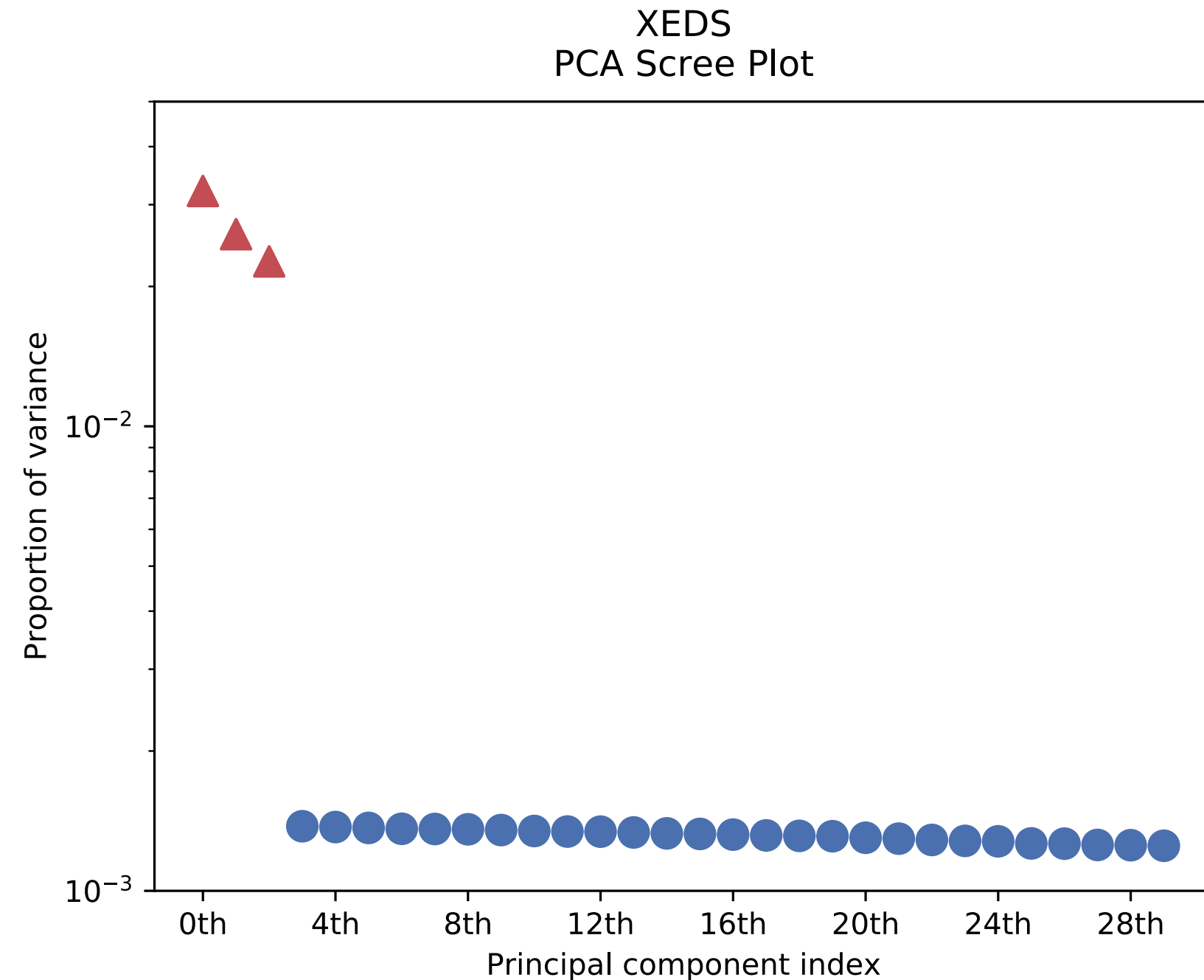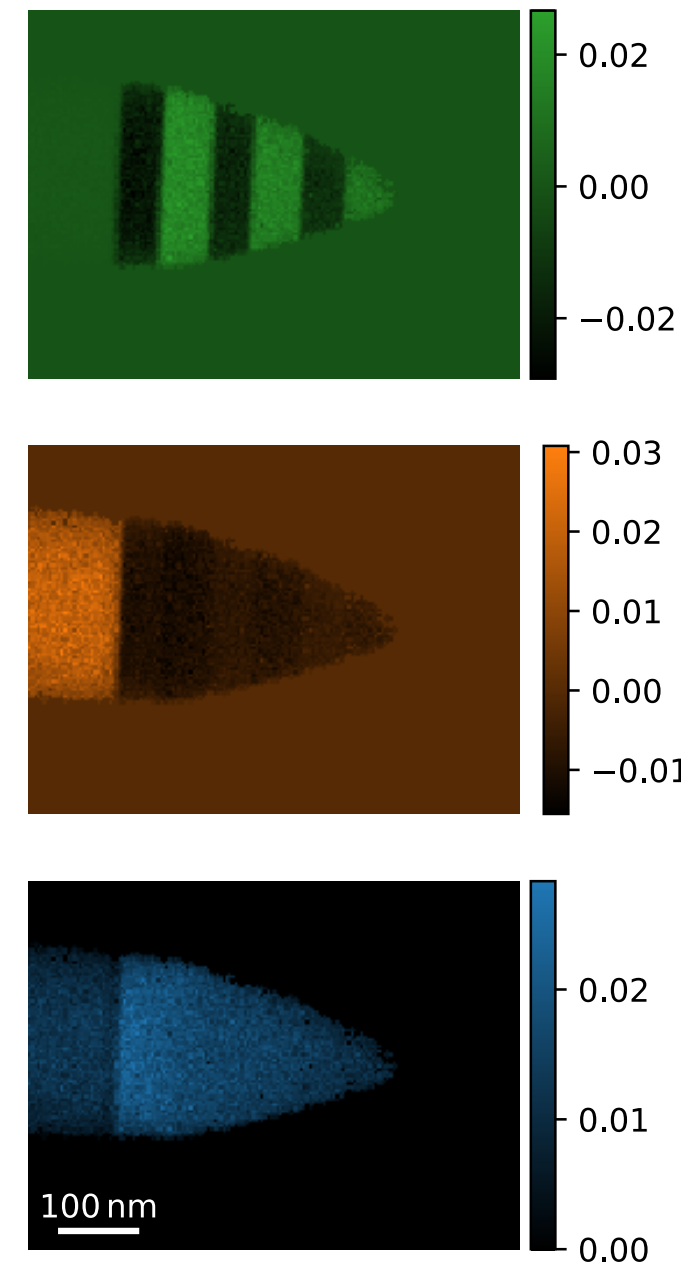