Nanowire Recognition, Width, and Height Analytical Library (NaRWHAL)

Yatong Ge[1], Nicholas Montoni[2], Arushi Prakash[1], Wesley Tatum[3]

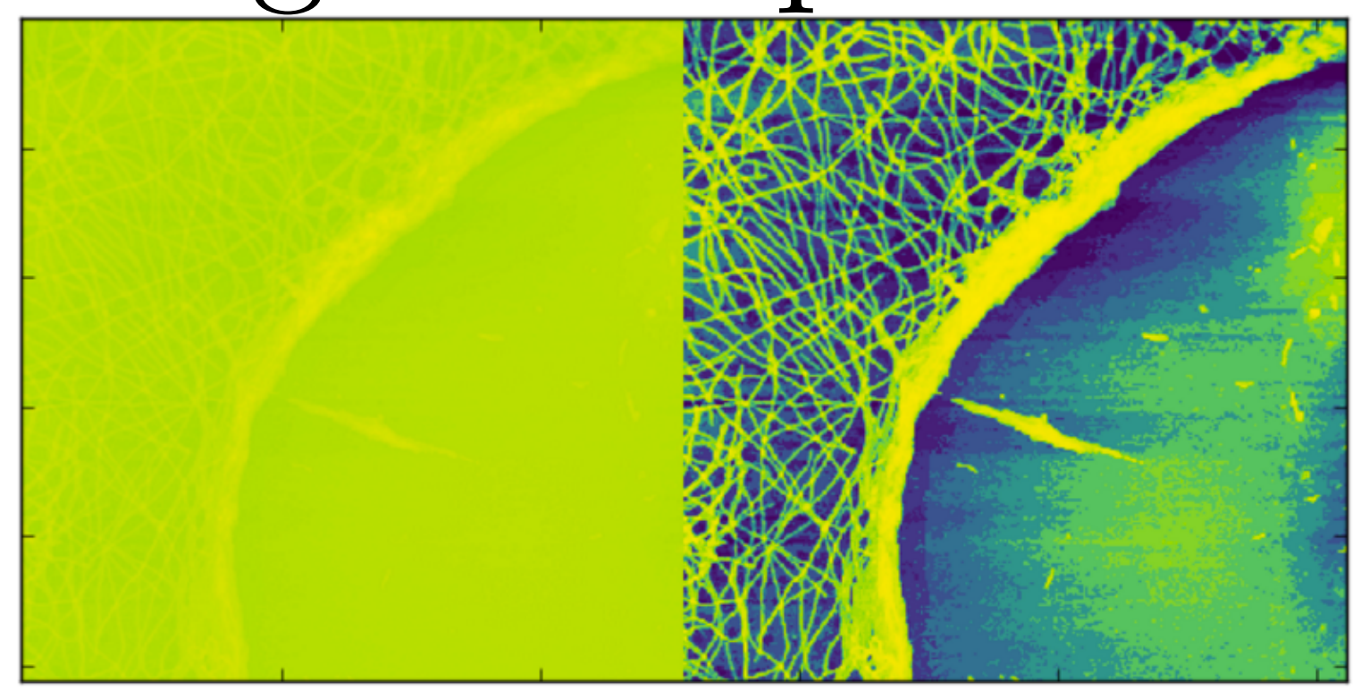
1. Chemical Engineering 2. Chemistry 3. Materials Science and Engineering

