# 4-dimensional cell tracking, analysis, and visualization using LongTracker, ImageJ, and FluoRender

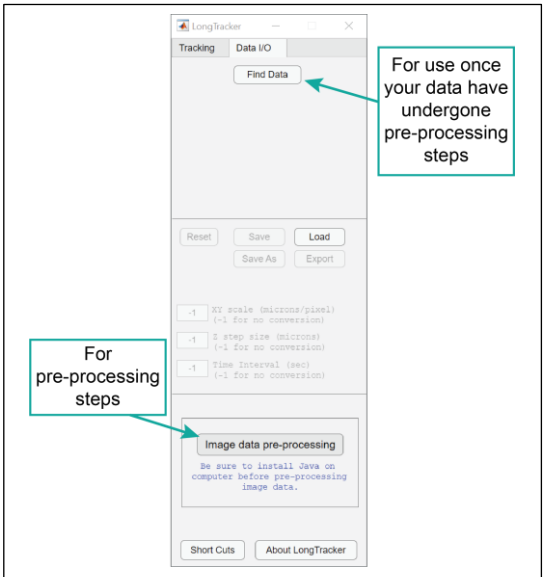
***Before you start:***

**• Install LongTracker:**

1. **Make sure Java is installed** on the computer: *www.java.com*
   a. To verify if Java is installed go to: *www.java.com/en/download/help/version_manual.html*
2. **Copy the LongTracker files into a folder** on the computer (23 files total). *Sample folder name: C:\Matlab Files\Longtracker*
3. **Install MATLAB version 2022b or higher, and the MATLAB toolboxes:**
   a. **Image Processing Toolbox**
   b. **Signal Processing Toolbox**
   c. **Statistics and Machine Learning Toolbox**
4. After installation is finished, **open MATLAB**, click on the **HOME** tab, and press the **Set Path** button.
   a. Press the **Add Folder** button.
   b. Find the folder containing the LongTracker files then press **Select Folder**.
   c. Press the **Save** button, then the **Close** button.

## 1. Open up LongTracker

1. **Open MATLAB**
2. **Type longtracker** at the MATLAB Command Window prompt and press Enter. The LongTracker interface should open (screenshot).
3. **Click on Image data pre-processing button**

