CellTracker Documentation

Michael Deng

April 22, 2016

Abstract

CellTracker is software designed to aid in the analysis of single cell timelapse microscopy videos. The software enables users to track single cell gene expression over time using custom segmentation and motion tracking algorithms.

1 Background

The CellTracker algorithm is designed to operate on fluorescence microscopy images and may not perform optimally for phase-contrast or other imaging modalities.

One of the key features of CellTracker is that it allows for long-term single cell gene expression measurement across multiple channels. It allows the user to measure gene expression in one color channel using object regions defined by another object channel.

The general workflow works as follows: the user inputs the microscopy images and settings for the analysis. The user then saves the program output and views it in the Viewer. From the viewer, the user makes additions or corrections the program output, and saves the data for the relevant trajectories.

2 Algorithm

Segmentation

There are two segmentation algorithms that can be used in CellTracker, EdgeWater and ThreshWater. Both algorithms begin by smoothing input images using a Gaussian filter in order to remove noise.

The EdgeWater algorithm then detects edges in the image using the Canny edge detector. From these edges, the algorithm will then find cells from the connected edges.

In contrast, the ThreshWater algorithm detects cells from an adaptive threshold. The resulting foreground pixels (cells) are further separated using watershed segmentation.

The algorithm also allows for filtering of cell objects by a variety of parameters, including size, fluorescent intensity, position, and shape.

Background Correction

The ability to correctly account for background fluorescence is vital to the analysis of single-cell gene expression. Accordingly, CellTracker provides multiple algorithms in order to subtract the background from the cell objects. Both of these objects are adaptive in order to account for nonuniform background in the microscopy images.

The first algorithm, Gauss, corrects for background illumination using Gaussian blurring. Each image is blurred using a Gaussian filter with a large kernel size in order to blur away all the cell objects. The resulting blurred image is used as the background for the image.

The second algorithm, SKIZ, corrects for background illumination using a method called skeleton-by-influence-zones. This algorithm calculates watershed transform of the distance transform of the image, resulting in watershed ridge lines that denote minima in the image. The background for a given cell object can then be calculated from the intensity of its surrounding SKIZ region.

Motion Tracking

The motion tracking algorithm tracks cells from frame to frame using a sum of finite differences minimization algorithm. The underlying assumption is that the frame rate of microscopy is fast enough that the cell objects will remain relatively the same from frame to frame.

We can further develop this notion mathematically by defining a cell object in terms of k metrics. These metrics can include object size, intensity, position or shape. We then define the difference vector D as:

$$D = (\Delta m_1, \Delta m_2, ..., \Delta m_k)$$

where Δm_i denotes the difference in the ith metric between two frames.

We can summarize this difference vector as a single term, the cost C:

$$C = \sum_{i=1}^{k} \frac{\Delta m_i}{\sigma_i}$$

We cannot simply add each of the differences in order to calculate a cost. We instead non-dimensionalize each value by a scaling factor σ_i that denotes the average change in a given metric from frame to frame. The goal is to minimize this cost in order link objects from one frame to the next.

We can do this in the context of large microscopy images with many cells by developing a cost matrix. This matrix has elements C_{ij} that denote the cost of linking the ith object in the first frame to the jth object in the next frame. In order to find all linkages, we apply the Hungarian Assignment Algorithm. This algorithm returns the optimal mapping from the set of i objects in the first frame to the j objects in the second frame that has a global minimum cost.

This cost minimization algorithm can also be expanded to link cell objects not just in adjacent frames, but across multiple frames as well. This is useful when a given object will disappear from the field for a few frames at a time, and the tracking algorithm can account for gaps such as these.

It is interesting to note that the Hungarian Assignment Algorithm is a global minimization algorithm. Thus, the output will be the optimal linking. However, sometimes a greedy, or local minimization algorithm will perform better than the global version. In addition, the greedy algorithm has the advantage of being less computationally expensive. The CellTracker program has options to implement both variations for motion tracking.

