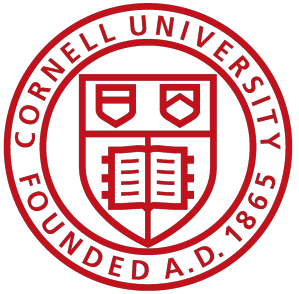


T2: Basic Data Visualization with Nion Swift

Table of Contents

In this tutorial you will learn:

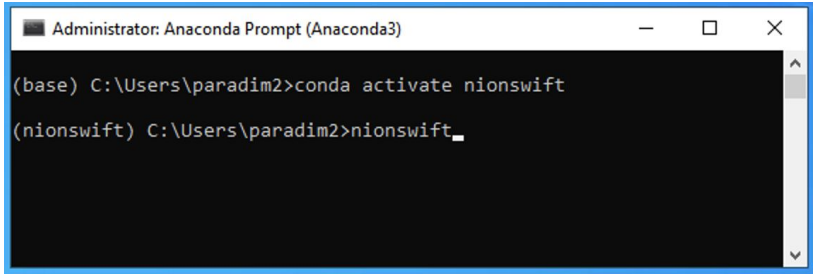
- How to install and launch a GUI for basic data processing, Nion Swift
- How to load EM data into the Swift interface
- Basic* STEM image processing, including:
  - Image calibration
  - Line profiles
  - Fourier techniques
  - Basic* image stack registration
  - Image crop, rotation

* these techniques are useful for quick data screening or on-the-fly processing, but full quantitative data analysis (image registration, filtering, EELS analysis) should be performed externally (e.g., Tutorials T3 and T7).

Getting started

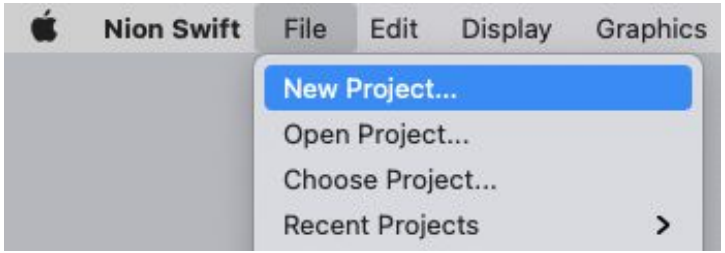
Launching Nion Swift on the VM

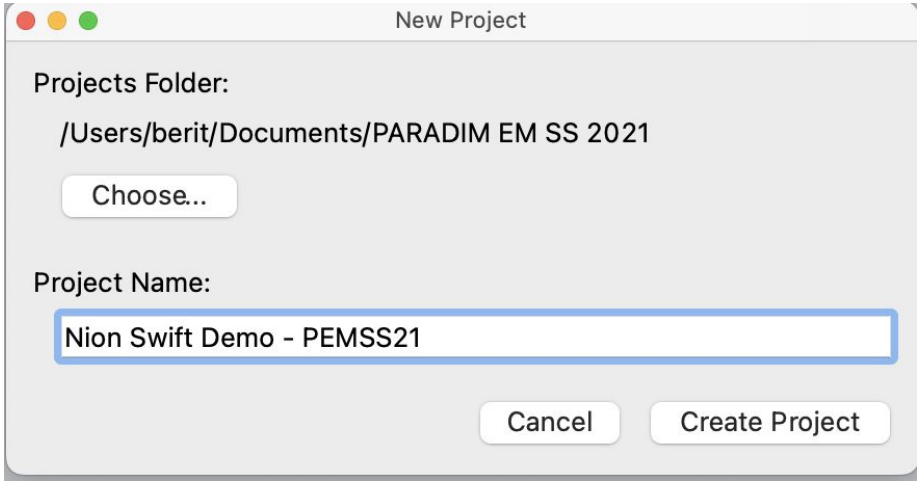
Log into your VM and launch the terminal. Enter '`conda activate nionswift`' to activate the swift python library environment. Then enter '`nionswift`' to launch the program. It might take 1-2 minutes to launch.



If you want to install Nion Swift on your own computer for later use, see the instructions at the end of the tutorial. Note that you will also need to install some extra plugins for EELS or 4D-STEM analysis.

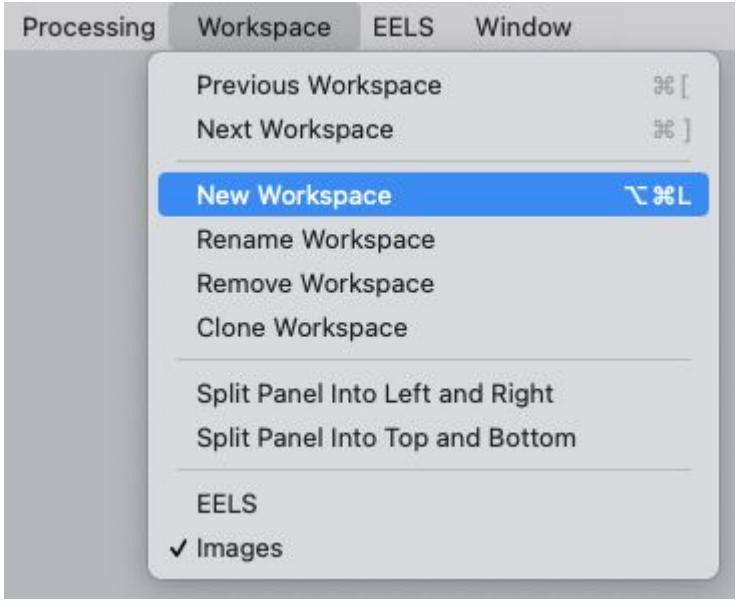
Starting a new project





Create a new project (you will be automatically prompted the first time you launch Nion Swift). It's a good idea to save them in a consistent location with descriptive names so you can easily go back to your work later. The projects will save automatically as you go, so you shouldn't need to worry about saving them manually as you work.

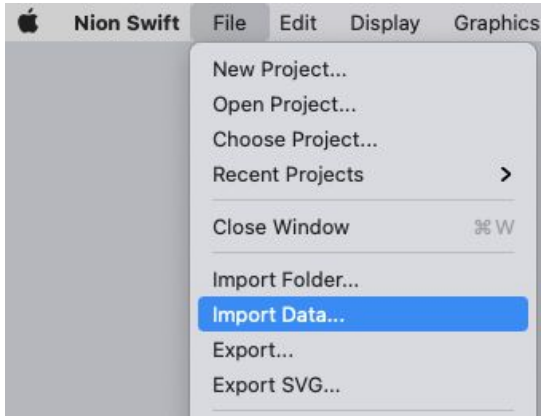
Using the workspaces

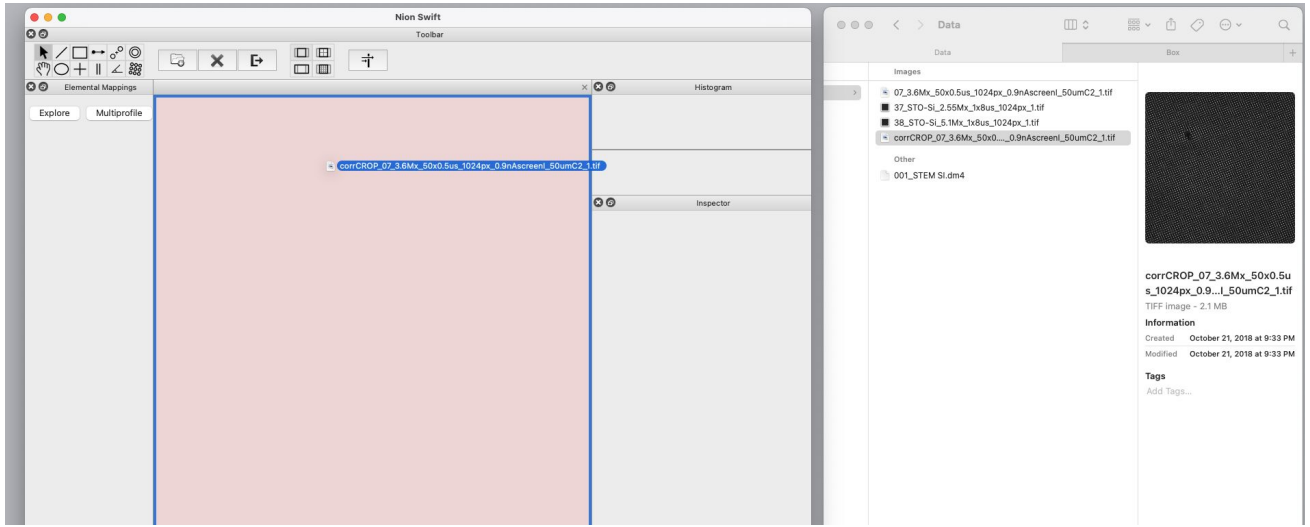


Nion Swift operates in a workspace format, so you can pull in multiple data sets, manipulate them, and save your work as you go. You can create multiple workspaces and switch between them within a single project (more on that later).

For now, create a new workspace.