University of Washington, Seattle, Washington

NaRWHAL

Backgrounding

Histogram Equalization

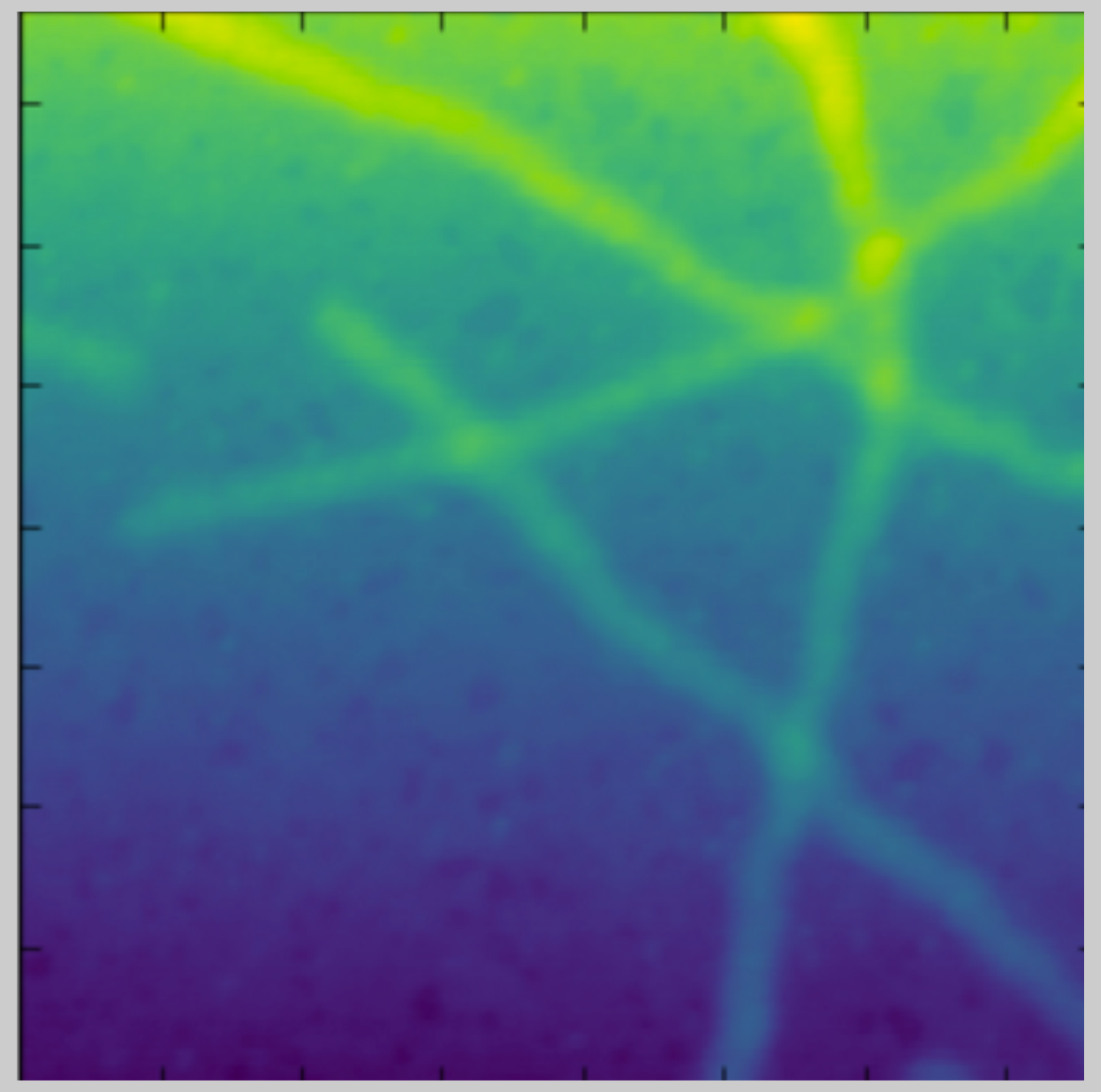


The backgrounding function looks at instances of intensities above or below 3 sigma, removes them and iterates 10 times to reach a convergence. It then calculates the rms of this background and removes that gradient. The histogram equalization just spaces the intensities of the histogram of intensity values such that the cumulative distribution function is approximately linear, rather than a few sharp increases.