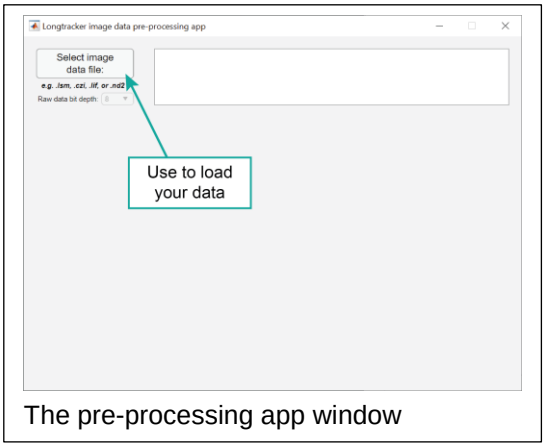


The LongTracker interface window

## 2. Pre-processing in LongTracker

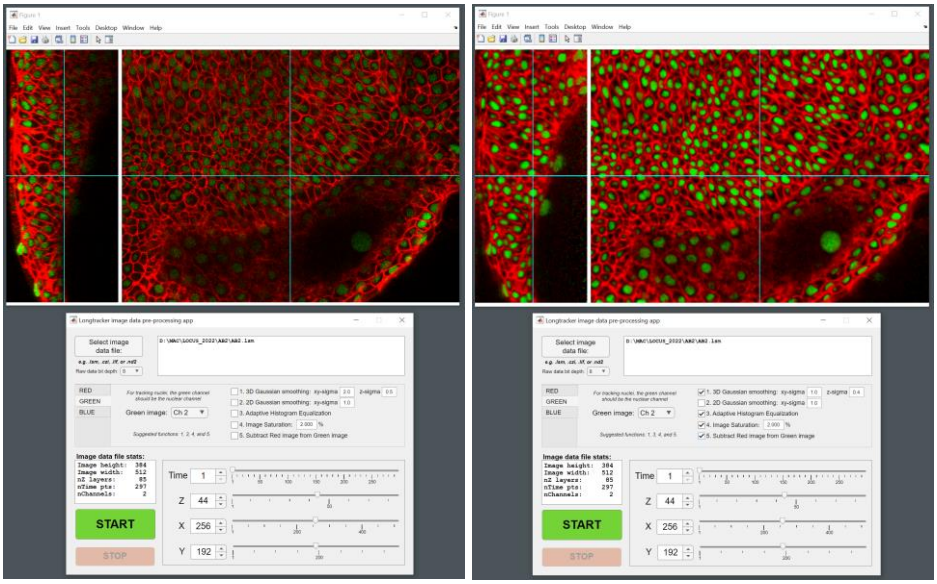
The goal of this is to prepare datasets for cell tracking. Raw data are often noisy, and signal varies over z and time. The pre-processing steps generate a set of images used only for tracking and help to enhance signal to make it easier to identify nuclei for tracking.

1. **Click Select Image Data File** and verify the raw image data bit depth
   a. If raw image data bit depth is not correct, the processing will not work correctly
   b. This will work with .lsm, .czi, .lif, and .nd2 files



The pre-processing app window

2. **Assign the channels**: for tracking, we will want the membranes as the red channel and the nuclei in green.
3. **Process the membrane (red) channel first**, using the (1) 3D Gaussian smoothing, (3) Histogram equalization, and (4) Image saturation.
4. **Process the nuclear (green) channel second**, using the (1) 3D Gaussian smoothing, (3) Histogram equalization, (4) Image saturation, and (5) Subtracting membrane (red) channel from nuclear (green) channel.
5. **Click on the big green Start button**.



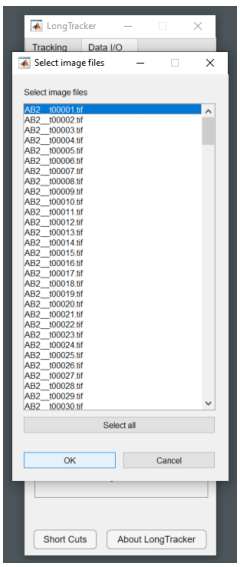
Data before (left) and after (right) pre-processing steps

6. **Make a New Folder to save the processed data**.
    a. The resulting New Folder will also hold the raw data in 8-bit tif format, with files separated by timepoint and channel. This is useful since many confocal file formats have everything saved in one large file, and one often needs to work with the membrane or nuclear channels separately for data visualization and presentation.
    b. The file name will already include an underscore following the prefix listed.

## 3. Cell tracking using LongTracker

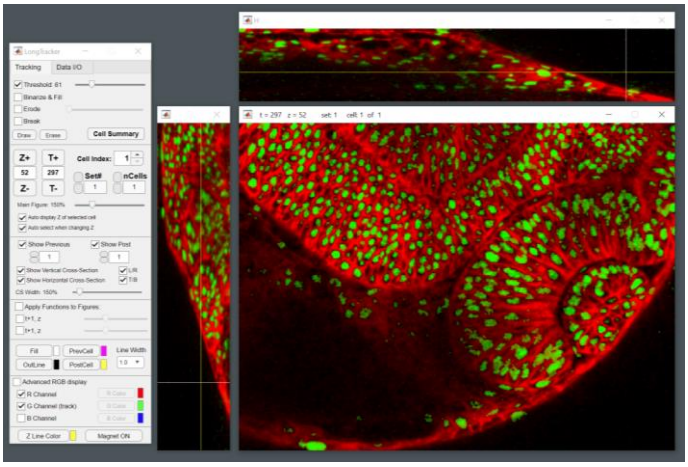
We can now open the pre-processed datasets in LongTracker and begin tracking. Tracking can be carried out either forward or backward in time.

1. **Load the pre-processed dataset.**
    a. Go back to the **LongTracker interface** and **click on Find Data**.
    b. **Select the Folder** containing the pre-processed dataset
    c. A window will open with a list of the image files contained within; **click on Select all** and **click OK**.



