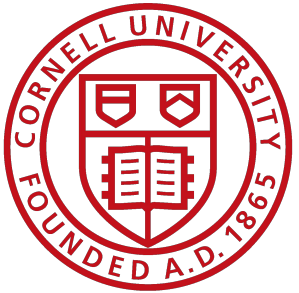
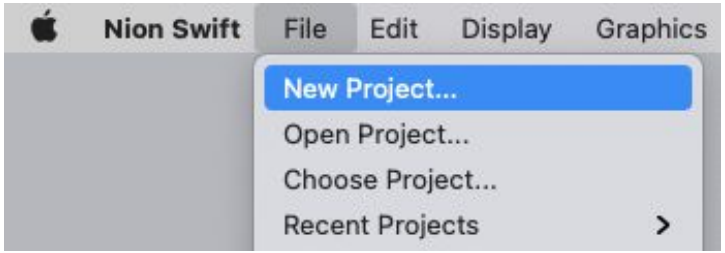
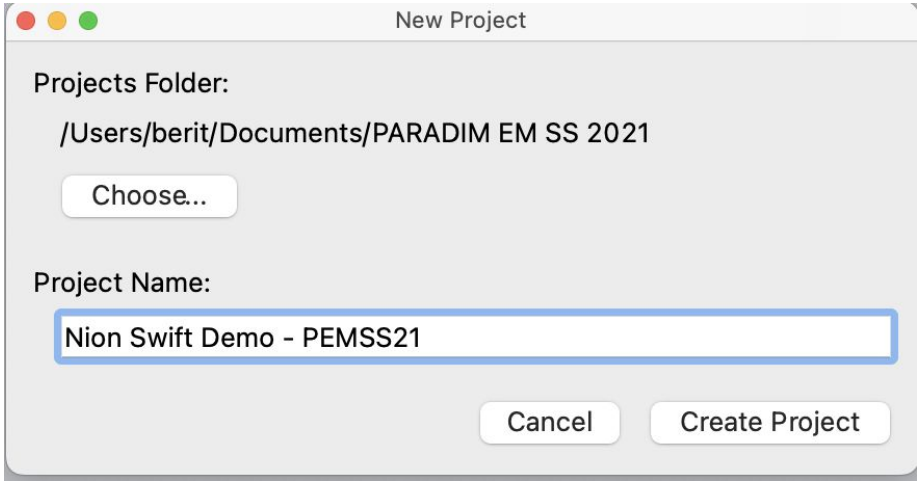


T4a: Basic EELS with Nion Swift

Getting started

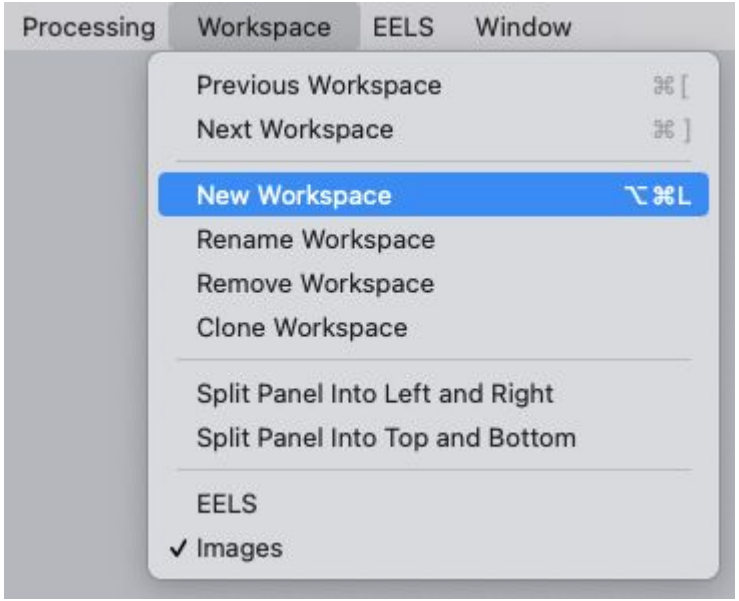
Starting a new project





Create a new project for EELS, the same way you did before.

Using the workspaces



Create a new workspace format called “EELS”. This will allow you to design and customize a different setup for EELS vs imaging data processing.

