Open *MPAv7.m* and select a czi file. Make sure that the user initialized variables BF (brightfield) and DNA (stain for the nucleus; e.g., Hoechst or DAPI) correspond to the proper channel number in the stacks.

Manual Mask Alignment

A window entitled ‘Figure 1: Manual Mask Alignment’ appears, along with a set of moveable cross-hairs. If the mask or image is too dark, try increasing the brightBoost (default: 1). Left-click the left end of the top micropipette (see cross-hairs below) in order to properly fit the image (green) to the mask (purple). Once the mask is properly fit, the program can properly segment the image into its 18 pockets and the user can systematically apply the correct threshold to each nucleus.

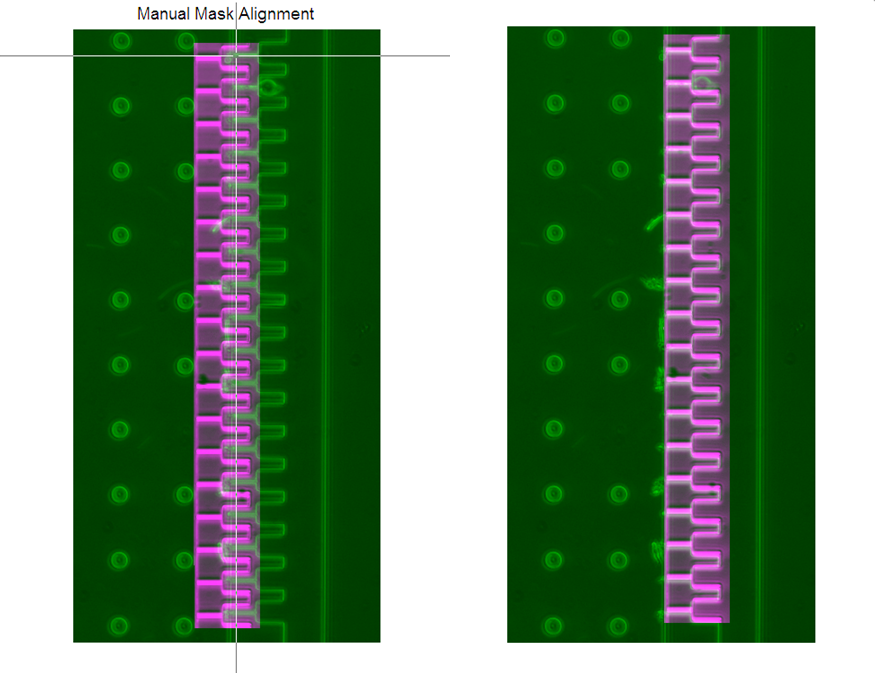


Figure 1. Manual Mask Alignment

If the image is rotationally misaligned (Figure 2A), left-click the left end of the top micropipette channel (yellow ‘left’ arrow) and then (Figure 2B) middle-click the left end of the bottom micropipette channel (yellow ‘middle’ arrow). Once the rotational error has been corrected and the image is in line with the mask (Figure 2C), left-click the left end of the top micropipette channel (yellow ‘left’ arrow) once again to re-shift the image. (Figure 2D) When the image is properly aligned, right-click to move forward.

