**p53Cinema single cell analysis package**

**Introduction**

How to get the most out of p53Cinema?

**Getting started**

**0. System requirements**

* Matlab 2013 or higher.
* Matlab image analysis toolbox.
* Imaging data. The software is designed to process grayscale TIF files.
* The panel package from Matlab exchange (only for plotting cellular genealogies).

**1. Suggested file directory structure**

The p53Cinema will accommodate a wide variety of directory structures. One strategy that has worked for us is to work within an experiment folder, with subdirectories for the raw image files and tracking files:

Experiment folder

Subdirectory: RAW\_DATA

Subdirectory: Tracking (The quantification scripts iterate over all .mat files in a specified directory. Having a directory exclusively dedicated to have tracking files will prevent potential errors during data quantification).

When you start Matlab, it is convenient to change directory to the experiment folder:

> cd <experiment\_path, String>;

For example:

> cd ‘O:\\Sysbio\LahavLab\Microscopy\M20181020\_p53DynamicsMovie\’;

**2. Creating a database file**

A database file is a tab delimited file linking image filenames with their metadata.

The database file requires the following columns (the order is not important):

* filename
* group\_label
* channel\_name
* position\_number
* timepoint

Normally it is possible to extract all the relevant information from the filename. Use the function createDatabaseFile as a starting point to generate this table.

> database = createDatabaseFile(<rawdatapath, String>, <outputFile, String>);

For example:

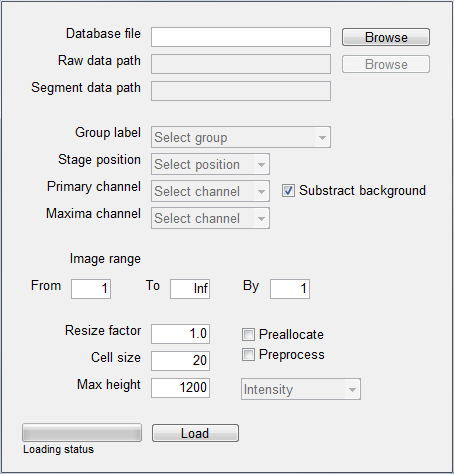
> database = createDatabaseFile(‘RAW\_DATA’, ‘M20181020\_database.txt’);

**3. Open the tracking software**

Use the following command to open the software in Matlab

> master = p53CinemaManual\_object\_master;

This will open the fileManager window



Give the program your database file

Select the RAW\_DATA directory

Choose ***group*** and ***stage position***

***Primary channel***: This is the channel that you will *visualize* while tracking. Background substraction is recommended unless you are visualizing brightfield images.

***Maxima channel***: This is the channel that the software uses to *predict* the location of individual cells. In most cases this will be the same as the primary channel (nuclear marker).

You should check ***preallocate*** and ***preprocess*** to get the most out of the software. In particular, preprocess will use the ‘Maxima channel’ to identify cells in each images by identifying local maxima in images in based on intensity or shape criteria.

You can load only a subset of images if you will, or load all images if you set the ‘To’ field to Inf. This is useful to test how well the software is identifying cells. You can also partially track datasets, quantify data and complete cell tracks later.

The ***cell size*** parameter is the size of the window used to blur images to identify single local maxima associated with individual cells. A higher number will identify fewer local maxima per cell, but it may fail to distinguish between cells that are next to each other.

Click ***Load*** to start tracking

Change this number to match the size of your monitor.

This will open four windows:

1. imageViewer

Cell fate events are annotated as rings surrounding centroids. Yellow denotes division events. If there is a blue star, it represents that two cells divide at the same centroid (linked sisters).

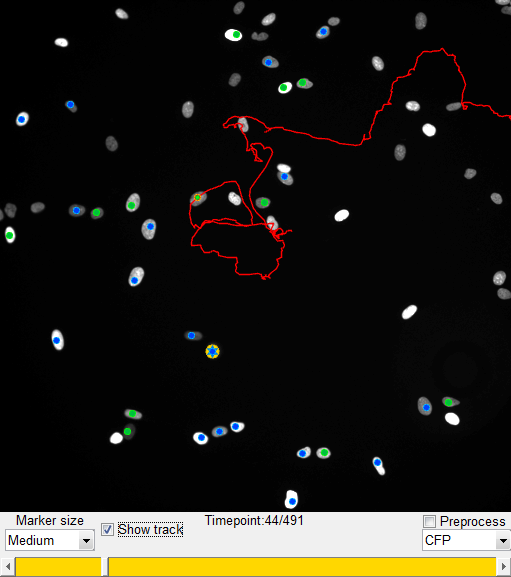
Change the current channel to visualize. Note that because it is not preallocated, the software will read images one by one, and thus it is rather slow. Sometimes looking at other channels is helpful to solve ambiguities in tracking.

