30 April 2014

New Tracking Code (NTC) Instructions

The code has two phases, Segmentation and Tracking. The Segmentation was developed by Shawn Garbett and Sam Hooke. The Tracking code was developed by Stephen Hummel. As of April 2014, the code runs on MATLAB 2014A. Questions, comments, and concerns should be addressed to the parties responsible for that section.

Segmentation (There is a "README" file in the segmentation folder with this information)

The goal of the segmentation code is to properly segment images. This is not a trivial task and uses several steps. There is a balance between statistics and time since there is a manual component to the segmentation to generate a training set. You need at least 200 objects in nucleus, debris, and under-segmented categories, but it can be time consuming at least initially.

The segmentation process occurs in three steps and alternates between MATLAB and R which is accessed through a terminal.

It should be noted that the you should attempt to select pre-division and post-division cells during the create classifier step. This data will be used in the tracking section to generate a Naive Bayes Classifer for detected potential mitotic events.

Steps to create classifiers:

1. Create training set with Well B02

   ```
   SegmentReview(1, '~/Work/Images', '~/Work/CellAnimation/segmentation/segment/training_2010-05-01-001.mat');
   ```

2. Convert training set into csv file:

   ```
   SetToCSV(objSet, [name of csv file])
   ```

3. Use training set to create classifiers

   From Terminal:

   ```
   perl Train.pl args:
       1. [directory containing csv file of training set]
       2. [name of csv file]
   ```

4. Classifiers are in CellAnimation/segmentation/segment.

   Called model[category].Rdata.

Steps to segment an image stack:

1. Change parameters to reflect your well at beginning of
   - a. LocalNaiveSegment
   - b. LocalFinish
   - c. LocalGMMSegment
2. Run LocalNaiveSegment in Matlab
3. From Terminal -

   ```
   perl Classify.pl          args:
       1. location of csv files
   ```

        2. location of classifiers
        3. imageNameBase
        4. startIndex
        5. endIndex
        6. framestep
        7. digitsForEnum
    4. Run LocalFinish in Matlab
    5. Run LocalGMMSegment in Matlab
        Output .mat files are in [wellname]/output

Steps to track image stack:

1. From GUI editor or text editor, change parameters in assayAltSegment.m to reflect your well.
2. Run assayAltSegment in Matlab

Tracking

The New Tracking Code (NTC) is a relatively simple algorithm focused around a single script that calls several other script functions. All files must be in the same MATLAB folder for the code to work.

The primary script is titled "NewTracking.m". Lines 10-30 are user input lines where the user directs the script to the images, ensures image name is correct, format, as well as several processing factors such as range multiplier and time resolution.

The script then pulls all of the .mat files from the segmented images and reformats the data into a large matrix / dataset so that features can be recalled at any point easily. The cells / objects in every image are then examined using a Naive Bayes Classifier. The classifier is derived from the segmentation review .csv file. The classifier is done using Area, Major Axis Length, Minor Axis Length, Eccentricity, Equivalent Diameter, and Intensity. The user needs to ensure the naive bayes classifier is correctly defined online 96. This should not be an issue but the user should be aware of it. The cells are classified as either "dividing" or "nucleus". This is a string added in the last column of the dataset.

The high confidence tracks are then generated using a nearest neighbor algorithm. This is done in "InitialTracks10" function which looks for cells in the designated range that are not along the edge of the image, not labeled as dividing, and exist throughout the length of the timeframe.

The high confidence tracks are then processed in the "HighConfidDataB10" in order to extract the characteristics of the tracks. This includes the mean distance, mean area change, mean eccentricity change, mean major axis length, mean minor axis length, mean solidity change, and mean intensity change of every cell from image to image of the high confidence tracks. The high confidence tracks and the mean change data are both exported as .csv files.

An area filter is then processed over all objects / cells in the dataset in order to remove debris. The user designates the size in the user input section.

The next function, "Potential" runs a range search for every cell from one frame to the next in order to determine potential matches. The range distance is determined by the mean distance calculated in the high confidence tracks and the range multiplier. For each potential option a probability distribution function is then calculated using a negative log likelihood for the changes in morphological features. This data is then added to the columns in the dataset and exported as a .csv file.

The script then uses the naive bayes classifier string tags for dividing cells in the "MitoticOptionsv2" function to determine mitotic options. For more than two cells tagged as dividing the function generates various dividing options. These options probability distributions for these potentially mitotic cells are then also calculated.

The "MainArray_STE" function then takes the mitotic options and the potential options to generation the necessary array for the integer programming. This process is done for every image in the stack and the array plus the probability distribution functions are used in the integer programming. MATLAB is removing the binary integer programming function this process then uses a mixed integer function. The upper and lower limits are consequently set so that results are binary, ones and zeroes. A one represents a selected object for a track. All of the selections are exported by image number for later reference.

The "MatchProcess" function then works its way through each image using the selected information, the potential matches, and the mitotic options to generate tracks. The potential matches are then processed using the "GlobalTracks" function to generate a track ID. The tracks are constantly cross referenced against previous position to ensure continuity in the tracks. The matches and the global tracks are also both exported as a .csv.

The tracks are then analyzed using "ExtractMitoticData" in order to pull out and format the track information for use in the fractional proliferation function in R. This data is extracted as a .csv file.

The .m files in the "Tracking Autocorrection" folder are still under development and will be implemented shortly.