PARADIM
AN NSF MATERIALS INNOVATION PLATFORM

EELS analysis (lite)

Table of Contents

In this tutorial you will learn:

- Understanding the dimensionality of EELS spectrum images (SIs)
- Identifying elements in a spectrum using online resources
- Preliminary background subtraction and on-the-fly elemental mapping
- Semi-quantitative analysis (line profiles)
- RGB image stacking

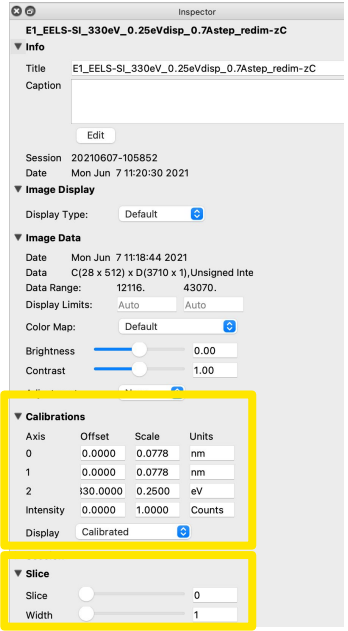
For robust, rigorous, quantitative, and reproducible results, you should use the full EELS analysis notebook in tutorial T4b. But good analysis takes time, and it's useful to have tools that can help you do some quick checks while you are running an experiment or to help you decide which data to focus on. Nion Swift has some nice, basic EELS processing capabilities that we will explore here. Later you will see how, by putting in a little effort, we can improve upon certain aspects of this analysis.

Example 1: STEM-EELS map

Loading the spectrum image (SI)

Load the data file “E1a_SmBaMnO_onDSO_E_y_x.dm3” using “Import Data…” or drag-and-drop (it may take a few minutes to download if you are accessing via cloud-based storage).

Many EELS detectors today have 2048 energy channels.
Our K2 direct detector has 3710!

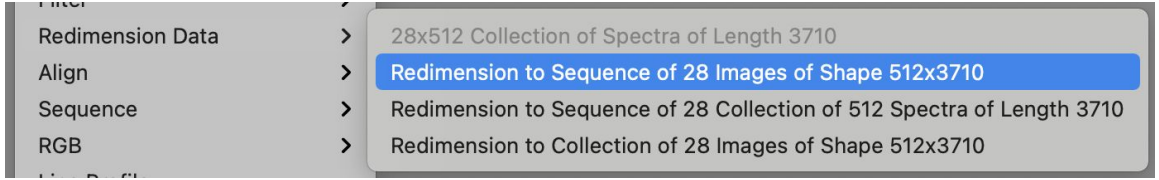


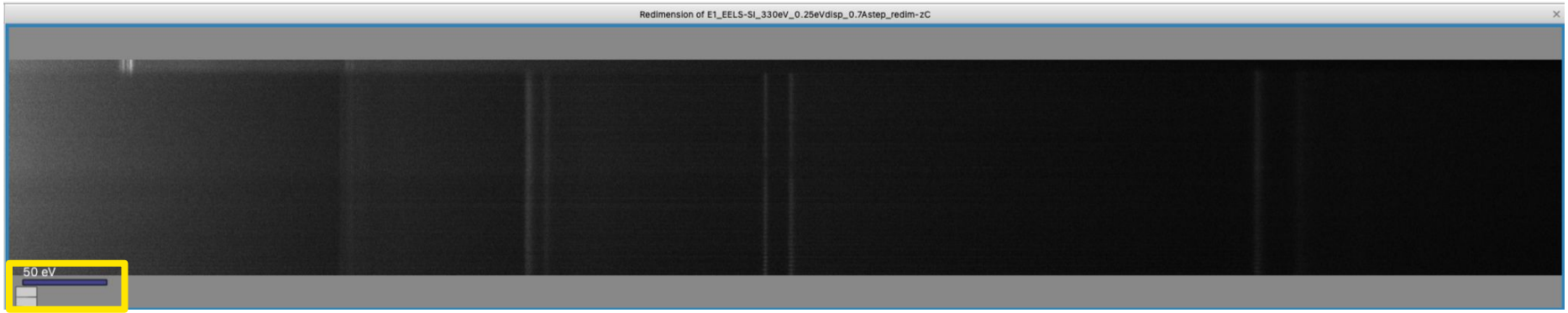
Notice the dimensions of this object in the Data Panel: this is a spectrum image, a three-dimensional data cube with dimensions of *x* (distance), *y* (distance), and *E* (energy loss). Swift can read the metadata from .dm4 and .dm3 files (the most common for EELS), so you’ll notice the units are already calibrated, in this case each *xy* pixel is ~0.8 Å and each energy channel is 0.25 eV (this is the *dispersion)*. CAUTION: these calibrations are handy, but it’s best practice to also have an external record of the spatial and energy units for every file in your notebook, filename, etc.

Scroll through the slices or and see how the image updates. What you are seeing here are two-dimensional planes of energy-loss, conceptually similar to an EFTEM image for a single energy channel.

