

## **Image Processing Protocol**

*Software Requirements, Getting Started, Processing Instructions, and FAQ*

Corresponding GitHub repository: <https://github.com/yellenlab/Cell-Array-Counter>)

### **SUMMARY OF ANALYSIS**

The goal of this analysis is to extract a set of MetaMorph® images, and through a series of alignment step(s), cropping, filtering, and cell/bead tracking algorithms extract individual brightfield and fluorescent images of each apartment complex and produce heat maps and efficiency data of an entire chip, along with collection of individual cell property data if applicable. We currently have developed a custom automated imaging system with the ability to image all single cell capture sites multiple times using brightfield and a desired fluorescent filter. These images are processed using custom developed MATLAB® scripts, which isolate and properly identify multiple apartment complexes if present in single images. With individual apartment complexes isolated, we further identify cells within a hydrodynamic trap and apartment and whole chip efficiencies and trends can be extracted, among other information. This data is presented in multiple full device heat maps and in automatically generated reports, aiding in device optimization and design. Cell properties extracted include approximate size (based on radius of fitted circle), location, and brightness. MetaMorph Microscopy Automation and Image Analysis Software is currently used for creating automated imaging processes, and ImageJ is used as a tool for post-image processing and alignment step(s) if desired.

### **SOFTWARE REQUIREMENTS AND DOWNLOAD INSTRUCTIONS**

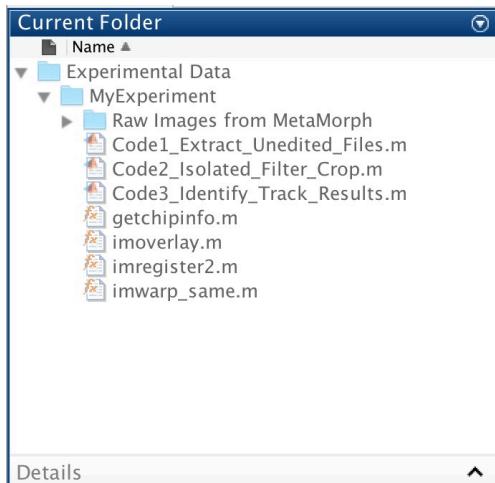
- MATLAB, version R2017a or newer
  - Go to folder with newest version and appropriate OS (Mac or Windows) of the following codes:
    - Code1\_Extract\_Unedited\_Files.m
    - Code2\_Isolated\_Filter\_Crop.m
    - Code3\_Identify\_Track\_Results.m
    - getchipinfo.m for Mac OS or getchipinfo\_Windows.m for Windows OS
    - imoverlay.m
    - imregister2.m
    - imwarp\_same.m
- ImageJ
  - Is free and open source, and can be downloaded at: <https://imagej.nih.gov/ij/download.html>
  - Also need to download the Template Matching Plugin found here:  
<https://sites.google.com/site/qingzongtseng/template-matching-ij-plugin>
  - Follow instructions for installation provided on website. Need to download 6 total files:
    - javacv.jar
    - javacpp.jar
    - opencv.jar
    - cvMatch\_Template.java
    - Align slices.java,
    - Platform dependent library files (see website)
  - NOTE: We have had issues getting the plugin to install correctly on Mac OS platforms, but Window's platforms have had success
- (Optional) Fiji
  - Same as ImageJ, but used for post-processing photo stitching if desired

## GETTING STARTED

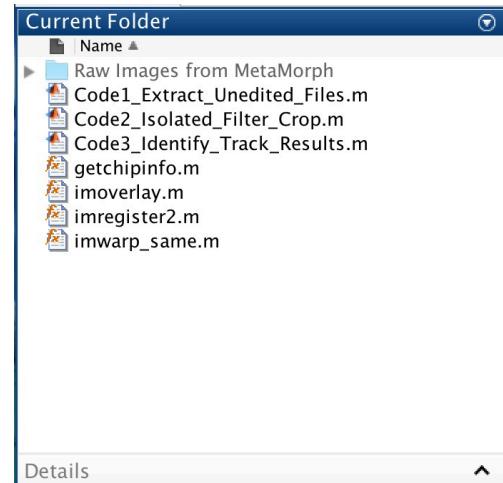
1. Find the most recent version and download the 7 MATLAB codes:
  - Code1\_Extract\_Unedited\_Files.m
  - Code2\_Isolated\_Filter\_Crop.m
  - Code3\_Identify\_Track\_Results.m
  - getchipinfo.m for Mac OS or getchipinfo\_Windows.m for Windows OS
  - imoverlay.m
  - imregister2.m
  - imwarp\_same.m
2. Open MATLAB, and make a new folder wherever you would like data to be stored for an experiment. For clarity, as an example we'll call this folder MyExperiment.

NOTE: Organization is very important, so make sure this folder is named properly. The name does not affect the experimental results and analysis procedure, but is good practice.

3. Copy/Drag the folder containing all raw images from the experiment generated using MetaMorph. This folder should include brightfield and fluorescent images, and likely will contain a .txt file with information entered before running the experiment.
4. Drag and drop the downloaded MATLAB codes from step 1 into the MyExperiment folder.  
**IMPORTANT:** make sure that the working directory is the MyExperiment folder, and not a parent or sub-folder.<sup>1</sup>



**INCORRECT:** Working directory is not the folder containing codes and raw images.



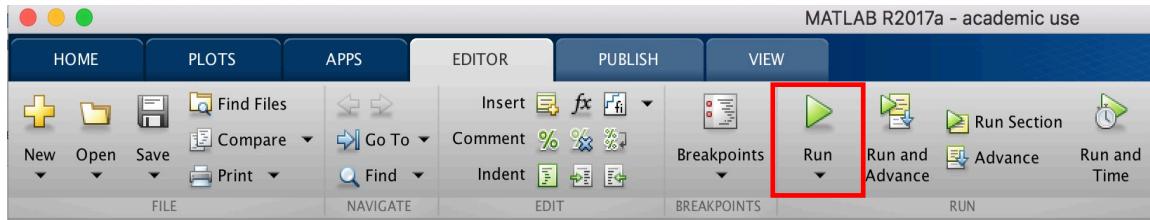
**CORRECT:** Working directory is the folder containing codes and raw images.

---

<sup>1</sup> The analysis codes often look for saved data and images within the working directory, and automatically save to the working directory. Not being in the working directory will result in improperly saved data and can produce errors. If you're more Matlab savvy, then you can add the appropriate codes/folders to the pathway in order to run all codes. However, all future instructions assuming the correct configuration shown

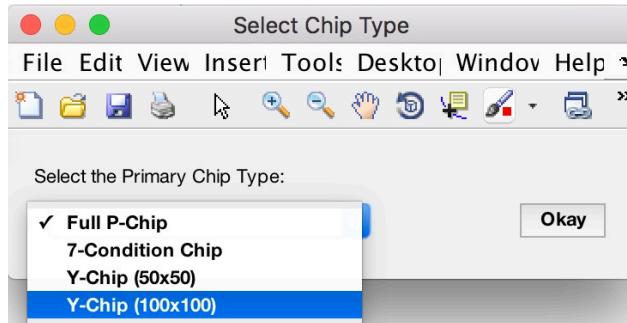
## PRIMARY ANALYSIS OPERATION INSTRUCTIONS

- Now in the working directory, open the code entitled `Code1_Extract_Unedited_Files.m`. Click **Run** in the Editor tab.

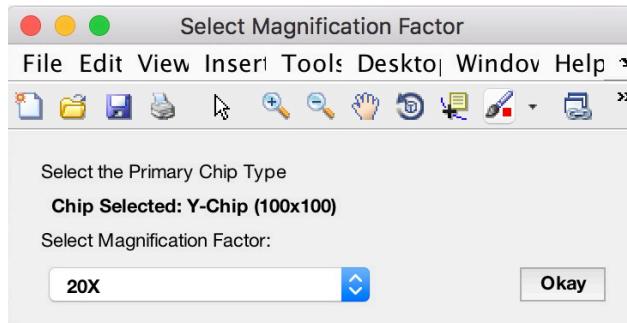


Follow the onscreen prompts to select the chip type, magnification, fluorescent filter, and the image file type that the raw image files from MetaMorph were saved as.

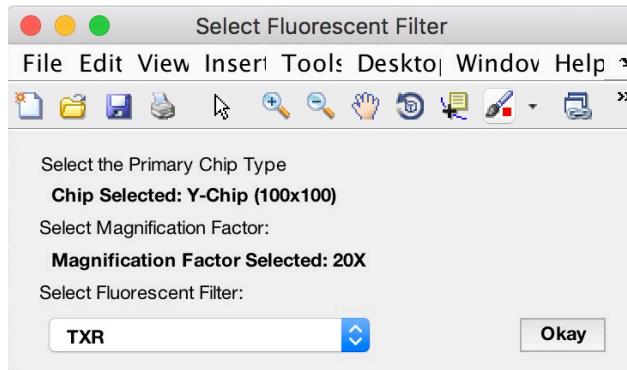
You will first be prompted to select a supported chip type in a drop-down menu:



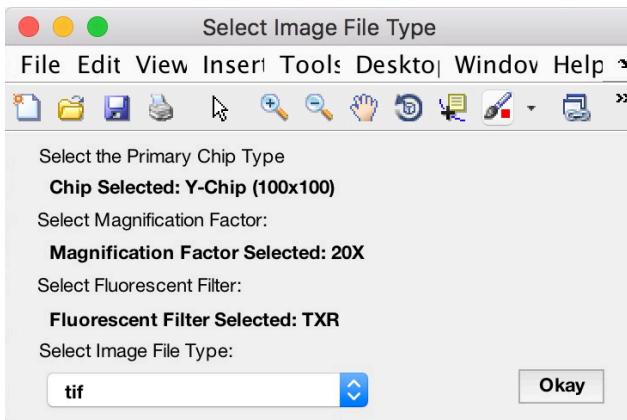
Press **Okay** after selecting a chip type. This will close the current window and open a window with a drop-down asking for the magnification factor. Only certain magnification factors are supported and if available, the magnification factor will be listed in this drop-down menu for the selected chip type:



Press **Okay** after selecting a magnification factor. The window will close and open another window prompting the user for the fluorescent filter type. Select the correct fluorescent filter type from the drop-down menu:



Press **Okay** after selecting filter type. Lastly, select the file type the images were saved as (either .tif or .jpg)<sup>2</sup>.



Press **Okay** after selecting image file type.