3 Using the Software

The Input GUI

When beginning an analysis, you will first encounter the input GUI (Figure 1). This GUI will allow you to enter the relevant images, as well as the settings you would like to use for segmentation, background correction, and motion tracking.

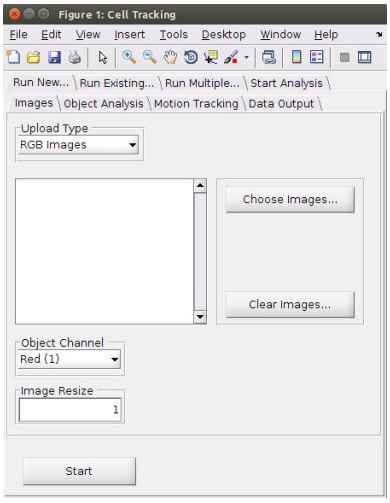

Figure 1: The input GUI.

This GUI consists of four components: Images, Object Analysis, and Motion Tracking.

The Images tab controls the images that will be used in the analysis. The user can upload images in one of three forms: as a list of RGB images, as three lists of split-channel images, or simply as grayscale images. The user then defines the color channel that will be used for object detection. The user can also choose to downsize the images in order to decrease computation time.

The next tab is the Object Analysis Tab (Figure 2). This tab allows the user to select between the two segmentation algorithms discussed above. In addition, the user can select a background correction algorithm, and can choose to filter out cell objects by metric.

The specific settings can be seen in Table 1.

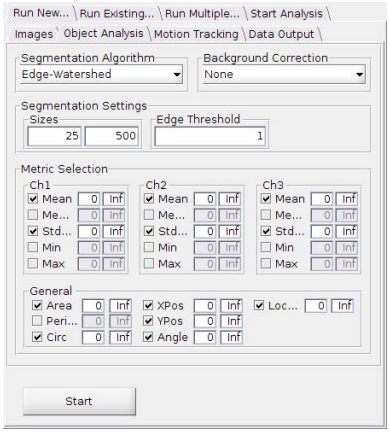

Figure 2: The Object Analysis tab.

Setting	Description
Segmentation Algorithm	Which segmentation algorithm to use
Background Correction	Which background correction algorithm to use
EdgeWater Sizes	Min and max sizes for cell objects (area)
Edge Threshold	Threshold multiplier. Values greater than 1 make edges harder to detect and less than 1 make edges easier to detect.
ThreshWater Sizes	Min and max sizes for cell objects (radius)
Metric Selection	Min and max values of object metric to detect

Table 1: Object Analysis settings.

The Motion Tracking tab controls the settings for the motion tracking algorithm (see Figure 3). The specific settings are described in Table 2.

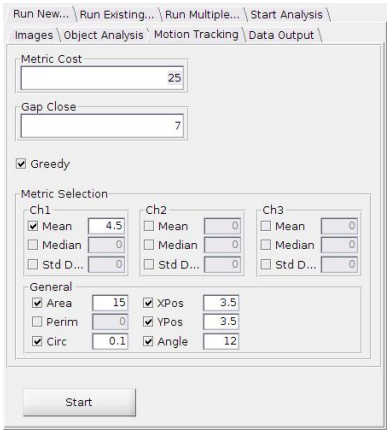

Figure 3: The Motion Tracking tab.

Setting	Description
Metric Cost	Maximum cost that can form a link
Gap Close	Maximum number of frames apart objects can be linked
Greedy	Whether or not to use greedy algorithm
Metric Selection	Object metric to use for motion tracking along with average metric variation

Table 2: Motion Tracking settings.

The final tab, the Data Output tab, allows the user to store the program output as a .mat file for convenient re-upload.

The Viewer

The output of the CellTracker program can be viewed in the Viewer GUI (see Figure 4). The viewer consists of three main parts: the image/graph view, the track/metric selection, and the options tabs.