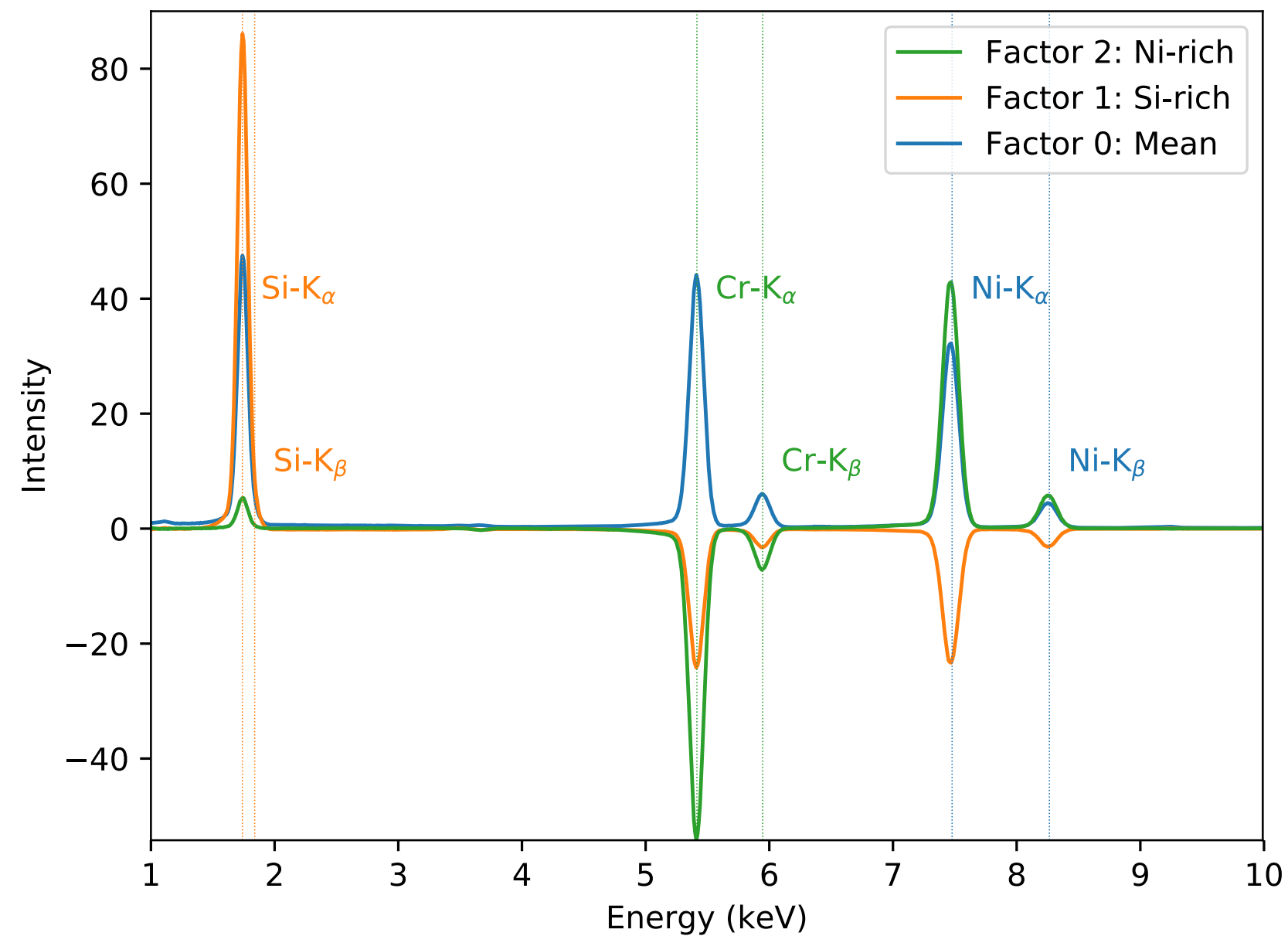
Single pixel counts in the single digits
Cr and Ni visible, but noisy