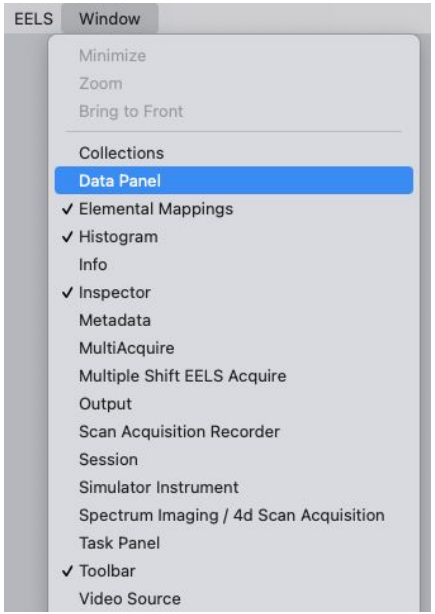
Opening a data file

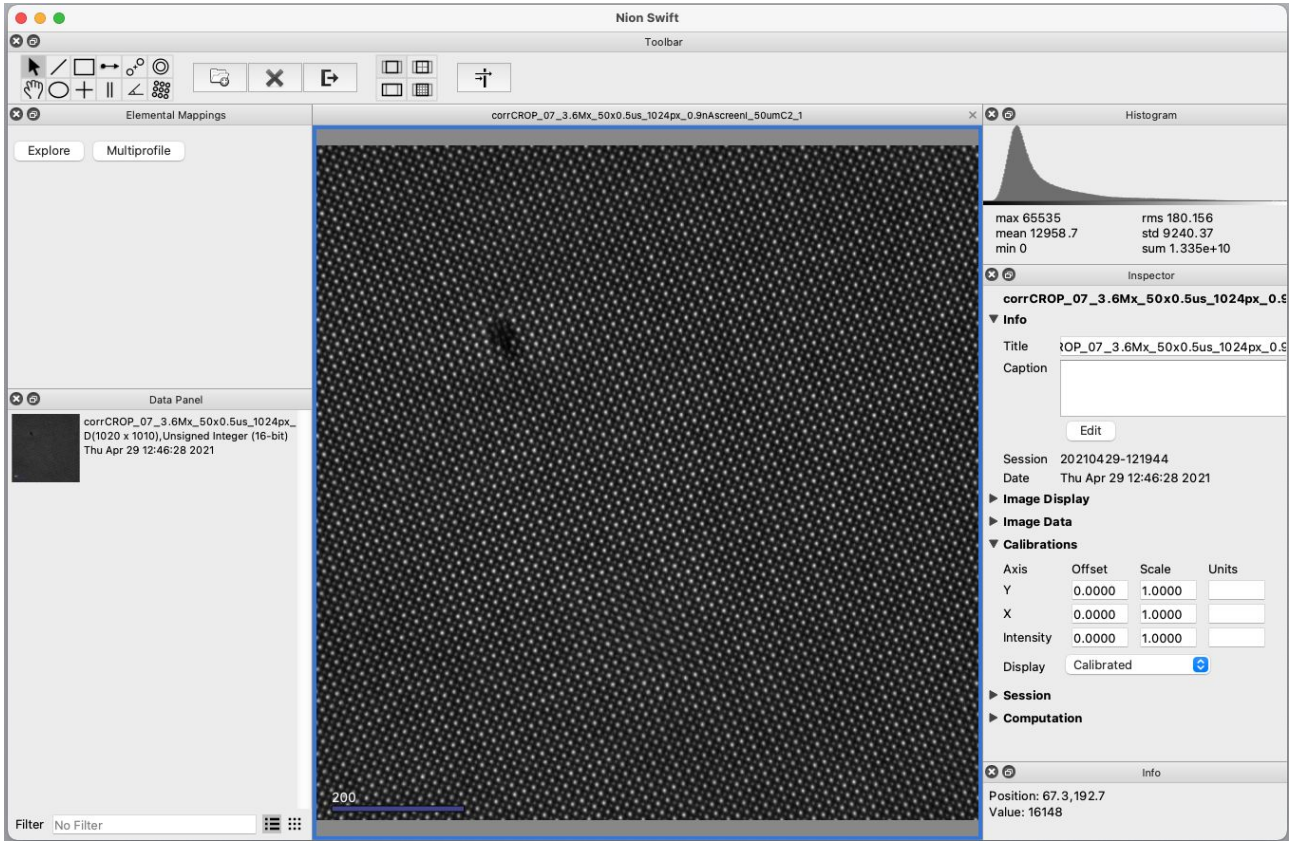




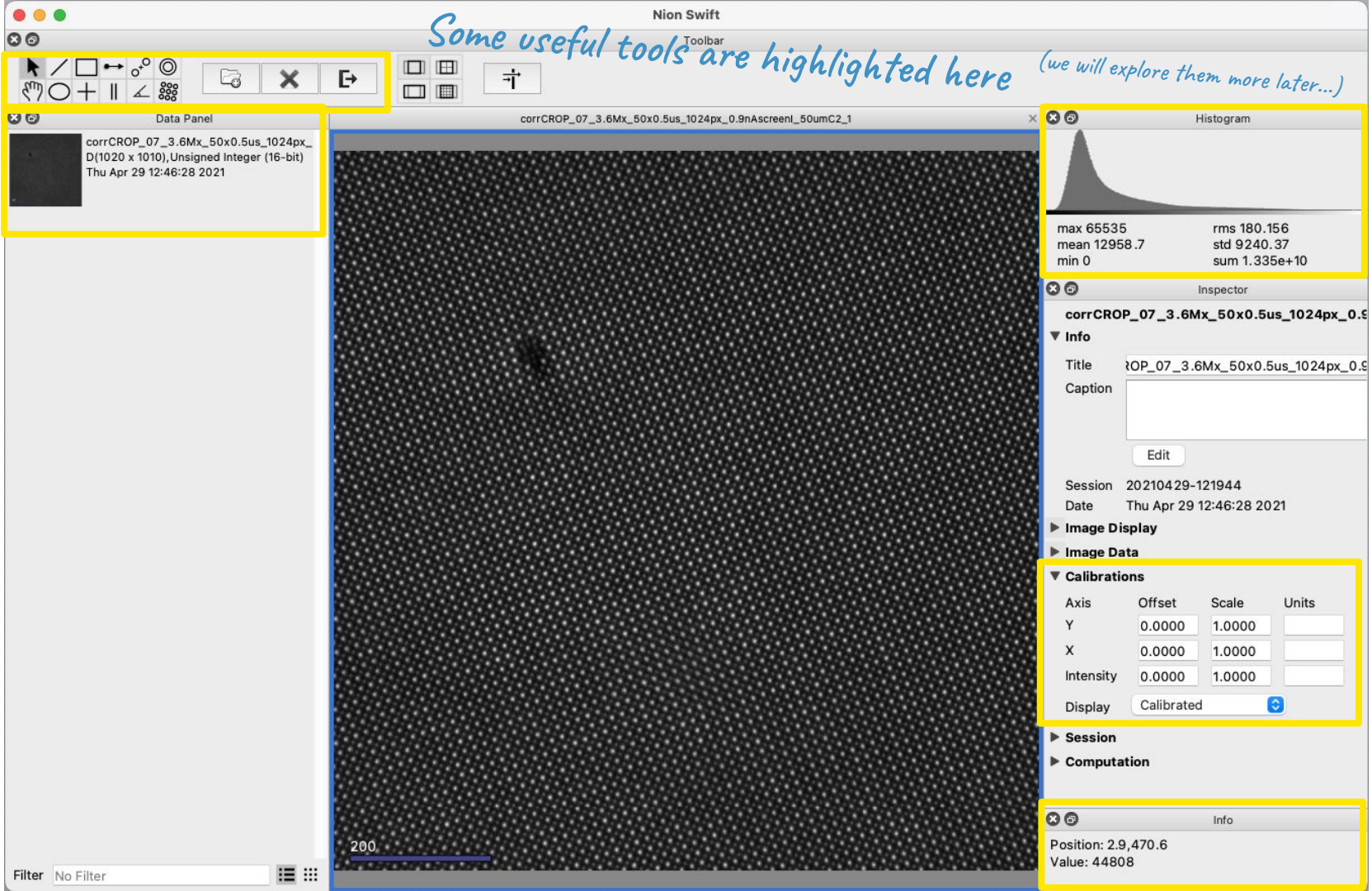
Load the data file “corrCROP_07_3.6Mx_50x0.5us_1024px_0.9nAscreenI_50umC2_1.tif” using “Import Data…” or by drag-and-dropping the file from your computer folder into the workspace. Swift will highlight an area in red where the new data item is going to “land” (but don’t worry, you can always move it around later).

Opening a data file





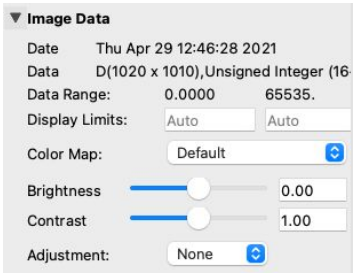
Open the “Data Panel” to see/browse all the items you’ve loaded into this project. Their dependents will also appear here.

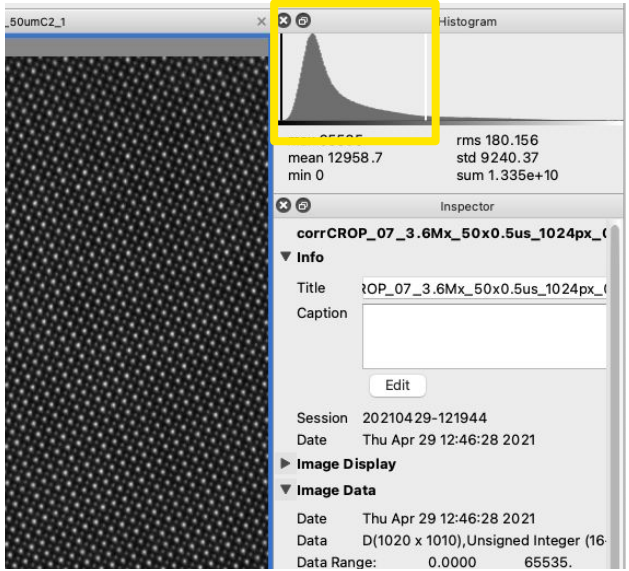

Nion Swift
Toolbar
Some useful tools are highlighted here
(we will explore them more later...)
Data Panel
corrCROP_07_3.6Mx_50x0.5us_1024px_
D(1020 x 1010),Unsigned Integer (16-bit)
Thu Apr 29 12:46:28 2021
corrCROP_07_3.6Mx_50x0.5us_1024px_0.9nAscreenI_50umC2_1
200
Filter
No Filter
Histogram
max 65535
mean 12958.7
min 0
rms 180.156
std 9240.37
sum 1.335e+10
Inspector
corrCROP_07_3.6Mx_50x0.5us_1024px_0.9
Info
Title
ROP_07_3.6Mx_50x0.5us_1024px_0.9
Caption
Edit
Session 20210429-121944
Date Thu Apr 29 12:46:28 2021
Image Display
Image Data
Calibrations
Axis Offset Scale Units
Y 0.0000 1.0000
X 0.0000 1.0000
Intensity 0.0000 1.0000
Display Calibrated
Session
Computation
Info
Position: 2.9,470.6
Value: 44808

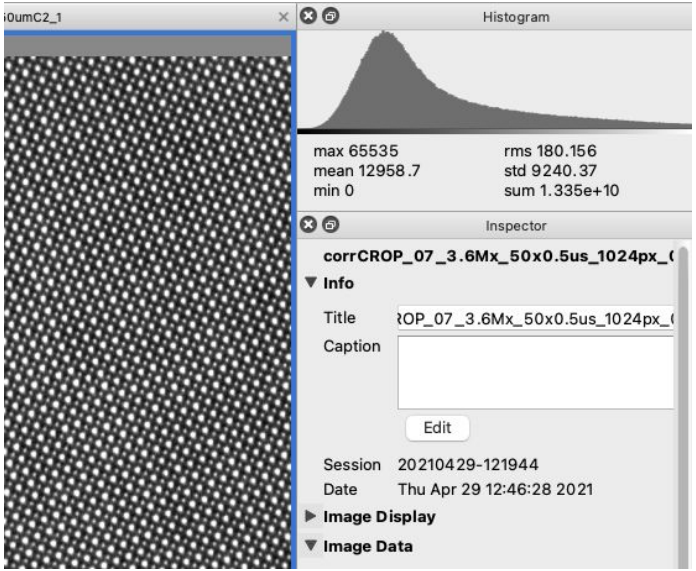
Measuring and adjusting

Example 1: STEM lattice image

Brightness, contrast, cropping







Use the “Brightness” and “Contrast” sliders in the Image Data drop down (in the Inspector window) to adjust the image.
Better practice: Directly crop on the image histogram by clicking and dragging over the range you want to limit. Double click on the bottom of the histogram to reset it.

Example 1: STEM lattice image

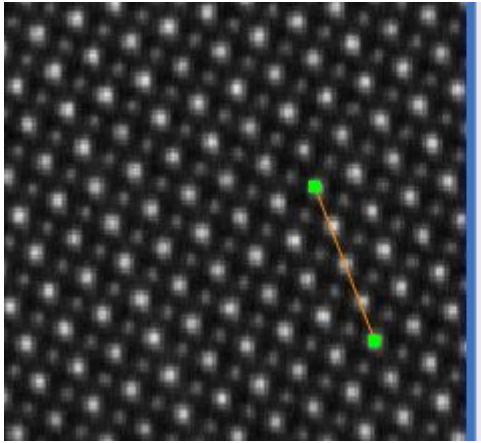
Scale calibration

Pro tip: you can easily zoom in and out on images with the plus and minus buttons on your keyboard

Calibrations

Axis	Offset	Scale	Units
Y	0.0000	0.0254	nm
X	0.0000	0.0254	nm
Intensity	0.0000	1.0000	

Display: Calibrated



Graphics

Line: None

X0	21.390 nm	Y0	16.497 nm
X1	21.950 nm	Y1	17.934 nm
L	1.54245 nm	A	-68.5220°

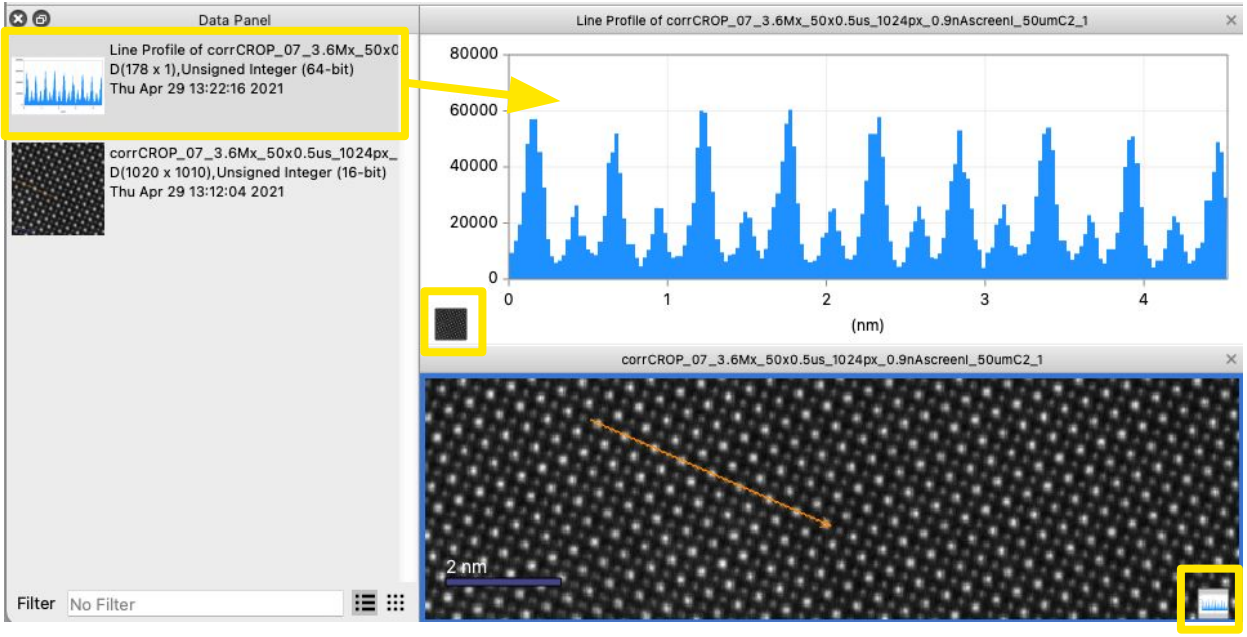
Display: Calibrated

If you know the image calibrations, you can input them directly here. Best practice is to calibrate off a known value in the image (e.g., lattice constant), see Slides 20-21 for how to extract this from the FFT. For now, use the values shown above.