The user will be prompted to select the folder where the raw MetaMorph are saved. This folder should contain both the brightfield and fluorescent images. On a Mac, a window will pop up where the user should select the folder without a formal prompt (example image shown on following page). On Windows, this title bar will ask the user to select the folder.<sup>3</sup>

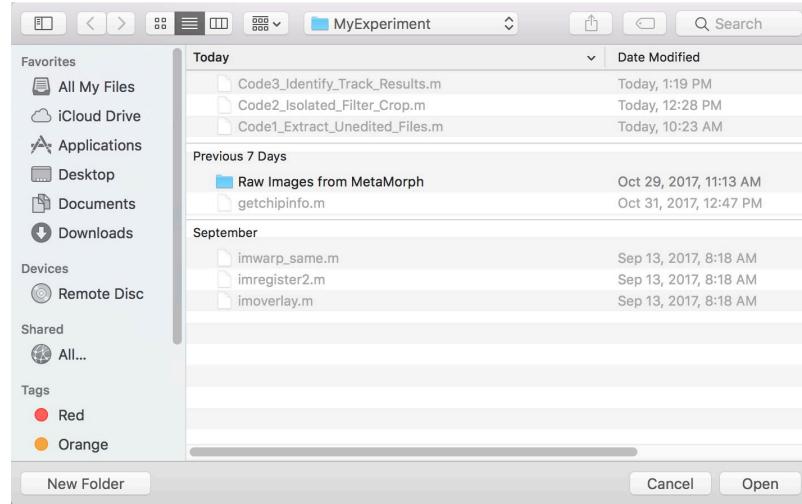
Lastly, the user will be prompted to either delete or keep the original MetaMorph folder. Note that all images are saved and no images are lost, however deleting the original folder can save space if original folder is large. Keeping or deleting the original folder has no effect on results or processing speed.




---

<sup>2</sup> Almost always raw images from MetaMorph will be saved as .tif files. If they are not, they probably should be for image quality and best resolution.

<sup>3</sup> Outside of another pop-up dialog box on a Mac, it was easier to just have the pop-up window. There was no great way to prompt the user of a Mac, and on Windows the title bar was the easiest place to put this prompt.



In the photo shown above, the user would choose the folder “Raw Images from MetaMorph” and click **Open** to select the original from Metamorph on a Mac.

At this point, the code will separate the brightfield and fluorescent images into separate folders, respectively called “BFUnedited” and “FluoroUnedited”.

6. (OPTIONAL) If the original brightfield images appear to be somewhat misaligned with each other (easily checked by clicking through brightfield images), an alignment step may be necessary – this process is done through ImageJ. If images appear to be well-aligned, an initial alignment may not be necessary and skip to step 7. If an alignment is desired, follow the following instructions:

### **Alignment in ImageJ**

- i) Open ImageJ, and upload the brightfield images from the “BFUnedited” folder as an image sequence. To do this, go to ImageJ and select **File → Import → Image Sequence**. Navigate to the “BFUnedited” folder, and:
  - If on Mac, click on the folder and select **Choose**.
  - If on Windows, open the folder and select first image in folder when sorted by name (image with st\_000\_apt\_000 is first). Click **Choose**. Click **OK** in following menu with only “Sort Names Numerically” checked.
- ii) After the images are finished loading, select a large rectangle over most (middle  $\frac{3}{4}$  of image generally works) of the FIRST image (st\_000\_apt\_000) in the image sequence, which should have apartment complex 00/00 in the lower left. Go to **Plugins → Template Matching → Align slices in stack**. Click **OK** twice. Let run
- iii) Once finished, save the **Results** table (NOT the Displacement Log) in a temporary place (e.g. the Desktop), by clicking **File → Save As**. It will save as an Excel file. Open the Excel file (click “Yes” to the warning), and resave it with the name "DisplacementLogIso" as a .txt file. Open that .txt file and DELETE THE FIRST LINE WITH TEXT. Drag and drop that .txt file into the working directory in MATLAB. If no file given or name isn't exactly "DisplacementLogIso.txt", no correction will be made and images will not be aligned, though the code can still run to completion<sup>4</sup>.

---

<sup>4</sup> I wish this was simpler, however reading from an Excel with text is inconsistent, and the process outlined here is tested to work.

	A	B	C	D
1	Slice	dX	dY	
2	1	2	0	-1
3	2	3	-1	-4
4	3	4	-1	-5
5	4	5	-1	-4
6	5	6	-1	-8
7	6	7	-1	-5
8	7	8	-1	-3
9	8	9	-1	-4
10	9	10	0	-2
11	10	11	0	0
12	11	12	0	4
13	12	13	0	4
14	13	14	0	5
15	14	15	0	2
16	15	16	0	0
17	16	17	0	0

Excel Results File

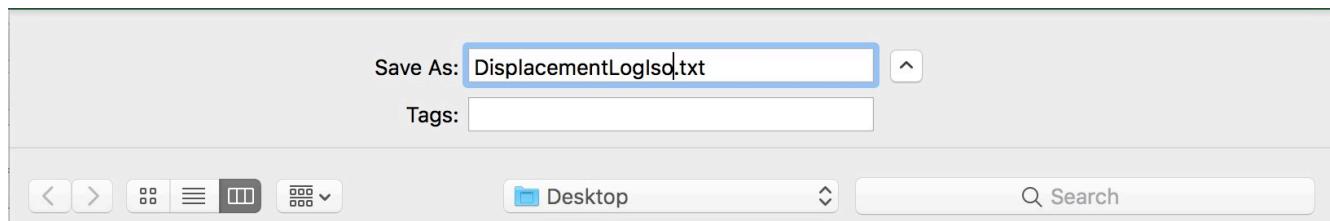
The Excel file will have first row with column headers, and starting at row 2 will have N-1 rows where N is the total number of brightfield images.

Column 1: Brightfield image number in stack.

Column 2: Image within stack that will be matched to image in column 1. Will always be 1 higher than entry in column 1 within the same row.

Column 3: Number of pixels that the image in column 2 is shifted horizontally to match image in column 1.

Column 4: Number of pixels that the image in column 2 is shifted vertically to match image in column 1.



How to save the Excel file as a .txt file for the first alignment.

DisplacementLogIso.txt			
1	Slice	dX	dY
2	2	0	-1
3	3	-1	-4
4	4	-1	-5
5	5	-1	-4
6	6	-1	-8
7	7	-1	-5
8	8	-1	-3
9	9	-1	-4
10	10	0	-2
11	11	0	0
12	12	0	4
13	13	0	4
14	14	0	5
15	15	0	2
16	16	0	0
17	17	0	0
18	18	0	0

How the Excel file should look after being saved as “DisplacementLogIso.txt” as a tab delimited.txt file.

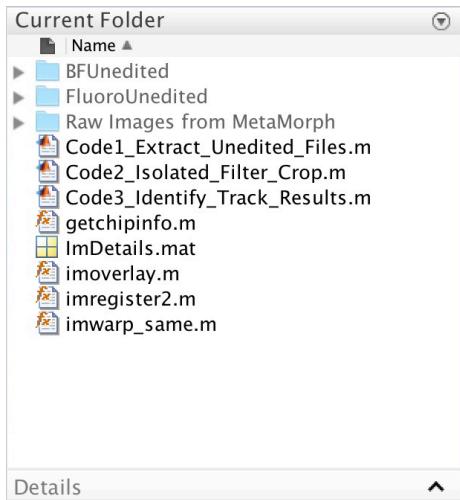
**THE FIRST LINE OF COLUMN HEADERS IS STILL SHOWN AND SHOULD BE DELETED**

DisplacementLogIso.txt — Edited			
1	2	0	-1
2	3	-1	-4
3	4	-1	-5
4	5	-1	-4
5	6	-1	-8
6	7	-1	-5
7	8	-1	-3
8	9	-1	-4
9	10	0	-2
10	11	0	0
11	12	0	4
12	13	0	4
13	14	0	5
14	15	0	2
15	16	0	0
16	17	0	0
17	18	0	0

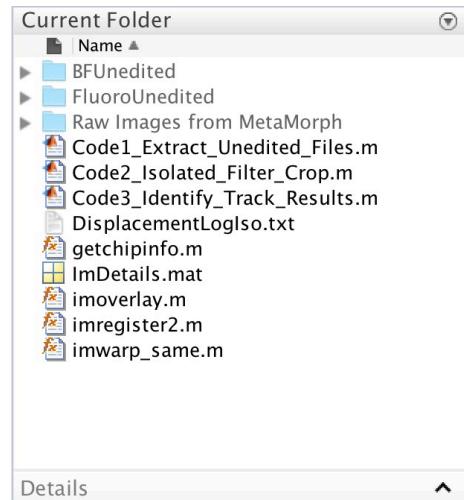
How final .txt file should look *after* deleting the first line.

**FIRST LINE HAS BEEN DELETED AND FILE IS SUTABLE FOR USE**

7. After copying DisplacementLogIso.txt into the working directory, the working directory should look like the following:



Working directory without an initial alignment step.



Working directory with an initial alignment step.

8. Regardless of if an alignment is provided, in the working directory open the code entitled Code2\_Isolated\_Filter\_Crop.m. Click **Run** in the Editor tab.
9. This code will load in all brightfield and fluorescent images. The user will be shown a fluorescent image and be asked if any white dots (speckling) are present. If present, the image will be filtered to account for this, and the user should click **Yes**. If there is no speckling, select **No**. The images will then be filtered appropriately for cell counting. An example of an image with and without speckling are shown in Appendix A.
10. The user will be prompted to select a region of an apartment complex such that the code can properly isolate and crop around each apartment complex. Follow instructors below for appropriate chip type:

For the Full P-Chip or 7-Condition chip this will be the alignment mark near the trifurcation. Select the appropriate alignment mark based on the prompted message. Remember that the first number indicates the street (columns), and the second number indicating the apartment (rows) (e.g. apartment 01/00 is the second column, lower most complex). See Appendix B for pictorial representation.

For the Y-Chip, select the center of the apartment complex for complex 00/00. The user is then asked to select the center of the apartment complexes 01/00, 02/00, and 03/00, with the first number indicating the street (columns), and the second number indicating the apartment (rows). The apartment complexes listed refer to the lowest complexes in each column, going from left to right. Lastly, the user is prompted to select the center of apartment complexes 00/01 and 00/02, or the center and uppermost complexes in the left column within the image. See Appendix B for pictorial representation.