Check ‘preprocess’ to substract background in this alternate channel.

You can choose to show the full track of the current cell (red) or not.

You can change the size of the circles using this drop menu.

Centroids are annotated as colored circles. The red centroid is the current cell. Green centroids are cells present at the start and end of the image sequence. Blue centroids are absent in either the beginning/end of the image sequence (incomplete tracks)



contrast

zoomMap

cellTracker.

**3. Start tracking**

*How do you create and edit tracks?*

Use tracking mode toggle button to open and close individual tracks.

1. Activate ‘tracking mode’ and click a cell that doesn’t have centroid to open a new track.
2. Track this cell (described in detail below).
3. Deactivate the ‘tracking mode’ to finish this track (you can always edit it later).
4. To track a new cell, reactivate ‘tracking mode’ and click a place that doesn’t have a centroid.
5. To edit a pre-existing track, reactivate ‘tracking mode’ and click a pre-existing centroid.
6. The p53Cinema software uses the activation/deactivation cycles of ‘tracking mode’ to know whether you continue tracking a cell, or open/edit another track. Once you get experience, you can use the ‘Tab’ as a shortcut for clicking the ‘tracking mode’ button twice.

*How does tracking work?*

Every time you click the software predicts where the cell is in the next image using a nearest neighbor criterion. The software also automatically links centroids when you scroll through the image sequence or when you use the keyboard shortcuts ‘>’ and ‘<’.

If the centroid linkage is working well, you can propagate tracks without the need of manual scrolling. The software will stop when there is a point of ambiguity in the tracking, in which case you can indicate which the correct centroid is and continue tracking. If you uncheck ‘Play tracking’, the software will automatically construct the track and jump to the point of ambiguity. In some cases, this can save a lot of time.

*How do you delete centroids and correct mistakes?*

Click the ‘backspace’ button to delete a centroid. The track should switch from red to green color. This means that the track exists, but there is no longer a centroid for such track in the current frame. If you click ‘backspace’ while a track is green, you will delete the whole track (think of a centroid in a frame as a shield for the track). You can also use the ‘delete forward’ and ‘delete backwards’ buttons to delete all centroids from a certain frame all the way to the end/beginning of the image sequence.

*How do you annotate cell fate events?*

To annotate division events, click the ‘division’ button or the ‘space’ bar. To annotate death events, click the ‘death’ button or the ‘Ctrl’ key. You can use the ‘up’ and ‘down’ keys to jump between division events. Use the ‘track sister’ button to start a new track from a division event and link sister cells. Linked sisters will show a yellow circle with a blue star.

When p53Cinema fails to identify a centroid, you can manually impute it using the mouse right click. This is common when there are differences in intensities in nearby cells, and it can be alleviated by choosing a smaller ‘cell size’ in the fileManager, at the expense of potentially more centroids within individual nuclei.

*How do you annotate cell fate events?*

Use the ‘save tracks’ button to save current tracks. The software does NOT save tracks by default, so be sure to save them before you close the GUI. Use the ‘load tracks’ button to load them into a new tracking session. You can use the centroidsBackup script to convert .mat tracking files into tab-delimited files. This is particularly useful to interface with other packages or to integrate tracking with your custom scripts:

> centroidsBackup(<rawdata\_path, String>);

For example:

> centroidsBackup(‘tracking’);

p53Cinema will also load tab-delimited centroids with the following columns:

* cell\_id
* centroid\_col
* centroid\_row
* timepoint
* division (0 or 1)
* death (0 or 1)
* value (it is ok if this is a column of 1)

*Note about keyboard shortcuts*

The use of keyboard shortcuts requires that the imageViewer is the active figure. Clicking a button/checkbox/slidebar will make keyboard shortcuts irresponsive. To use them again, just click a centroid in the imageViewer and continue tracking.

**4. Quantify data**

*Obtain single cell tracks by sampling pixels around centroids (recommended to get started)*

This method will:

1. Read images corresponding to a particular channel and substract background, for each tracked position and timepoints.
2. Sample an area around centroids as defined by the blurRadius parameter to quantify the intensity of individual cells.
3. Aggregate information from all tracking files in single cell matrices of dimensions NxM. N is the number of single cells in the dataset and M is the number of timepoints in the movie.
4. Generate single cell matrices for division events and death events.
5. Consolidate information from linked sisters so that both sisters contain intensity and cell fate information from mother tracks.