How many components to choose?

- **PCA orders components by "described variance"**

- ***a priori* we know there should be three components**
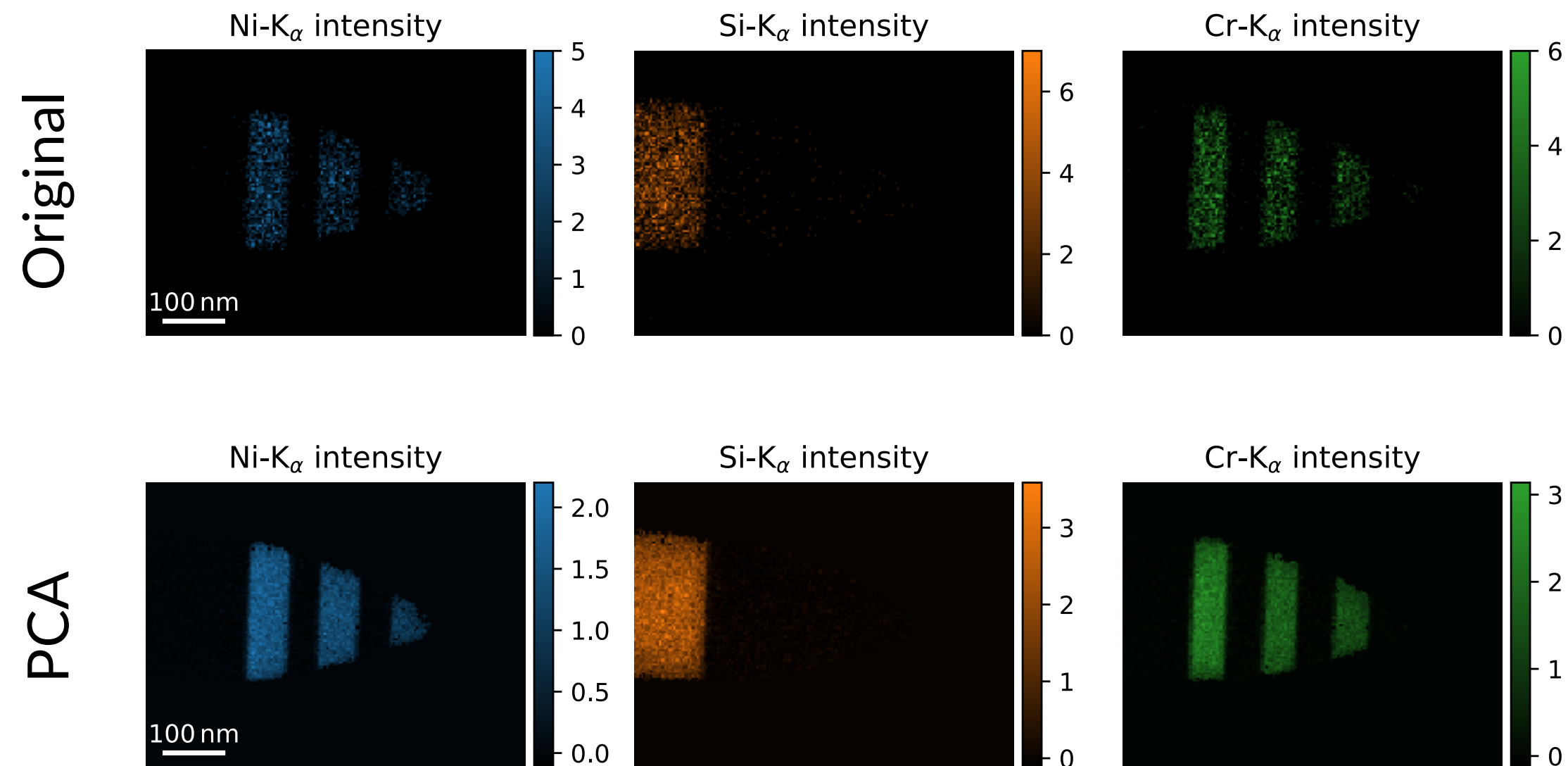
- **Three important components confirmed**

