CellTracker Demo

Michael Deng

April 22, 2016

1 Running the Analysis

1.1 Background

This demo outlines the process of running a CellTracker analysis on a set of 20 sample microscopy images. These images consist of three channels: a nucleic marker in the red channel, YFP-E2F1 expression in the green channel, and a phase image in the blue channel. A sample composite RGB image is shown in Figure 1.

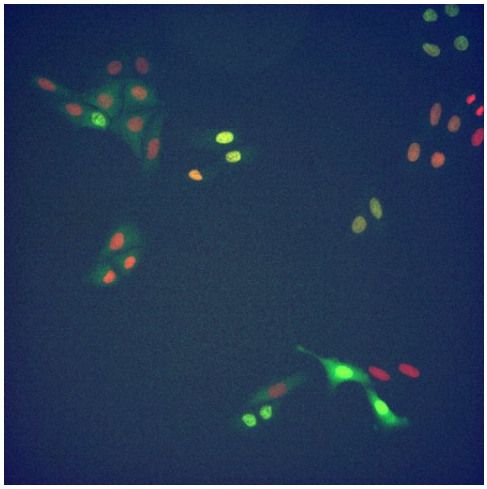

Figure 1: A sample composite microscopy image.

This demo will describe the process of detecting the cell nuclei in these images and measure the nuclear YFP expression throughout the duration of the video. To begin the analysis, type the following in the MATLAB command line:

```
>> runTracking;
```

The GUI shown in Figure 2 will appear.

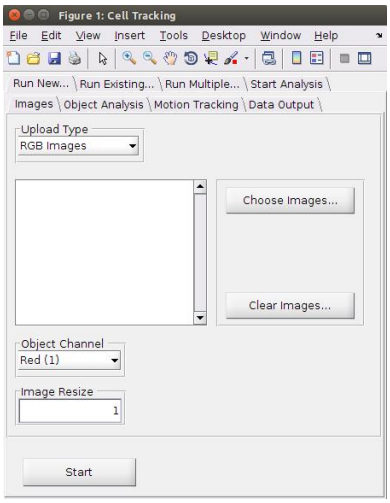

Figure 2: The input GUI.

1.2 Inputting Images

The microscope images are in the form *im1ch#t##.tif*. Each frame is split into a three channels—one for each color. Thus, in order to upload them, we should choose **Split Channel Images** as our **Upload Type**. Selecting the **Choose Ch1 Images...** button will bring up a dialog box that will allow you to select the images for the analysis. Select multiple images using Shift or Ctrl-Click.

The cell nuclei are in the red, or first channel, and thus we can leave the **Object Channel** selection as default.

The microscope images are 512×512 pixels, which is a sufficient size for the analysis. Thus, we do not need to downsize the image and leave the **Image Resize** setting at 1.

After entering these settings, the GUI should look like Figure 3.

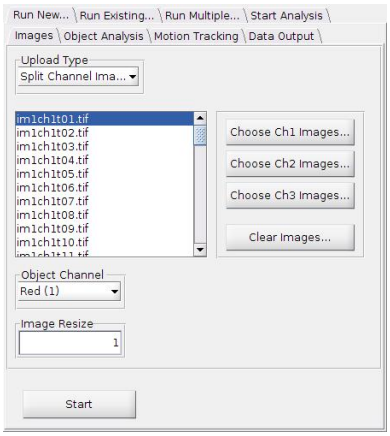

Figure 3: The input GUI after inputting images.

1.3 Object Analysis

For our segmentation, we will choose to use the **Edge-Watershed** algorithm. In addition, we will choose **Gauss** as a **Background Correction** algorithm.

By quickly glancing at our images in any image viewer, it appears that the smallest cell nuclei have an area of 125 and the largest nuclei have an area of 300. Thus, we choose 75 and 400 as the minimum and maximum sizes for our segmentation algorithm. Because are cell nuclei are very clearly separable from the background, we can choose a higher edge threshold than 1. Although this makes it more difficult to detect edges, it also decreases the likelihood of detecting false edges. We will choose 2 as an edge threshold.