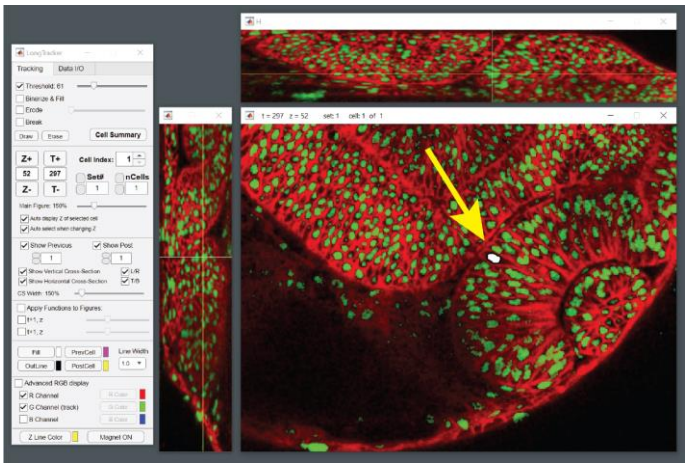
Finding data and selecting images

2. **Switch to the Tracking tab** on the LongTracker interface (instead of Data I/O).
3. **Set up the Tracking windows**
    a. **Main Figure can be zoomed** to 150% or whatever is comfortable on the monitor
    b. **Turn on Vertical and Horizontal cross sections**; these will line up with the Main Figure. If "Magnet ON" is set at the bottom of the LongTracker interface window, the cross sections will move with the main figure window if that window is moved.



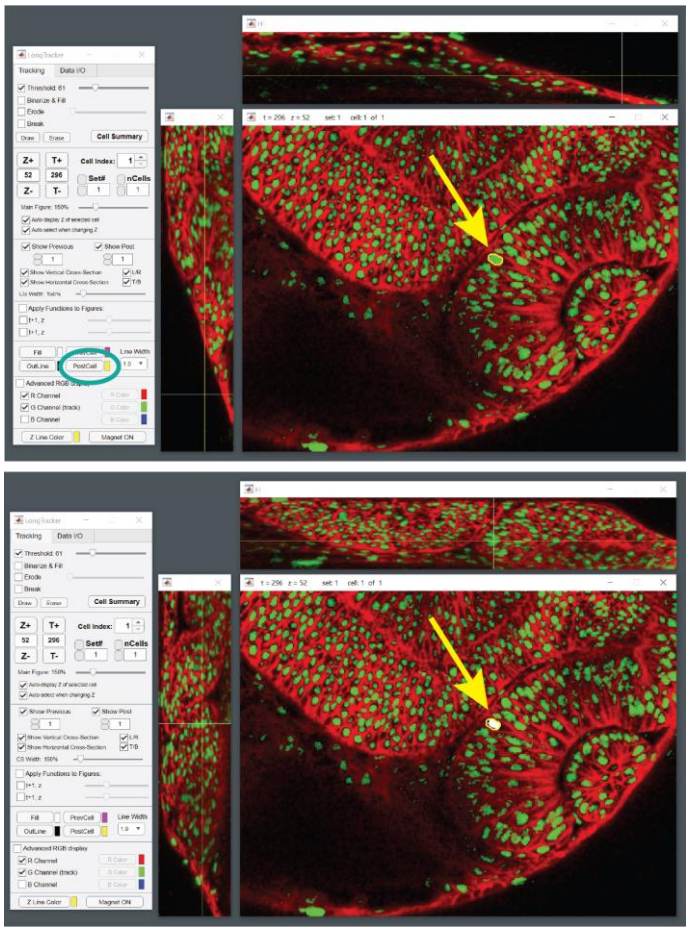
Tracking set-up with LongTracker interface, Main Figure, and Vertical and Horizontal cross-sections.