XEDS
PCA Scree Plot

Result of PCA - Factors and Loadings

Signals are non-physical; Elements mixed between factors
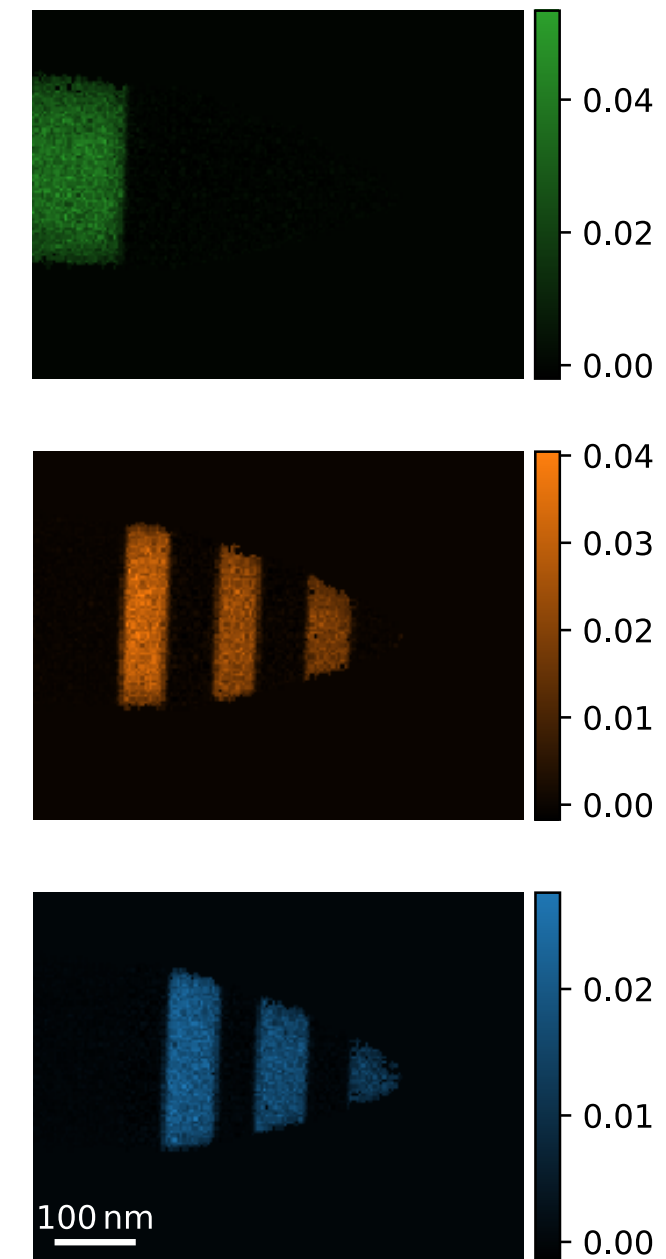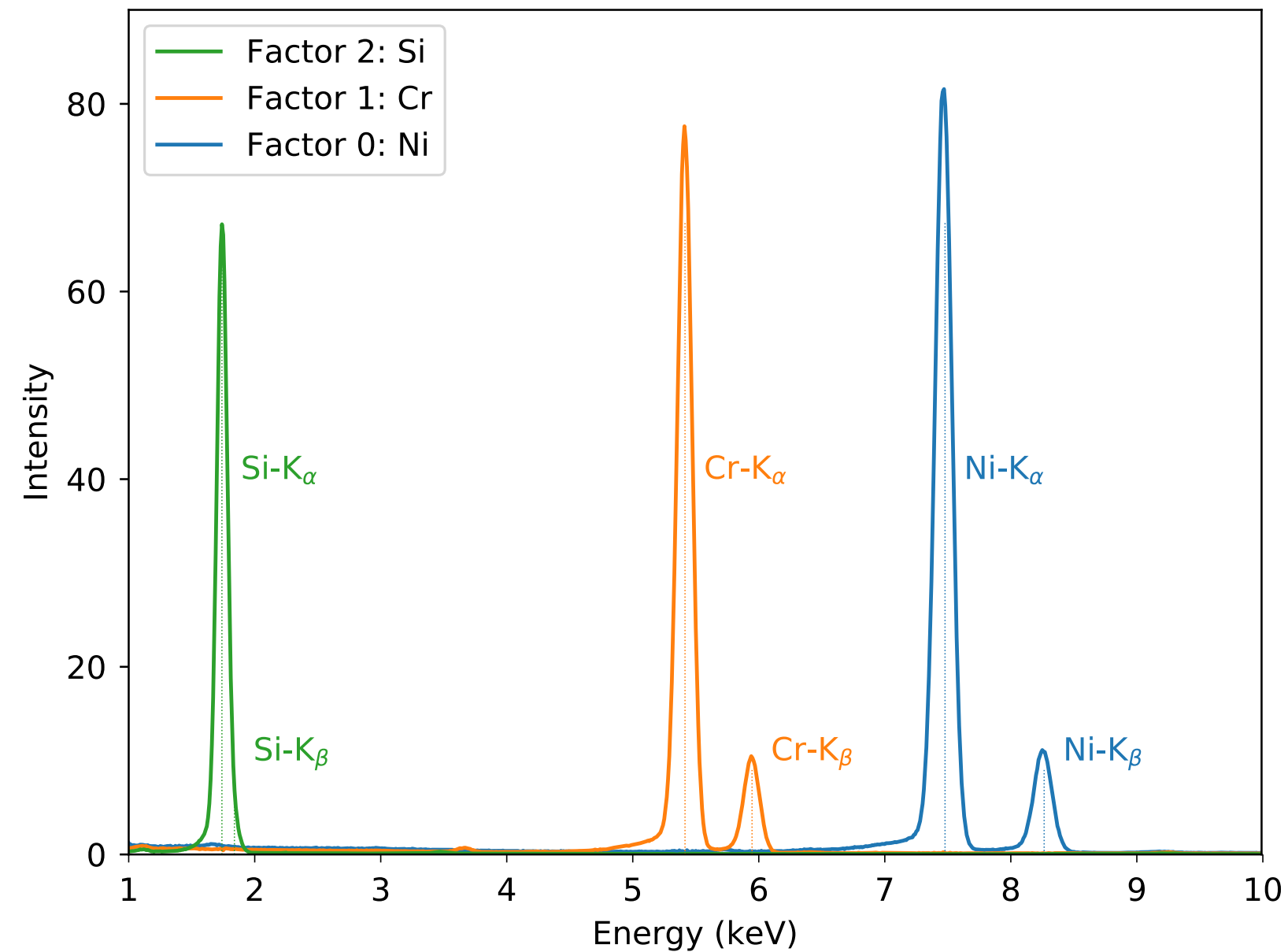*Drastically enhances S/N ratio in "loading" maps*