Example 1: STEM-EELS map

Image registration

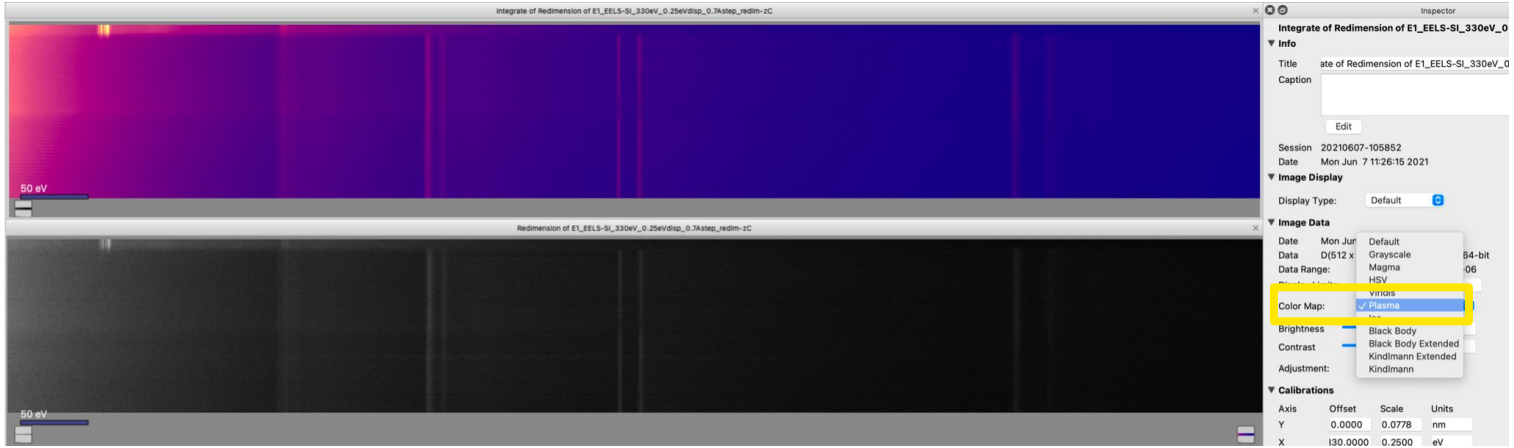




Redimension the spectrum image to a sequence of 28 images of shape 512x3710. The output data item is now a visualization of the energy loss spectrum (note units on the scale bar!) across one spatial dimension of the image. Now, you can clearly see the different bright lines in the spectrum (EELS edges), and perhaps you can see how they trade off in the *x* dimension. Note that this is only displaying a single slice in the *xE* plane: you can scroll through *y* slices using the Sequence index in the Inspector.

Example 1: STEM-EELS map

Redimensioning the data



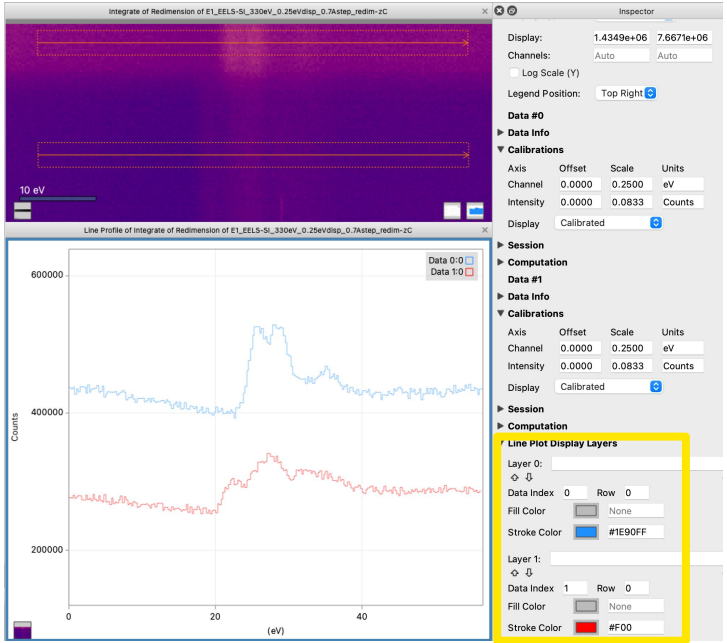
In addition to adjusting the contrast on the histogram, sometimes you might find that different color maps are helpful for revealing subtle variations. Perceptually uniform colormaps (like plasma, magma, and viridis) are usually the best choices.