4. **Sets and Cells:** we use these to keep cells together that are related by mitosis.
    a. **Sets** = each independent cell we track is its own set; the first cell being tracked is assigned Set #1, Cell Index #1.
    b. **Cell Index** = if we encounter our tracked cell undergoing mitosis, the other daughter cell can be tracked as the second cell in the set (a partial track that arises at the mitosis timepoint), as Set #1, Cell Index #2.
    c. **nCells** = number of cells allowed per set; this is flexible and can be changed as necessary
5. **Cell Tracking**
    a. Start at the desired timepoint (T), either the first timepoint if tracking forward or the last timepoint if tracking backward.
    b. **Threshold** to improve visualization of individual nuclei. Other binary functions (**Binarize and Fill**, **Erode**, and **Break**) are provided to help with identification of single nuclei.
    c. **Place cursor over nucleus of interest.** The crosshairs will provide a location in the XZ and YZ cutaways to ensure that you are in the optimal z-slice for your particular nucleus (in the middle of the nucleus).
    d. **Click on the nucleus of interest.** It will be masked in white with a black outline (depending on the color selections in the LongTracker interface).



Cell selected at the final timepoint (t297) to track a cell backwards through the dataset.

e. **Use the arrow shortcut to move forward or backward in time by one timepoint.** In the example, a cell is being tracked backward in time. The mask from the final timepoint (t297) is now a yellow outline (PostCell color), so one can identify the same nucleus at t296. If one is tracking forward in time, the mask will become a magenta outline (PreCell color). These colors are customizable. The Pre/PostCell masks ensure that the same cell is being tracked between timepoints; this is how we determine whether the temporal resolution of the dataset was appropriate. If the nuclei do not overlap between timepoints, higher temporal resolution (shorter time between z-stacks) is required.

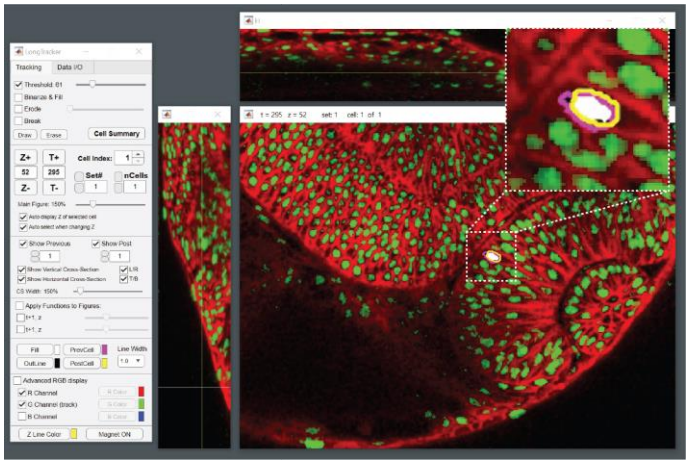


Moving back one timepoint (t296): the mask from the last timepoint (t297) becomes yellow (PostCell). The same nucleus at t296 is chosen with confidence.

f. **Click on the same nucleus at the adjacent timepoint.** The nucleus from the current timepoint will now be masked in white with a black outline and the yellow outline mask is still visible.

g. **Use the arrow shortcut to move another timepoint forward or backward.** The mask from the adjacent timepoint is now a yellow outline again. Use the Vertical and Horizontal cross sections to ensure that you are still in the largest part of the nucleus in z, and if necessary, move up or down in z before selecting the nucleus at the next timepoint.

