Image analysis pipeline

This document describes our image analysis pipeline used for analyzing the microfluidic data. This pipeline contains a series of MATLAB scripts, and some steps generate intermediate data structures (as in .mat or .xls format). The functions in the pipeline are listed as the following table, which will be explained in the next sections in details.

**The steps A0 to A4 perform shift correction for the on time-lapse image stacks and image pre-processing.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Function | Input data structure | Output data structure |
| A0\_enter\_global\_var | Define general variables for the following steps. | NA | NA |
| A1\_stack\_shift\_correction | Generate shift correction data for an image stack. | image stack (.tif) | shift correction data (.mat) |
| A2\_stack\_crop | Crop raw image stack into ROI image stack with and shift correction. | image stack (.tif),  shift correction data (.mat) | ROI image stacks (.tif) |
| A3\_stack\_crop\_multi | From an ROI image stack, crop multiple microfluidic chambers region into small ROI image stacks. | ROI image stacks (.tif) | small ROI image stacks  (.tif) |
| A4\_stack\_deback | For each small ROI image stacks, remove the microfluidic chamber background of phase channels, and crop the fluorescence channels accordingly. This generated the debacked image stack. | small ROI image stacks (.tif), an image with empty microfluidic chamber (.tif) | debacked image stacks (.tif) |

**The steps B0 to B3 perform cell segmentation, kymograph algorithm and cell cycle construction**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Function | Input data structure | Output data structure |
| B0\_stack\_preseg | For each image in the debacked image stack, perform 2D segmentation for cell outline and cell core region. This generates a segmented image stack. | debacked image stack (.tif) | (1) Segmented image stacks (.mat)  (2) Segmented image panel (.jpg) |
| B1\_stack\_kymoseg1 | Convert segmented image stacks into kymograph. Find cell separators on kymograph. | segmented image stacks (.mat) | (1) Cell separator (.xls)  (2) Separators/kymograph plots (.jpg) |
| **Manual curation after B1:** By inspecting the Separator/kymograph plots, the user can manually correct the separator if there are errors. The correction is done by modifying the .xls file sheet. | | | |
| B2\_stack\_kymoseg2 | Correct the cell separator by importing the manually curated data. | Cell separator (.xls) | (1) Cell separator (corrected) (.xls)  (2) Separators/kymograph plots (corrected) (.jpg)  (3) Separator list (.xls) |
| **Manual curation after B2:** By inspecting the Separator/kymograph plots, the users can manually remove or add separators by modifying the separator list. | | | |
| B3\_stack\_kymoseg3 | Convert the corrected separator data into cell cycle data; also add fluorescence intensity information on cell cycle data. | (1) Cell separator (corrected) (.xls)  (2) Separator list (.xls)  (3) Segmented image stacks (.mat) | Cell cycle data (.mat) |
| B4\_stack\_collect | Collect all cell cycle data within the same field of view (FOV). | Cell cycle data (.mat) | Cell cycle FOV dataset (.mat) |
| B5\_combine\_dataset | Combine all cell cycle datasets with the same experimental condition | Cell cycle FOV dataset (.mat) | Cell cycle combined dataset  (.mat) |

**The steps C1 to C4 generate data structures for cell cycle analysis and lineage analysis**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Function | Input data structure | Output data structure |
| C1\_collect\_cell\_cycle | Collect cell cycles and append intensity data on each cell cycle. Generates a data structure cc\_array. | Cell cycle combined dataset (.mat) | Cell cycle array (.mat) |
| C2\_collect\_MCL | Collect mother cell lineage (MCL), and append intensity data on the lineage.  Generates a data structure MCL\_(dataset\_name). | Cell cycle combined dataset (.mat) | Mother cell lineage data 1 (.mat) |
| C3\_analyze\_MCL | Analyze the ATP dynamics on mother cell lineage. Perform EMD and wavelet analysis. Generate a data structure  MCL\_data(dataset\_name) and  MCL\_sumary(dataset\_name). | Cell cycle combined dataset (.mat) | Mother cell lineage data 2, 3 (.mat) |
| C4\_tree | Construct tree structure of each microfluidic chamber. This generates a tree\_(dataset\_name). | Cell cycle combined dataset (.mat),  cell cycle array (.mat). | Triad dataset (.mat) |

**Input data:** Our image stacks typically contain three channels:

c1: phase contrast image

c2: 405nm excitation with 540nm emission

c3: 488nm excitation with 540nm emission

The channel c1 is used for cell contour and the channels c2, c3 are for fluorescence sensors. The raw images are .TIF files located in a single folder.