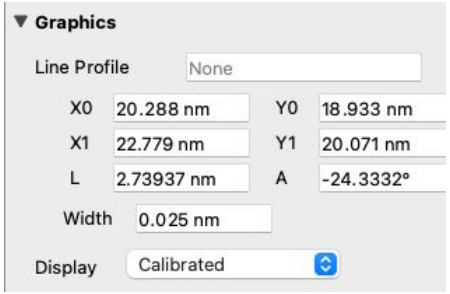
Once the image is calibrated, you can use the Line tool to measure distances. Here we can confirm that our calibration worked: the distance across 4 unit cells is 1.54 nm (a = 0.386 nm). Note you can easily change between calibrated and pixel values.

Example 1: STEM lattice image

Line profiles



You can also manually set the position and integrated width of the line profile in either pixels or calibrated units.



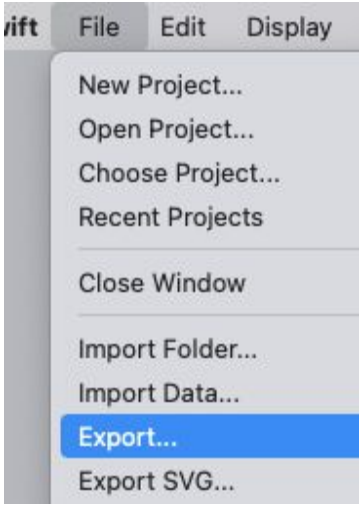
Use the Line Profile tool to analyze image intensity across a specific region. The measurement output (intensity line profile) will appear as a new item in the Data Panel. You can drag and drop it into the workspace. Swift will keep track of the data item parents and dependents. An icon of the parent data will appear in the bottom left corner, icons of the dependent data items will appear in the bottom right. You can right click on these icons to easily “Reveal in Data Panel” or go to them directly within the workspace.

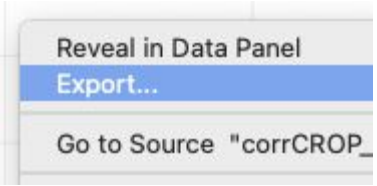
📍 Checkpoint!

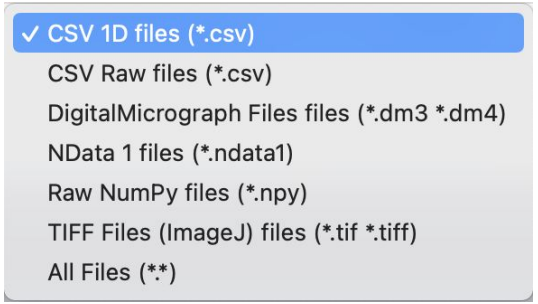
About how much brighter are the heavy atoms (A-sites) in this image than the lighter atoms (B-sites)?

Example 1: STEM lattice image

Exporting data items





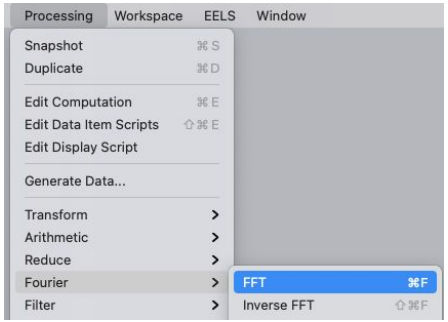


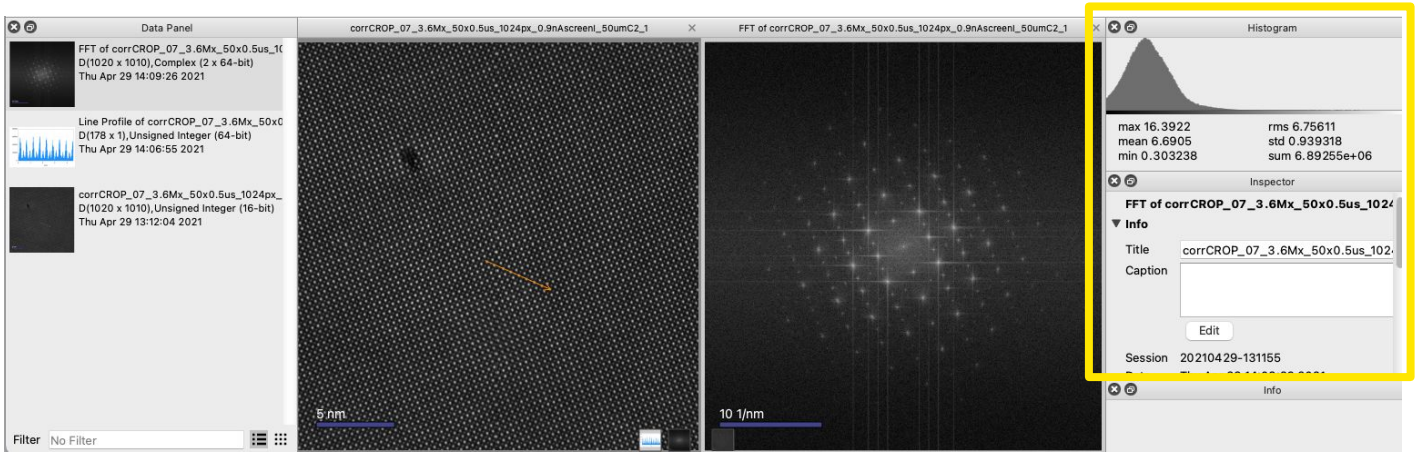
You can export data items from “File → Export…” or by right-clicking on the data and “Export…”. A variety of export formats are supported. You can also export display items as scalable vector graphics (SVG), which can be handy when preparing figures or other presentations.

Fourier-based analysis

Example 1: STEM lattice image

Fourier analysis



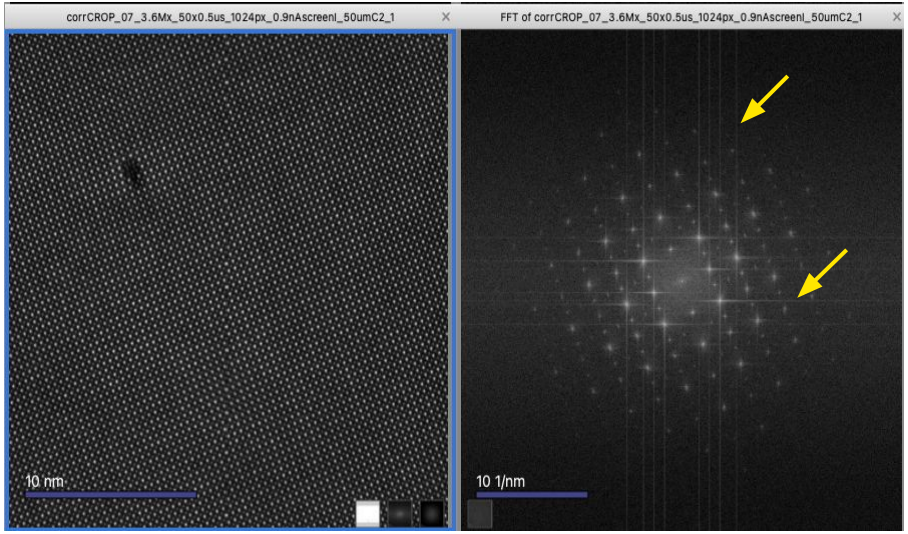


The Fast Fourier transform (FFT) is a very helpful tool for many kinds of image analysis. Take the FFT of an image using “Processing → Fourier → FFT”. The FFT will appear in the Data Panel and as another dependent of the original image. Drag and drop the FFT into the workspace. Notice that the Inspector and Histogram windows update for whichever data item you have selected.

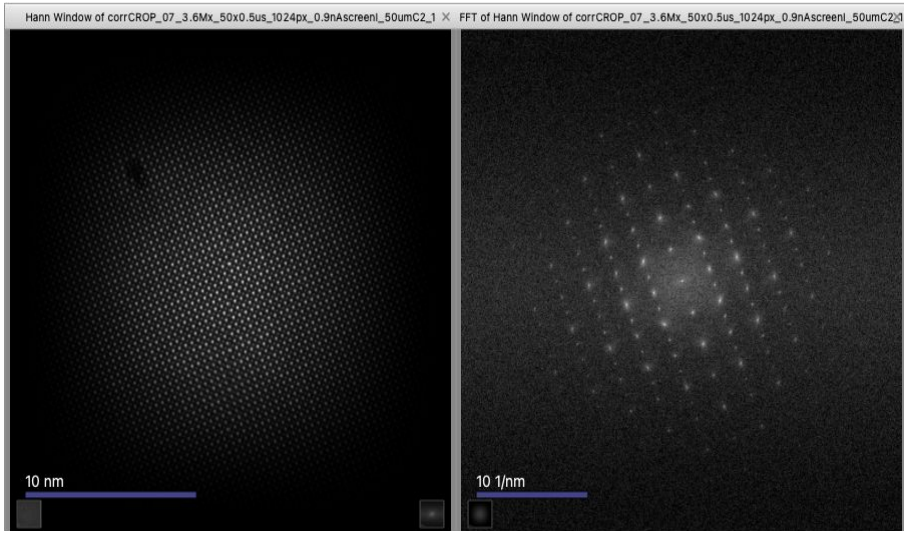
Example 1: STEM lattice image

Fourier analysis: Hann window

Original image and FFT



Hanned image and FFT



The vertical and horizontal streaking in the FFT are caused by the abrupt image boundaries. One way to reduce those contributions is to first apply a Hann window (“Processing → Fourier → Hann Window”) to damp out the edges of the image and then take the FFT.