We do not need to filter out detected objects by any different metrics, so we will leave the **Metric Selection** section untouched.

After inputting these settings, our **Object Analysis** tab will look as in Figure 4.

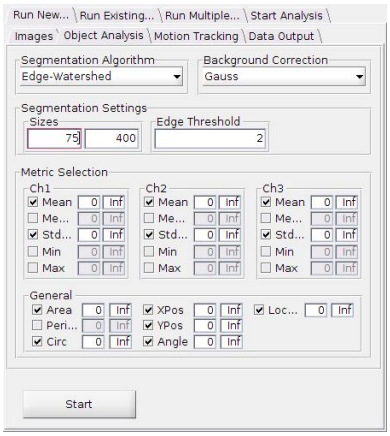

Figure 4: The input GUI after object analysis.

1.4 Motion Tracking

Next, we will enter settings for our **Motion Tracking** algorithm. The **Metric Cost** is the maximum cost that two cell objects have and still be linked in adjacent frames. We will leave it at its default value of 25. The **Gap Close** parameter defines the maximum number of frames apart two objects can be and still be linked. We will also leave this parameter at its default value of 7. We will also use the greedy algorithm instead of the optimal algorithm.

The metric selection defines which parameters we will use to track the cell nuclei over time. Thus, we expect these measurements to stay (relatively) constant from frame to frame. We have chosen the nuclear channel mean intensity, the area, circularity (measure of shape), x and y position, and angle. We will also use the default variances for these metrics. These metrics define how much we believe each metric will change on average from frame to frame.

It is often best to generate these average variances by first running an initial rough analysis on a set of microscopy videos. Then, the average variance can be calculated from the rough analysis and can be used to inform all subsequent analyses.

After inputting these settings, our **Motion Tracking** tab will look as in Figure 5.

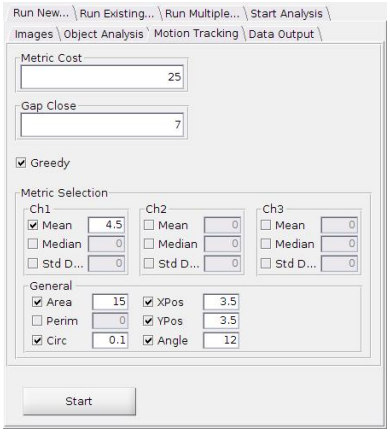

Figure 5: The input GUI after motion detection.

1.5 Outputting Data

We finally move to the **Data Output** tab. Here we can choose a file name for our data output. The results of the analysis will be output as a MATLAB .mat file. This file can be quickly reloaded so that the results can be viewed without having to run the analysis over again.

Once the output file is chosen, select the **Start** button at the bottom of the GUI.

1.6 Analysis in Progress

Once the analysis is started, you will see the progress of the analysis in the subsequent GUI (Figure 6).

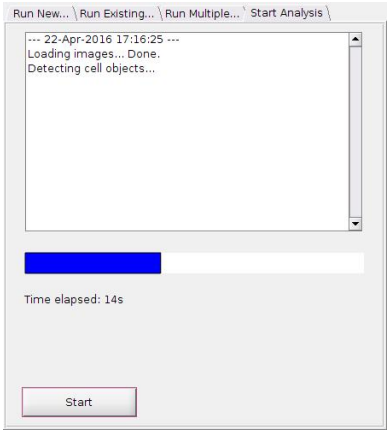

Figure 6: An analysis in progress.

2 The CellTracker Output

After the analysis is finished, you will see the output in the GUI in Figure 7.

