Big-Fish Pipeline

Preparation:

1. Download and install big-fish pipeline following the instructions from <https://github.com/fish-quant>
2. Read the documentation of bigfish package from https://big-fish.readthedocs.io/en/stable/
3. Run the example notebook via jupyter notebook on <https://github.com/fish-quant/big-fish-examples>
4. Read the documentation file while reading the codes in the example. Try to understand the codes:D

Modules:

1. Create the folder for A SET of IMAGEs and download all the images
2. Parameter files and output folder preparation
3. Cell and nucleus segmentation (ImageJ)
4. Convert external cell and nucleus mask to cell and nucleus label
5. Spot detection
6. Cellular Analysis (this step need the result of step 3 and 4)
7. Coordinate Analysis
8. Store the data and statistics of interesting items (number of RNA in each cell, etc.) in csv files
9. Draw boxplots and scatter plots for the results (with package seaborn)

Input:

1.A set of qHCR-FISH images with at least 3 channels ---- cell, nucleus and rna. The images have the suffix:.tif. (You may try the images of other suffix. But you need to go through the scripts and make necessary changes)

2.A seed parameter file.

Output:

1.The data and statistics of interesting items (number of RNA in each cell, etc.) in csv files

2.Boxplots and scatter plots for the results

TERMs:

Important or confusing terms are list here to avoid confusion.

1. A SET of IMAGEs:

Each time we process A SET of IMAGEs. We set the same parameters of A SET of IMAGEs

1. Channel:

One image has more than 3 channels, which are different ‘aspects’ of the area in the image. We need to at least have 3 channels: cell, nucleus and rna.

1. Section:

There are equal number of sections in each channel. One section is one cross-section of a slice. The total number of a sections in an image = number of sections in each channel \* channel number.

Procedures:

! Detailed comments are stored in the scripts.

! Please strictly follow the rules for creating folders and naming files that will be stated in the Procedures.

! Be careful about the PATH variables in each script.

Module 1:

1. Create the input and output folders for A SET of IMAGEs

Make sure you are in a “grandparent” folder (FISH etc.) where all the scripts for this pipeline are stored.

Create a parent folder with the name of this set of images (picture\_2 etc.), then add 4 child folders in the parent folder.

The 4 child folders are named: acquisition, analysis, statistics, plot

The path of these 4 folders will appear in the scripts as acquisition\_folder, analysis\_folder, statistics\_folder, plot\_folder respectively.

1. Add all the images in [acquisition] folder

! Please create the folders strictly following the above structure! Otherwise, you will have to change many path variables in the script. Or you will meet tons of bugs.

Module 2:

1. Add the seed parameter file (parameters.txt) to [analysis] folder.

! DOUBLE CHECK to make sure you have changed every parameters you need to change including input and output folder (otherwise, you may mess up the results of this set of images to the sets you previously worked on).

! The number after nuc\_channel (xxx\_channel) is the index of the channel. If nuc is the first channel of the image, then enter “1”. Double check if you have input the correct channels indices in this part!

The filename will be changed automatically when we generate the new parameter file for each image, so don’t worry about the filename in parameters.txt now :D

1. Use parameters\_and\_folder.py to generate parameter files for each plot.

There are 2 functions in parameter\_and\_folder.py:

make\_parameters\_file(acquisition\_folder, initial\_parameters\_file):

Create parameter files for each folder based on the seed parameter file: parameters.txt

The parameters files are named as image name + ‘\_parameters.txt’

! To avoid chaos, no change will be made in the parameters file for each image if they have already existed. If you already create the parameter files for each image but need to change the parameters for some reasons, PLEASE first delete ALL the parameter files except for the seed parameter file. Otherwise,no change will be made in the parameter files for each image. DOUBLE CHECK the content of new seed parameter file. Then modify the seed parameter file: parameters.txt and run parameters\_and\_folder.py again.

make\_analysis\_folder(acquisition\_folder, analysis\_folder):

Create child folders for each image in [analysis] folder. Also create a folder in each child folder with the name of spot detection channel (kcna1.etc). Future analysis related to this spot detection channel will be saved in this folder.

! To avoid chaos, no change will be made in the child folder for each image if they have already existed. If you already create the child folders for each image but need to change the folders ENTIRELY (not just change some files inside it). PLEASE first delete ALL the child folders.

Module 3:

1. Create cell mask and nucleus mask for each image using ImageJ and save them as cell\_mask.tif and nuc\_mask.tif in each child folder for each image.

Steps to create cell\_mask.tif with ImageJ:

1. Go to Image > Adjust > Brightness and Contrast > Auto to see the images more clearly. DO NOT click apply, otherwise you may ruin the projection of the images.
2. Go to Image > Stacks > Z project > Projection type: Max Intensity (you may change the type if needed) to get the projections of each channel.
3. View the cell channel to see where the PV cells are.
4. Use ‘Freehand selections’ in the tool bar and draw the contour of each cell CAREFULLY. If you find some cells with strange shapes, you may go back to the original sections to check their shape.

! Hold Shift to select multiple cells for one time. If you mess up one cell, you have to cancel the selection, and draw the contour for ALL cells in the image again. It takes some time and practice to obtain the contours with high quality. So please be patient. But trust me, contours with high quality will help you A LOT in the following analysis.

1. Right click the area within the cells > Create Mask
2. Save it as cell\_mask.tif to the child folder for this image.
3. Don’t close the cell mask you just made. You will need it in the next step.