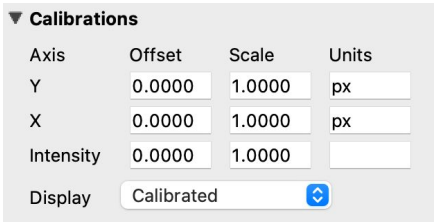
PARADIM
AN NSF MATERIALS INNOVATION PLATFORM

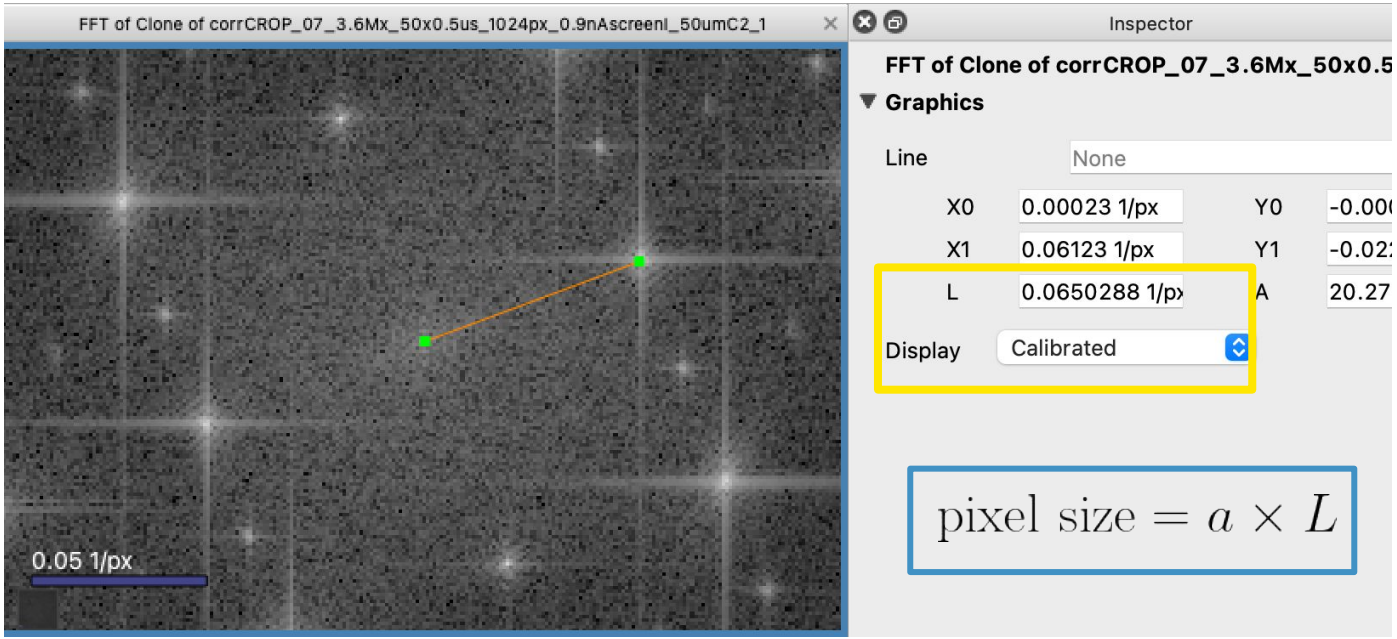
📍 Checkpoint!

Can you think of any potential disadvantages to a Hann-windowed FFT?

Example 1: STEM lattice image

Fourier analysis: pixel calibration (method 1)

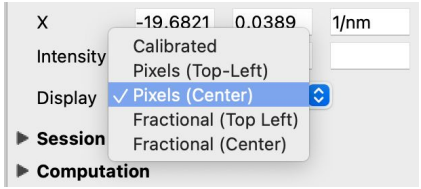


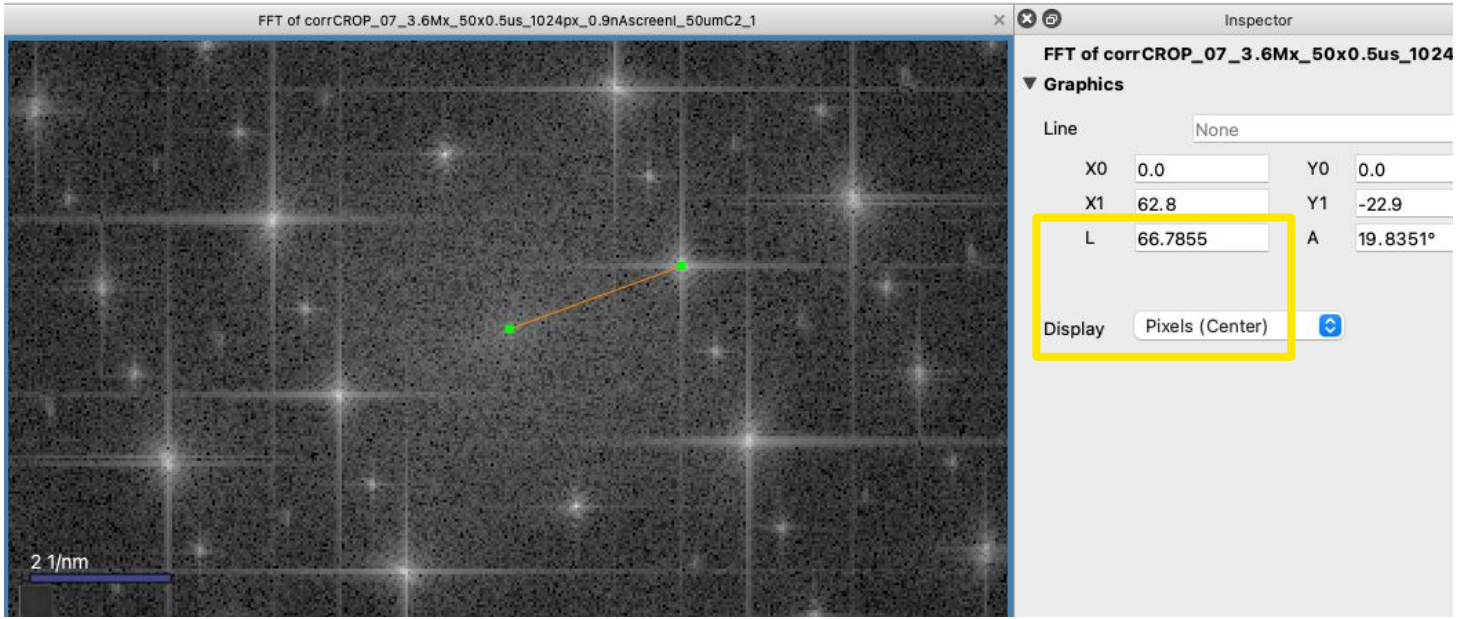


We can use the FFT to precisely calibrate the pixel size of a STEM image of a crystal with known lattice spacing. In the Inspector → Calibrations, reset the X and Y scale to be 1.00 px each (this would be the default if you loaded an image that had not yet been calibrated). Draw a line from the center spot out to a Bragg peak with known lattice spacing. The length of the line should have units of 1/px (if not, change the Display to "Calibrated"). To get the calibrated pixel size, multiply the known spacing of that peak *a* (in Å or nm) by the length of the line *L* (in inverse pixels). Recall that for this material *a* = 0.386 nm. Go back to the STEM image and update your calibration as before.

Example 1: STEM lattice image

Fourier analysis: pixel calibration (method 2)





If you want to directly override a different set of calibrations, go to the Inspector → Calibrations and change the Display to Pixels (Center). Draw a line from the center spot (you can manually input X0 and Y0 to be more precise) to a primary Bragg peak and write down the measured length (L). This time, you need to account for the total image size, so divide this length by the pixel dimensions of your image (in this case use the *x* dimension: 1020). Then use the same formula before with the correctly scaled *L*. Check that both methods give you about the same value we used previously.

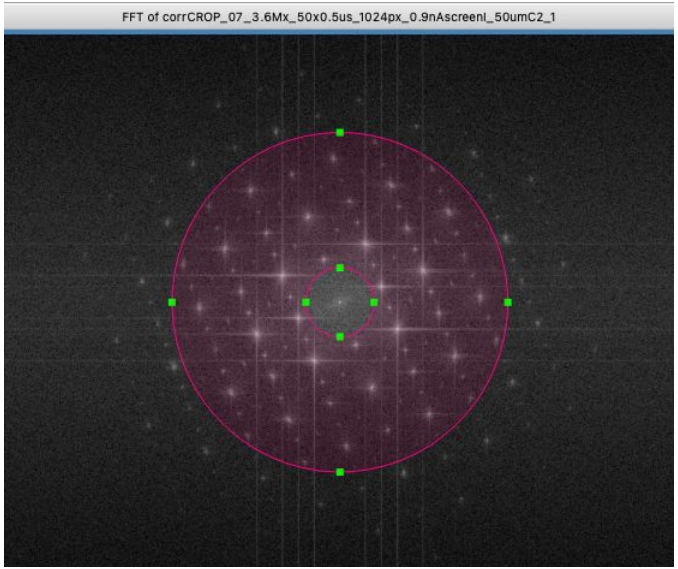
📍 Checkpoint!

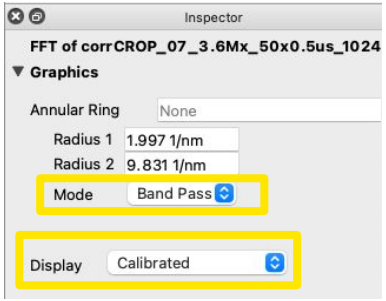
Do both methods of scale calibration give you the same pixel size value?
Does it agree with what you were given at the beginning of the tutorial?

Example 1: STEM lattice image

Fourier analysis: Fourier filtering

By filtering different parts of the information in the FFT, we can highlight different aspects of the original image. For example, using high-, low-, or band-pass filters can be used to increase or limit the contributions of the lattice or background variations.





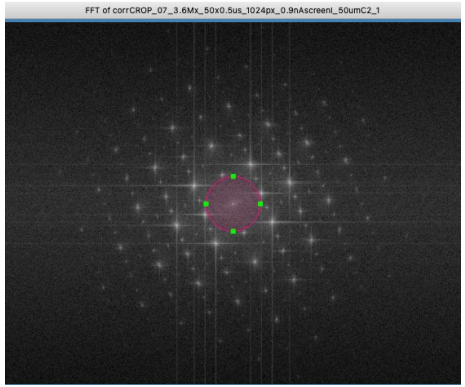
To apply a Fourier filter, click on the image FFT (raw) and go to “Processing → Fourier → Add Band Pass Filter”. The pink overlay will show you which frequencies in the FFT will get used; click and drag on the green squares to adjust the size. You can toggle between low, high, and band-pass filters by selecting the filter overlay and changing the “mode” in the Inspector. Notice that you can also toggle between calibrated units and pixels. How do the (calibrated) radius values change as you make the inner circle smaller and larger (hint: think about converting the calibrated units from 1/nm to nm)? Understanding how a Fourier mask size relates to real space dimensions is incredibly important!

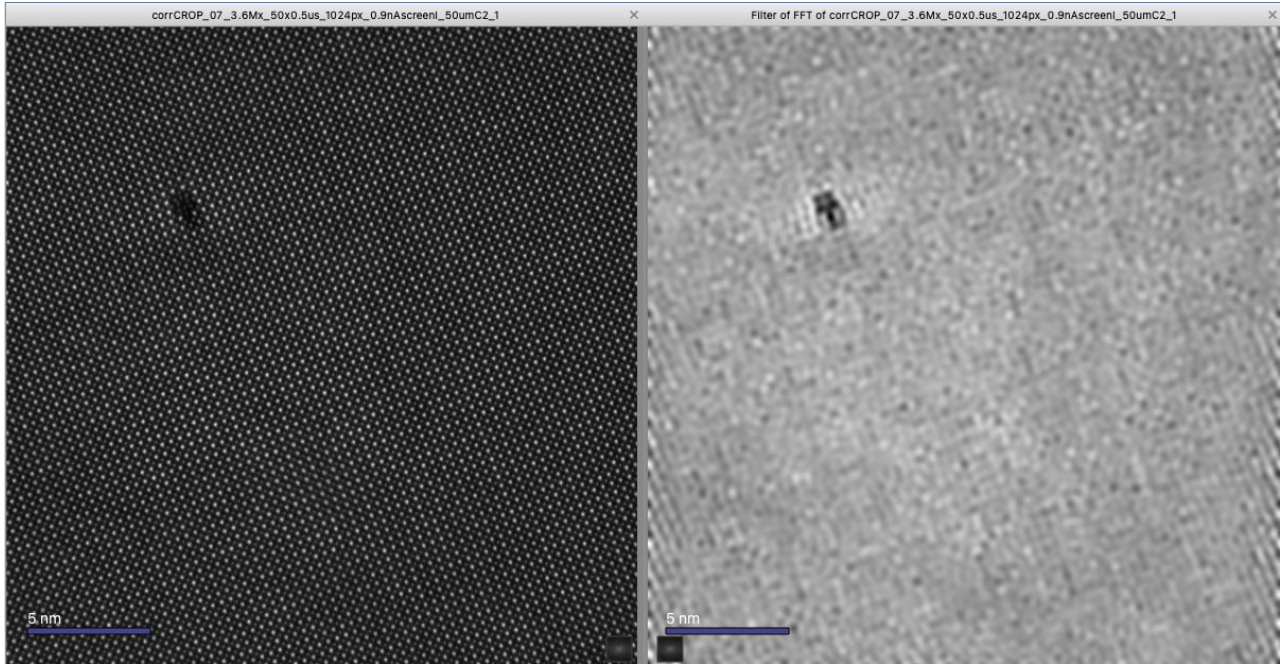
⚠️ A note on nomenclature

In the following slides, we use the designations for 'low-pass' and 'high-pass' filters that match Nion Swift. ***Here, you can think of the 'high-pass' (low-pass) filter as ignoring (passing-by) the high frequencies and keeping only the low (high) frequencies.*** *Traditionally, the opposite convention would be used, such that a high-pass (low-pass) filter lets through / keeps the high frequencies and blocks the low (high) frequencies.*