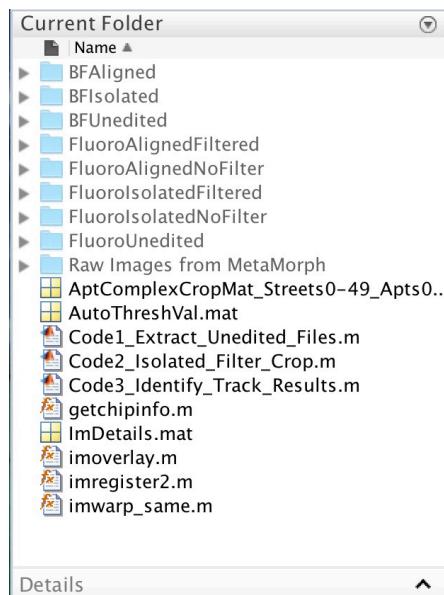
Upon clicking all necessary points, all windows will close and the code will isolate and save individual brightfield and fluorescent images of each apartment complex.

- Follow the process in step 6 for alignment with ImageJ, except now if:
  - on Mac, find the folder “BF Isolated” and select **Choose**.
  - on Windows, open the folder “BF Isolated”, select the FIRST image in the folder, which will be complex 00/00. Click **Choose**. Click **OK** in following menu with only **Sort Names Numerically** checked.

### **Critical Alignment Step in ImageJ**

Starting in step 6 ii), follow the same procedure, except save the Excel file with the name "DisplacementLog" as a .txt file (**THIS IS IMPORTANT!**). Once again, DELETE THE FIRST LINE WITH TEXT, and drag and drop the edited .txt file into the working directory in MATLAB. If no file given or name isn't exactly "DisplacementLog.txt", no correction will be made and images will not be aligned, though the code can still run to completion. This step is critical to accurate tracking and failure to complete it will not yield good results!

The working directory should now look like the following:



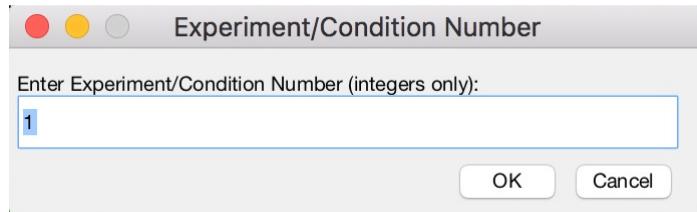
- Regardless of if an alignment of single apartment complexes is provided (though it should be for best results), in the working directory open the code entitled `Code3_Identify_Track_Results.m`. Click **Run** in the Editor tab.
- Follow the user input prompts that will automatically open as follows<sup>5</sup>:

Experiment number: allows the user to denote an individual experiment from another if more than one experiment was performed on a given day<sup>6</sup>.

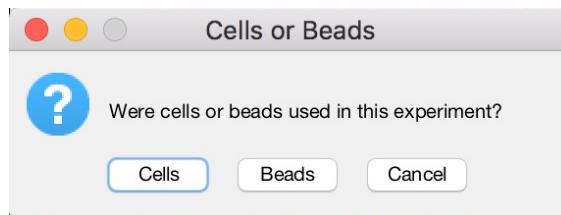
---

<sup>5</sup> If the user selects **Cancel** at any point, code will need to be run fully again.

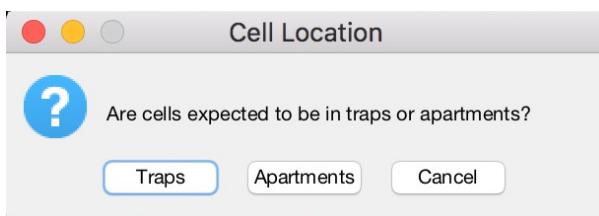
<sup>6</sup> **Important Note:** If the user wants to run the same data set again after all processing has finished but with different analysis parameters, click **Run** `Code3_Identify_Track_Results.m` and change the experiment number. This will allow the user to fully run the same data set again without overwriting any previous processing. If the experiment number is the same as a previously processed data set, the user will be prompted asking if they want to overwrite this data set. Once a data set is overwritten, it will be permanently overwritten – there is no backup storage. For more information, see Post Processing.



Choose whether cells (no matter the kind) or beads (fluorescent polystyrene particles) were used:

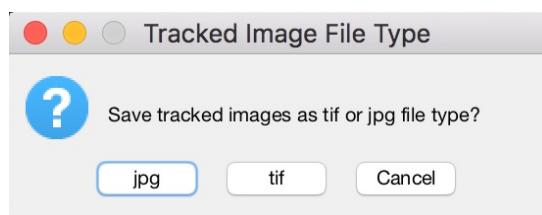


Choose where cells/beads are expected to be found (title of question box changes based on if cells or beads are chosen previously).



Choose the file type that all images generated through MATLAB will be saved as. Tracked images are those produced after tracking cell(s)/bead(s) location(s), overlaying fluorescent false coloring, and indicating if a cell was found in specified regions.

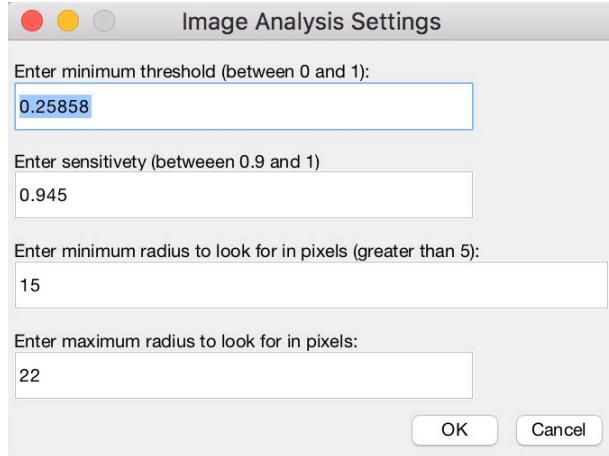
NOTE: Saving as .tif will use much more space than saving as .jpg (.tif file size can be a 5x larger file size, which over thousands of images can be use a large amount of hard drive space).



Choose the image analysis parameters that will be used for tracking the cells/beads. These are critically important for accurate tracking, and default values represent first approximations<sup>7</sup>. They are listed in order of variability that they may change between data sets from top to bottom.

---

<sup>7</sup> If still unsure about what to put for these values even after reading through following page, see the ranges for where these values may lie for a data set similar to those presented. However, these values often need tuning to give best results but should remain similar once determined for a single data set providing all other imaging setting remain unchanged.



Threshold: Used for binarizing fluorescent images, a necessary process to track cells/beads. Pixels range in values from 0 (black) to 255 (white), and the threshold value as it is displayed represents a value that multiplied by 255 is the cutoff to make pixels above it white, and those below it black. This value should generally never be above 0.4, and is commonly 0.3 to 0.4. Needs to be between 0 and 1, else an error is thrown. Default value is calculated using MATLAB's built in `graythresh` command. Cases for default value:

- If lower than 0.3, manually change to 0.3.
- If above 0.3 and below 0.4, leave as is.
- If greater than 0.4, code automatically makes default 0.4.
- If value is close to or less than 0.2, cells may not be bright enough to analyze. Use best judgement, and raise threshold to 0.25. Results may not be accurate due to dim cells if the default value is this low.

Sensitivity: Determines how sensitive the tracking process is while finding cells/beads<sup>8</sup>. Needs to be between 0 and 1, with 0 meaning it will never find anything unless it is a perfect circle, and 1 being it will find many, many circles (including partially visible circles) and increases risk of false detections. Value should be between 0.9, and 1, though really should never be above 0.95 for most analyses. Generally, doesn't need to be changed, but can be raised or lowered slightly (down to 0.94 or up to 0.95) if necessary.

Minimum and Maximum Radius: These are the minimum and maximum radii that the tracking algorithm uses when finding circles. These values are in pixels (not microns), thus tuning needs to be done initially for each chip type and magnification<sup>9</sup>. These may rarely be changed (though likely not ever by a user) and are pre-tuned such that unless the cell type or bead size changes significantly, they are accurate.

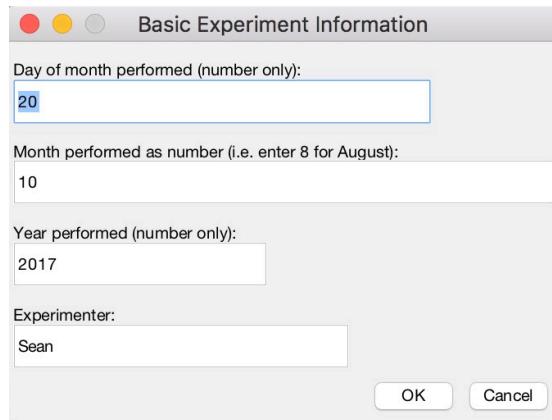
Click **OK** after entering values.

---

<sup>8</sup> This is not the same as eccentricity, though it is similar. The algorithm `imfindcircles` uses a circular Hough transform, which will find circles based on eccentricity (more or less) for a given sensitivity, but then report the corresponding circle and radius it finds, and not an arbitrary shape it may detect while looking for circles.

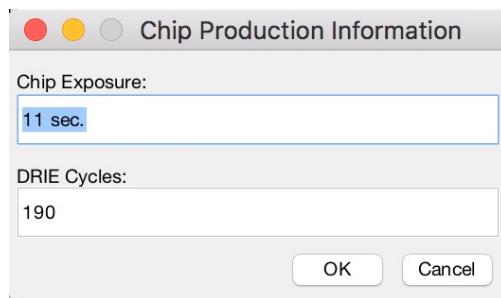
<sup>9</sup> Values are not in microns because the pixel/micron ratio changes between chip types and magnification factor, thus a cell/bead (say of 10 µm diameter) in one chip imaging setup may have a radius of 16 pixels, 8 in other, and 24 in another. Because this ratio is not constant, there is no conversion factor that can be universally used.

Enter date of experiment (default is current date), and who performed the experiment:



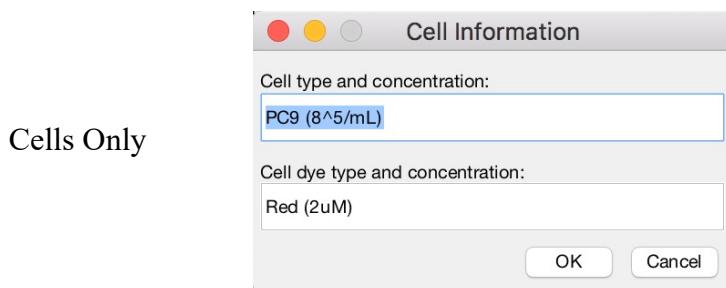
Click **OK** after entering values.