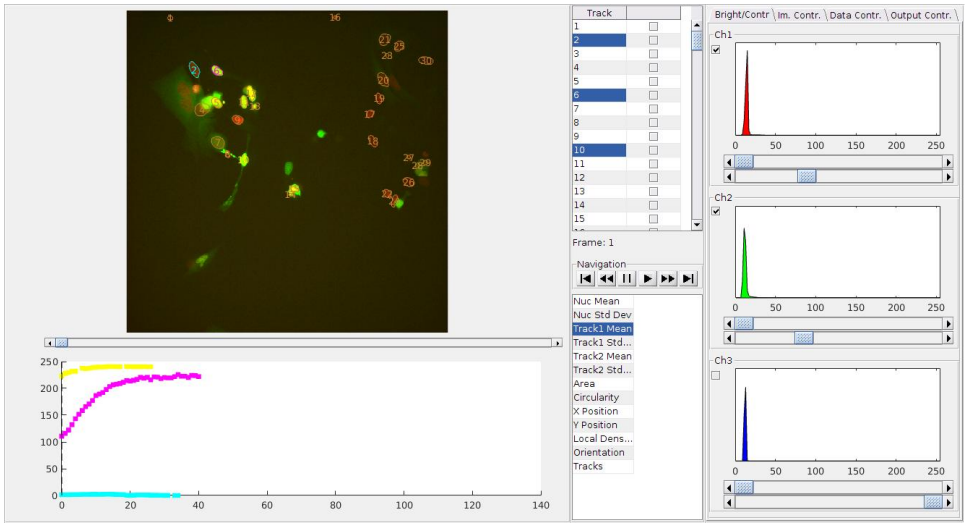

Figure 4: The output viewer.

The image/graph view consists of the microscopy image of the current frame as well as a graph of selected metrics for selected trajectories. The two can be mutually resized by dragging the bottom of the image.

The track/metric selection allows the user to select which trajectories or metrics to view in the image/graph. The user can select multiple trajectories using Ctrl-Click.

The options tabs allows the user to control various options of the viewer. The first, Bright/Contr, allows the user to control the brightness and contrast of each color channel by manipulating the pixel intensity histogram. The Im. Contr. tab allows the user to control how they want to view the trajectories on screen. The Data Contr. tab allows the user to make manual adjustments to the microscopy data in order to make corrections in the event

of incorrect tracking. The final tab, Output Contr. allows the user to output trajectory data.

Editing CellTracker Output

Often times it is necessary to modify, add, or delete cell objects or trajectories because of errors in the object detection/tracking algorithm. This can be done manually by creating new objects by drawing them with the "New Object" button.

In order to edit the outputted object trajectories, we must first distinguish between the object index and the trajectory index. In each frame, there are many objects, and each object is assigned a unique object index. These can be viewed by unchecking the "Show Trajectories" box under the "Im. Contr." tab. When cell objects are linked into a trajectory, the corresponding object indices for each frame are combined into a single list. In turn, the trajectory itself will have its own index, which may not be equal to the object index in that given frame.

Thus, in order to modify a trajectory, we must find the correct object index to incorporate into the trajectory. Once this is found the trajectory can be modified by switching to the Table view under the "Data Contr." tab. Each row of the table corresponds to a trajectory, and each column corresponds to frame number. Thus, the value of the given trajectory at a given time point must be filled with the correct object index.

In addition, it is also possible that the program will incorrectly incorporate a section of one trajectory into the trajectory of another. You can resolve this issue quickly by copying that data from one trajectory to another. To do this, highlight the cells corresponding to the frames of the trajectory you would like to copy over. Then click the "Copy" or "Cut" button under the "Data Contr." tab. Then, click the trajectory you would like to copy into and click "Paste".

Because it is common to manually modify the output of the tracking algorithm in the above way, you can save the modified data by clicking the "Save All Data" button under the "Output Contr." tab. The changes will then be reflected in the subsequent runs of the program.