Example 1: STEM lattice image

Fourier analysis: Fourier filtering

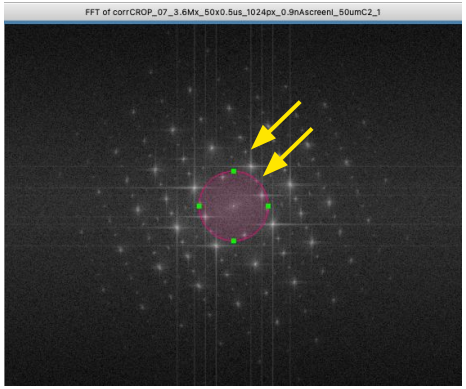


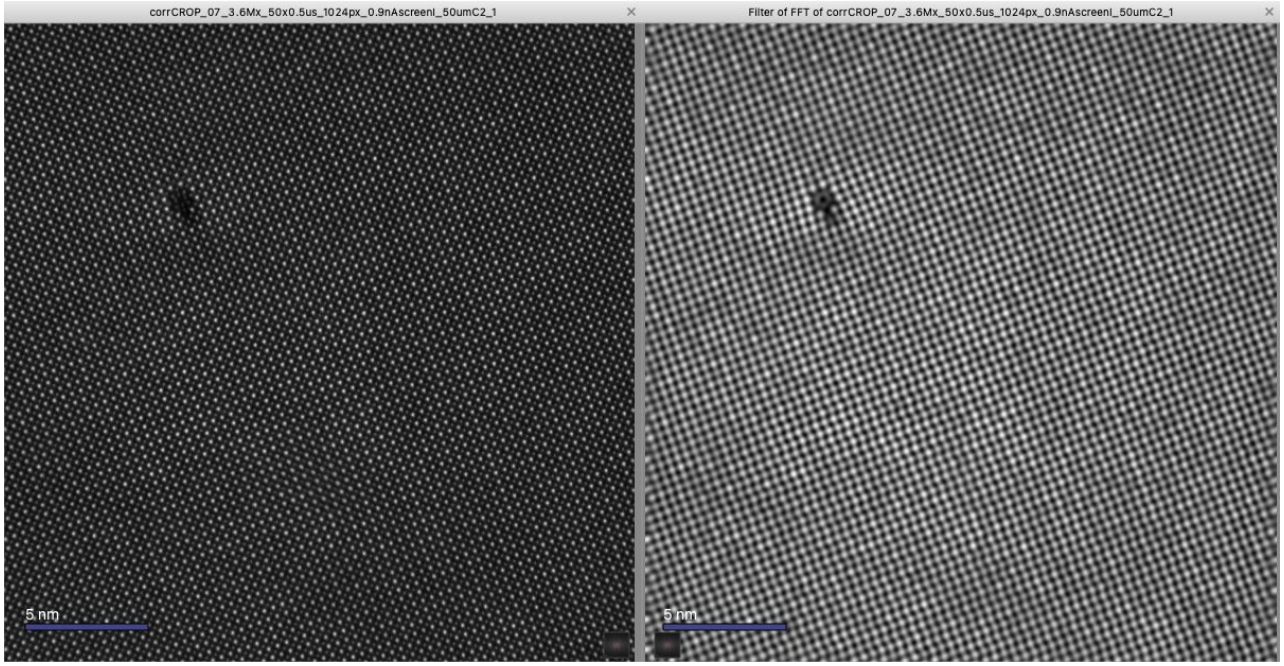


Create a **high-pass filter** and set the inner diameter just inside the first set of primary peaks in the FFT. Use “Processing → Fourier → Fourier filter” add drag the output into the workspace next to the raw STEM image. Notice that all the periodic (e.g., lattice) information seems to have been removed from the image. The Fourier filter has suppressed the higher frequency information, which in this case we have set to include all the bright crystalline peaks.

Example 1: STEM lattice image

Fourier analysis: Fourier filtering

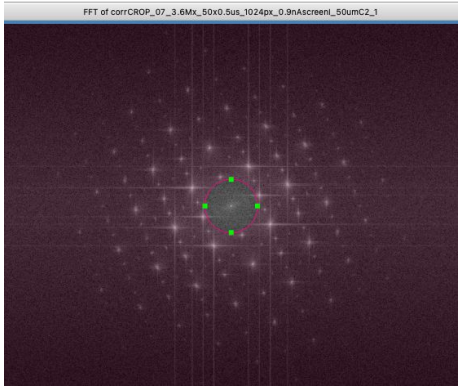


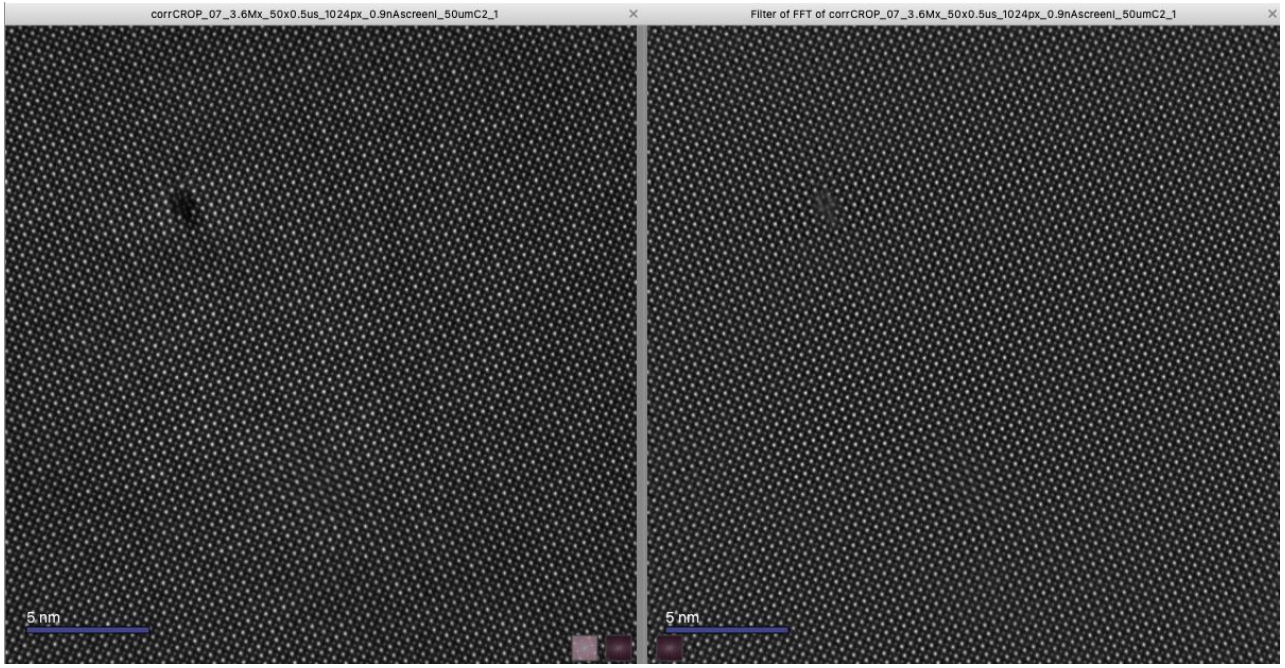


Adjust the inner diameter to just outside the first peaks and watch how the filtered image updates. Now we can see only the most fundamental periodicity in the image, which in this case highlights the A-sites of each unit cell. Depending on how carefully you set the inner diameter of your filter, you may notice an additional left-right zigzag ordering in the A-site rows of the filtered image. How should you precisely adjust your filter to either include or exclude these distortions? Slowly expand the diameter of your frequencies and watch how the output changes. What kind of filter do you need to also see the B-site columns in the output?

Example 1: STEM lattice image

Fourier analysis: Fourier filtering

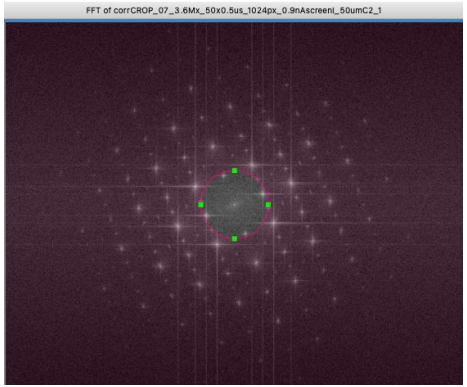


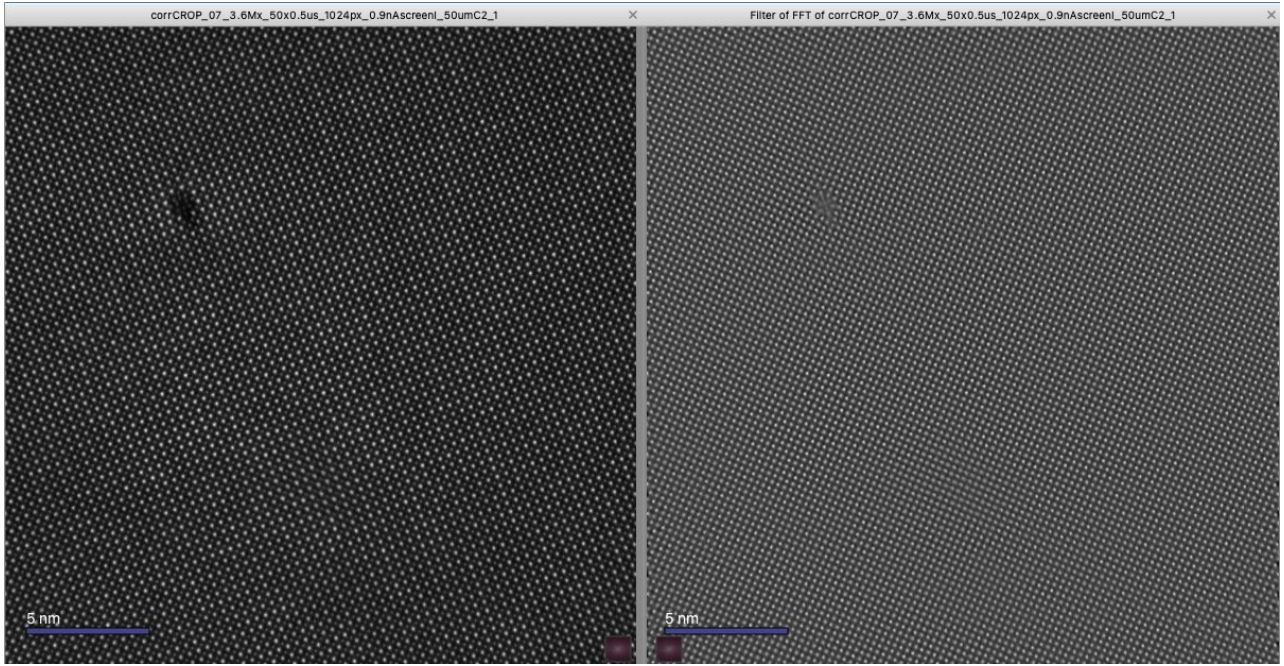


Change to a **low-pass filter** and set the inner diameter just inside the first set of primary peaks in the FFT. Notice that main periodicities of the image (e.g., the lattice) are preserved, but that the larger-scale variations in background have been suppressed in the filtered image. In this case, those contrast variations arise largely from the mixed cation doping, so filtering out those contributions has actually *removed* potentially useful information!