**A0\_enter\_global\_var:** The A0 step simply specifies the common parameters (such as path and image stack frame number) for the image analysis. By execute this script, these parameters are loaded into the Matlab workspace as variables.

**A1\_stack\_shift\_correction:** The A1 step perform shift correction for image stacks (typically the phase contrast channel is used). To estimate the image shift for each frame, we find a fixed feature (such as a small bubble embedded in microfluidic PDMS) and track the movement of this feature. This allows us to know the xy shift of image (the rotation movement is usually negligible and not considered).

Practically, we crop a small region from image stacks and track the center of mass of the fixed feature (either brighter or dimmer than the background). The feature tracking can perform for each frame or for every k frame where k is an integer. The xy-shift data will be obtained by interpolation for every frame. The shift correction data is exported as data\_interpolate.mat, and the parameter used in shift correction is exported as param\_Stack\_shift\_Correction.mat, in the same folder of images stacks.

**A2\_stack\_crop:** The A2 step is to crop the region of interest (ROI) from the raw image stack, with the shift correction data generated in A1 step. The export .TIF files are arranged in three separate folders (for channel c1, c2, c3, respectively). The ROI data are used for visual inspection (for example, by ImageJ) and used in the following steps.

**A3\_stack\_crop\_multi:** The A3 step is to crop image stacks for each microfluidic chamber. Our microfluidic device has multiple identical chambers, and for typical field of view (under 100X objective) there are 8 to 12 chambers with cells. To crop the regions of these chambers, we specify each chamber using its tip coordinate as reference point (orange circle in Figure 1). By specifying the distance of left, right, up, and down with respect to this reference point, the script crops a smaller image stack. The procedure is performed in parallel for all microfluidic chamber in the same field of view. For each microfluidic chamber, the cropped data is saved as .TIF files in separator folders.

|  |  |
| --- | --- |
| A picture containing text, indoor  Description automatically generated | Figure A3. Crop multiple microfluidic chambers from image stack. The reference point of the middle chamber is labeled in orange circle. By specifying the distance of left, right, up, and down with respect to this reference point, the script crops a smaller image stack (yellow rectangular). |

**A4\_stack\_deback:** The A4 step remove the microfluidic features from the phase contrast image and leave the cell features. This procedure greatly facilitates the following image analysis and is referred to as the “deback” step. Practically, we need to provide an empty microfluidic chamber and align the reference points between the data chamber with cells (acquired from step A3) and an image of empty chamber. The deback algorithm performs another shift correction to align the empty chamber with the data image stack, and the microfluidic background is subtracted for the entire data image stack. After deback procedure, data is saved as .TIF files in a new folder.

|  |  |
| --- | --- |
| A picture containing text, lined  Description automatically generated | Figure A4. Demonstration of “deback” process.  (a) An empty microfluidic chamber  (b) A normal microfluidic chamber with cells  (c) The inverted phase image after deback step. The processed image becomes similar to a fluorescence image with uniform signal in the cells. |

**B0\_stack\_preseg:** The B0 step perform pre-segmentation for cells, using the processed phase image from A4 step. We use adaptive threshold method for the segmentation. The adaptive threshold method converts the processed phase image (16-bit data matrix) into a binary matrix (1 or 0 for each pixel). For each pixel we compare the intensity with the neighboring pixels. A pixel is assigned to value 1 in the output matrix if it satisfied two criteria: (i) significantly brighter than the neighborhood and (ii) brighter than the minimal level.

|  |
| --- |
| Let be the input data matrix. Let be a Gaussian kernel, we define two matrices and by    where is the Hadamard product defined by .  The criterion (i) is  The criterion (ii) is  The parameter are chosen based on the input data matrix. Pixels satisfied both criteria are assigned as 1 in the output result. |

We use two parameter sets for the segmentation. One parameter set is for labeling the cell area (referred to as “outer contour”), and the other parameter is for separating cells in the next step (referred to as “inner contour”). Example of outer and inner contours are shown in below. The segmentation data is saved as mask\_record.mat.

|  |  |
| --- | --- |
| A picture containing text  Description automatically generated | Figure B0. Figure panels comparing raw input data (R) and segmentation result (S).  In segmentation panels, the pixels within inner contour are labeled with white color, while the pixels between outer and inner contours are labeled with gray color. |

**B1\_stack\_kymoseg1:** The previous segmentation gives binary matrix as output. For typical image analysis methods, the segmented area is cut into “cells” based on the pixel connectedness. However, the pixel connectedness is not robust, and this method could give erroneous call for cell division and merging for multiple times. Biologically, cell can only divide one time and this time point need to be decided properly.