Result of PCA - Denoising

Line intensities extracted from model with top 3 PCA elements;
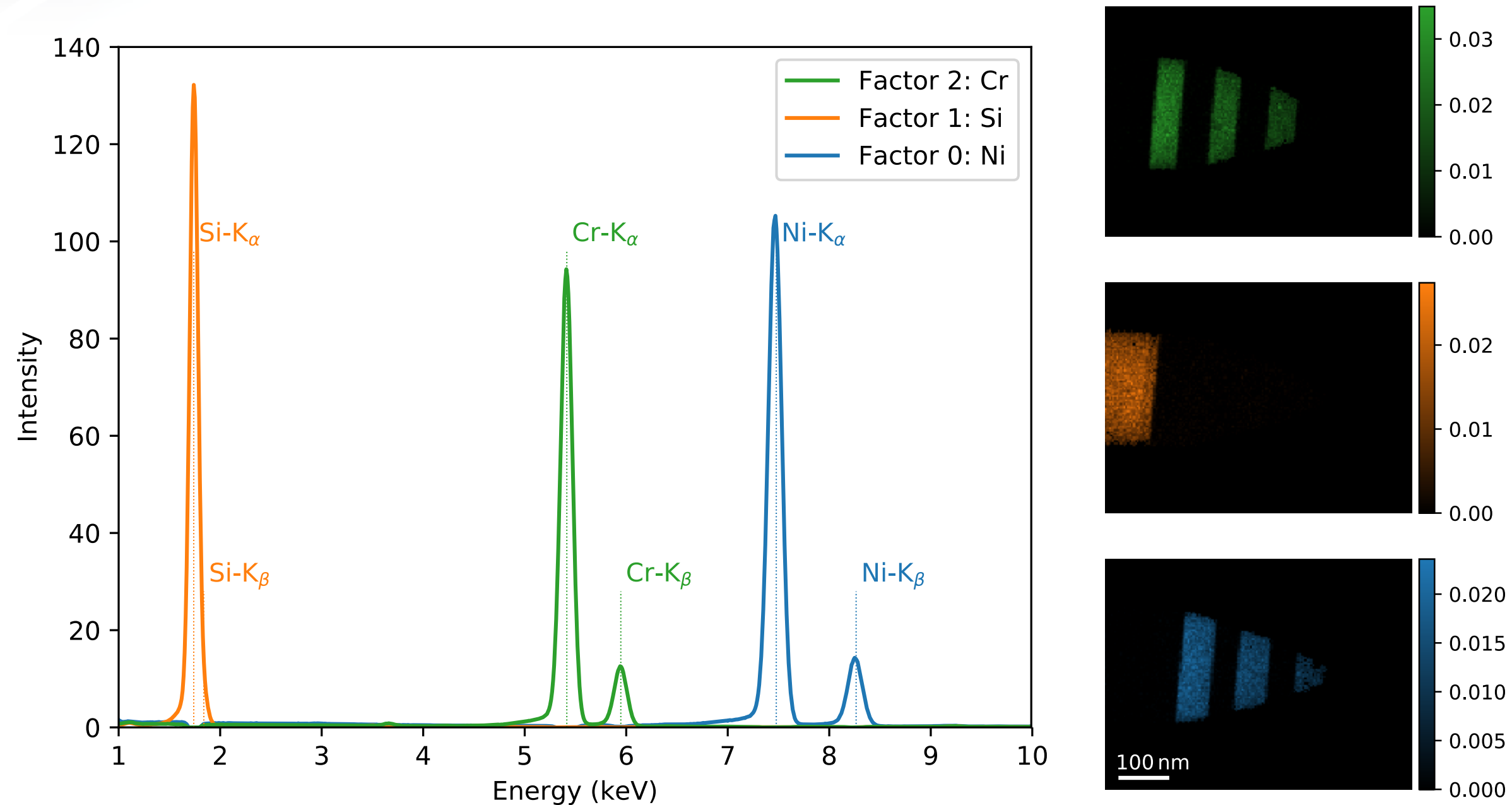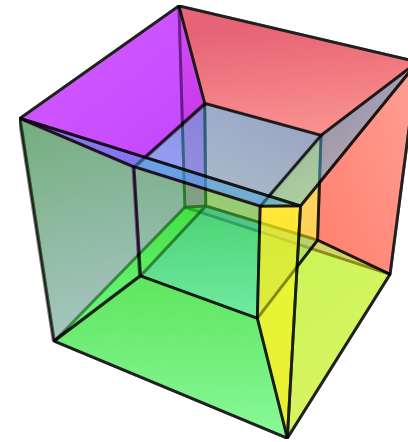Drastically enhances S/N ratio in line intensity maps

Result of PCA + BSS

One component for each element (phase)
Much more interpretable results

Result of NMF

One component for each element (phase)
Drastically enhances S/N ratio in "loading" maps

Thank you!

Onto the demo…
Any questions?

joshua.taillon@nist.gov