Nearest Neighbor Analysis of STM images

By: Kristen Johnson

Updated: 4/1/2022

This a tutorial for how to use MATLAB to determine the distribution of bound nearest neighbors in an image. This procedure assumes that the user has already fit the image to with a grid and determined the state of each molecule. You must have the following variables loaded into MATLAB:

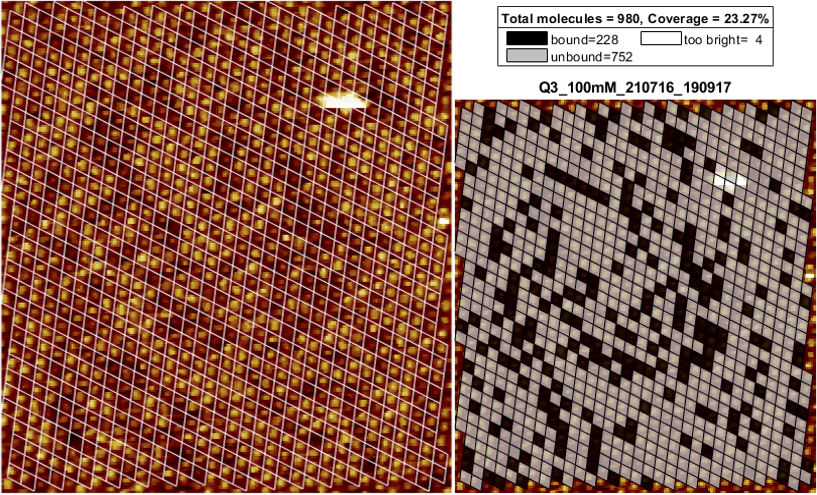
*pgon* - 1 x number of grids cells polyshape variable, the hold coordinates of all grid vertices.

*im* - m x n array, image variable.

*state/stateOut* - 1 x number of grid cells array, holds state value of each cell usually 0=bound, 1=unbound, 2=do not count.

*mps/midps2* - 2 x number of grid cells, holds x,y coordinate of the midpoint of each grid cell.

Before continuing you should be able to recreate the following figure that has all of the different molecule types color coded.



*Figure 1: Example of data you need before doing nearest neighbor analysis.*

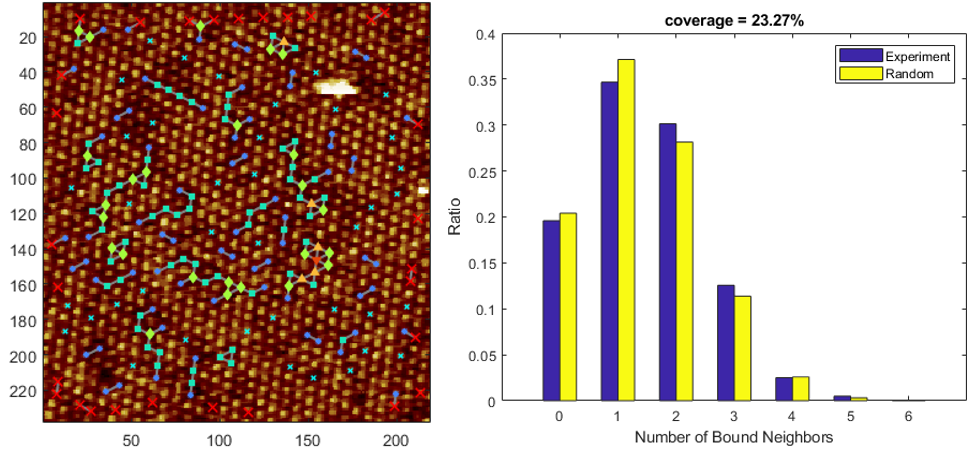
Once you have results like in Figure 1 then continue with the following:

1: run stage 5: Nearest Neighbor analysis in the singleImageCounter\_v1\_8.mlx file.

This section has the built in function that finds all nearest neighbors and outputs both an image with overlaid color-coded grid, a histogram of experimental and theoretical counts, and a table variable with the counts and ratios of each type.

Input=[row number from state\_ind variable states to consider in neighbor count], mps, pgon, state, state\_ind, im, countResult, colormap

See the function documentation at line 431 in the singleImageCounter\_v1\_8.mlx file. Output should look like figure 2.

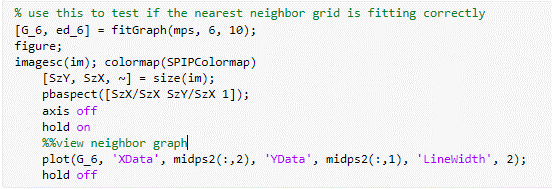


*Figure 2: Example output of the function nn\_calc.*

2. run state 6: to save the results into an excel file and matlab variable.

Troubleshooting:

The function may not correctly fit a grid to the data. It is currently set up for 6 nearest neighbor CoOEP images if your image has a different number of nearest neighbors you will have to update the function, calc\_nn. Calc\_nn is an embedded function in the singleImageCounter\_v1\_8.mlx file. It begins on line 431. You will need to update line 460. This line calls another embedded function, fitGraph, the input variables are mps and the number of neighbors, and a search distance in pixels (optional). Check the options are correct by running in a new section the code show in the following figure 3.



*Figure 3: Code example for troubleshooting the full nearest neighbor grid.*

**

*Figure 4: Example of correct hexagonal grid.*

Once the grid looks correct, like figure 4, update line 460 and then run stage 5: nearest neighbor analysis as normal.