For mother-machine dataset, cells are aligned in one row and there is a significant advantage for tracking cell cycle and cell division. We use an algorithm called “kymograph segmentation”, starting with visually inspect the microfluidic chamber with time (see figure below). In the kymograph figure panel, the cells are separated by virtual red lines called “separators”. The separators follow a couple rules:

1. A new separator corresponds to a cell division event (yellow circle).

2. Separators cannot stop in the middle of kymograph (since divided cells cannot merge again).

3. Separators either exist in the bottom of the kymograph (cell exit of microfluidic chamber) or extend to the right edge of the kymograph.

Tracking separators on kymograph is easier than tracking cells on kymograph, since the latter has branching structure caused by cell division. Our goal is to use algorithm to automatically find separators and with some manual curation. Once separators are constructed, the information of cell division and cell cycle can be extract from the separator.

|  |
| --- |
| A picture containing text  Description automatically generated |

In practice, we constructed the kymograph by following procedure: (1) For each time frame, take the 2D segmentation matrix and perform averaging of each row. This allows us to convert the 2D matrix into a 1D column vector, where the vector with lower intensity correspond to the region with no cells or constricted cells. (2) Concatenate the 1D column vector of each time frame and forming a kymograph matrix. (3) For visualization purpose, we inverted the kymograph such that the cell region is dark, and the separator region is bright.

A customized algorithm is used for detecting the separators. Briefly, for each column we find peaks for the bright region. Then, peaks between adjacent column were linked together and forming the separators. As demonstrated in the figure below, each separator is a list of peaks across multiple time frames with no breaks in the middle. The separators either exist in the lower part of kymograph or extend to the last column of the kymograph.

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence |
| Figure B1. Kymograph and separators. Cell regions in kymograph are showed in dark color. Separators are indicated by random color. Red circles are short separators (by algorithm error) which will be filtered out in the following analysis. |

All the intermediate data of this step is exported in MATLAB structure kymoseg\_rec1.mat. The kymograph and separators are exported in .jpg figures for manual inspection. In addition, the kymograph linking data is also exported in .xls files “*peak\_linking\_original.xls*” (used as an original copy) and “*peak\_linking\_corrected.xls*” (used for manual curation step).

**Manual curation after B1:** In some cases, the algorithm may identify erroneous separators. The error can be visualized by the kymograph/separator plots exported from B1 step. To correct these error, manual curation can be applied by modifying the “*peak\_linking\_corrected.xls*” file. The file contains a table for each linking steps, and this table can be curated.

**B2\_stack\_kymoseg2:** This step continues the kymograph segmentation procedure. The algorithm reads the file “*peak\_linking\_corrected.xls*” and re-analyze the separator. All the intermediate data of this step is exported in MATLAB structure kymoseg\_rec2.mat. The corrected separators and kymographs were exported again in a new folder. In addition, in this step the algorithm generates a list for separators and export as “*sep\_manual\_check.xls*” and allow us to manually remove the short/erroneous separators.

**Manual curation after B2:** In this step, the user could manually remove the erroneous separators by modifying the file “*sep\_manual\_check.xls***.**

**B3\_stack\_kymoseg3:** This is the last step for kymograph segmentation. The algorithm read the file “*sep\_manual\_check.xls”* and the data fromkymoseg\_rec2.mat to generate the final separator structure. The pre-segmented data were cut by the separator and generate a list of cells. Cells divisions and are inferred from the separator structure. Cell cycle and cell lineage are obtained similarly. In addition, the fluorescence data (c2 and c2) were recorded on each cell cycle.

All data are exported as kymoseg\_rec3.mat. In addition, the algorithm also exported figures showing the cell segmentation result and cell cycles (see below) for visual inspection.

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence |
| Histogram  Description automatically generated |
| Figure B3. (Up): Cell segmentation figure. Cells are colored from blue to red, according to the order in microfluidic chamber. (Down): Cell cycle figure. Each cell cycle is labeled by a random color, while the circle indicates the center of mass of cells at time t. |

**B4\_stack\_collect:** This step collects the cell cycle data of microfluidic chambers within same single field of view (FOV). The data is exported as “cc\_ensemble\_(datasetname).mat”.