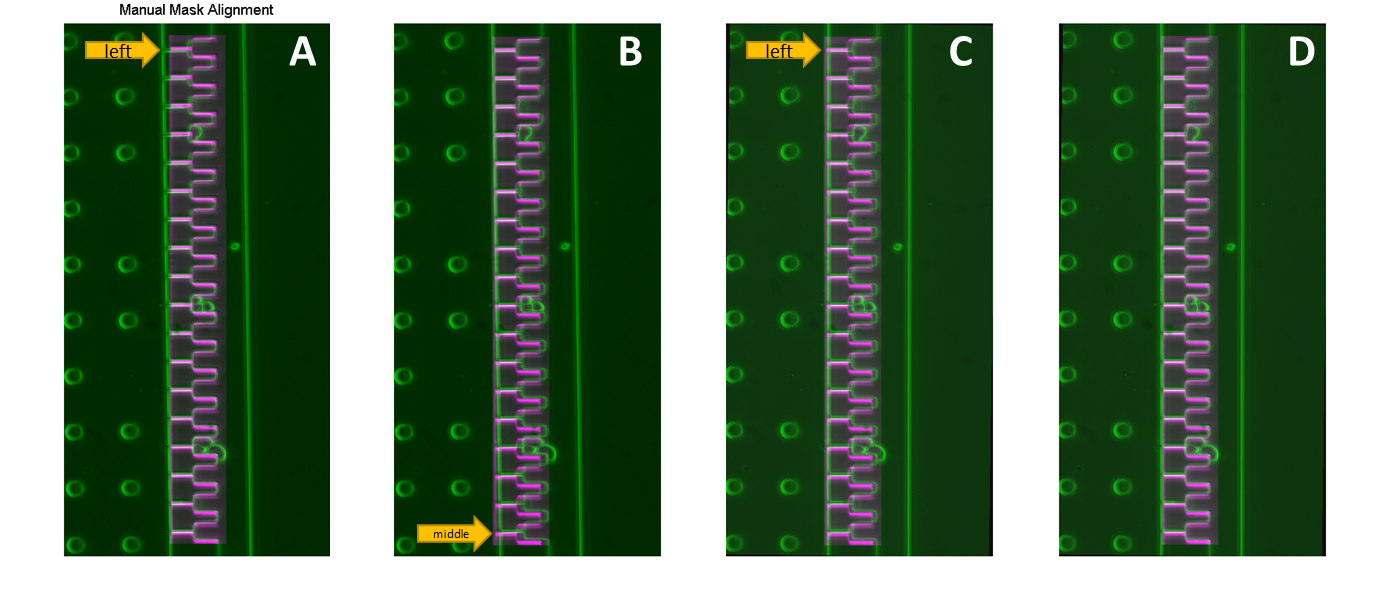


Figure 2. Compensating for rotational error

Setting the Micropipette Entrance and Thresholding the Nucleus

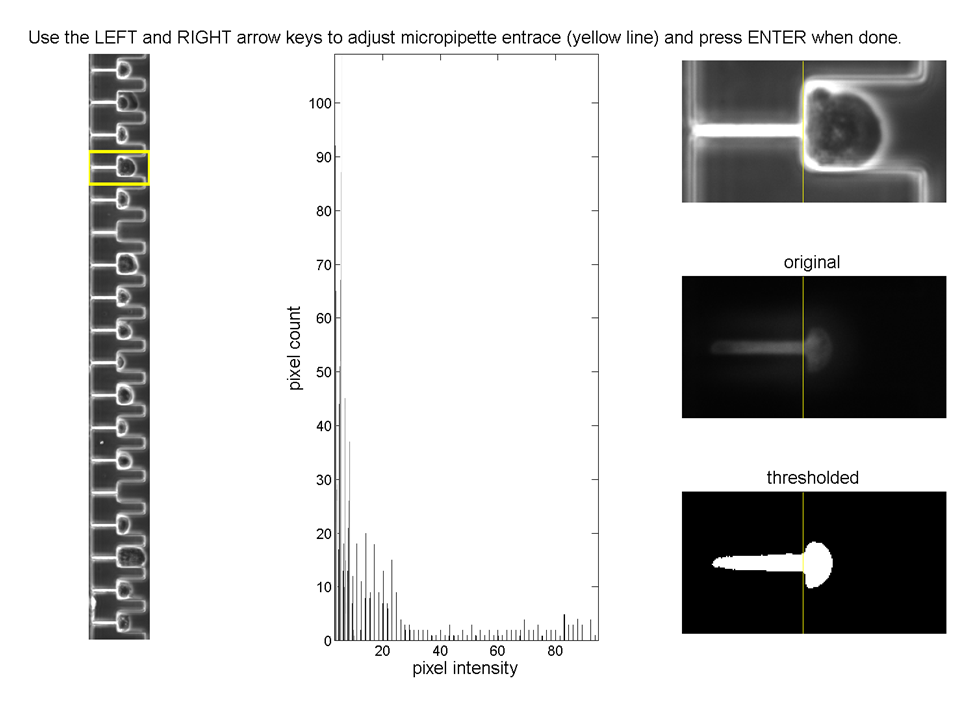
Another figure window appears (Figure 3). This is the main user interface used to threshold each nucleus in its respective pocket. 

Figure 3. Manual threshold adjustment

A yellow box (left panel, currently the fourth pocket down) indicates the pocket and cell being analyzed. The left and right arrow keys adjust the ‘starting line’ indicating the entrance of the micropipette channel. Press ‘enter’ when the yellow starting line corresponds to the entrance of the micropipette. The program saves the position of line for subsequent pockets, limiting the amount of user adjustment.

After pressing enter, set the nuclear threshold by clicking within the middle panel, a 60-bin histogram of image intensity values in the pocket. The selected value determines the binary threshold. Additional erosion and dilation processing steps (applied during execution of *theshIt.m* within the program) automatically smooth boundaries and remove spurious pixels within the thresholded image (Figure 3, right panel, bottom). The left mouse button previews the image sequence with the applied threshold (Figure 3, right panel) at a playback speed specified in the program. To adjust the playback speed, change the variable nthFrame. This allows visualization of every nth frame during playback. For large files with many frames, it may be particularly beneficial to increase nthFrame.