To improve the SNR, we can integrate across the full depth of this new sequence similar to how we integrated a sequence (stack) of STEM images with “Processing → Sequence → Integrate” (note that because the integral dimension is not orthogonal to the interface, the changes will be correspondingly blurred) in *x*. Find the EELS edge that extends across both sides of the interface and zoom in on the high SNR integrated profile. Although the edge is present on both sides, what changes do you see in the fine structure of the edge? One of the most powerful capabilities of EELS mapping is extracting subtle, spatially localized chemical and bonding differences.

Example 1: STEM-EELS map

Redimensioning the data: spectrum view

Another way to compare between the fine structure changes is to look at the spectrum as a line plot. Make two different line profiles on this edge to compare their characteristic shapes (look back to T2 if you forget how to adjust the line fill and stroke).

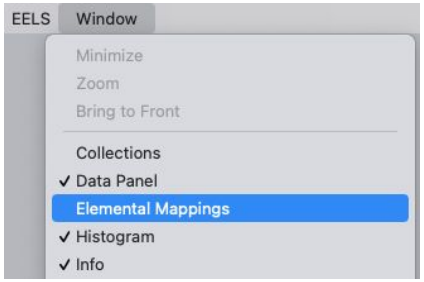


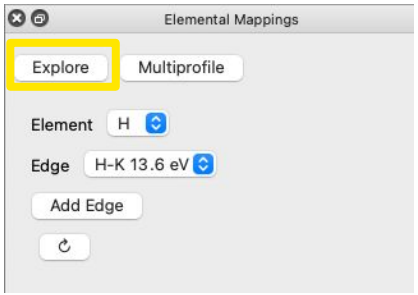
Notice that here, the scale of our line profile now has an arbitrary offset in the energy loss dimension. To map other kinds of variations (e.g., elemental concentration) it might be easier to go back to our original data cube…

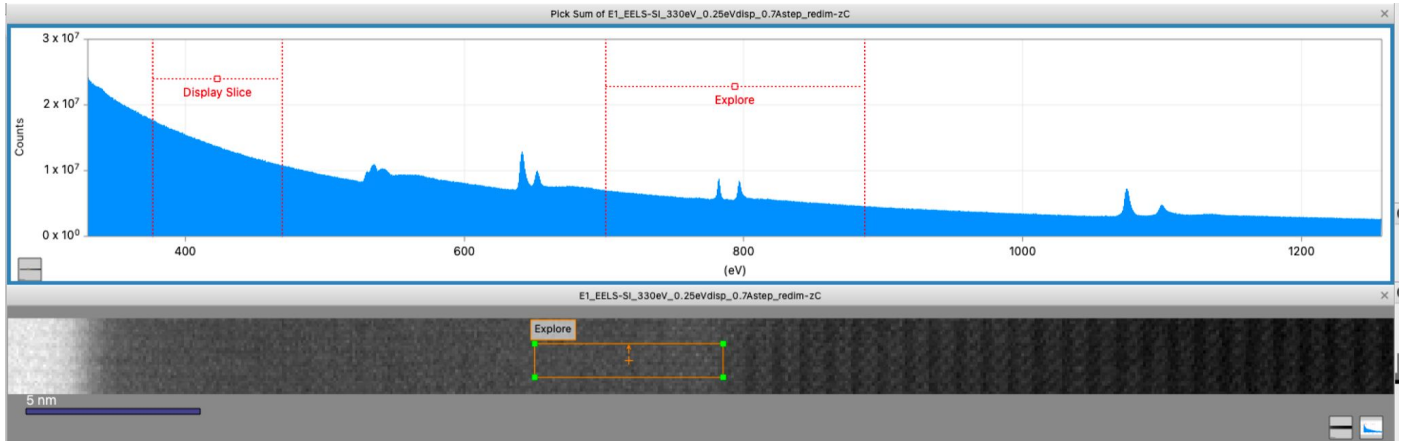
Example 1: STEM-EELS map

Selected area spectrum analysis

Add the “Elemental Mappings” window to your workspace. Select the original spectrum image in the workspace and click “Explore”.