**B5\_combine\_dataset:** This step combines the datasets of previous B4 step (for example, same experimental conditions from different FOV or different days). The combined data is exported as “combined\_dataset\_(datasetname).mat”.

**C1\_collect\_cell\_cycle:** This step collects all cell cycles from the B5 step and appends the measurement data, including cell size, fluorescence levels, and the mother/daughters of each cell cycle. The data is exported as “CC\_(datasetname).mat”.

**C2\_collect\_MCL**: This step collects the information of mother cell lineage (MCL). In a microfluidic chamber, the mother cell is defined as the cells located in the terminal position. The catenated data from all cell cycles of mother cell is collected. The data is exported as “MCL\_combined\_(datasetname).mat”.

**C3\_analyze\_MCL**: This step further analyzes the mother cell lineage data. The trajectory of ATP is subjected to empirical mode decomposition (EMD) to find the baseline, and the signal (with baseline subtracted) is subjected to the wavelet analysis. The data are exported as “MCL\_PS\_(datasetname).mat” and “MCL\_summary\_(datasetname).mat”.

**C4\_tree**: This step analyzes the tree structure of each microfluidic chamber. The tree structure is constructed. Next, triad structures (mother-daughters) are found, and the cell cycle data of triads are assembled for following analysis. The data is exported as “Tree\_(datasetname).mat”.

Post analysis

The image analysis pipeline described in above generates basic dataset for cell cycles, mother cell lineages, and tree structures. The post analysis scripts are used for visualization and for analyzing various types of correlations.

|  |  |  |
| --- | --- | --- |
|  | Function | Input data |
| P1a Cell kymograph | Create kymograph plots | Phase and fluorescence data from “debacked” images |
| P1b Movie | Create images similar to the kymograph plot, but export as image stacks for the movie. | Phase and fluorescence data from “debacked” images |
| P2a Lineage\_plot | Plot cell size and ATP dynamics of mother cell lineages. | MCL\_combined\_(datasetname).mat |
| P2b Wavelet scalogram | Plot scalogram from wavelet analysis. | MCL\_combined\_(datasetname).mat  MCL\_PS\_(datasetname).mat |
| P3a CC\_analysis | Analyze the cell cycle datasets and save as ATP\_stat.mat. | CC\_(datasetname).mat |
| P3b Boxplot for cc | Creating boxplots for cell cycle statistics. | ATP\_stat.mat |
| P4 Tempogram | (1) Generate tempogram statistics, save as “tempogram\_summary”. (2) Plot tempogram. | CC\_(datasetname).mat |
| P5 ATP\_GR\_correlation | Generate correlation plot of ATP statistics with single-cell growth rate. | ATP\_stat.mat |
| P6 ATP\_2D\_statistics | Generate 2-dimensional distribution plot for ATP statistics and single-cell growth rate | ATP\_stat.mat |
| P7a Sibling and Triad statistics | From tree structure data, assembling sibling statistics (mother cell could be missing) and triad statistics (must have complete mother-daughters). Save as sibling\_stat.mat  and triad\_stat.mat | Tree\_(datasetname).mat |
| P7b ATP\_stat\_type | The script “cc\_array\_different\_types.m” extract the cell cycle data with different cell types (i.e. pole classes relative to the mother cell lineage). The result is saved as “cc\_array\_type”. This data structure is further analyzed by script “ATP\_stat\_type.m” and export as ATP\_stat\_type.mat | sibling\_stat.mat |
| P7c Cell\_pair\_analysis | (1) “correlation\_parent\_daughter.m” generate scatter plots between statistics of parent and daughters. (2) “correlation\_siblings.m” generate scatter plots between statistics of siblings. | ATP\_stat\_type.mat |
| P8a Decorr\_data | This script creates the data structure “ATP\_traj\_data.mat”, which containing ATP trajectories between siblings. | Tree\_(datasetname).mat |
| P8b Decorr\_statistics | (1) The script “Corr\_decay\_data.m” analyzes the data structure “ATP\_traj\_data.mat” and calculate the correlation for each time points after cell division. The result is summarized as “corr\_data\_ensemble”. (2) The script “Plot\_corr\_decay.m” analyzes the “corr\_data\_ensemble” data and plot the decay curve. | ATP\_traj\_data.mat |