Example 1: STEM lattice image

Fourier analysis: Fourier filtering



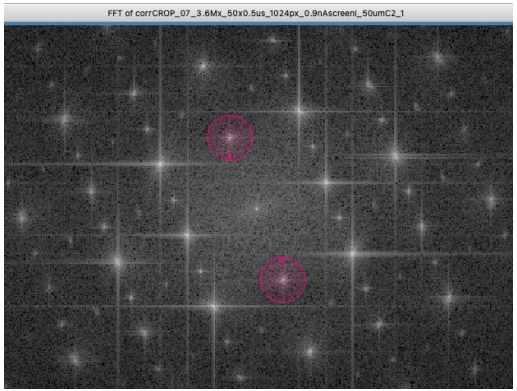


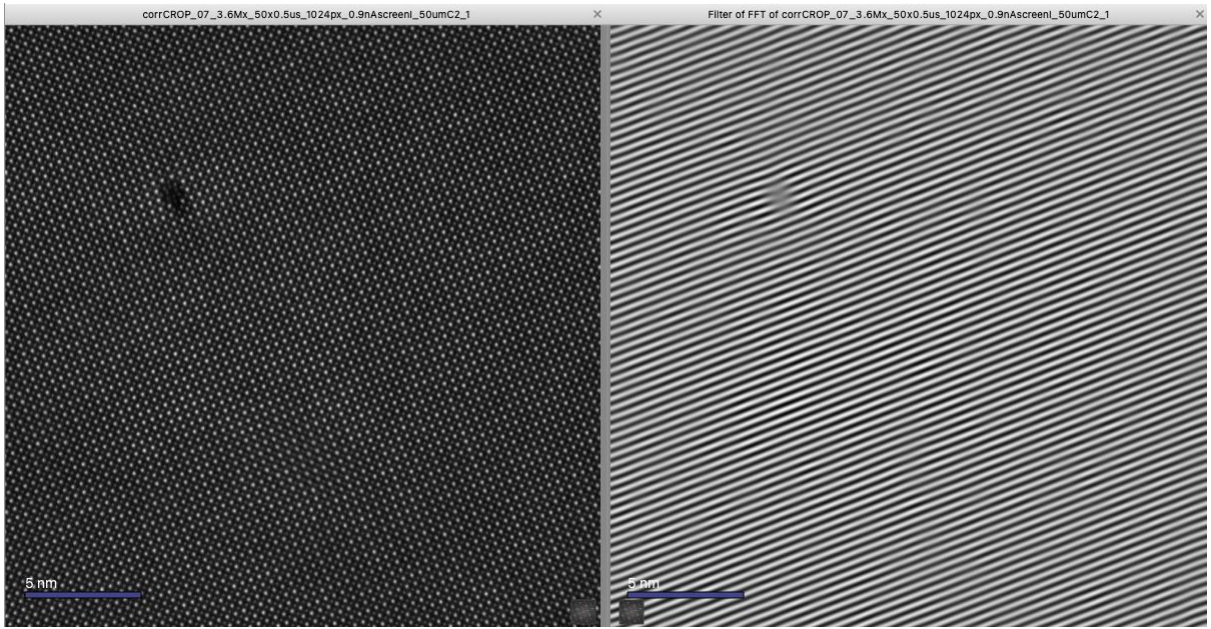
Adjust the inner diameter to just outside the first peaks and watch how the filtered image updates. Now, the A- and B-site columns have almost the same contrast! Can you explain that?

Fourier filtering is very dangerous if you don't take the time to think carefully about what you're doing to your image.

Example 1: STEM lattice image

Fourier analysis: Fourier filtering

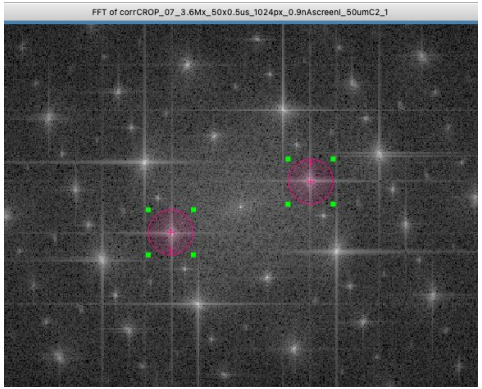


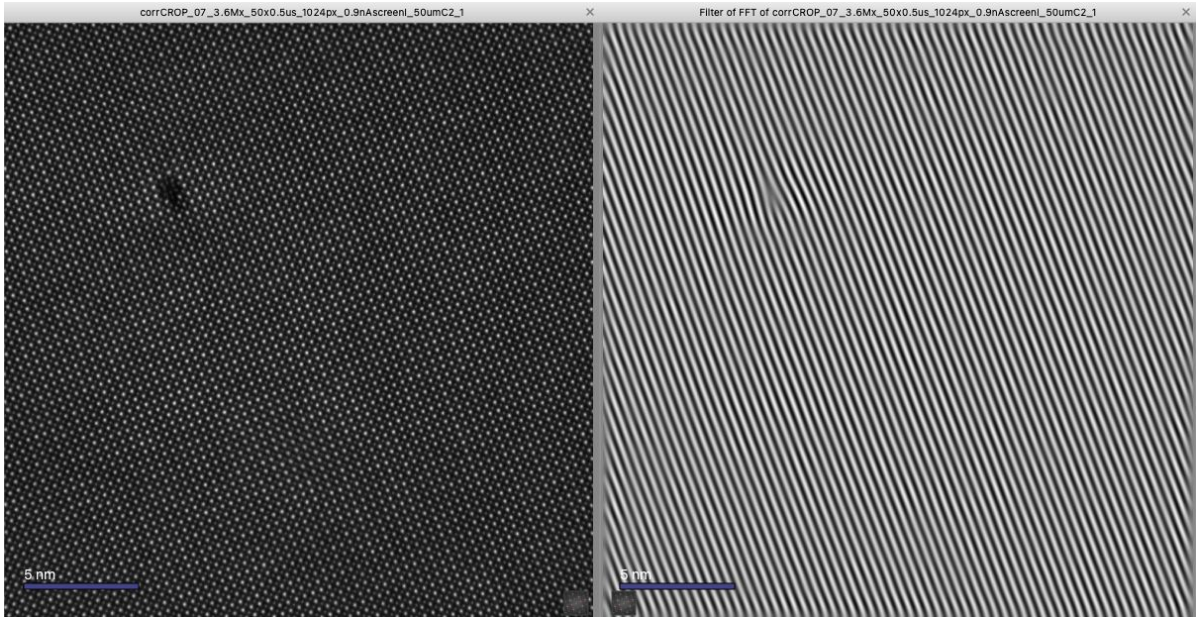


Click on the pink filter and delete it. Go to “Processing → Fourier → Add Spot Filter”. Drag and re-scale one of the spots to cover the primary [010] spot without overlapping any of the other peaks or streaks in the FFT (Hint: you only have to move one spot and the other will automatically follow symmetrically. Hold “shift” while re-sizing the spot to keep it circular). The output should automatically update, but if not, use “Processing → Fourier → Fourier Filter” to regenerate a fresh one. Think about the result: what are we actually mapping with this type of Fourier filter? How could this be useful for quantum materials analysis?

Example 1: STEM lattice image

Fourier analysis: Fourier filtering

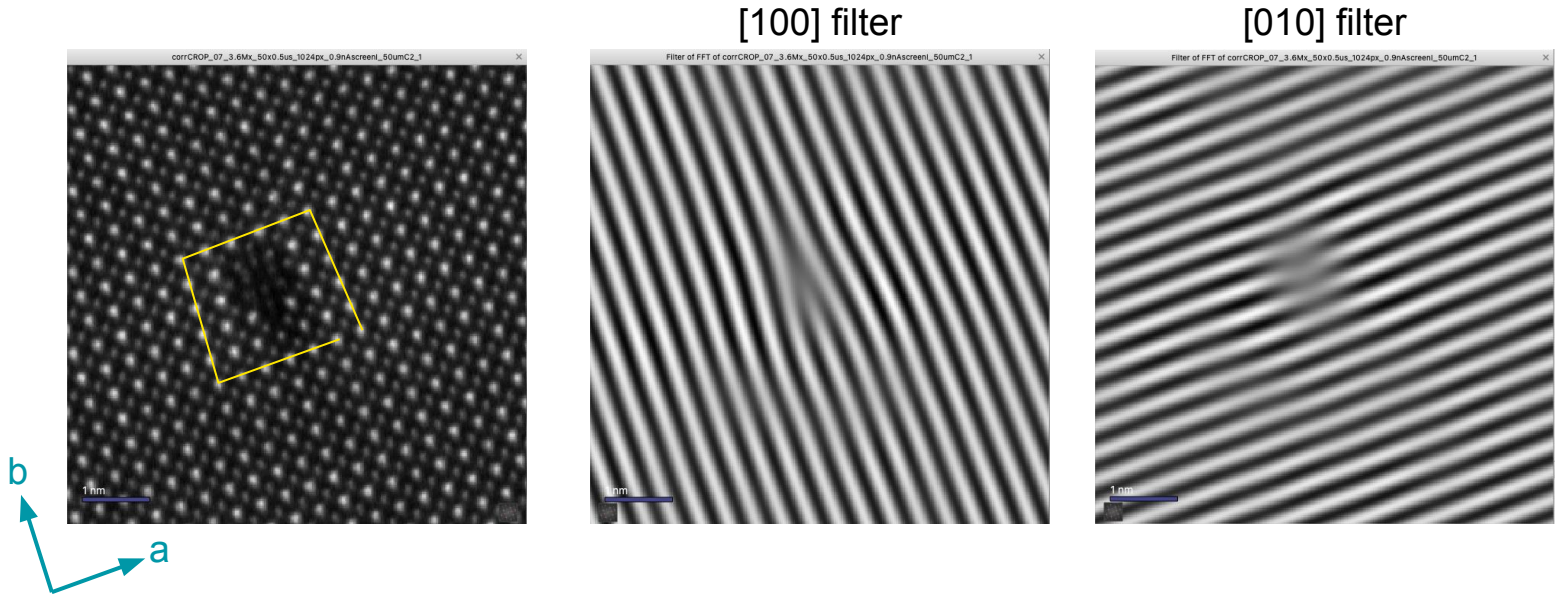




Move the filter to the [100] primary spot. How does the output change? How does the lattice defect appear in the filtered fringe map?

Example 1: STEM lattice image

Fourier analysis: Fourier filtering



The spot filter highlights a single frequency of plane waves, or “lattice fringes”, in the image. Looking at how different planes vary in the image can be a useful tool for analyzing different types of defects or other lattice variations. Here, we can easily highlight a simple edge dislocation. Tracing the Burgers vector directly on the lattice tells us ***b*** = [100]. We can easily see this from the fringe maps as well: the [100] fringe maps shows two lattice planes merging into one at the dislocation, while the [010] fringes are relatively uninterrupted.

Example 1: STEM lattice image

Fourier analysis: Fourier filtering

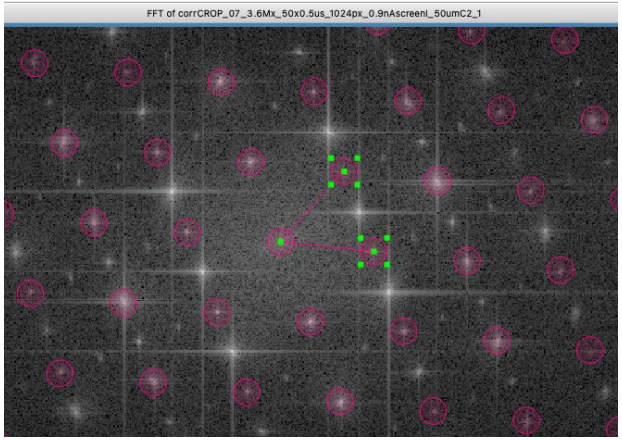
📍 Checkpoint!