Enter the chip exposure time from Photo2 and number of cycles from the DRIE:

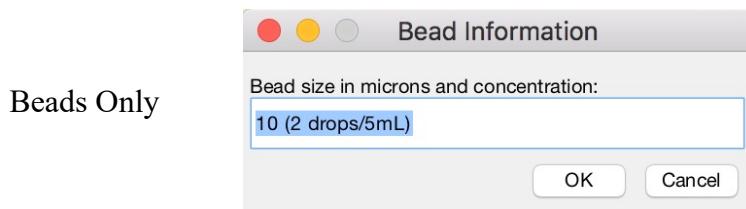


Click **OK** after entering values.

If cells were selected, enter the enter the cell and dye type and concentrations:

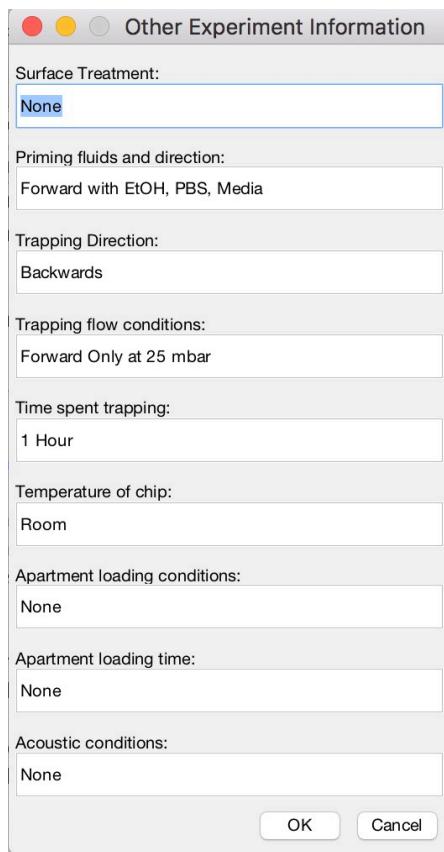


If beads were selected, enter the enter the bead size and concentration:



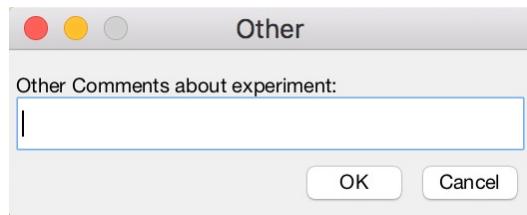
Click **OK** after entering values.

Enter other experimental information:



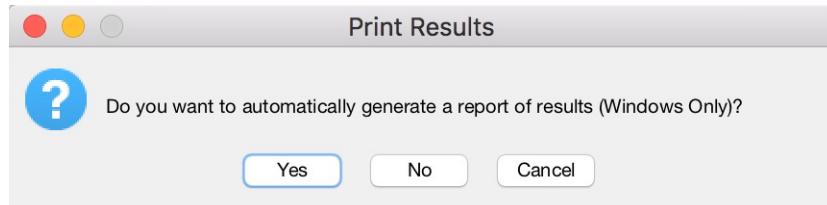
Click **OK** after entering values.

The user will then be asked to enter other comments that someone should know regarding this data set (e.g. light changed during imaging, cells looked unhealthy at start, chip was dirty in location xx/xx, etc.).



Click **OK** when finished, regardless of if other comments were given or not.

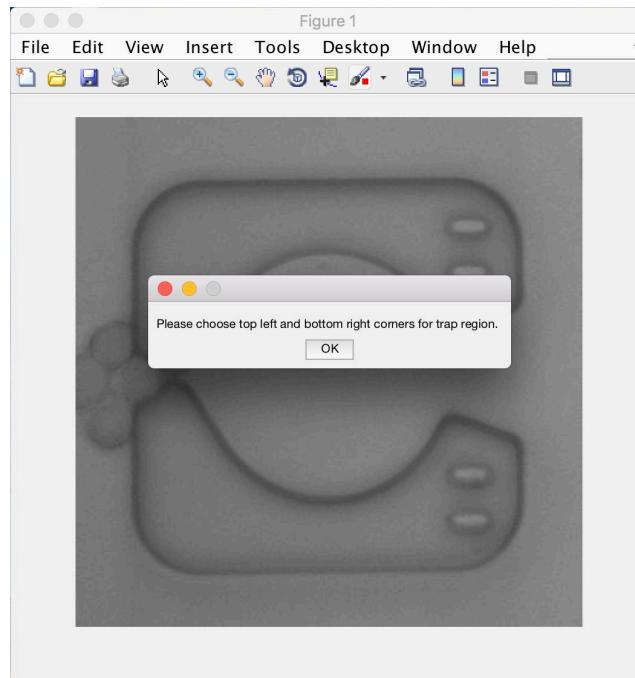
If the user is running a Windows OS (either on a native hard drive or through a Virtual Machine), they will be prompted about if they want to automatically generate a Word report of the data processing. Select **Yes** if desired (encouraged). This choice will not affect the data processing and final figures created.



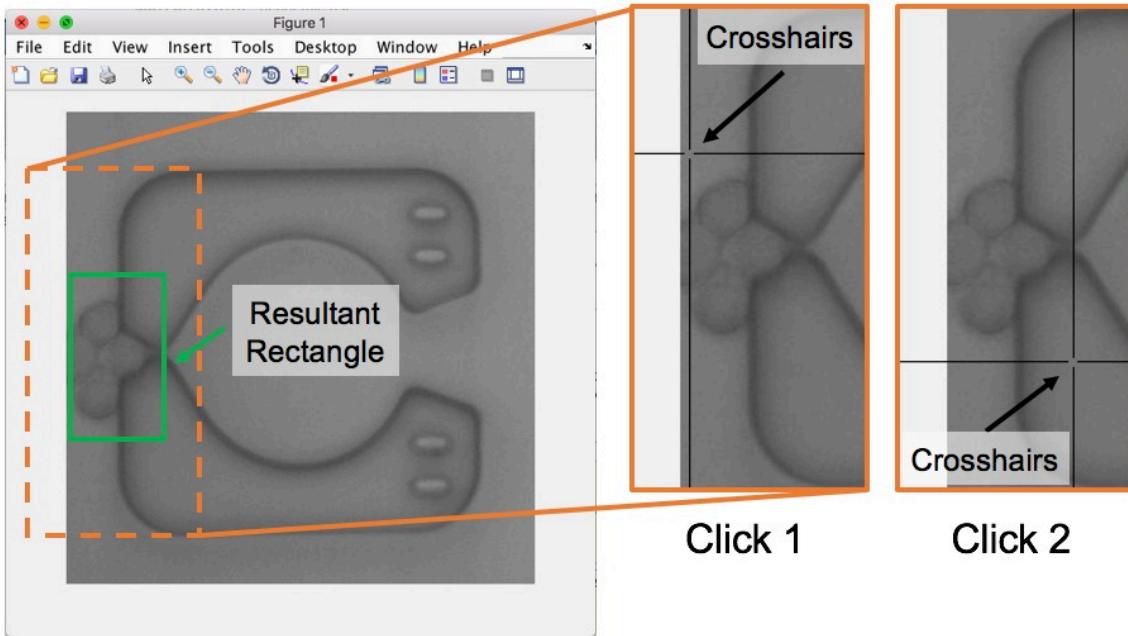
If on a Mac OS, if prompted with this message, click **No**, else an error will occur with the code. See FAQ if this occurs to remedy the situation. This feature has been finicky depending on the machine and Window has only been tested on machine's using Windows 10.

14. The user will be prompted to select the top left and bottom right corners for the trap region, meaning that the user is selecting the top left and bottom right corner of a rectangle that will be used for determining if cells/beads are in the trap. Click **OK** to proceed to selecting this region.

A figure of a single apartment complex will be shown. Select the region to be somewhat generous so that stacking of cells/beads in the trap can be accounted for in efficiency results. Two clicks are required and (at the moment), no user feedback is given after the first click. Crosshairs are added for ease in finding selection locations. After the second click, the figure window will close.

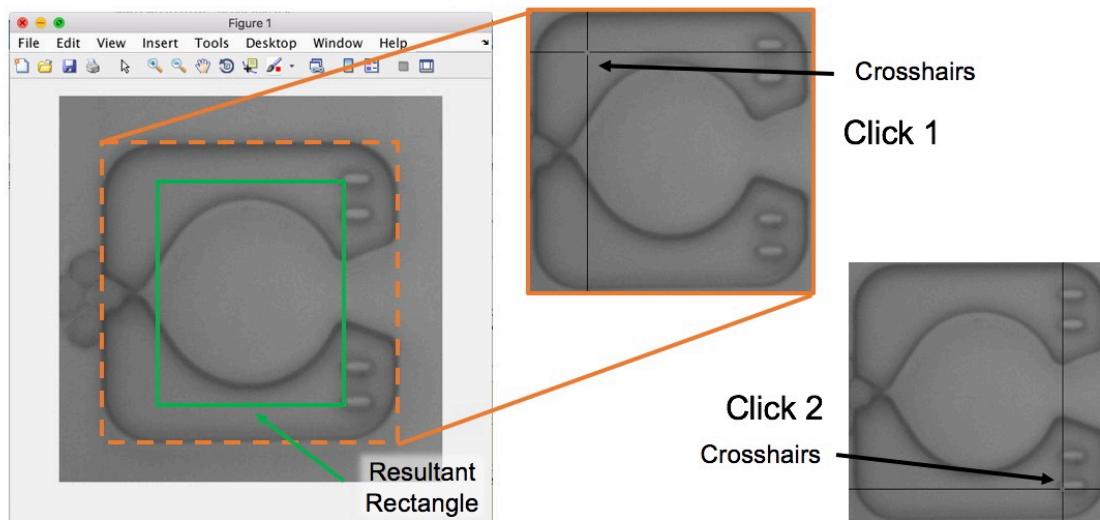


An example selection process is shown below. Click 1 is top left of rectangle, click 2 is bottom right of rectangle. Note that green rectangle is shown only for visualization in this image, and is not shown when running the code.



15. After the figure window closes from step 14, the user will be prompted to select the top left and bottom right corners for the apartment region, meaning that the user is selecting the top left and bottom right corner of a rectangle that will be used for determining if cells/beads are in the apartment. Click **OK** to proceed to selecting this region.

Like step 14, a figure of a single apartment complex will be shown. Select the region so that it only includes the apartment region, with a slight border<sup>10</sup>. If the apartment region is circular, make sure this rectangle will include most of the apartment. Two clicks are required and (at the moment), no user feedback is given after the first click. Crosshairs are added for ease in finding selection locations. After the second click, the figure window will close.




---

<sup>10</sup> A slight border around the apartment crop region will help to detect cells that may only be partially visible in the cropped region.