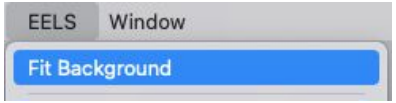




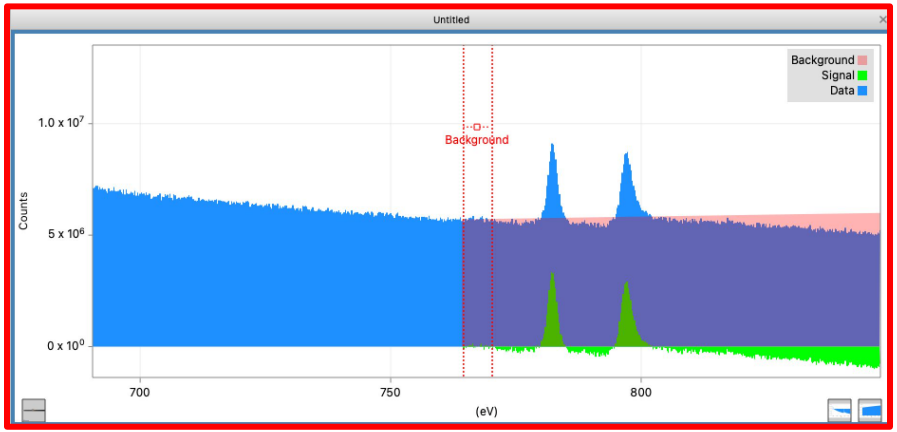
Drag the new Data Item “Pick Sum of…” into the workspace. Drag the small orange box around and see how the spectrum changes. Play around with adjusting the Display Slice and see how the image changes (notice that the spectrum does not!). Use the EELS Atlas (eels.info/atlas) to see if you can identify the elements in each region of the sample (hint: Swift will try to suggest elements for you based on the energy range of the “Explore” slice in the “Elemental Mappings” window).