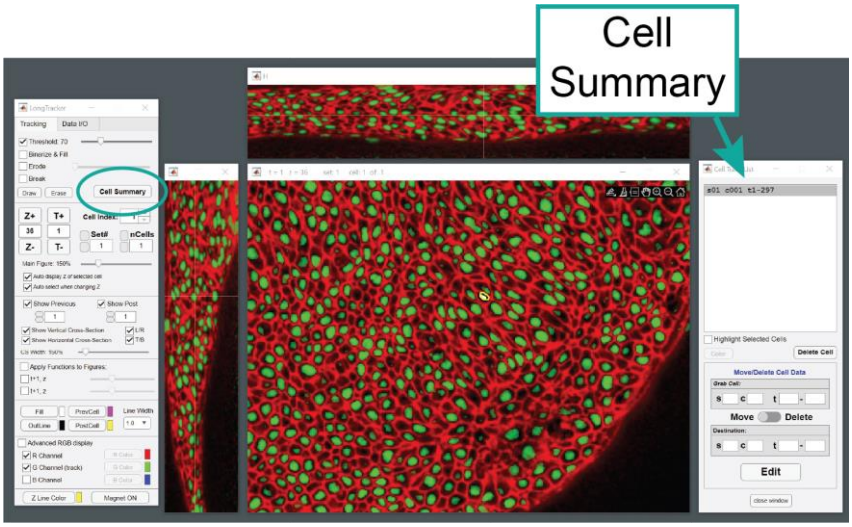
h. **Do this for a few timepoints.** One can check the tracking by moving back and forth in time using the arrow shortcuts. If one goes to a timepoint where a nucleus has been selected and there is a nucleus selected in the timepoints immediately before and after, one can see all three masks (magenta outline, yellow outline, and white with black outline) simultaneously.



One can check the tracking by moving through time. Here, the nuclear masks are visible at the timepoint, and the timepoints immediately before and after (Pre/Post Cell).

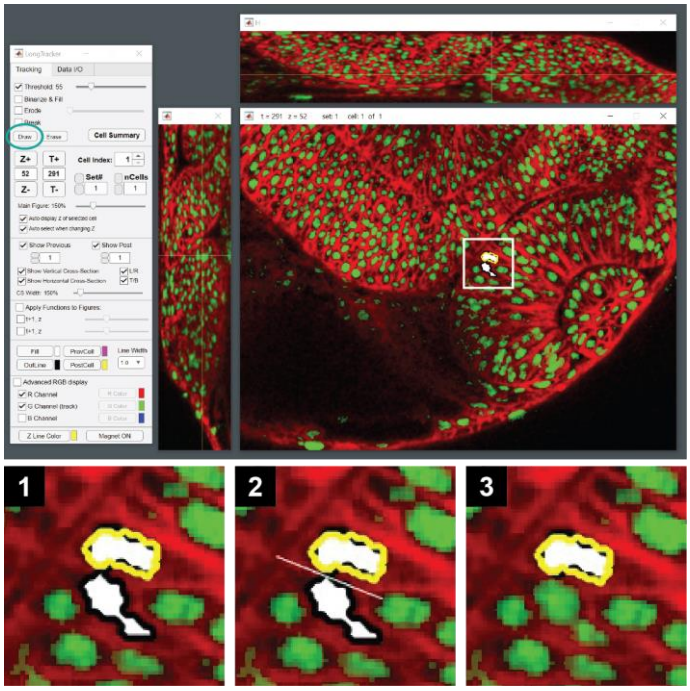
i. **Keep going to completely track a cell through the dataset.** If one encounters a mitosis, one can increase the "nCells" and then change Cell Index to track the other daughter cell.

j. **Cell Summary.** The Cell Summary window can be used to keep track of how many cells you have tracked, and for each cell, what timepoints have tracking information. For example, if you accidentally skipped one timepoint, that will be apparent in the Cell Summary.




The Cell Summary window is helpful to monitor tracked cells and identify inadvertent gaps in data.

6. **Cell tracking issues**
    a. **Multiple nuclei appear "attached":** Pre-processing does not solve all issues with the signal, and it is not uncommon to click on a nucleus and see another "attached" nucleus (screenshot). In this case, the "Draw" function can be used to break the nuclei. Click on "Draw" and move over to the window. Click to start a straight line that will cut the two nuclei apart. Click again to end the straight line. The nuclei will be broken apart. Proceed with tracking.