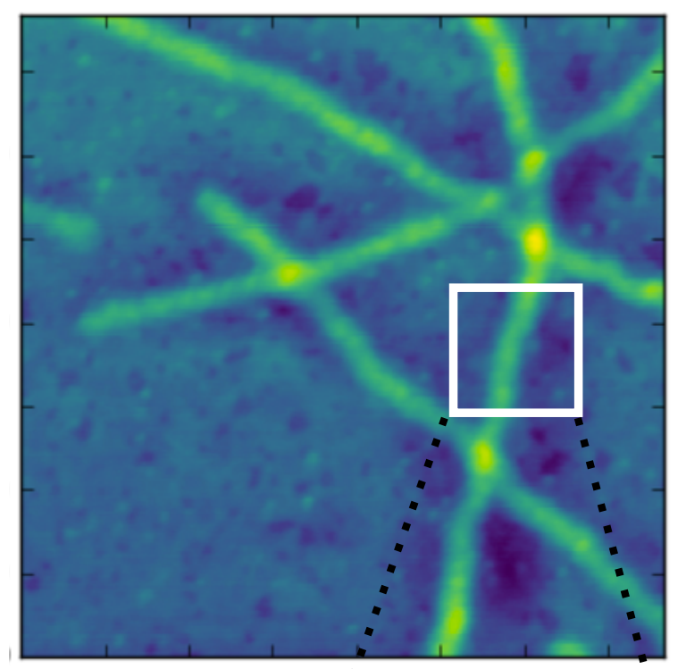
Problem Statement

In the field of nanoscience, there is a great need for rapid and automated analysis of images aquired from instrumentation such as scanning probe microscopes. As direct, human-driven analysis is prone to inaccuracies, we propose to develop a user interface that allows scientists to choose regions of images for analysis. The tool must incorporate backgrounding, choice of analysis method, and output of desired observables.