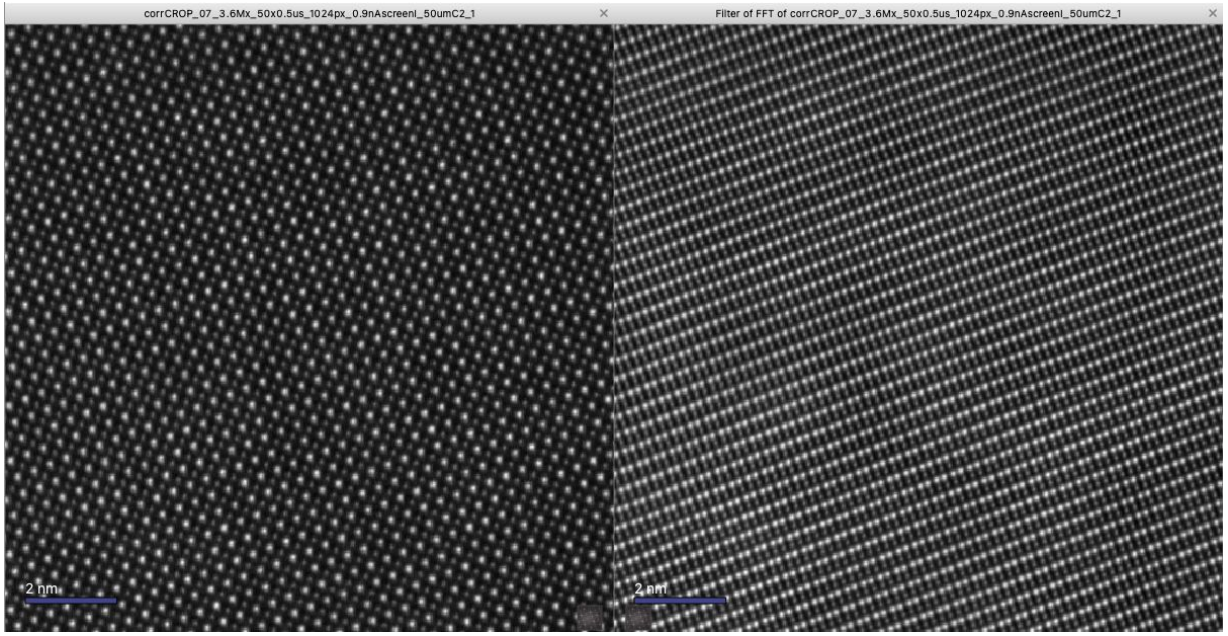
Play around with the various parameters of the spot filter.

- How does changing the spot size affect the output fringe map (Hint: this relates to the previous question about converting the mask size in 1/nm back to real space units)?
- Which other FFT peaks are sensitive to the edge dislocation? How could a systematic description of this relationship be useful for materials analysis?
- What about higher-order spots? What is the relationship between the [100] and [200] fringe maps?

Example 1: STEM lattice image

Fourier analysis: ⚠️ Fourier filtering ⚠️





As a fun art project, create a lattice filter ("Processing → Fourier → Create Lattice Filter"). The filter is now an extrapolated array of spot filters in two directions. Drag the two control spots to the half-order [1½0] and [1-½0] spots. Compare the raw STEM image and the filter output in a clean region (away from the dislocation). What???!?!?? We've magically converted our (001) projection into something that looks more like a (101)! This is an extreme example, but should illustrate that when used irresponsibly, filters can create all kinds of fake information.

Tread carefully!

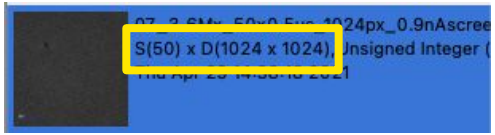
PARADIM
AN NSF MATERIALS INNOVATION PLATFORM

Image registration (lite)

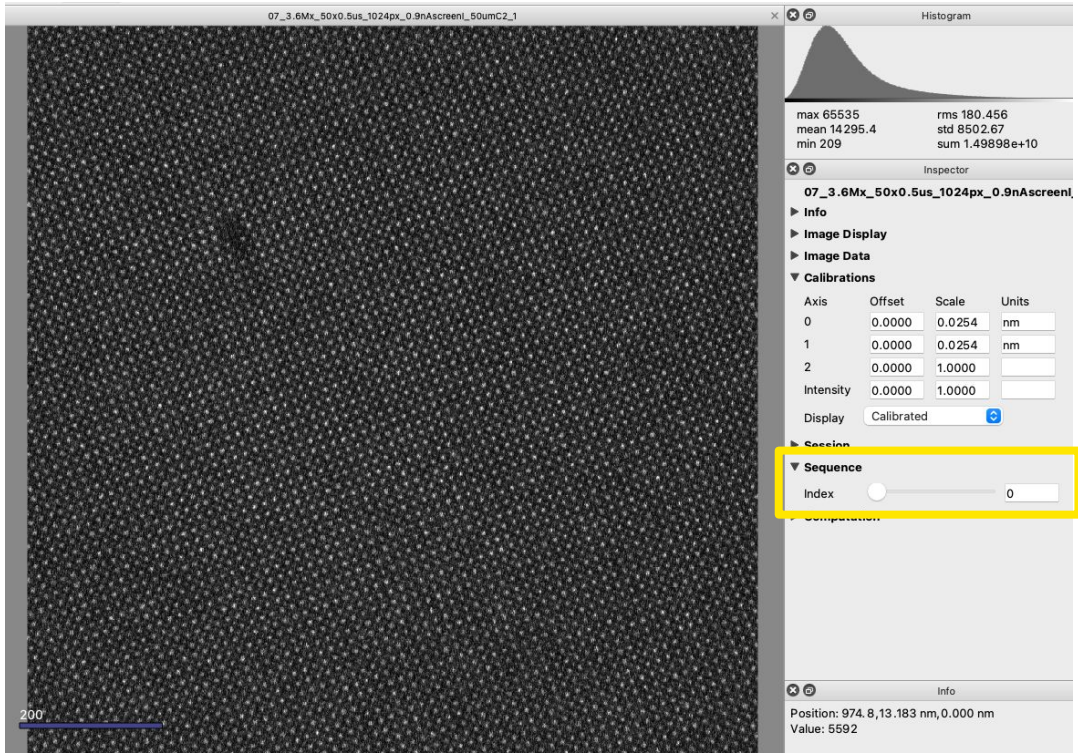
*for full image registration, see **T7***

Example 1: STEM lattice image

Image registration



Each data item shows its shape. This image stack is being read as a sequence (S) of 50 images with individual dimension (D) 1024 x 1024 pixels.

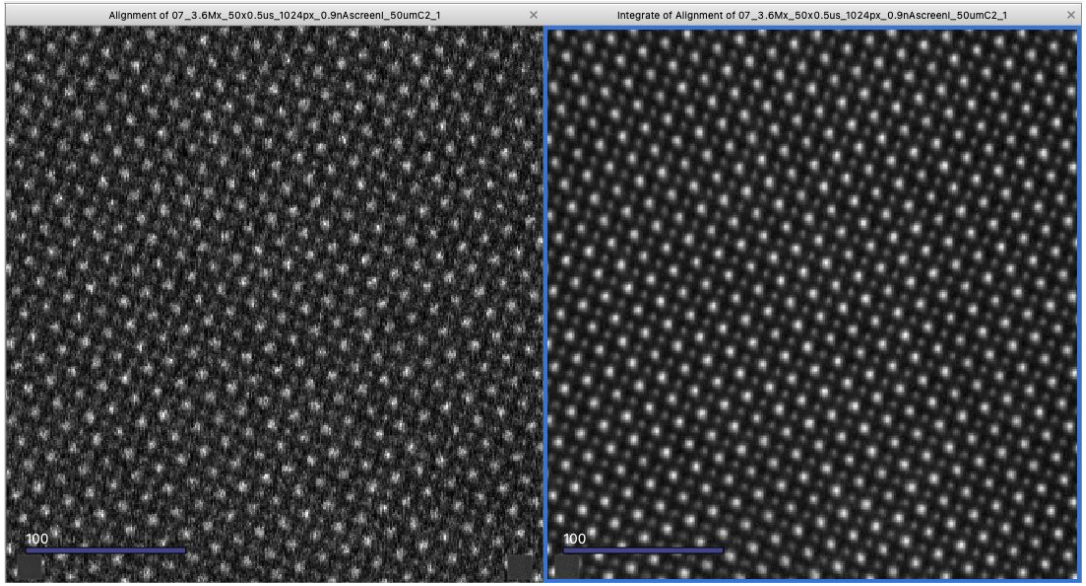


For high-resolution images, we often acquire a stack of rapid images frames which we then align and sum to produce a high signal-to-noise-ratio (SNR) image without artefacts or distortions from the scan or sample drift. Import the raw image stack "07_3.6Mx_50x0.5us_1024px_0.9nAscreenI_50umC2_1.tif" and calibrate it using the same values as before. The Inspector now includes a slider to move through the image sequence. Move the slider around and make sure you understand what is going on in this image stack.

Example 1: STEM lattice image

Image registration

For best results, final image registration should be performed using the Rigid Registration notebook (see session T7). But for preliminary analysis, it can be helpful to have a quick tool at your disposal. With the raw image stack selected, use “Processing → Sequence → Align (Fourier)” or “Processing → Align → Align Sequence/Collection (Fourier)” (depending in your version of Swift). It may take a few seconds... drag the resulting data item into the workspace. Scroll the through the aligned image sequence and the raw image sequence -- what do you notice?

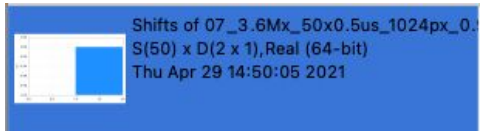


Select the aligned stack in the workspace and use “Processing → Sequence → Integrate” to sum the aligned stack. Drag it into the workspace and compare with a single image slice from the raw stack. What do you see?

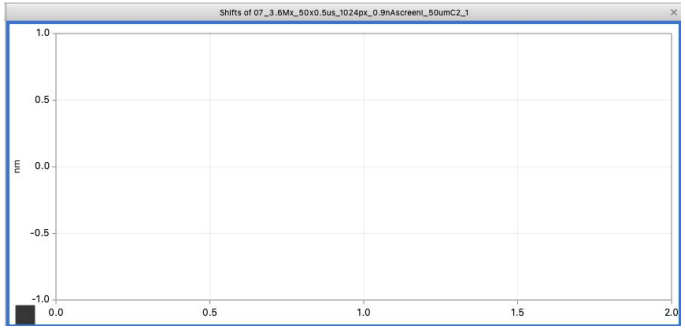
Example 1: STEM lattice image

Image registration

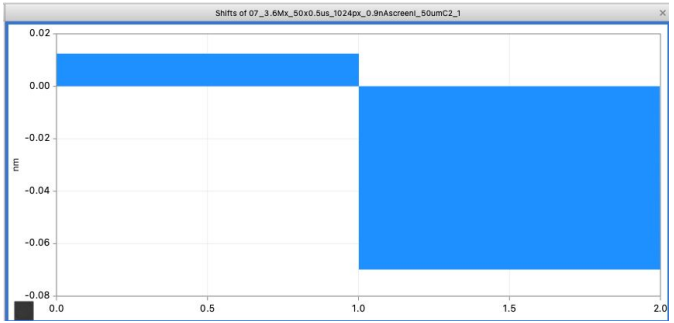
Sometimes the image shifts themselves can be important metrics for understanding your experiment -- especially if you are working under cryogenic or other *in situ* conditions.



Index 0



Index 10

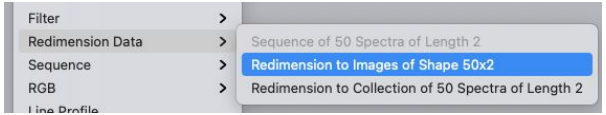


Select the raw image stack again and run “Processing → Sequence → Measure Shifts”. Drag the new data item into the workspace. Notice from the description that it is also sequence (S) of 50 shifts, each with two dimensions (the *x* and *y* shifts). The first shift pair will always be (0,0), because this method takes the first image to be the initial reference point. Scroll through the sequence and see what is displayed for each shift pair in the sequence.

