alt text

MorphoSegger

MorphoSegger is a Matlab package for imaging analysis of bacterial cells that combines two powerlful tools, SuperSegger and Morphometrics. SuperSegger provides machine-learning based segmentation (with training GUI, foci and fluorescence analysis) and Morphometrics that interpolates contours at subpixel resolution for robust cell size and morphology estimation. Fiji macros are integrated in Matlab through MIJ to expand the range of tools for pre-processing and analysis. This tool is most suited for analysis of bacteria in agar pads or CellASIC ONIX platforms.

Installation

Before running MorphoSegger, make sure to get familiar with SuperSegger, Morphometrics and Fiji macros.

Requirements:

Matlab:

- Curve fitting Toolbox
- Statistics and Machine Learning Toolbox
- Deep Learning Toolbox
- Global Optimization Toolbox
- Parallel Computing Toolbox

Others:

- Running installation of Fiji and MIJ
- Adapted code of SuperSegger and Morphometrics included in MorphoSegger package (see below for a list of modifcations).

Setting up MIJ:

The easiest way to install MIJ is via the repository. Click *Update > Manage update sites* on Fiji and check the repository BIG-EPFL. It might be useful to add the ImageJ-Matlab repository as well. Finally, add the path for Fiji scripts in Matlab: `addpath '/Applications/Fiji.app/scripts'`. Type `Miji` in Matlab to make sure it loads properly.

Note: Set Matlab memory settings to accommodate your RAM: *Preferences > General > Java Heap Memory*

Modifications to SuperSegger and Morphometrics:

Morphometrics:

- The C libraries were updated, now it works with MAC and LINUX. To run it in specific distribution, unzip the MM_mexfiles for the appropriate OS and add them to the path (Remove other ones from path to prevent problems). In the MAC version, adjust privacy settings to prevent the system blocking the libraries.

- Several bugs of command line scripts were fixed. The simply_segment_cl.m script works now to segment binary masks via simple threshold. The false_pos.m file was used instead of a more complicated version in simply_segment_cl.m
- The option to save cells_table_ID was added in the command line version, previously only in the GUI.
- A script morphometrics_mask_cl.m was created to run specifically in binary masks. The run_parallel.m script was added to run Morphometrics using Parallel Computing toolbox from Matlab. It runs multiple XY positions in parallel (multiple cores in a laptop computer).
- Several bugs were in the view_contours function. The mesh and contours visualizations can now be saved as a .tif stack to inspect the quality of the analysis.
- The parameter files in morphometrics can now be loaded as a structure and modified in the processExp pipeline file using the v2struct function (added to the package).
- A script cleanMorphometrics.m was added to delete temporary files *Gparent.tif and *Gsegt.tif to prevent errors when re-running Morphometrics.

SuperSegger:

- trackOptiStripSmall was modified to save the binary mask after segmentation. It saves the info present in the *seg.mat files.
- intCheckForInstallLib was updated to recognize the Deep Learning Toolbox (previously known as Neural Network Toolbox)
- In BatchSuperSeggerOpti the clean_flag user input prompt was commented since clean)flag is always used in MorphoSegger.
- In BatchSuperSeggerOpti the line closing the parallel job was commented so the parallel pool is still active when running Morphometrics.
- A line was added in trackOpti so the **cell** directory is no longer generated since is not being used.
- The temporary images generated in SuperSegger are deleted in the pipeline script processExp to save space.

Usage

Note: *check this youtube tutorial to get quickly started.*

1. Check all requirements above, then download the tool:

```
git clone https://github.com/batflorez/MorphoSegger.git
```

1. Prepare a **processExp** pipeline script for an individual experiment adjusting the different variable settings. The experiment consists of a folder with numbered .ND2 files for this version, but it can be easily modified.

The most commonly modified variables are:

Selects part of the time lapse that will be extracted from the ND2 files. This is useful to reduce processing time and prevent dealing with dead cells towards the end of the time lapse:

```
    % Select frames to analyze
    t_start=1;
    t_end=60;
```

If you have observed shifts between channels, SuperSegger allows you to correct it by setting the pixel shifts in X and Y. Simply input those values in the last 2 columns of the matrix which are the X and Y pixel shifts respectively. A way to calculate those values is by manually shifting the image with the Channel Aligner tool from MicrobeJ that runs in Fiji. If you want to automatically calculate those values using fluorescent beads data, refer to intAlignIm function in SuperSegger.

```
    %Alignment constants
    CONST.imAlign.Phase     = [ 0.0000    0.0000    0.0000    0.0000];
    CONST.imAlign.GFP       = [ 0.0000    0.0000    0.0000    0.0000];
    CONST.imAlign.mCherry   = [0.04351   -0.0000    0.7200    0.5700];
    CONST.imAlign.DAPI      = [0.0000     0.0000    0.0000    0.0000];
```

Advanced segmentation parameters in SuperSegger need to be modified for each experiment. An easy way to check if the paramters are appropriate for the dataset is to run the trainingGui function, double-click in modify parameters, load a single tif image and click run. The slide bars will allow you to test different parameter values and observe segmentation quality, and the final values can be input in the processExp script:

```
CONST.superSeggerOpti.PEBBLE_CONST = 1.3;          %Useful for removing
debris
CONST.superSeggerOpti.INTENSITY_DIF = 0.3;         %Useful for removing
debris
CONST.superSeggerOpti.remove_microcolonies =false; %Deletes clusters of
cells.
CONST.superSeggerOpti.remove_debris = 1;           %Deletes debris and
bubbles
CONST.superSeggerOpti.MAX_WIDTH = 1e15;            %It sets cell length to
split cells
CONST.superSeggerOpti.MAGIC_RADIUS = 7;            %Contrast enhancement.
Deletes areas between cells
CONST.seg.OPTI_FLAG = false;                       %Segments cells by
shape
CONST.regionOpti.MIN_LENGTH = 8;                   %Minimum length of
cells
CONST.trackOpti.MIN_AREA=80;                       %Threshold to filter
small particles
```

Set this variable to the proper time interval. Important when running Foci Analysis:

```
CONST.getLocusTracks.TimeStep = 5;
```

This parameter sets the minimum area of a cell in Morphometrics. It helps clean up segmentation errors occurred in SuperSegger.

```
params.f_areamin = 80;
```

1. Once the settings are defined, the pipeline can run in Matlab by typing:

```
% 1. Preparing files            - preprocess
% 2. Convert file names         - naming
% 3. SuperSegger Segmentation  - supersegger
% 4. Clean up  files            - cleanup
% 5. Convert images to stack   - imag2stack
% 6. Morphometrics              - morpho

>>procesExp(1,1,1,1,1,1)
```

The pipeline will run and process ND2 files, converting them to tif, renaming to SuperSegger format and finally performing segmentation and Morphometrics calculations on the output binary masks from SuperSegger. The pipeline is modular, so different steps can run independently, just make sure to change the appropriate boolean input variables.

It takes around 70 min to run an experiment with 12 XY positions, phase and fluorescence, mid cell density, 60X and around 90 time points in 6 cores. For more details about changing parameters of Morphometrics or SuperSegger please refer to their respective documentation.

Contributing

Pull requests are welcome. For major changes, please open an issue first to discuss what you would like to change. Please make sure to update tests as appropriate.

License

license mit

MIT license