16. After the figure window closes, and the code will begin the analysis. A new figure window will open showing the brightfield image with fluorescent false coloring that results from binarizing the image, along with boxes corresponding to the trap and apartment regions selected in steps 14 and 15 and red circles around cells/beads that were tracked. This image will update somewhat quickly as the code processes each image through a for loop. These images are saved in a folder entitled “Tracked\_Images\_Exp#”, where # refers to the experiment number entered in step 13. The brightfield images alone with fluorescent overlay are saved in the folder “BFEdited\_Images\_Exp#”. These images can be used for stitching.

The box line color for the trap and apartment regions selected will be red if there was (were) no cell(s)/bead(s) found in that region, and green if there was a (were) cell(s)/bead(s) found in that region.

NOTE: To increase the processing speed, make the figure window very small and minimize it. The only consequence of this is that tracked images are saved at the size a user makes the figure window<sup>11</sup>.

IMPORTANT: If you see that the image analysis parameters don't appear to be accurate, be careful with wanting to change the analysis parameters right away. Often the first few streets will have the most build-up of beads/cells, which are also the most difficult to track accurately. Wait until the code has processed at least four or five streets and if then the code appears to be missing many beads/cells it should be able to pick up, then the image analysis parameters need to be changed.

To force-stop the MATLAB code, then:

- if on a Mac, press **control+C**
- if using Windows, press **Ctrl+C**

Click **Run** again for `Code3_Identify_Track_Results` and you will start the entire analysis over again, and will need to follow all prompts. Change the image analysis parameters as desired and repeat this process as necessary.

17. Once the tracking has finished, the figure with tracked images will close, and the following figures will be automatically generated and displayed (in order):

- Trap Heat Map: displays number of cells/beads found in defined trap region.
- Apartment Heat Map: displays number of cells/beads found in defined trap region.
- Bypass Heat Map: displays number of cells/beads not in defined trap or apartment regions.
- Row and Street Efficiency Line Plots: The number of cells/beads are found along each street and row, and efficacies of single, double, and three+ are found and displayed as line plots within one figure.
- Brightness Histograms: Three histograms of the average, maximum, and total cell/bead intensities across all cells/beads tracked within the entire chip within one figure.

These figures may take a few minutes to be created. After finishing generation, if using a Windows OS and the user selected to create an automatically generated report of the results, MATLAB will automatically open Word and create the document. The user will be asked if they want to save the report.

---

<sup>11</sup> To save tracked images, the code takes the current MATLAB figure and saves it directly, including the size of the figure window. It unfortunately does not save this image as the original size of the image it is showing.



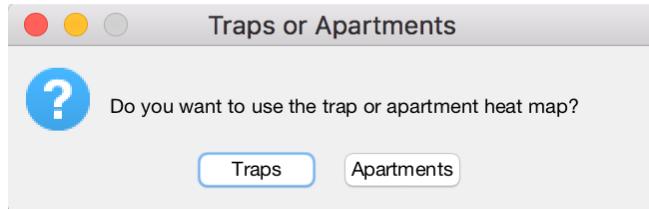
Upon selecting **Yes**, the user will be asked where to save the report. The report name is already pre-determined including the type of chip, date performed, and the experiment number.

If the user selects **No**, the report will remain open but unsaved. It can be saved manually still.

18. The user will be prompted if they want to view any specific traps or apartments. If the user selects **No**, the program is terminated and the primary analysis is complete.

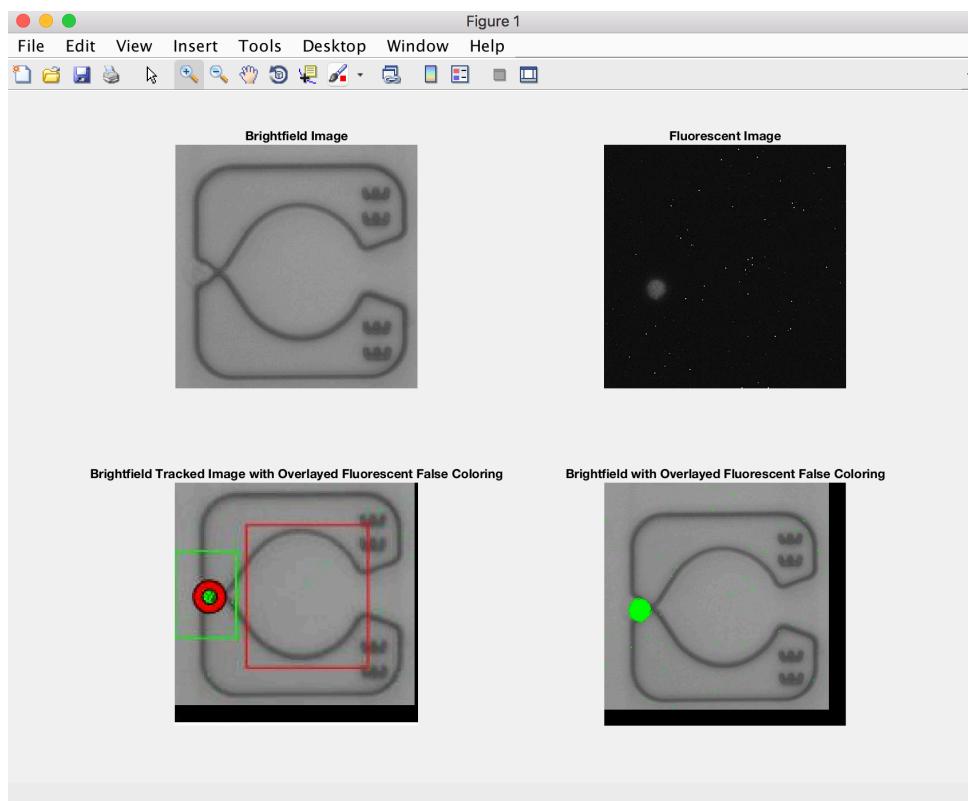


If the user selects **Yes**, they will then be prompted asking whether they want to view the trap or apartment heat map in order to select a complex they wish to view.



Depending on if **Traps** or **Apartments** are selected, the corresponding heat map will open after clicking **OK** through on a pop-up menu. Click on the trap or apartment complex you would like to see images for (crosshairs are added for guidance), and the heat map figure will close. This will open a figure window showing the original brightfield image (top left), the original fluorescent image (top right), the tracked image (bottom left), and the fluorescent image overlaid with false coloring (bottom right).

An example is shown below:



## Interpretation and Troubleshooting

Based on the original brightfield and fluorescent images, it is up to the user to determine if there are one or multiple cell(s)/bead(s) in the trap/apartment regions, and then based upon the tracked image, deduce if the image analysis parameters were satisfactory and/or if that image of tracked correctly.

If the false colored fluorescent image shows little fluorescence but the cell/bead appear bright in the original image, the minimum threshold for binarizing the fluorescent images was too high. In general, if there is little or no false color fluorescence but a cell/bead is clearly present, the minimum threshold was too high.

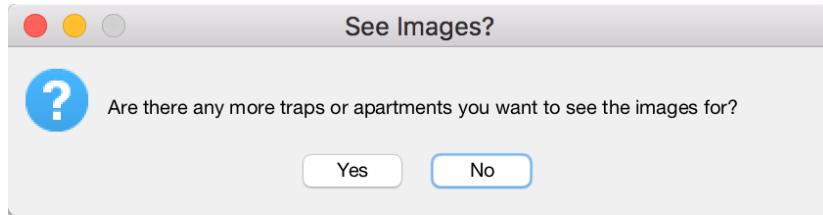
If the original brightness appears to be very low and too much fluorescence appears in the false colored image, the minimum threshold was too low. This is often a problem when cells/beads are too dim, and can result in more incorrect identifications than desirable than if cells/beads are too bright.

If the original fluorescent image is clear and has one or more bright cells/beads and the false colored fluorescent image appears to match the original fluorescent image well, but the cell/bead was not tracked, the sensitivity is too high, and needs to be lowered.

If too many cells/beads appear to have been tracked, the sensitivity may be too high. This should be checked with other images if this is the case to be sure.

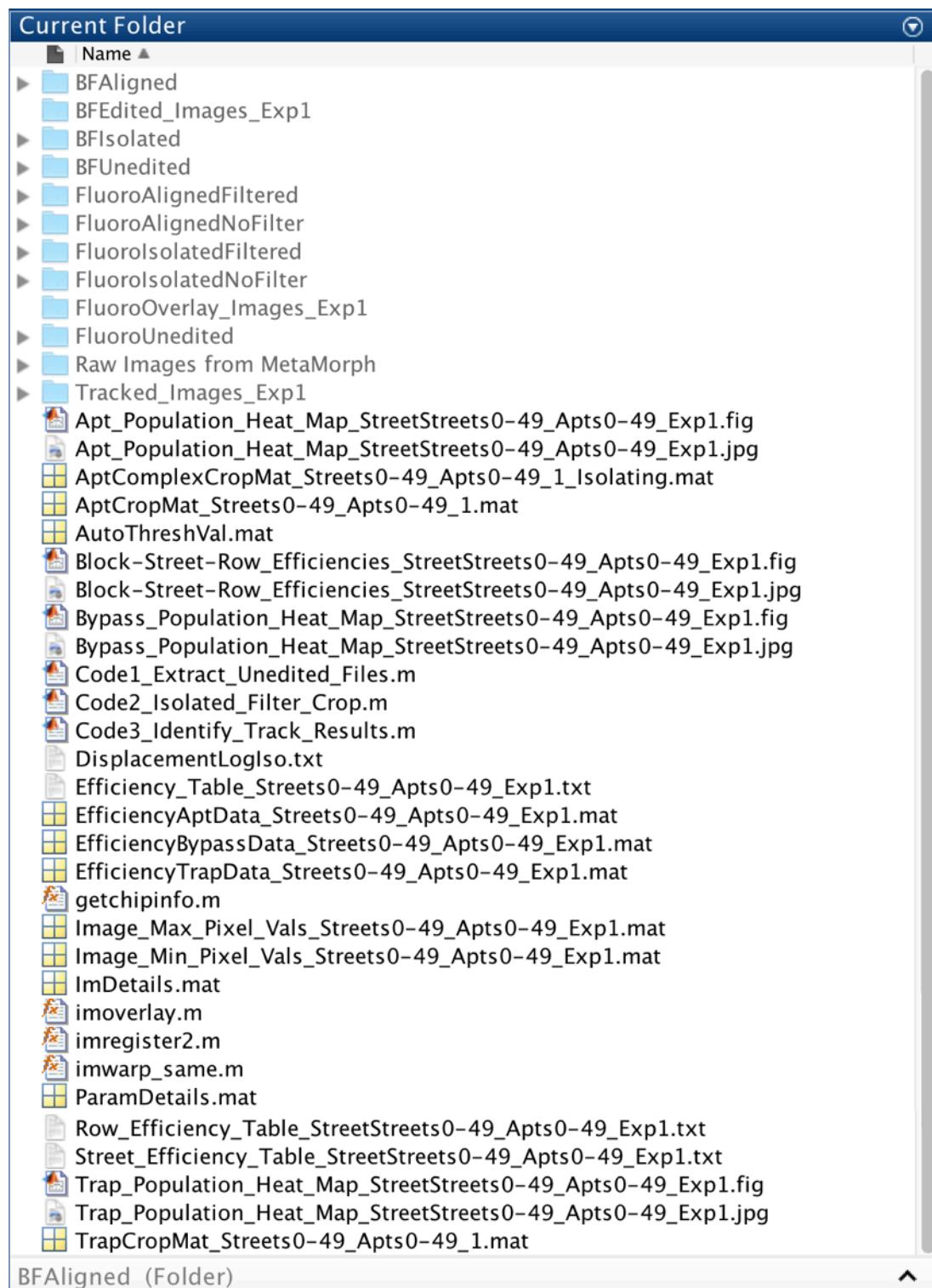
It is important that if any issues appear present to check this across multiple complex to see if that complex was not tracked properly, or if the data overall was not analyzed correctly and needs to be re-analyzed!