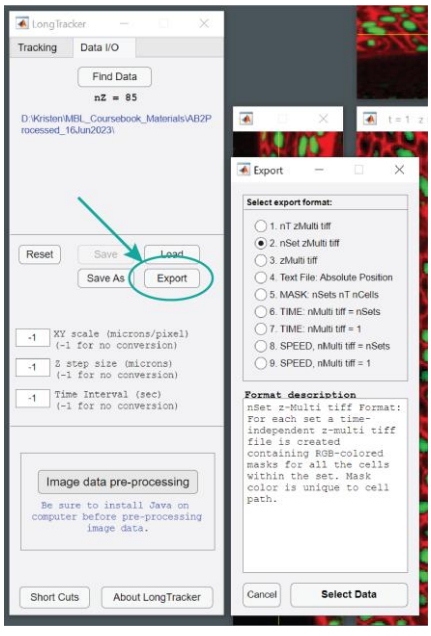
When selecting a nucleus, two nuclei may become marked (1). Use the "Draw" function to draw a line between the two nuclei to separate them (2), and the nucleus originally selected will remain selected.

    b. **Nucleus seems to disappear from the dataset:** Depending on dataset quality, this may also happen. This could be biological: depending on your process of interest, cells may undergo apoptosis. However, this may also be due to the quality of the dataset or apoptosis due to imaging. If a cell cannot be tracked completely through the timecourse, it is listed as a "partial track" and can only be partially evaluated.

## 4. Exporting data from LongTracker

Once cells are tracked, LongTracker provides several options for data export. Here we cover some of our most used export formats.

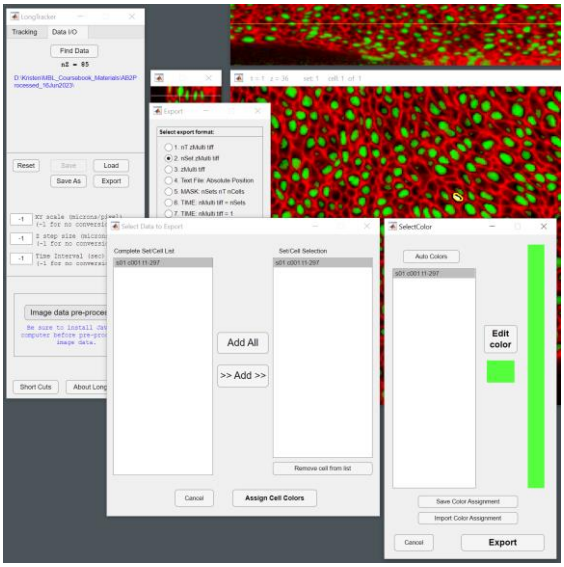
1. Once you are ready to export some or all of your data, go back to the LongTracker interface and **switch from the Tracking tab to the Data I/O tab**.
2. **Click on the Export button**. A new window will appear with the Export format options. When each one is selected, a description of the format appears in the Export window.
3. **Commonly used export formats**
    a. **nT zMulti tiff (1).** This will generate a set of 3D tif stacks, one for each timepoint, with the nuclear mask on the correct z-slice.
    b. **nSet zMulti tiff (2).** This will generate a single tif stack of all of the nuclear masks over time (the trajectory); this is ready for 3D visualization in FluoRender.



LongTracker Export Window

c. **Text file (4).** This will generate a text file suitable for import into Excel containing the 3D centroid positions (x, y, z) of the tracked nuclei at each timepoint. This is useful for quantification of trajectory length and cell speed.

4. **Data export and customization**
   a. Once you select the format, **click on Select Data**. When exporting data, there are many options for selecting the data and colors.
   b. A new window will appear allowing you to select the data (specific cells and sets) you would like to export. Add the Sets and Cells you would like exported and **click on Assign Cell Colors**.
   c. A new window will appear which allows you to assign specific colors to specific cells. **Click on Edit Color** and a new color edit window will appear. Once you are finished selecting colors, you can save the color assignments, if desired.
   d. Once you are finished selecting the data and colors for export, **click on Export**.



Export interface allows customization of which cell tracks to export and what color(s) to assign.

**5. Qualitative visualization and analysis of cell tracking data**

Once data are exported, they can be processed in many ways for visualization using ImageJ. For visualization, we use both ImageJ (2D over time) and FluoRender (3D). ImageJ macros are very useful for this part, as there are many z-stacks and timepoints to process, and it is helpful to semi-automate the process.