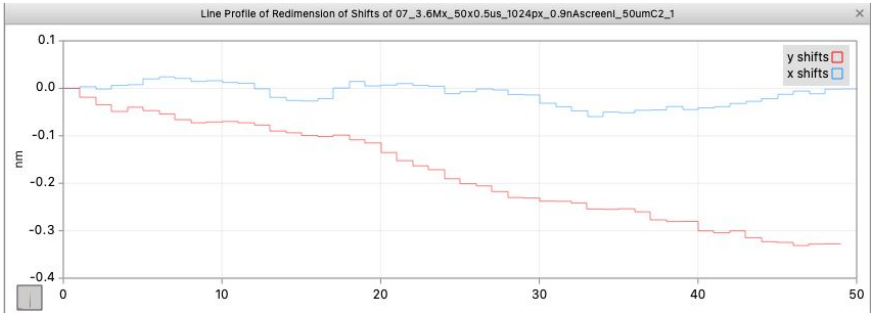
Example 1: STEM lattice image

Image registration

Rather than thinking about these shifts as an (*x*, *y*) coordinate for each image, we can also think of these shifts as tracking the *x* and *y* positions over the course of the stack. This might be easier to visualize if we **redimension the data**.



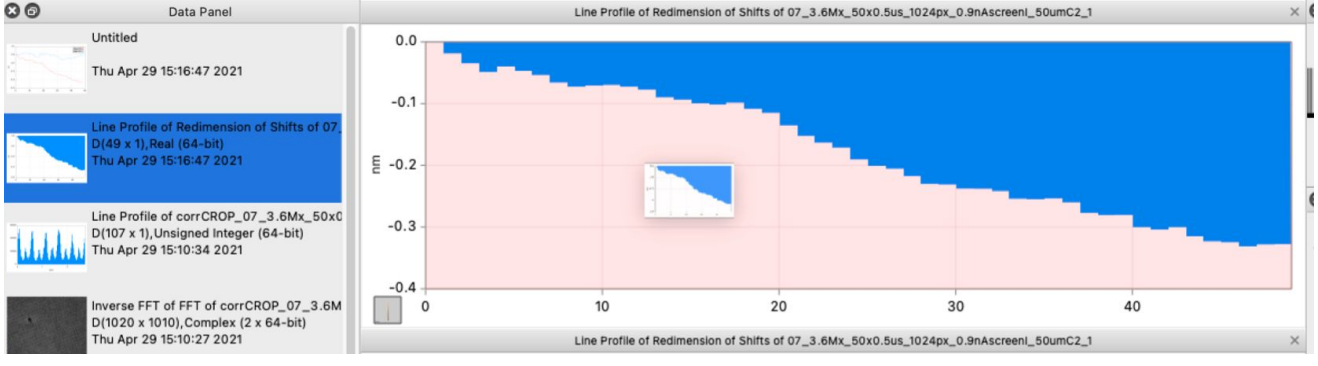
Select the stack of shift pairs in the workspace and use “Processing → Redimension Data → Redimension to Images of Shape 50x2”. Drag the new item into the workspace. Now we can clearly see the trend in shifts of *x* and *y* as the two columns of the “image” (Note that it’s rotated sideways here for display purposes). Which dimension has drift of larger magnitude? Try taking a line profile across each of the dimensions.

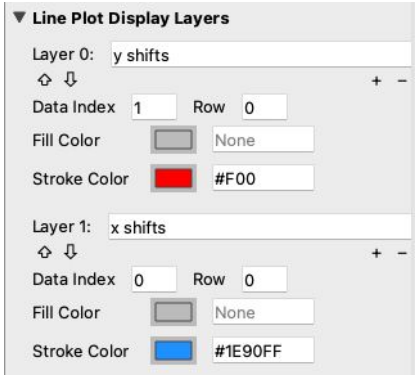


Example 1: STEM lattice image

Some handy tricks

When you drag items into the workspace, Swift gives you a preview of the added window in light red. To combine two data sets of similar type (e.g., plot two line profiles on the same axes), drag one data set so that the red window fills **only the graph area** (not the full window).





Zoom in (out) on line plots or spectra by holding ⌘ and click-dragging on the axis right (left). Click and drag to pan left/right or up/down.

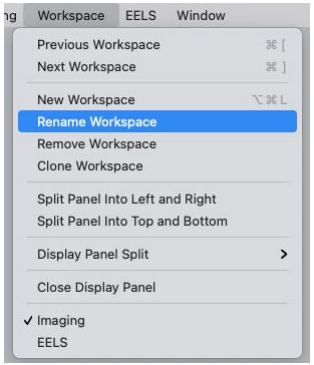
You can adjust the display settings (color, line fill, etc.) at the bottom of the Inspector window.

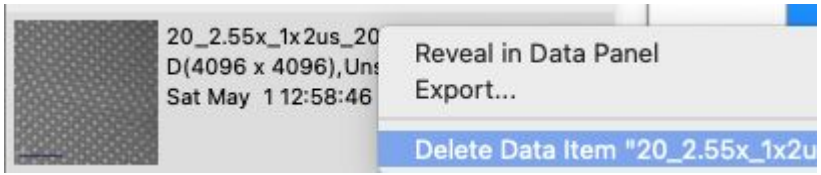
📍 Checkpoint!

Which dimension (x or y) had larger magnitude of sample drift?

A fresh start

PHEW!
Things are getting a little crowded in the workspace… let's get a fresh slate.





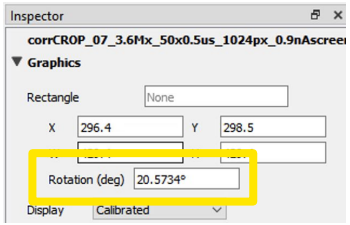
Swift lets you save multiple workspaces, so you can easily switch back and forth between different analyses while keeping things (relatively) organized. Under the "workspace" menu, rename this workspace something descriptive (e.g., "Imaging"). Then create a new workspace and rename it "EELS" for the next example.

Remember that you can also group different experiments or data sets into separate projects under the "File" menu. Workspaces within a single project will share data files (images, FFTs, line profiles, etc.). To keep the memory under control, it's good practice to start new projects when the data sets aren't related, or to periodically remove data items from the data panel (right click → "Delete Data Item", or select + delete).

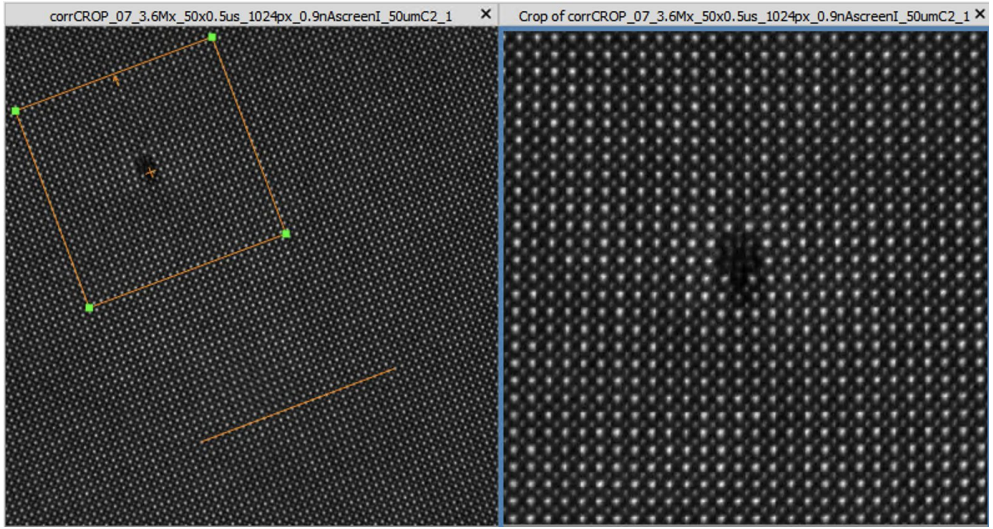
Rotating and Cropping

Make it pretty

We typically acquire STEM images such that the crystal lattice is not exactly aligned with the scan (can you think why? Hint: think back to the Hanned FFT). For some kinds of analysis or publication, we may want to rotate or crop the image by some arbitrary amount. Let's go back to the original lattice image and create a cropped view of the edge dislocation.



Remember that you can always use the line tool to take quick measurements of things like angles on the image!



Draw a box on the STEM image. In the Inspector, set the angle of your box to match the angle of the lattice. Drag and resize the box so it's square and centered on the dislocation. With the box selected, use "Processing → Transform → Crop" to create a new image of just that region. Tweak the shape and orientation of the box until you are happy: note that the cropped image will continue to update as you change the parent object (the box). When you are happy with it, save 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.

Wrapping up

Continue to play around with the other data sets provided in the folder. What kinds of interesting materials effects can you observe or quantify using these kinds of analysis techniques? Try to calibrate your images take line cuts across interfaces, do some lattice order filtering, etc.

Installing Swift on your own computer

You can continue to use Swift on the VM during the summer school, but if you'd like to install it on your own computer you can follow the instructions provided here. Ask questions if you run into any trouble!

Installation

Install directly via

http://nion.com/swift/files/NionSwift-0.15.6.dmg (macOS)

http://nion.com/swift/files/NionSwift-0.15.6.msi (Windows)

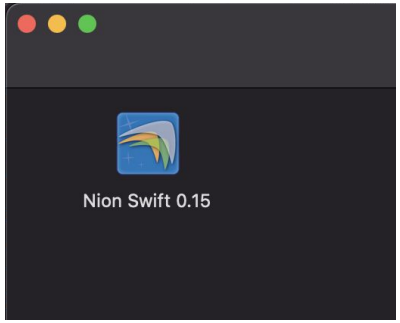
OR

Follow command-line install instructions: https://nionswift.readthedocs.io/en/stable/installation.html#installation

If you already have a Conda environment installed on your computer (see *Python3 Installation Instructions* for more info on installing Conda).

Launching Nion Swift

As an application

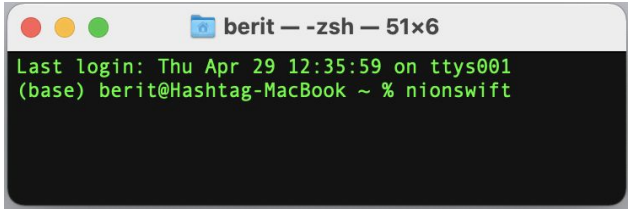


Note: the first time you launch the Nion Swift application, you will need to bypassing the security on your computer:

Mac: right-click on the icon and select “Open…”, when the warning comes up choose “Open Anyway”

Windows: Select “Run anyway” when the warning appears

From the terminal

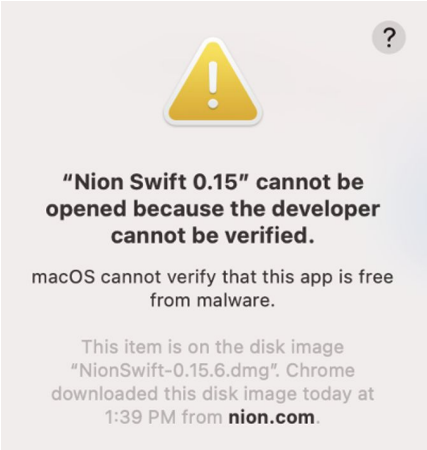


Launching Nion Swift

Detailed instructions: macOS

Open the .dmg installer and drag the Nion Swift app to your hard drive (somewhere you can find it again, e.g. “Applications”).

Right-click (or ctrl+click) → “Open” on the app to start it. Because the app did not come from the Apple store it will sulk:



Click “Open anyway” to open the app the first time. After this, you can just click on the app itself to start normally.

Installing packages

You will also need to add some extra packages if you want to do things like EELS or 4D-STEM analysis.

For command line installation, follow the steps provided at the bottom of the installation webpage:

```
$ conda install -c nion nionswift-eels-analysis
$ conda install -c nion nionswift-experimental
```

For direct installs, download the two plugin folders provided on the Box.

Copy these folders into your Nion Swift Library:

On Mac: '`/Users/you/Library/Application Support/Nion/Nion Swift/PlugIns/`'

On PC: '`C:\Users\#userName\AppData\Roaming\Nion\Nion Swift\PlugIns\`'