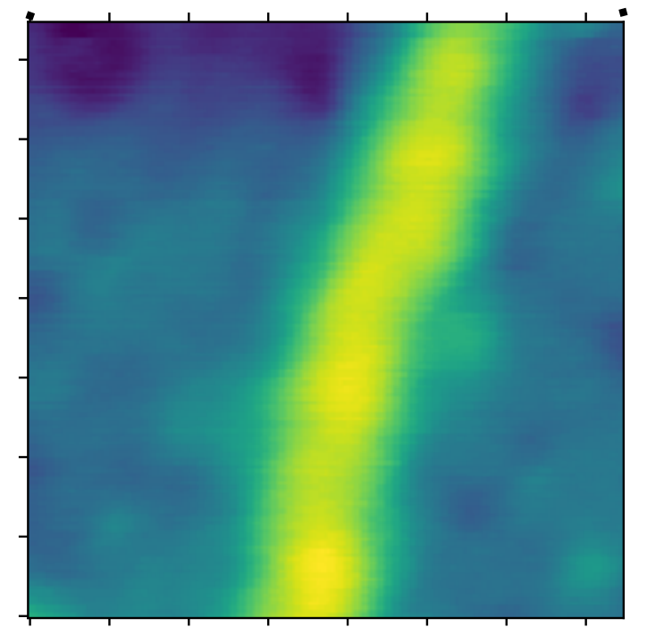
Raw AFM Data



Backgrounded Image

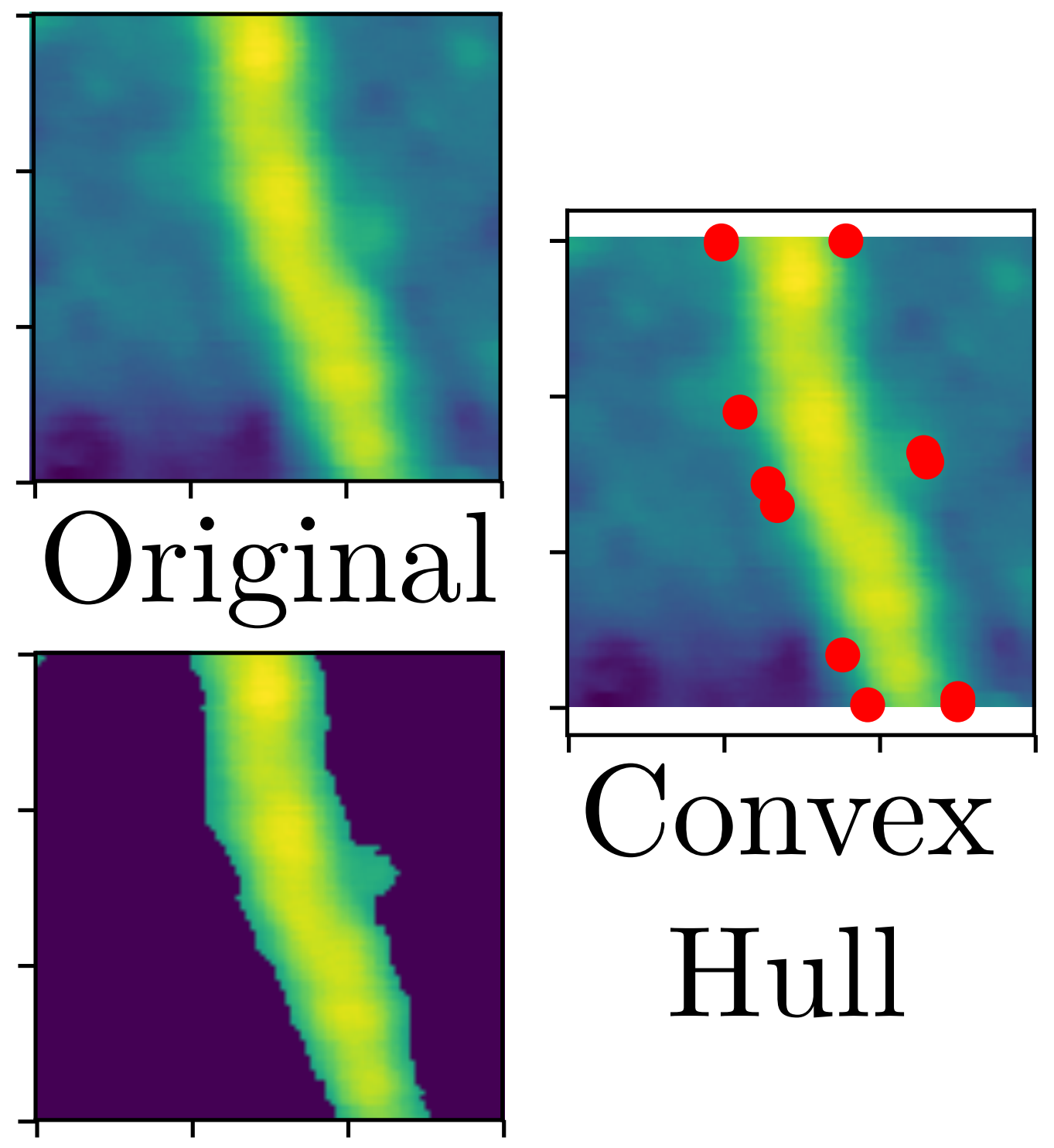


Contour Detection



Selection

The contour detection process uses Open CV thresholding to determine where there is and is not a nanowire. As the nanowires are the tallest objects, the lower bound is placed at half the nanowire height to exclude excess background. We then exploit another Open CV algorithm that uses the contours to determine whether or not there is a convex shape present. The points that define the convex hull are plotted on top of the image.