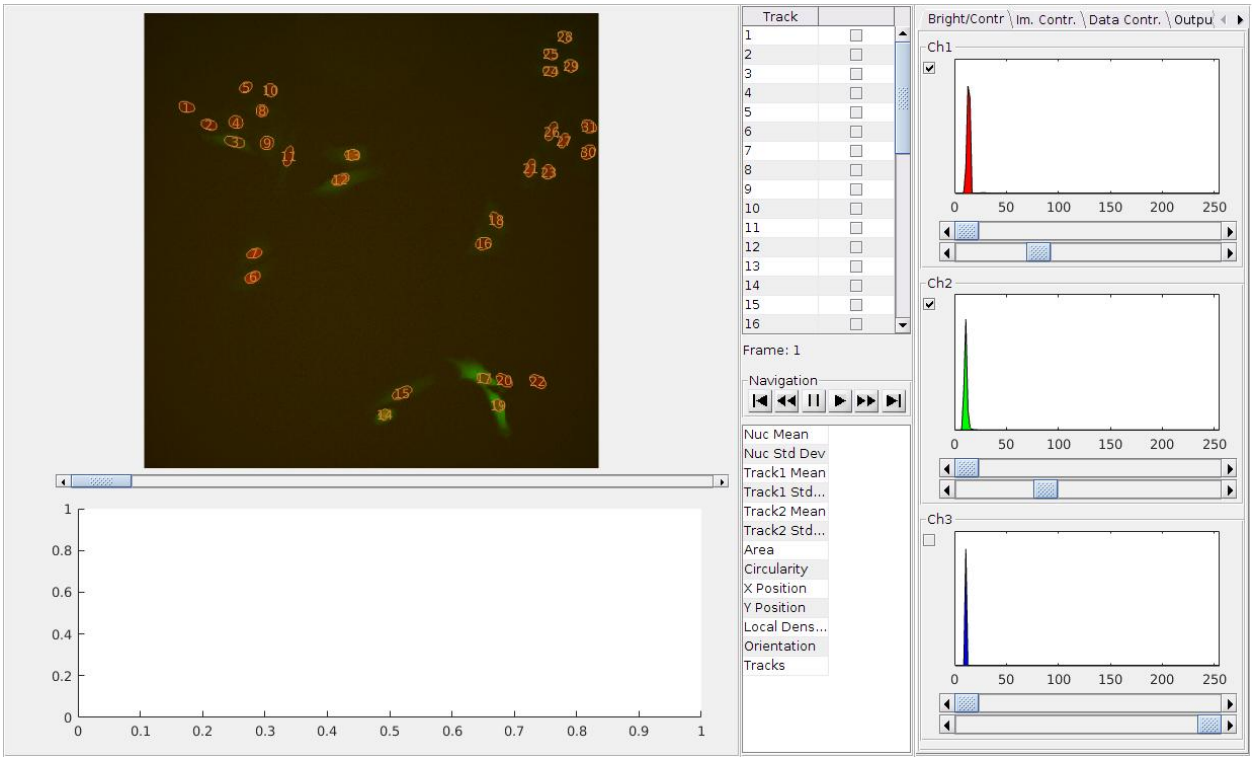

Figure 7: CellTracker output.

Initially, it may be hard to see the cell nuclei (red) and the YFP-E2F1 (green) in the image. We can adjust the brightness/contrast in the image by playing with the sliders under the histograms on the right. In addition, we can shrink or enlarge the microscopy image by dragging the bottom of the image.

2.1 Tracking Survival Events

We will begin by taking a look at cell 6. We can visualize the cell and its measurements by selecting Track 6 in the top list and Track1 Mean in the bottom list. By scrolling along from frame to frame, we can see that this cell undergoes mitosis at frame 14. In addition, by scrolling further, we can see that the mitotic event results in daughter cells 38 and 39.

2.2 Adding New Objects

We also notice that cell 3 has a gap at frame 14. This is because the segmentation algorithm was unable to correctly segment the object at that frame. In order to correct this mistake, we will attempt to create a new object to add to the analysis.