Once done, the user will be asked to click **OK** to close this figure window. They are then asked if they would like to view any more traps and/or apartments.



If **Yes**, this step will repeat back again to this window, or unless the user closes the figure window improperly, in which an error will occur. This does not affect the data and simply terminates the program. If **No**, the program is terminated and the primary data analysis is concluded.

Once the program has concluded the primary analysis, the working directory should look like the directory shown on the following page. Note that heat maps have been saved both at MATLAB figure files (.fig) and as .jpg image file types. To open .jpg file types, right click on the image in the working directory, and select **Open Outside MATLAB**. Simply double-clicking will not open .jpg image file, but will open a .fig file type.

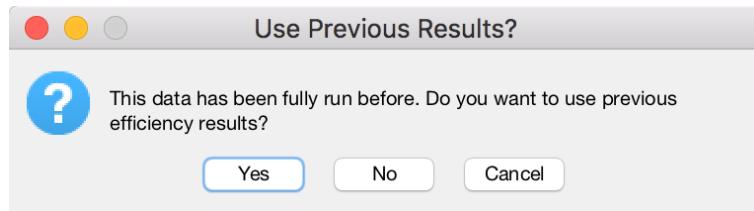


## POST-PROCESSING ANALYSIS

While most of the analysis available occurs during the primary analysis the first time processing a data set, there are some options only available after a data set has been fully processing once and efficiency results are available. These options are given below.

**Custom Region Heat Maps and Efficiencies** – allows the user to define a custom solid rectangular region of complexes and creates trap, apartment, and bypass heat maps, as well as extract efficiency data from the defined region

1. For a data set that has already been processed, open the version of `Code3_Identify_Track_Results.m` that was used to process that data set, and click **Run**.
2. Enter the experiment number that is associated with the previously processed data set.
3. If this data set has been fully processed, the user will be prompted if they want to use the efficiency results from the previously processed data.



Select **Yes**. If the user selects **No**, this will mean they intend to overwrite the data associated with this experiment number. If **No** is selected, the user will be notified that they are about to overwrite data with the following prompt:



Selecting **Yes, process again** will take the user back to step 10 in the primary analysis immediately after they would have selected the experiment number, and images associated with this data set will be processed again. Doing this will immediately begin to overwrite data and the previous processing iteration will not be recoverable after proceeding even part way through the primary analysis.

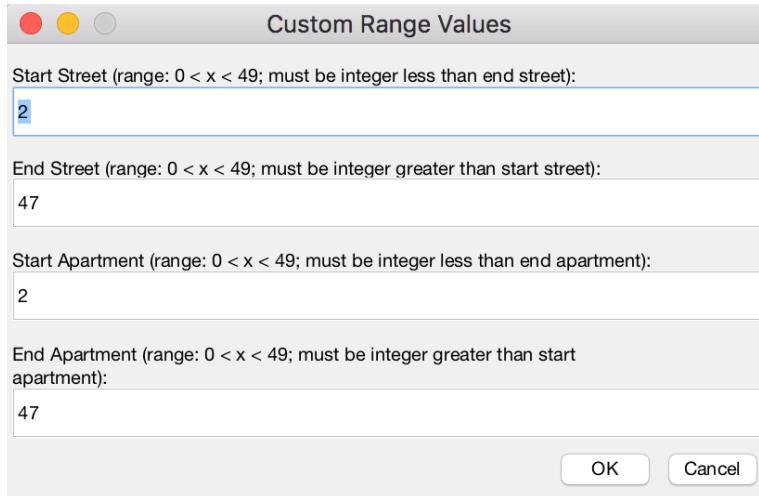
Selecting **No, don't overwrite** allows the user to continue using the previous efficiency results.

4. Provided that the user intends to continue and has not decided to overwrite previous data, the user will be asked if they wish to view a custom region of this data set. The image below was associated with a Y-Chip (50x50), but depending on the chip type, the proper name associated with the data set will be substituted appropriately.



Select Yes.

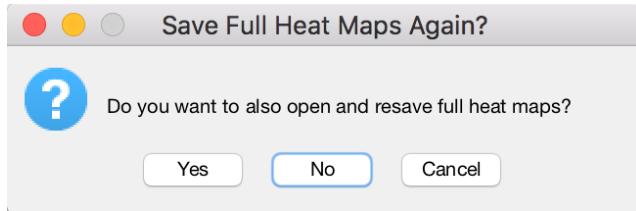
- The range values to produce the rectangular custom region need to be entered in the appropriate user input dialog box.



The default values are chosen such that an outside border of two apartment complexes thick will be removed if the user chooses to use these values. Often the outermost complexes and the start and ending complexes tend to have the worst efficiencies throughout a chip, and excluding them from the efficiency results can increase the single-cell capture efficiency by more than 1%<sup>12</sup>.

Values that are eligible must be within the ranges specified by the prompt about each dialog box. If the input is not valid, an error will be thrown and the user will need to input correct values for values which were improper.

- The following prompt asks if the user want to open the full heat maps again (say, for comparison to the custom defined region).



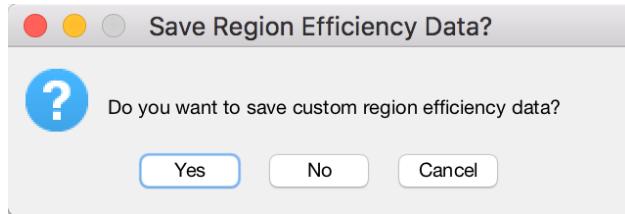

---

<sup>12</sup> We have seen this even increase single-cell trapping efficiencies by more than 4%.

Select **Yes** to reopen these full heat maps through MATLAB, or select **No** to not view these full heat maps. Either option does not affect the custom region results and heat map creation.

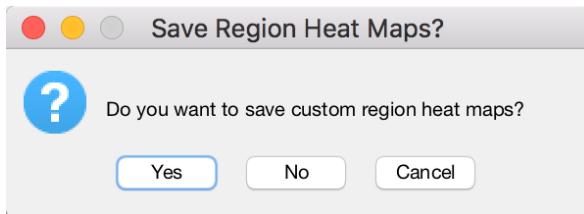
7. The trap, apartment, and bypass heat maps will now be generated, and efficiency results displayed in the Command Window. Do not close these heat maps while they are being generated and/or saved to not cause the code to have an error or possibly cause a problem saving these heat maps.

A window will open asking if the user wants to save the efficiency results from the custom region to a table.



Select **Yes** or **No** as desired. The table will be saved with a name indicating that it contains efficiency results for the region defined with values entered in step 5.

8. Lastly, a window will open asking if the user want to save the custom region heat maps.



Select **Yes** or **No** as desired. The heat maps will be saved with a name indicating that they represent the heat maps for the region defined by the chosen values in step 5.

The user will be prompted if they want to view any specific traps or apartments as was done at the end of the primary analysis. See step 18 of primary analysis operation instructions regarding this option.

#### **Process to Create Another Custom Region** – for if the user wants to create more custom regions

After the code is finished running after creating the first custom heat maps through the process described previously, simply click **Run** again for the same code, follow the prompts making sure not to overwrite previous data and once reaching step 5, enter the desired region. This process can be repeated for as many custom regions as the user desires.

#### **Create MATLAB Generated Word Document if Not Initially Selected to be Created** – if the user initially selected to not create the automatically generated Word document on a Windows OS, this process will guide the user through how this document can be created for a data set that has already been processed

While short, this process involves going into the code and running a specific section. This will only work if the following are true:

- The last time `Code3_Identify_Track_Results.m` was run it was completed, and was not run and then terminated before completing.<sup>13</sup>
- The data set that a Word report is desired for has been fully processed.
- The computer is running on a Windows OS.
- All figures saved as .jpg have not been deleted.
- Efficiency data matrix has not been deleted.

Provided that the user has not deleted anything from the working directory, the final two points should be true automatically.

1. Provided the above conditions are met, type the following into the Command Window:  
`PrintToWordDoc = 1; CheckforEfficData = 0;`

Press **Enter/Return** on the keyboard.

2. For the data set and processing results for which the user desires to generate a Word document, if not already open, open the corresponding `Code3_Identify_Track_Results.m` used to process the data set originally.
3. Scroll down to the section with commented title section<sup>14</sup>: %% Printing to Word Doc

This section will look like the following when the user clicks anywhere within the section:

```

2510
2511 - %% Printing to Word Doc
2512
2513 - if PrintToWordDoc == 1 && CheckforEfficData ~= 1
2514 - % Re-rounding number so they'll be evenly spaced on Word Doc.
2515 - Num0N = {3, 1}; Num1N = {3, 1}; Num2N = {3, 1}; Num3N = {3, 1};
2516 - for Numiv0 = 1:3
2517 -   if Num0(Numiv0, 1) == 0
2518 -     Num0N{Numiv0, 1} = '0.0000';
2519 -   elseif Num0(Numiv0, 1) < 10
2520 -     Num0N{Numiv0, 1} = sprintf('%0.4f', Num0(Numiv0, 1));
2521 -   elseif Num0(Numiv0, 1) >= 10 && Num0(Numiv0, 1) < 100
2522 -     Num0N{Numiv0, 1} = sprintf('%0.3f', Num0(Numiv0, 1));
2523 -   elseif Num0(Numiv0, 1) >= 100

```

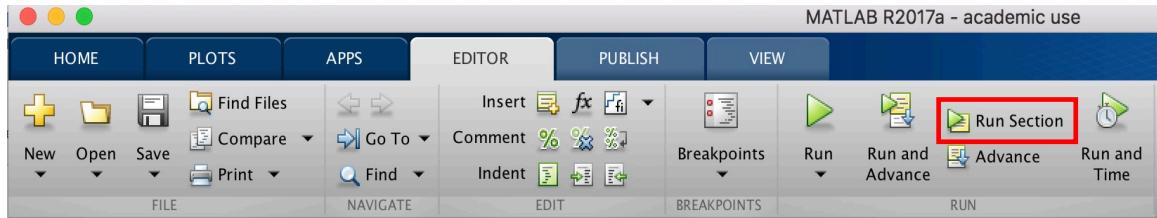
Click anywhere in the section, but do not write or delete anything. If the cursor is within the section, the entire section will be highlighted in a yellow hue. Note that the line number may have changed since this picture was taken.