Example 1: STEM-EELS map

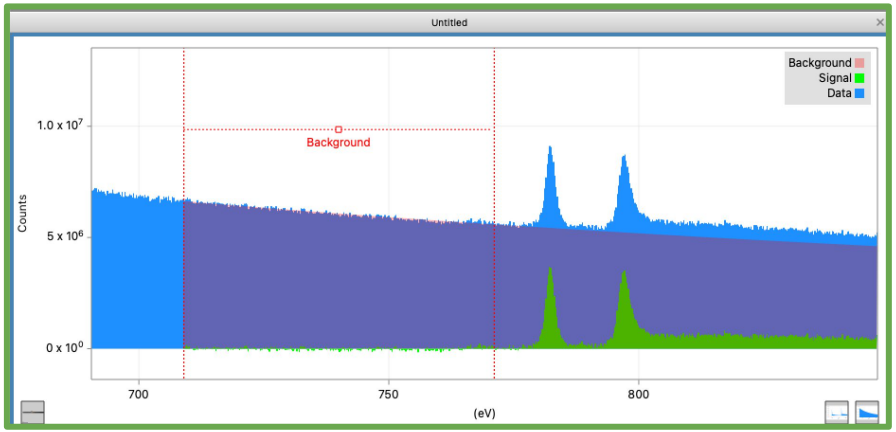
Background subtraction



A bad background fit



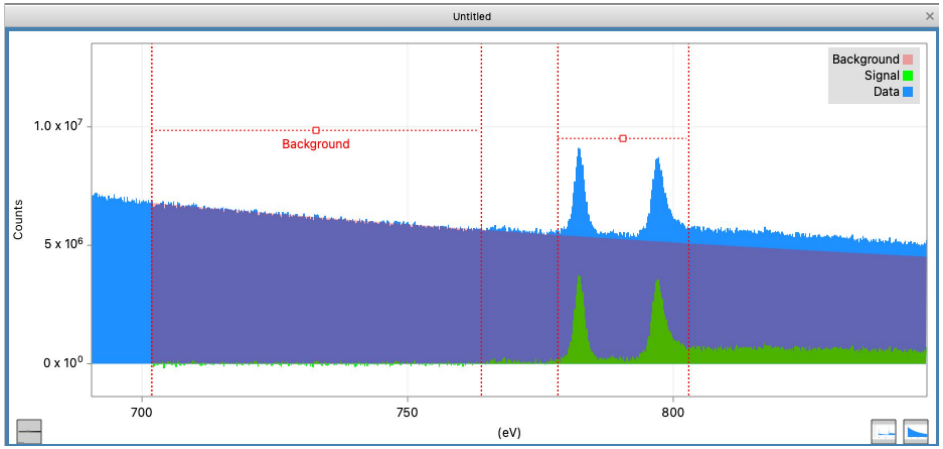
A good background fit

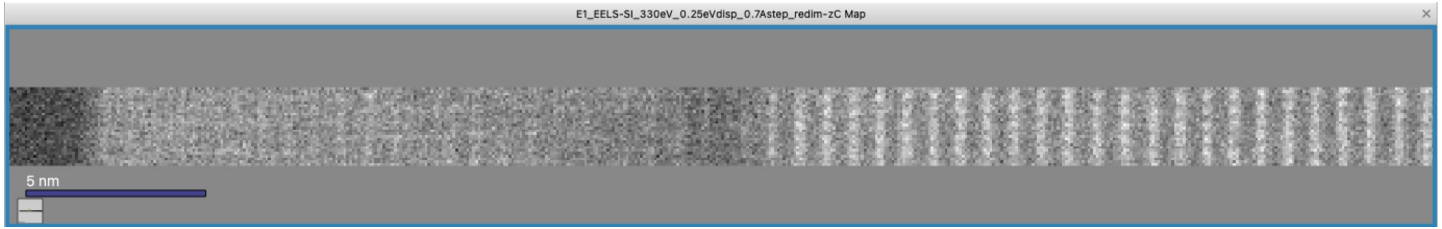


Click and drag to make a new energy window on the Pick Sum spectrum over the background region in front of the sharp peak at ~783 eV. With that energy window selected (solid lines), use “EELS → Fit Background” to model the inelastic power law decay. Play with the bounds of the Background window until you get a good background fit: the “signal” pre-edge should be flat and near 0 intensity within the fit window. Check that the background fit looks good as you move the “Explore” box around.

Example 1: STEM-EELS map

Elemental mapping

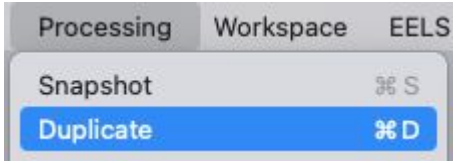


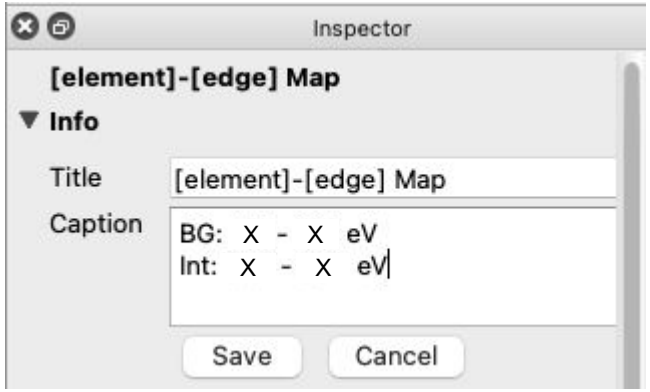


Click and drag on the background subtracted spectrum make a new energy window that includes the two sharp peaks near 783 and 797 eV. With that energy window selected (solid lines), use “EELS → Map Signal” to generate an elemental map. Drag the new Data Item into the workspace and tweak the integration window to get good SNR in the map.