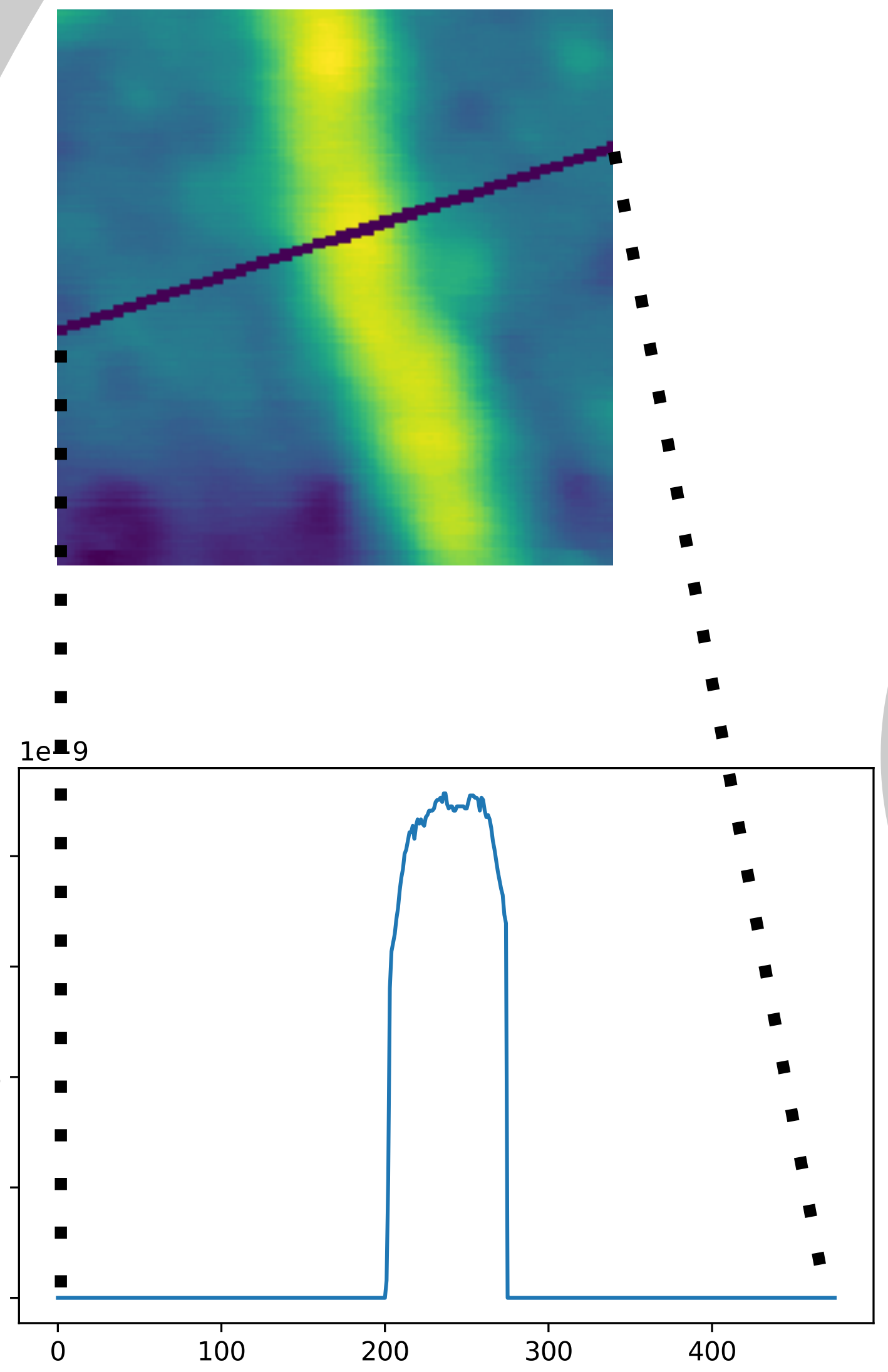


Original

Convex Hull

Thresholded

Profiling and Measurement



To determine the profile of the nanowire, we employ an open source tool in scikit-image called profileline. Given a set of contours, this tool draws a line that bisects them. Exploiting the fact that this will draw a line down the middle of wire, we compute the profile along the line exactly perpendicular to the line drawn by scikit-image.

Future Work

- Offer more backgrounding and analysis options
- Compare backgrounding/analysis methods
- Train the software to recognize nanowires
- Correlate measured observables to synthesis methods
- Automate the process
- Expand to varied nanostructures

About the GUI

Tkinter is Python's standard GUI package. It is a thin object-oriented layer on top of Tcl/Tk (scripting languages). Our GUI contains point and click interactions that enable a user to upload an AFM image of nanowires, manually select nanowires of interest, remove background noise and measure features of the nanowires. The GUI is designed to disseminate as much information about the process as possible, including the removed background and the profile of nanowire used for feature calculation. Keeping in mind that users may have different sensitivities to color, the GUI allows you to change the color scheme. Only sequential color schemes are employed to ensure that heights in the AFM image are portrayed correctly. Messages guide the user through the process.

References

Van der Walt, *et al.* "Scikit-image: Image processing in Python." *PeerJ*, 2014.

Nundy, S. *et al.* 2000.

“Python GUI Programming (Tkinter)” https://www.tutorialspoint.com/python/python_gui_programming.htm

"How to embed a Matplotlib graph to your Tkinter GUI” https://pythonprogramming.net/how-to-embed-matplotlib-graph-tkinter-gui/

This work makes use of OpenCV.