4. Under the Editor tab in MATLAB, click **Run Section**, shown boxed in red.

---

<sup>13</sup> For this method to work, the code relies on using stored variables from a completed code, some of which are not stored by default. Every time this code is run, all variables are cleared and it pulls from purposely saved variables, hence why some values need to only be entered once, while others need to be entered every time.

<sup>14</sup> A commented title section refers to a MATLAB comment – denoted by a %% – that sections off a portion of a code until another section is started by using a %% comment. These sections will have a yellow hue when the cursor is inside them, and can be run individually without running the entire code.



A Word document will then automatically be created and be compiled with appropriate information concerning the data set.

5. Like if the user originally selected to create a Word document, they will be asked if they want to save the report.



Upon selecting **Yes**, the user will be asked where to save the report. The name is already pre-determined including the type of chip, date performed, and the experiment number.

## APPENDIX A – Example Fluorescent Images with and without Speckling

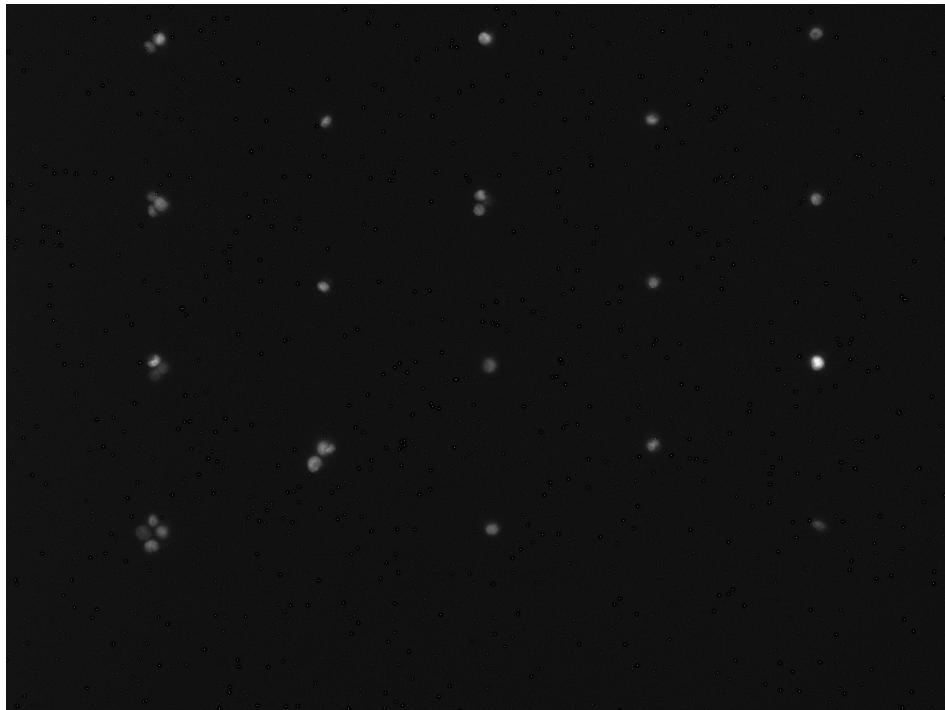


Image with speckling (very faint white pixels that are not cells). Will need to zoom in to see. Speckling will likely not be visible on printed copies.

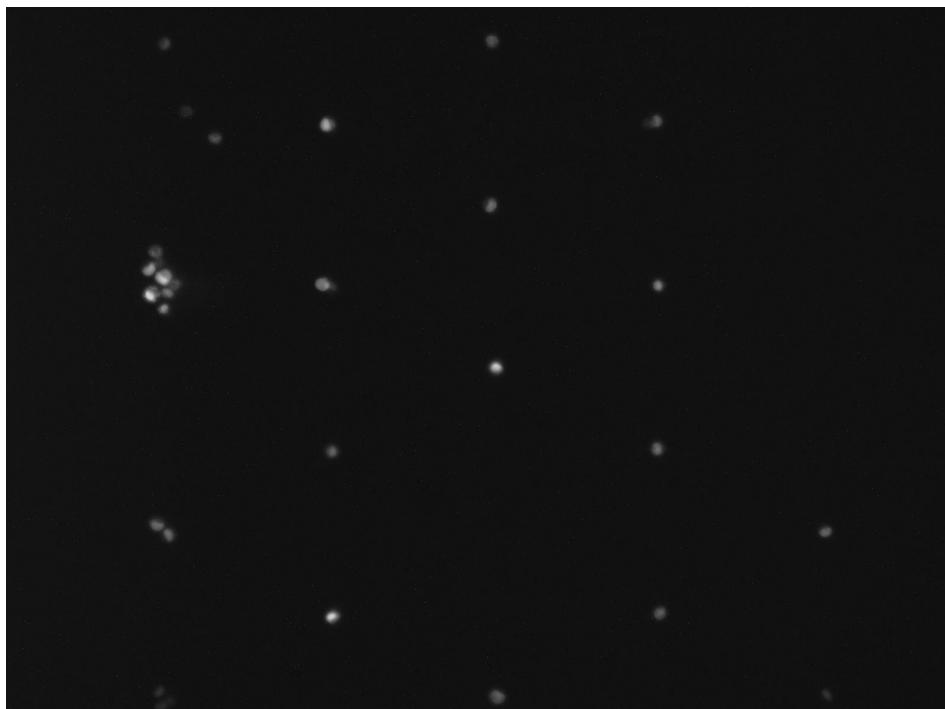


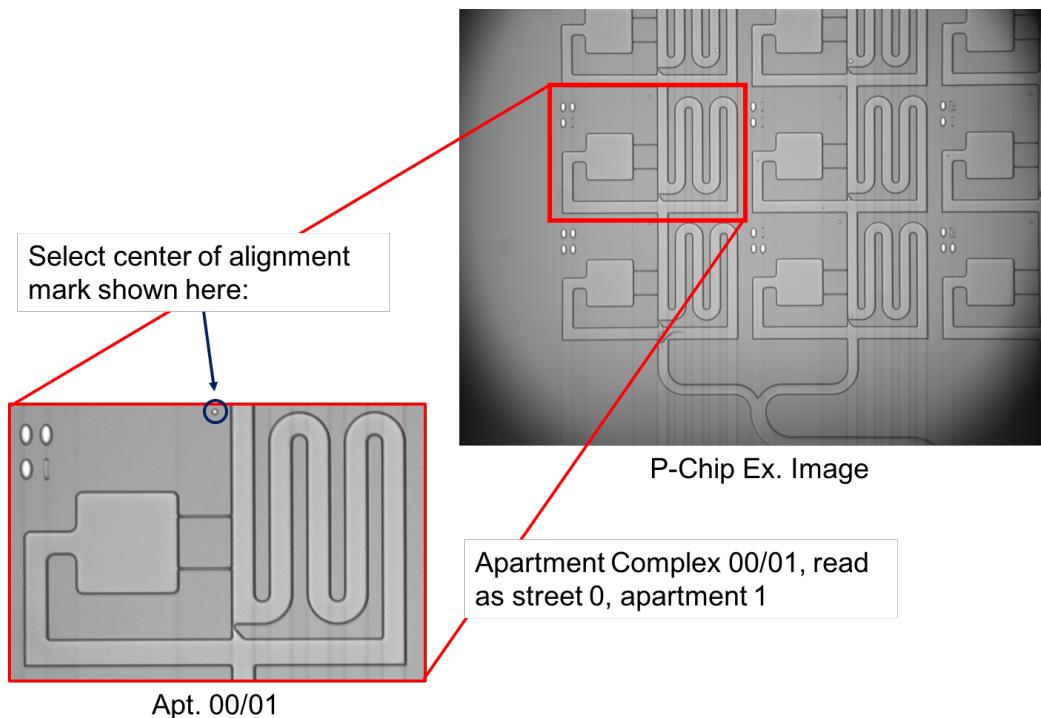
Image without speckling<sup>15</sup>.

---

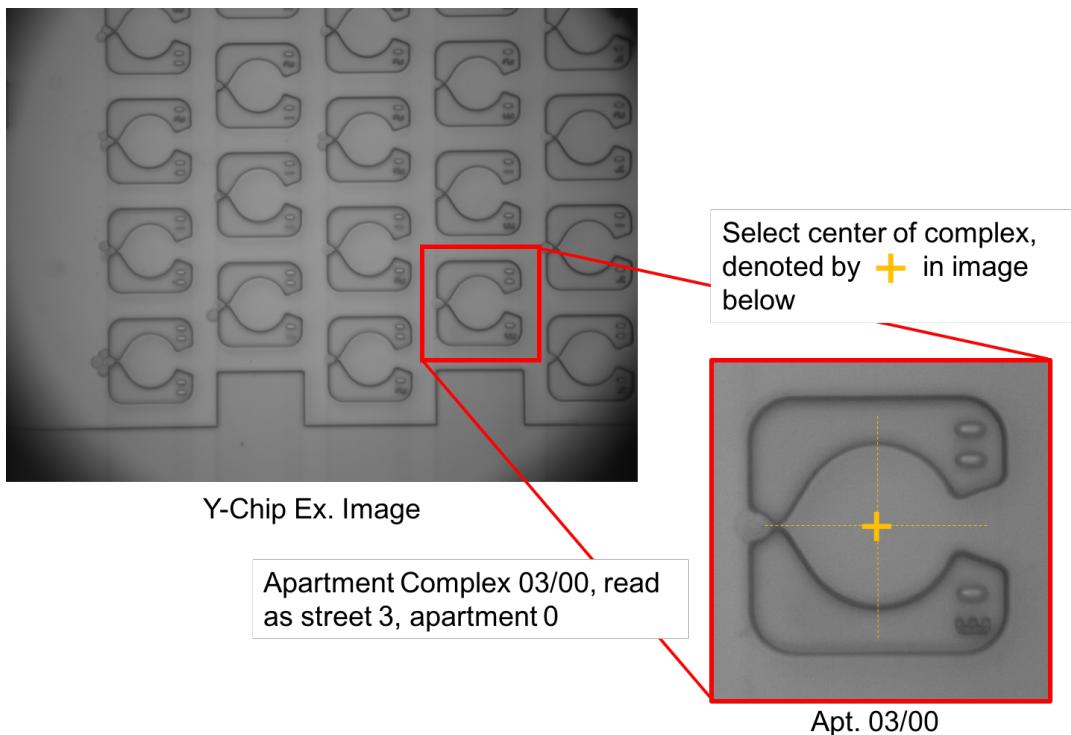
<sup>15</sup> Images shown are not from the same data set.