Missing data is has a value of -1. This method measures only one channel at a time. You have to run it independently to measure every channel in your dataset.

> measurements = getDatasetTraces\_fillLineageInformation(<database\_file, String>, ...

<rawdata\_path, String>, <tracking\_path, String>, <flatfield\_path, String>, ...

<channel\_name, String>, <blurRadius, Integer>);

For example

> measurements = getDatasetTraces\_fillLineageInformation(‘M20181020\_database.txt’, ...

‘RAW\_DATA’, ‘tracking’, ‘’, ‘YFP’, 7);

You can access single cell matrices using:

> traces\_YFP = measurements.filledSingleCellTraces;

> divisions = measurements.filledDivisionMatrixDataset;

> annotation = measurements.cellAnnotation;

*Obtain single cell tracks by locally segmenting cells*

This method conducts a local segmentation in a segmentationChannel and quantifies multiple parameters for each individual cell in a collection of measurementChannels.

> measurements = getDatasetTraces\_localSegmentation(<database\_file, String>, ...

<rawdata\_path, String>, <tracking\_path, String>, <flatfield\_path, String>, ...

<measurementChannels, cell of strings>, <segmentationChannel, String>);

For example

> measurements = getDatasetTraces\_fillLineageInformation(‘M20181020\_database.txt’, ...

‘RAW\_DATA’, ‘tracking’, ‘’, {‘YFP’, ‘CFP’, ‘Cy5’}, ‘CFP’);

You can access single cell matrices using:

> traces\_YFP\_brightestPixels = measurements.singleCellTracks\_foci{1};

> traces\_YFP\_mean = measurements.singleCellTracks\_mean{1};

> traces\_Cy5\_mean = measurements.singleCellTracks\_mean{3};

> traces\_area = measurements.singleCellTracks\_area;

> traces\_solidity = measurements.singleCellTracks\_solidity;

*Obtain only cell fate event tracks (much faster)*

If you are only interested in division/death events, you can ignore intensity quantification using:

> measurements = getDatasetTraces\_ignoreTraces(<database\_file, String>, ...

<rawdata\_path, String>, <tracking\_path, String>, <flatfield\_path, String>, ...

<channel\_name, String>);

For example

> measurements = getDatasetTraces\_fillLineageInformation(‘M20181020\_database.txt’, ...

‘RAW\_DATA’, ‘tracking’, ‘’, ‘YFP’);

You can access single cell matrices using:

> divisions = measurements.filledDivisionMatrixDataset;

> annotation = measurements.cellAnnotation;

> centroid\_col = measurements.centroid\_col;

Note that since you don’t have intensity quantification, you can rely on the centroids matrix to know whether a cell was tracked in a particular timepoint (centroid\_col(i,t) > 0) or not (centroid\_col(i,t)= 0).

**5. Data analysis**

**5. Advanced tips**

*The database file allows you to annotate and organize your files without modifying filenames:*

Tip 1. Use the group\_label field to define different treatments.

> database = readDatabaseFile(‘M20181020\_database.txt’);

> database.group\_label(ismember(database.position\_number, 1:10)) = {‘non\_treated’};

> database.group\_label(ismember(database.position\_number, 11:20)) = {‘irradiated’};

> writetable(database, ‘M20181020\_database.txt’, ‘Delimiter’, ‘\t’);

Tip 2. Merge two parts of an interrupted experiment.

> database = readDatabaseFile(‘M20181020\_database.txt’);

> part1 = strcmp(database.group\_label, ‘M20181020\_part1’);

> part2 = strcmp(database.group\_label, ‘M20181020\_part2’);

> database.timepoint(part2) = database.timepoint(part2) + ...

max(database.timepoint(part1));

Tip 2. Link movie with immunofluorescence data, by merging timepoints from two datasets and consolidating nuclear marker channel names.

> database = readDatabaseFile(‘M20181020\_database.txt’);

> part1 = strcmp(database.group\_label, ‘Movie\_20181020’);

> part2 = strcmp(database.group\_label, ‘immunofluorescence\_20181021’);

> database.timepoint(part2) = database.timepoint(part2) + ...

max(database.timepoint(part1));

> database.channel\_name(strcmp(database.channel\_name, ‘DAPI’)) = {‘CFP’};