Example 1: STEM-EELS map

Elemental mapping

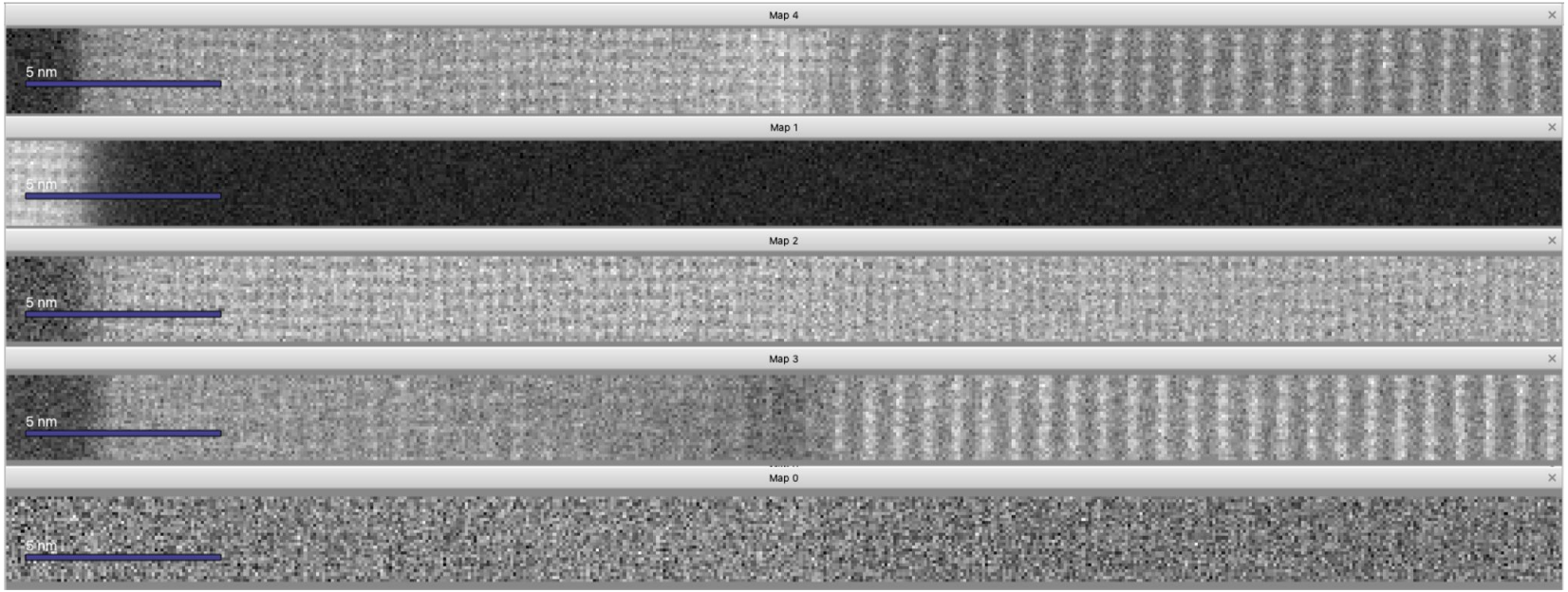




When you are happy with your elemental map, extract a static version using “Processing → Duplicate”. Note that the data item “Clone of…” doesn’t have any of the original item’s dependencies, i.e., it won’t update if you change the parameters in the spectrum. Select the Clone in the Data Panel and rename it (Inspector → Info) with the appropriate element and edge you’ve identified. For your records, add the background and integration window parameters in the Caption.

Example 1: STEM-EELS map

Elemental mapping

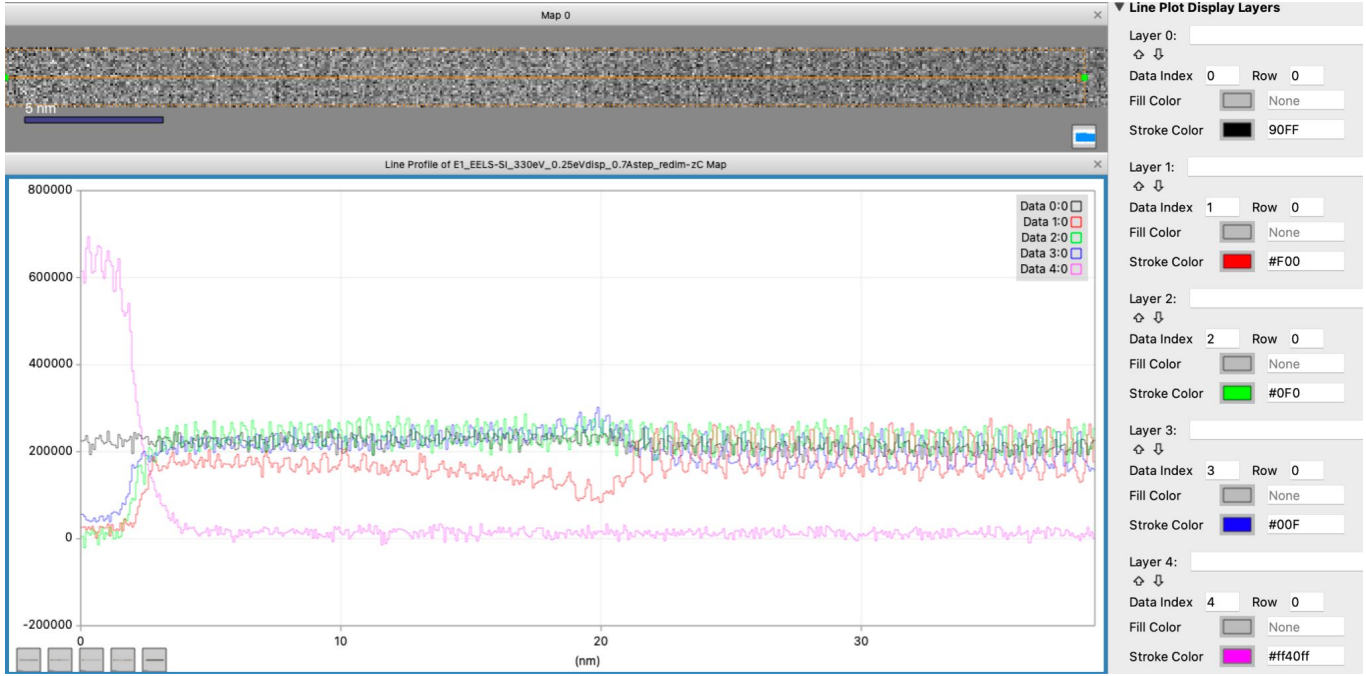


Repeat the above elemental mapping for all the EELS edges you identified. Save a clone of each one, appropriately labeled with element and edge. Remember to carefully check you background subtraction on each side of the interface for every edge!