1. **Visualizing nuclear movements in 2D over time**
   a. You will start with the data exported as **nT zMulti tiff (1).** There will be z-stacks containing the nuclear mask(s) for each timepoint.
   b. **Generate maximum intensity projections for each timepoint.** Because the visualization will be in 2D, first project the z-stacks to generate a single 2D image for each timepoint. This can be done manually, or with a macro (macro provided is "Z-projection.txt"). Before use each time, edit the macro to reflect the type of projection (maximum or average) and the z-slices you want projected (in this case, it will be the entire z-stack). Then save and install the macro in ImageJ (Plugins > Macros > Install). Then run the macro, and when

Z-projection.txt

```
dir = getDirectory("Choose a Directory ");
      list = getFileList(dir);
      setBatchMode(true);
      for (i=0000; i<list.length; i++) {
          path = dir+list[i];
 if (endsWith(list[i], "/")){
        //if "path" ends with "/", do nothing and skip to the next file
        }else{

open(path);


run("Z Project...", "start=1 stop=85 projection=[Max Intensity]");


dotIndex = lastIndexOf(path, ".");
          if (dotIndex!=-1)
              path = substring(path, 0, dotIndex); // remove extension
          myDir = dir+"Projected_TIFF"+File.separator;
          File.makeDirectory(myDir);

          sndir = dir+"Projected_TIFF"+"/";
          save(sndir+list[i]+"_proj.tif");
close();

}}

"Done"
```

Z-projection.txt macro: before installing in FIJI, edit the macro to reflect which z-slices to project and the type of projection (circled), and save.

prompted, select the folder with the data exported as nT zMulti tiff. This macro will generate a new folder ("Projected_TIFF") of individual projected images (one for each timepoint).

c. **Open up the folder of projected images as a tif stack in ImageJ** and save as the nuclear movement over time.

2. **Visualizing nuclear trajectories in 2D over time.** This is similar to the prior nuclear movement visualization, but now we sum the nuclear masks over time, which generates a trajectory (as opposed to a spot that moves around).
   a. You will start with the folder of projected images above ("**Projected_TIFF**"). These are maximum intensity projections of the nuclear masks over time.
   b. **Sum the nuclear masks using a macro** ("Z-proj_accumulate.txt"). No editing needs to be done manually with this macro. Just install it and run it; when it asks for the folder to use, choose the "Projected_TIFF" folder from the step above. This will generate a new folder ("zTIFF") which contains images of the nuclear masks accumulating. For example, image #1 will be the same, but image #2 will be the sum of image #1 and 2; image #3 will be the sum of images 1-3; etc.
   c. **Open up the folder of projected images as a tif stack in ImageJ** and save as the nuclear trajectory over time.

Z-proj_accumulate.txt

```
dir = getDirectory("Choose a Directory ");
      list = getFileList(dir);
      setBatchMode(true);
      for (i=0000; i<list.length; i++) {
           path = dir+list[i];
 if (endsWith(list[i], "/")){
       //if "path" ends with "/", do nothing and skip to the next file
       }else{

k=getNumber("How many time points?",1);;

//open(path);
run("Image Sequence...", "open="+path+" number="+k+" starting=1 increment=1 scale=100 file=[] or=[] sort");


for(y=1; y<=k; y++) {

 if (y<10){

run("Z Project...", "start=1 stop="+y+" projection=[Max Intensity]");

        dotIndex = lastIndexOf(list[i], ".");
          if (dotIndex!=-1)
               list[i] = substring(list[i], 0, dotIndex); // remove extension

          myDir = dir+"zTIFF"+File.separator;
          File.makeDirectory(myDir);

          sndir = dir+"zTIFF"+"/";
          save(sndir+list[i]+"_T00"+y+".tif");
        close();
        }

 if (y>9){
 if (y<100){

run("Z Project...", "start=1 stop="+y+" projection=[Max Intensity]");

        dotIndex = lastIndexOf(list[i], ".");
          if (dotIndex!=-1)
               list[i] = substring(list[i], 0, dotIndex); // remove extension

          myDir = dir+"zTIFF"+File.separator;
          File.makeDirectory(myDir);

          sndir = dir+"zTIFF"+"/";
          save(sndir+list[i]+"_T0"+y+".tif");
        close();
        }}

 if (99<y){
run("Z Project...", "start=1 stop="+y+" projection=[Max Intensity]");

        dotIndex = lastIndexOf(list[i], ".");
          if (dotIndex!=-1)
               list[i] = substring(list[i], 0, dotIndex); // remove extension

          myDir = dir+"zTIFF"+File.separator;
          File.makeDirectory(myDir);

          sndir = dir+"zTIFF"+"/";
          save(sndir+list[i]+"_T"+y+".tif");
        close();
        }

 if (y==k){
close();

} // y<10, y<100


}}

"Done"
```

