Pillar Tracking Guide

Example files:

- Ex_7um_10sTL_pillars.tif: timelapse of pillar tips
- Ex_7um_10sTL_cell.tif: timelapse of cell membrane, corresponding to the pillar images above.
- Ex_7um_10sTL.txt: parsed output from ParticleTracker script, run on the pillar images above.
- PillarForcededuct.m: MATPAB script for analysis of forces.
- PlotForcesOntoImage.m: MATLAB script for overlaying forces onto image stack
- Pillars.avi: output of pillar position with overlaid force vectors.

Set-up
1 – Install all ImageJ plugins by placing them in the Imagej/plugins/macros folder. For the matlab scripts, you can simply copy the entire folder over to your matlab directory.

Image-J section (tested with ImageJ version 1.53e)

- Stackreg: http://bigwww.epfl.ch/thevenaz/stackreg/
    - Note that this requires turboreg to also be installed.
    - Also, this causes an update issue in Fiji
- Particle Tracker has been incorporated into MOSAIC: https://sbalzarini-lab.org/?q=downloads/imageJ
    - Run Help->Update Fiji (or in case ImageJ2 run Help->Update...)
    - Click on "Manage update sites"
    - Find there and mark "MOSAIC ToolSuite"
    - Apply changes and Fiji should automatically download latest release of MosaicSuite.
    - Restart Fiji (as required) and after restart all functionality by MosaicSuite can be found in Plugins->Mosaic menu.
    - Documentation: https://sbalzarini-lab.org/docs/MosaicSuiteDoc/particleTracker.html

Notes:
The attached stack is processed, so steps 2 and 3 aren't needed. They can still be run though.
The subsequent steps all assume that you are working with a single-channel image. Pillar/membrane merges are a little more technical to handle.
Bleach correction helps for stacks where many frames are acquired.

2 – With the stack open, run the StackReg script, selecting the Translation option. This will permit an X-Y correction that adjusts for ambient stage drift. The other options involve methods of stretching and scaling, which artificially manipulates the distribution of intensity (where the pillars are).

You have the option of running StackReg on the entire FoV or on a sub-region (one or two cells). I prefer running it on the full FoV. Though it takes longer, it does provide the script more information to work from.

3 – Crop a cell-of-interest if you haven't already. Also note that when StackReg corrects for stage drift, it adds new areas of zero-signal to the boundary to correct for the shift. Since these regions have zero-signal, whereas your background is much higher, you will need to exclude these regions.

4 – Run the ParticleTracker script, selecting the Detect & Track option. Select a Radius, Cutoff and Percentile (good starting values being 7,0,10 for the iXon3 & 1um pillar diameters). Preview to ensure the fit looks good.

5 – For the bottom half, Particle Linking, set the link range to 1. This will prevent the plugin from skipping any frames. The displacement parameter dictates how close two neighboring particles/pillars can be. Since this is not just an XY adjustment (3 other vars), set it a little high. I generally do 20um for a 5-10s TL and 25-30 for a 15+s TL.

6 – Once the results window pops up, click Visualize All Trajectories to visualize the traces on each pillar. Each trace is color-coded. You should get a general idea if a single pillar is being identified by 2+ traces if there are multiple colors on a single pillar. To verify, select a region around a single pillar and click on the Trajectories in Area Info button.

7 – Select the region around the cell with sufficient background pillars: at least as many background pillars as ones that the cell is manipulating (useful for later).

8 – Click the Trajectories in Area Info button and copy all of the text to notepad.

9 – In notepad, check to see if there are any incomplete traces. Stitch together any partial traces and delete any redundant ones. As a general rule, if you have a redundant trace, then your Percentile value in step 4 was too high. File “Ex_7um_19sTL.txt” illustrates output from Particle Tracker that has been edited to link traces.

Matlab Section (tested with MATLAB version R2021a)

10 – Copy the completed textfile to your matlab directory.

11 – Open the matlab file PillarForcededuct.m.

12 – There are a couple of lines to edit before running the script:

- Change line 4: M=load(‘X.txt’); so that X refers to your *.txt file.
- Change line 7: Method = ‘Y’; so that Y refers to your pillar height
- Change line 8: time_scale = Z; so that Z is your time between images

13 – Run PillarForcededuct.m. You should get a plot as well as a list of values in the command window. Those values are average cell force, average force per pillar, greatest force per pillar, pillars in contact and number of frames used in the analysis.

14 – Compare values with the timelapse (number of pillars in contact, number of frames, etc) and check to see if forces reach a max on the plot.

15 – Generate image with force vectors plotted. This can be done using the PlotForcesOntoImage.m script.