Analysis of cell morphology and fluorescence / agarose pads / Unet

Alexandros Papagiannakis, Christine Jacobs-Wagner lab, Sarafan ChEM-H, Stanford University, 2023

This notebook provides an analysis example of the cell morphology and fluorescence statistics from the segmented cell labels agarose pads. The functions applied in this analysis are included in the *unet_snapshots* class included in the *snapshots_analysis_UNET_ghv.py* script.

In [ ]:

```
import os
import pandas as pd
import numpy as np
```

1. Adding the functions/classes to the path

In [ ]:

```
# includes all functions used for the analysis of the Brownian statistics
module_path = os.path.abspath(os.path.join('..'))
if module_path not in sys.path:
    sys.path.append(".../snapshots_analysis_UNET_ghv.py")

# Check the path to confirm the analysis functions are included
print(sys.path)

%run "....\snapshots_analysis_UNET_ghv.py"
```

2. Variable definition

The experiment folder includes the tif images of the segmented cell labels as well as the nd2 microscopy files.
The analysis is performed by directly reading the nd2 files and applying the cell segmentation masks. The class includes an nd2 reading function.

In [ ]:

```
experiment_path = ".../experiment_folder"
save_path = ".../results_folder"

def get_paths_and_folders(experiment_path):

    experiments = os.listdir(experiment_path)
    experiments = [x for x in experiments if len(x)==19]
    snapshots_paths = [experiment_path+'/'+x for x in experiments]
    unet_path = os.listdir(experiment_path)
    unet_path = [x for x in unet_path if 'phase_tif' in x][0]
    unet_paths = [experiment_path+'/'+unet_path+'/'+x+'_tif' for x in experiments]

    return snapshots_paths, unet_paths, experiments

snapshots_paths, unet_paths, experiments = get_paths_and_folders(experiment_path)
```

3. Medial axis definition, cell average statistics and diffraction limited particle positions

This code iterates over the agarose pads (experiments variable) and XY positions (iteration within the class) to draw the medial axis, measure the cell fluorescence and morphology statistics, and detect the position of diffraction limited particles.

In [ ]:

```
for ind in range(0,len(experiments)):
    experiment = experiments[ind]
    snapshots_path = snapshots_paths[ind]
    unet_path = unet_paths[ind]

    snap = unet_snapshots(unet_path, snapshots_path, experiment, save_path) # iitialize class

    snap.show_unet_masks(save=True, curate=True) # show the unet masks (not required)

    snap.get_bad_cells() # remove bad cells form the analysis

    snap.run_medial_axis() # draw the medial axis of all single cells

    snap.get_cell_mean_stats() # get the average fluorescence statistics for each single cell

    snap. apply_oned_coordinates() # map fluorescent pixels on the medial axis

    snap_par = particle_positions_snapshots(unet_path, snapshots_path, experiment, save_path) # initialize class

    particle_df = snap_par.getting_the_particles_in_snapshots(log_adaptive_parameters=[3.0, 1000, 95.0, 1.0, 9, -7.0, 0],
                                            min_particle_size=3, max_particle_size=90,
                                            min_particle_aspect_ratio=0.3, post_processing_threshold=90, box_size=7,
                                            metric='raw pixels', operation='sum',
                                            channel='mCherry',gaussian_fit_show =False) # detect the positions of particles
```