Z-proj_accumulate.txt macro used to generate nuclear trajectories from nuclear masks.

3. **Generating a membrane projection as a background for the nuclei and trajectories.** Visualizing nuclear movements and trajectories is not very helpful without the spatial and temporal context of the dataset. Here we will generate a grayscale average projection of the membrane on which to overlay the nuclear movements or nuclear trajectories.
   a. **You will start with the 8-bit raw image data** generated from the LongTracker pre-processing steps.
   b. **Generate average projections for each timepoint.** You may need to test a few average projections to see what may look reasonable for your dataset. You can test a few different options by opening up the first and last z-stacks and manually doing some average projections. Vary the number of z-slices and position in z-depth to see what is optimal for your data. You may use the macro listed above ("Z-projection.txt") and be sure to edit the macro to use an Average Projection, and what z-slices you want projected. For the membranes, it is unlikely to be the entire z-stack. Then install the macro in ImageJ and run it on the membrane data. As before, the macro will generate a new folder ("Projected_TIFF") of individual projected images (one for each timepoint).

c. **Open up the folder of projected images as a tif stack in ImageJ** and save as the average membrane projection over time.
d. **Overlay nuclear movements and trajectories on the membrane projections.** Open the average membrane projection tif stack and be sure to convert to RGB color to ensure that it is in the same format as the nuclear masks and trajectories. Open the 2D nuclear mask or 2D nuclear trajectory tif stack. Use the Image Calculator (Process > Image Calculator) and add the nuclei (first entry) to the membrane signal (second entry). This must be done in this order to maintain exported color assignments from LongTracker. If you want to overlay the trajectories on the membrane, add the nuclear trajectories (first entry) to the membrane signal (second entry).
e. **Save the new tif stacks as your nuclear movement and trajectories plus membrane files.** These can be converted to movie files in ImageJ.

4. **Visualizing nuclear trajectories in 3D.** It is often useful to visualize the entire trajectory at once in 3D, in order to qualitatively evaluate the movement in all 3 dimensions. For visualization, we use the free software FluoRender (*https://www.sci.utah.edu/software/fluorender.html*).
    a. **Start with data exported in the format nSet zMulti tiff (2).** This is a single z-stack with the entire 3D trajectory.
    b. **Import the trajectory data into FluoRender.** This will provide 3D visualization of the trajectory.
    c. **Import the final membrane timepoint into FluoRender.** This will provide spatial context for the trajectory, even if this only represents a single timepoint.
    d. **Customize the visualization.** Colors and orientation can be easily customized in FluoRender, as well as cutaways in any of the 3 axes.
    e. **Export the visualization.** Visualizations, including 3D rotations of the data, can be saved as tif stacks and movies.

**6. Quantitative analysis of cell tracking data**

It is often important to quantify cell movement. Starting with the export format **Text file (4)**, one can use the 3D (x, y, z) position of the nuclei at each timepoint to calculate total distance traveled (trajectory length) and speed. All of this can be accomplished using equations in Excel.

1. **Trajectory length.** To calculate trajectory length, use the 3D distance equation between each timepoint and sum distances. One can do this for part or the entire dataset. The 3D distance equation is:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

2. **Cell speed.** To calculate cell speed, simply divide trajectory length by time for part or the entire dataset.