Steps to create nuc\_mask.tif with ImageJ:

1. In the projection image

Go to Image > Overlay > Add Image >

Image to add (choose the cell mask image you created just now)

X location, Y location : 0

Opacity: choose a number you are comfortable with (sometimes 30)

> OK

1. Go to the nucleus channel to find the nucleus within the PV cells according to the cell masks we generated before.
2. Use ‘Freehand selections’ in the tool bar and draw the contour of each nucleus CAREFULLY. If you find some nuclei with strange shapes, you may go back to the original sections to check their shape.

! Tricks: If the image quality is not decent, then you may find the following tricks useful:

1. Go back to the original image. You may go to the cell channel, slide the Z bar to find the location of cells in some section, then go to the corresponding nucleus channel to see the nucleus more clearly. Overlaying the cell mask may help you locate the nuclei when nuclei are crowd.
2. Hold Shift when you slide C or Z bars to see different sections once you started make the contour of the nuclei.
3. If the the nucleus channel is too vague for you to distinguish the contour of a certain nucleus, you may draw the contour of the big black hole inside PV cells in cell channel instead. The big black hole is the nucleus. However, since nucleus channel is more precise (at most of the time), so use this method ONLY if nucleus channel is too vague.
4. One cell or One nucleus can have different size on each section. They are balls, so the size of cross-sections are different. Please choose the largest one. The reason is that, if the image quality is good, we use projection image to do segmentation, which will give us the area of the largest cross-sections. We want to keep it consistent!
5. It is a good habit to check the contour of nuclei with the background of cell channel to make sure you are doing good nucleus segmentation!
6. Right click the area within the nuclei > Create Mask
7. Save it as nuc\_mask.tif to the child folder for this image.
8. Save the max projection image (.tif) to the child folder for this image to make it convenient to check the projection if needed.

! The following modules are “scripts intensive”. In bigfish pipeline, each image are represented as an 4-D array (3-D for multiple channels). So it is a good habit to check the data type and format of the input and intermediate data. For example, the shape of the image array, the shape of dataframes. It will help you a lot when you meet some bugs.

! Please first understand the modules and functions in the scripts. You may also refer to the documentation, example and source code provided by bigfish package for more detailed information.

Module 4:

1. Use mask\_to\_label\_folder.py to convert binary mask (cell\_mask.tif and nuc\_mask.tif) to labels (cell\_label.tif and nuc\_label.tif). The labels are saved in both child folder for each image and the the folder with the name of spot detection channel (kcna1, etc.) for future analysis.

PS: Mask can tell you the areas of cell and no-cell. Label can tell you the areas of EACH cell ( cell1 and cell2 have different pixel value) and no-cell.

Module 5:

1. Use pipeline\_spot\_detection\_folder.py to generate spot detection results.

You will find

csv: kcna1\_spots.csv, kcna1\_clusters.csv

png: kcna1\_sd\_result.png

in analysis/child\_folder\_for\_each\_plot/spot\_detection\_channel\_name (analysis/3.1.23 Tsc2\_het\_sham\_L23\_colD\_image1.tif/kcna1 etc.).

The spot detection channel name may change.

1. Please check the result of spot detection with kcna1\_sd\_result.png. If the quality is bad, you may need to go back and adjust the parameters for spot detection (sad).
2. See the documentation of bigfish pipeline if you do not understand the content of each columns in spots.csv and clusters.csv.

Module 6:

Super excited that the results of segmentation and spot detection converge in this module!

1. Use pipeline\_celluar\_analysis\_folder.py to analyze each cell.
2. You will find

csv : kcna1\_summary.csv

png: image name + “cellular\_results” + cell index+ “.png” (kcna1\_3.1.23 Tsc2\_het\_sham\_L23\_colD\_slice2\_image1\_cellular\_results\_cell6.png)

npz: sd\_channel\_name + “results” + cell index + “.npz” (kcna1\_results\_cell\_5.npz)

in analysis/child\_folder\_for\_each\_plot/spot\_detection\_channel\_name (analysis/3.1.23 Tsc2\_het\_sham\_L23\_colD\_image1.tif/kcna1 etc.).

You may not be able to open the npz file which will be used in the next step.

1. Check the pngs to find if anything abnormal happens.
2. Items that we are interested in are stored in kcna1\_summary.csv for further analysis
3. You add or delete some columns in \_summary.csv by modifying the corresponding code in this script.

Module 7:

1. Use pipeline\_celluar\_coordinates\_folder.py to generate coordinates and some statistics of based on the results from cellular analysis.
2. You will find

csv: kcna1\_df\_features\_multicells.csv

in the same folder where spot detection and cellular analysis result are saved.

! The following 2 modules are designed for data with 4 groups (2 types of mice and 2 types of condition) . It is robust with some of the group lost. But if you have more than 4 groups or have the groups with other structure (4 types of mice with one condition, etc.), you may have to modify the code ‘structurally’.

Module 8:

1. Use to\_statistics.py to extract the data and statistics information of each group.
2. You will find csv files in statistics folder and boxplots in plot folder.
3. More information is included in the comments in the script.

Module 9:

1. Use draw\_plot.py to draw statistics plots to see the results and detect outliers.

You may exclude some sample or redo the analysis if something abnormal happens.

1. Go back to the documentation of seaborn some related bugs appear.
2. You will find plots with suffix .svg in the plot folder
3. More information is included in the comments in the script.