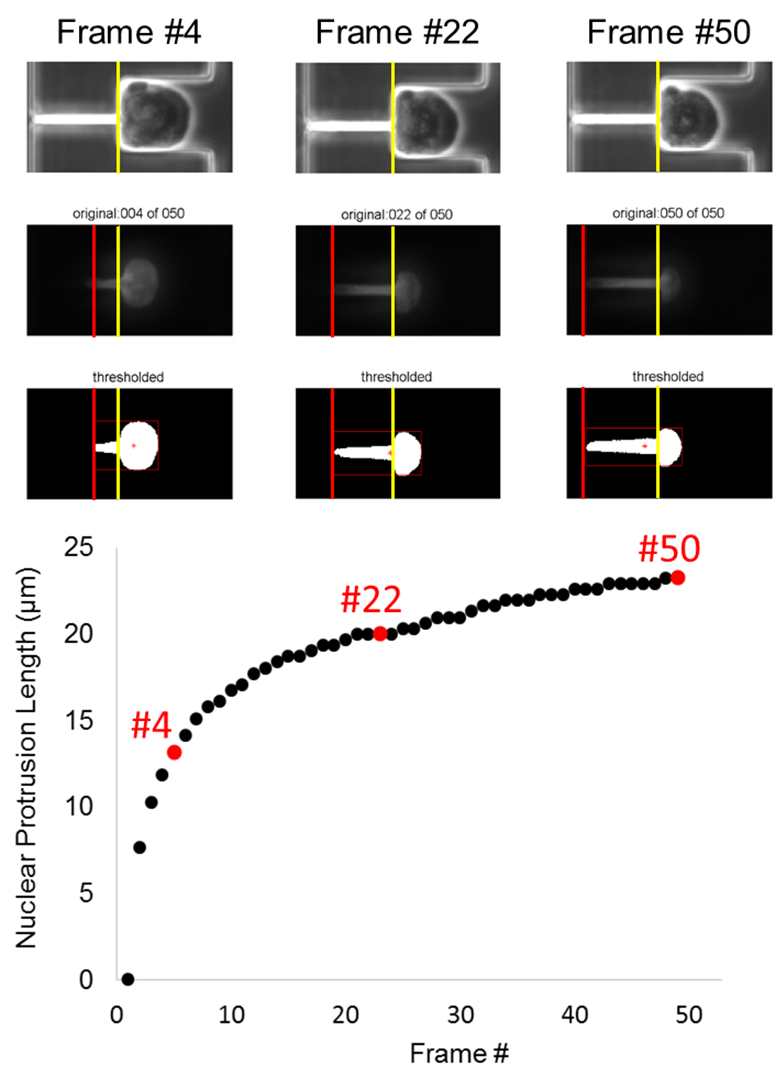


Figure 4. Computational Output

During the preview, make sure that the red line (Figure 4) matches up with the leading edge of the nucleus. If not, adjust the thresholding by left-clicking a different pixel intensity value. Note: the pixel count vs. pixel intensity histogram corresponds to the third to last frame (when a cell is most likely to be trapped in the pocket). This can be changed in the MATLAB script by changing the variable frameFromEnd from 3 to a value of choice.

Once satisfied, the right mouse button stores the nuclear threshold value and moves to the next pocket. The value that gets clicked is the value that gets stored, so it is important not to move the mouse around in between previews.

Once the user selects the micropipette channel entrance (yellow line) and nuclear threshold value, the program computes the nuclear protrusion length at each frame by drawing a bounding box around the nucleus and then calculating the distance between the left edge (red line) and the start of the channel (yellow line) (Figure 4). The values are automatically exported to an excel file where they can be plotted. The bottom panel (Figure 4) plots the nuclear protrusion length over time for a given cell with the red data points corresponding the frames above.

Data Export and Analysis

Once the threshold has been set on the last pocket by right-clicking, the program automatically exports an excel file with the name: MPAresults\_’filename’.xlxs. The spreadsheet has 18 rows of values corresponding to the 18 pockets and columns corresponding to the number of frames. The values correspond to the nuclear protrusion length in microns.

Not every pocket has an analyzable aspiration event and the user should consult the original czi to make sure the values corroborate the original. For a micropipette aspiration event to be valid, the pocket must be empty and then get filled by a cell that gets aspirated in the micropipette. On the spreadsheet, this appears as a zero followed by gradually increasing numbers. In the case of a zero, followed by a series of negative values that eventually become positive, use the last negative before the transition to positive values as the “zero” starting point of the aspiration. Exclude nuclei that fail to aspirate into the pipettes, but make a note of how many fall under this category.

Troubleshooting

1. Index error. E.g.:

Index exceeds matrix dimensions.

Error in MPAv6 (line 94)

img\_channels(:,:,:,DNA) = img\_channels(:,:,:,DNA) .\* DNABoost;

*Make sure that the channel numbers are allocated appropriately. DNA should be set to the channel number that corresponds to the nuclear signal.*

1. Images too bright or too dim

*Try changing the DNABoost or BrightBoost variables. Also a poor nuclear signal may result from the wrong channel being set as DNA.*

1. The program takes too long to run

Increase nthFrame. Also try running on different computer or running a different version of MATLAB. MATLAB R2014a works well.