You should find that 4 elements are strongly localized on either side of the interface, while one appears to be found everywhere.

Example 1: STEM-EELS map

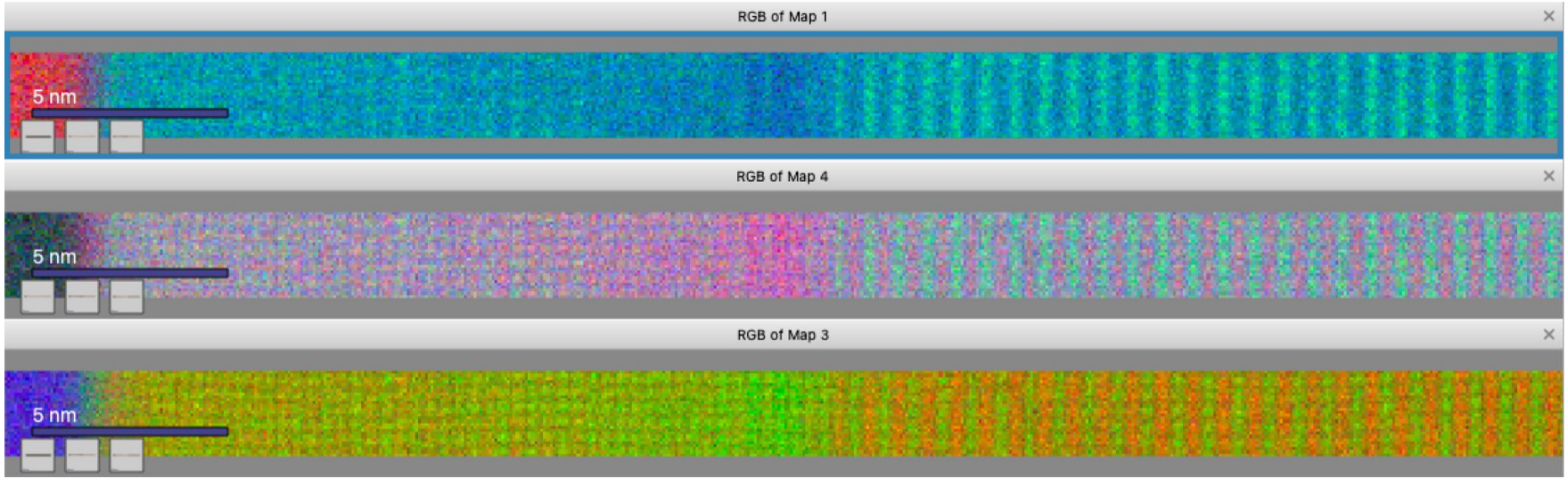
Line profiles (concentration)



At interfaces, we are often interested in possible diffusion or mixing of the elements. Draw a line profile across the interface on one of the elemental maps and set an appropriate integration width. Make identical line profiles on the other three elemental maps and combine all the outputs together into a single line plot. Adjust the legend to reflect the right elements, and recall that you can change the style of the line plot by, for example, putting the color assignment into the “Stroke” cell rather than the “Fill” cell.

Example 1: STEM-EELS map

Stack to RGB



Select three of the maps with localization (shift + click to select multiple items) and use “Processing → RGB → Make RGB”. Pull the RBG map into the workspace. The images will be assigned to red, green, and blue channels (respectively) in the order you select them. Play around with the order and see what version of RGB you think communicates the information from the three channels most clearly. It’s a good idea to rename each one as you go so you can remember which element is which color (e.g., “Red-A, Green-B, Blue-C” where A, B, C are the elements).

Wrapping up

Continue to play around with the EELS data set until you feel comfortable picking energy windows for fit and integration and understand the dimensionality of the SI data cube.

Once you have mapping parameters for all of the edges that you like, make sure you write them down -- they'll be a good starting point for the next part of the tutorial where we will more carefully process this data.

Basic EELS Processing in Python

Once you are satisfied with your mapping parameters, move on to the python notebook labeled '`T4a - Basic EELS Processing.ipynb`'. In this notebook, we will learn how to navigate the same EELS data set in python and improve the quality of our elemental maps created in Nion Swift.

Advanced EELS Processing in Python

Now that we have learned the basics of EELS processing in python, move onto the second python notebook labeled '`T4b - Advanced EELS Processing.ipynb`'. In this second notebook we will explore some more advanced EELS processing techniques such as background subtraction on low count data and Multivariate Curve Resolution (MCR).