To do this, we select the **Create Object** button under the **Data Contr.** tab. This will open a second window that will allow you to manually draw a boundary to define a new object. Draw a new object where cell 3 should have been. It may be helpful to zoom in on the image in order to draw the new cell nuclei.

When you return to the CellTracker Viewer, you will see this new object has been added to the output (Figure 8).

Figure 8: Manually added object.

Notice that the newly created object is indexed 29, but in red instead of brown. The red represents the object index. The newly created object is the 29th object in this frame. The red object index is showing because the object has no associated trajectory index, or brown index. Thus, we would like to add object 29 in frame 14 to cell 3, or the 3rd trajectory.

There are two ways to accomplish this. The first can be accomplished using the **Refactor Motion** button under the **Data Contr.** tab. This will go back and rerun the motion tracking algorithm on all unassigned cell objects. Ideally, the object we just created will be reincorporated into the correct trajectory in this way.

The second way is to manually add the newly created object ourselves. To do this, we switch to the **Table View** under the **Data Contr.** tab. Then, select Tracks as the metric to view. The table is organized into trajectories as rows and frames as column. Thus, we simply change the value in row 3 and column 13 to 29. Now we can see that our updated change has been incorporated into the microscopy image and the graph (Figure 9).

Now that the output data has been updated, the output file must be updated so that our changes can be saved. To do this, navigate to the **Output Contr.** and select the **Save Tracking** button. Overwrite the previous program output with this new updated version.

2.3 Saving Trajectory Data

Now that we've looked at and corrected some of our trajectories, we would like to save those trajectories for further data analysis. To do this, simply go into the Track list and check off the trajectories to be saved. Then, navigate to the **Output Contr.** tab and export the data. This can be done as both a .csv file or as a .xls file.

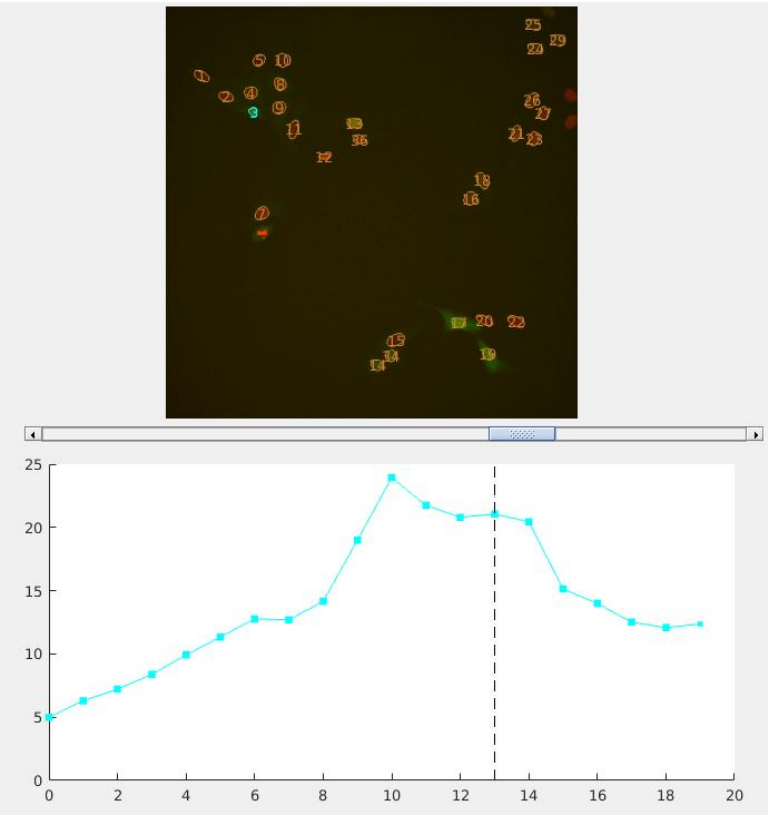

Figure 9: CellTracker viewer after new object incorporation.