## APPENDIX B – Example P-Chip Alignment Mark Selection and Y-Chip Center Point Selection

### Example P-Chip Selection:



### Example Y-Chip Selection:



## FREQUENTLY ASKED QUESTIONS AND TROUBLESHOOTING

I have a chip type or specific magnification that is not listed when I enter chip imaging information initially. What should I do?

Unfortunately, if it is not listed, the configuration is not supported and is not applicable to these codes. It should be noted that all instructions for magnification or chip type are based on lab naming convention and would need to be updated for other uses to adapt this protocol for another specific type of microfluidic single cell array configuration and/or platform.

Why is there white speckling in my fluorescent images?

While this hasn't been tested extensively, this can be the result of having too high a gain while taking fluorescent images, with speckling more likely to occur for higher exposure times as well. These speckles will usually appear in the same spots on all fluorescent images.

*But what if the spots are there even for low gain and low exposure?*

Make sure the exposure and gain are *actually* low (gain will be system specific, and an exposure no more than 1 second worked well for the lighting and camera system this work was developed for). If they are, then the camera may have dead pixels and/or the light source may also be failing, resulting in high exposures/gains needed to see cells/beads.

I don't have MATLAB 2017a or newer, will the code(s) still work?

They likely will not work because some built-in MATLAB functions may not have been introduced until at least version 2017a. Using older versions has not been tested and is not recommended for this analysis procedure.

Why do I have to download the 7 codes every time I want to run a data set?

Because these codes rely of saving all folders, images, data structures, and figures to the working directory, if there are multiple data sets in the same directory, this can lead to organization and saving problems while running the code. By putting each set of images in its own folder and having a new set of codes in that working directory, this keeps all the results (which can get long even if only processing a data set once) from that single data set in that folder, and overall helps keeps results organized over time. For more experienced MATLAB users, you can store all codes in a separate folder and add to the path to within the current working directory, however for those less experienced, adding the codes directly into the working directly will be easier.

Does having to keep adding these codes for each data set use up hard drive space?

Technically, yes. Practically speaking, hardly any. Because the code file sizes combined only makeup about 1/3 of a MB and because the images alone from MetaMorph® (as well as the images created from data processing) can take up space on the order of GB for some data sets, saving these codes only takes up a negligible amount of space.

What if I want to run the same set of images but using different analysis parameters (i.e. threshold, sensitivity, min and max diameter), and not delete my previous analysis with old parameters?

See footnote 5 on page 8. Click **Run** for the same `Code3_Identify_Track_Results.m` code you used for the original data set and change the experiment number. This will allow the user to fully run the same data set again without overwriting any previous processing. If the experiment number is the same as a previously processed

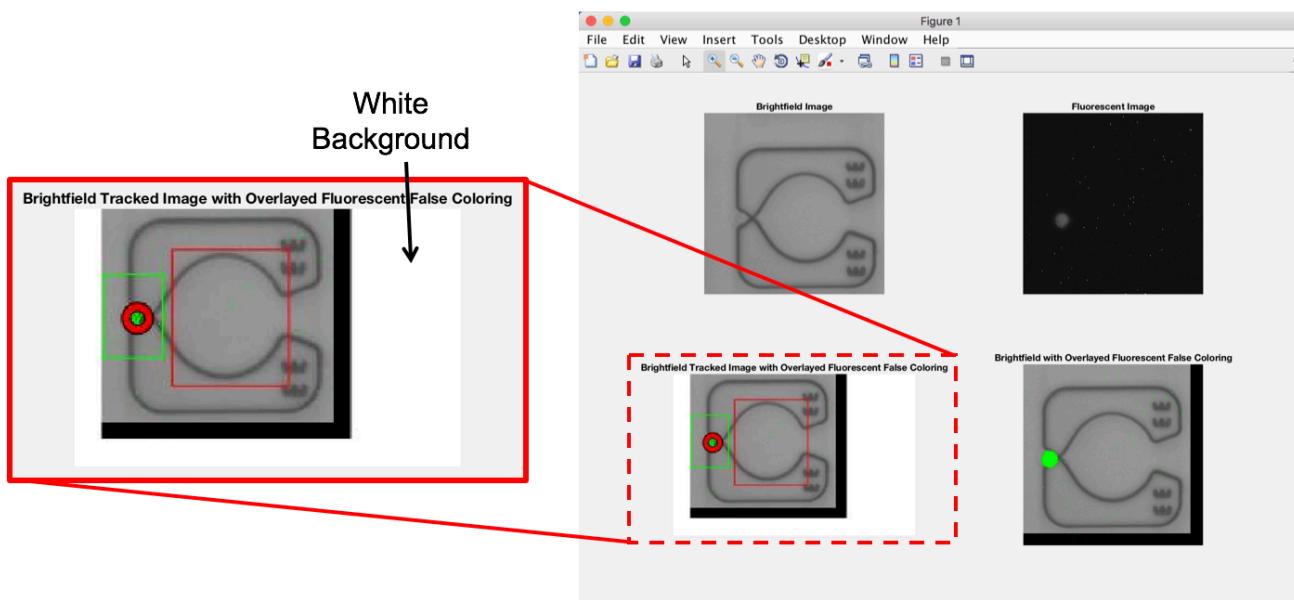
data set, the user will be prompted asking if they want to overwrite this data set. Once a data set is overwritten, it will be permanently overwritten – there is no backup storage.

I clicked Yes when prompted to create a Word document while on a Mac OS, and now the code had an error. What should I do? Can I still use the data analysis results?

If this is the case, the data and all figures have all already been saved, and nothing is lost. You can close all figure windows, and no report will/can be generated, since automatic report generation is not supported on a Mac OS. You can confirm the data has been saved by checking the working directory.

The tracked image (lower left) of the figure when I view a set of images for a single complex appears shifted in the figure window and/or has a white border. Why does this occur? Can I fix it?

An example of what this may look like is shown below:



This can occur because the tracked image size changes based on the figure window size during the main tracking process. If you make the figure window smaller during the main tracking, this tracked image is saved at this smaller size and with a white border (I'm not sure why MATLAB adds the white border, but it is default and not easily changed). This figure window attempts to crop this image to remove the white border, and the values it uses to crop aren't always accurate for removing the white border.

Unfortunately, outside of hard-coding cropping values each time based on the tracked figure window size, this problem isn't easily fixed now. However, the tracked image should still be visible and can be used effectively as needed.

Why do my tracked images have lower quality and/or have low resolution compared to my original brightfield images?

If this is the case, while running the data set you likely made the figure window very small to speed of the processing time. Because tracked images are saved directly from the figure window and the size of the figure window is preserved during saving, these images become lower quality and lose resolution due to binning. This binning creates the “fuzziness” that is often present in lower resolution images. This “fuzziness” is not seen in

the isolated brightfield images because those images are directly cropped from the original brightfield images, with no dramatic resizing taking place.

To avoid this problem, leave the figure window larger or at the original size during tracking. Tracked images will then be saved with the original resolution of the isolated brightfield image. However, the increased time to generate, display, and save these images may add multiple hours to the primary analysis, and may not be worth the added time to create clearer images.

*Why are some boxes on my heat maps black and not one of the legend colors?*

Boxes that are black are those which represent apartment complexes that physically do not exist on the chip, but are included in the heat map because it both makes the plots easier to generate, and allows the rows to be plotted as horizontal row without the offset physically present on the chip. These boxes are not included when calculating efficiency results.

*Why can't I open .jpg image when I double-click on them in MATLAB?*

MATLAB thinks you're trying to read in the image as a matrix, and not actually show the image. To open .jpg file types, right click on the image in the working directory, and select **Open Outside MATLAB**. Simply double-clicking will not open a .jpg image file, but will open a .fig file type.

*I'm running on a Windows OS and initially chose to not create the automatically generated Word document. How can I create it for a data set that has already been processed?*

See **Create MATLAB Generated Word Document if Not Initially Selected to be Created** subsection under in the post-processing instructions.

*The image analysis parameters don't appear to be accurate based upon the tracked images that are being shown. How can I change them without having to wait hours for the data to finish?*

Be careful with wanting to change the analysis parameters right away, since often times the first few streets will have the most build-up of beads/cells, which are also the most difficult to track accurately. Wait until the code has processed at least four or five streets and if then the code appears to be missing many beads/cells it should be able to pick up, then the image analysis parameters need to be changed.

If stopping the code is desired, then to force-stop the MATLAB code, then:

- if on a Mac, press **control+C**
- if using Windows, press **Ctrl+C**

Click **Run** again for `Code3_Identify_Track_Results` and you will be start the entire analysis over again, and will need to follow all prompts. Change the image analysis parameters as desired and repeat this process as necessary.

*How can I use viewing individual apartment at the end of the primary analysis for troubleshooting?*

See **Interpretation and Troubleshooting** in step 18 of the primary analysis instructions.

*How do I perform an alignment step using ImageJ?*

See **Alignment in ImageJ** in step 6 of the primary analysis instructions.

## AFFILIATIONS AND CONTRIBUTIONS

Image Analysis Process Developed by: Sean T. Kelly<sup>a, b</sup> and Jeffrey D. Motschmann<sup>a</sup>

MATLAB Code Developed and Written by: Sean T. Kelly

ImageJ Protocol Developed by: Sean T. Kelly and Jeffrey D. Motschmann

<sup>a</sup> Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA

<sup>b</sup> Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 49109, USA

## ACKNOWLEDGMENTS

This authors are thankful to support by NIH awards 1R01GM123542 and 1R21GM131279. This work was performed in part at the Duke University Shared Materials Instrumentation Facility (SMIF), a member of the North Carolina Research Triangle Nanotechnology Network (RTNN), which is supported by the National Science Foundation (Grant ECCS-1542015) as part of the National Nanotechnology Coordinated Infrastructure (NNCI).