Boomerang Catcher

Tomoki Ishibashi
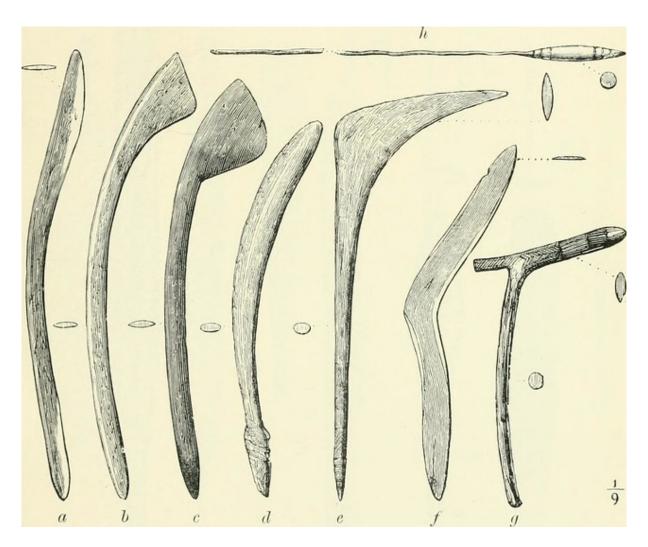
Contents

Boomerang Catcher (BC) is the ImageJ & python program set to analyze FRET in viral infected cells.

1 How to Use

1.1 Image Preparation

BC tries to analyze an image file named as `img.tif`.
BC requires `img.tif` to follow the rules;

1. File name should be `img.tif`
2. `img.tif` should be hyper-stacked

    1. Different images should be stacked into Z-direction
    2. In each Z-stack, two channels (channel 1 and 2) should be set
    3. `img.tif` should contain at least two Z-stacks

3. All images should be taken in same condition

1.2 Making Hyperstack

There is a lecture video (`1_how_to_make_hyperstack.mov`).

1. Gather image files into one directory

    - I recommend to follow the naming rule;

        - `1_1.tif` for image-1, channel-1
        - `1_2.tif` for image-1, channel-2
        - `2_1.tif` for image-2, channel-1
        - …

2. Open ImageJ, and click `File -> Import -> Image Sequence...`
3. Select **the directory** containing the images described in step 1
4. Click `Open`
5. A new window will open, and click `OK`
6. A stacked image will be imported
7. Select `Image -> Hyperstacks -> Stack to Hyperstack...`
8. Select `Order` as xyczt (default)
9. Change `Channels (c):` as 2
10. Change `Slices (z):` as the half number of the total images
11. Click `OK`

12. Select `Image -> Rename...`, change file name as `img.tif` and click `OK`
13. Save the `img.tif` by selecting `File -> Save As... -> Tiff...`
14. Select the directory where you want to save the `img.tif`, and click `Save`

1.3 Run ImageJ Program

1. Open the `img.tif` by dragging and dropping the file into the ImageJ control bar
2. Select `Plugins -> Macros -> Run...`
3. Select `190918_masking_measure.ijm` and click `Open`
4. Program will run and detecting cells
5. You will find windows named as `C1-img.tif`, `C2-img.tif`, `Result of filter2.tif`, and `Results`

   - `C1-img.tif` shows the detected cells in channel 1
   - `C2-img.tif` shows the detected cells in channel 2
   - `Result of filter2.tif` shows the detected area and area-ID
   - `Results` shows the 1st-step data

6. **Save the table file, `Results`, by selecting `File -> Save As...`**
7. Click `Save`
8. **If you need, save the images in your proper directories**

1.4 Run Python Program

2nd-step analysis will be performed by Google Colaboratory. This is the link to briefly understand Google Colaboratory.

In this step, you need to mount Google Colaboratory to your Google Drive. If you have not mounted it yet, follow this instruction.

1. Access to Boomerang Catcher Python Program
2. Click the `Connect` in the upper right
3. Click the `>` mark on the left side of the window
4. New tab will appear, and click `File`
5. By clicking `Upload` or dropping the `Results` to the left side tab
6. After finishing the upload, click the grey triangle mark of the top row
7. After running the first row, click the next grey triangle
8. After running the second row, click `Refresh` in the left side tab, and you will find `out.csv`
9. Click right button of your mouse on the `out.csv`, and click `Download`
10. Save the file your directory
11. In the `out.csv`, you can find `Ch2/Ch1-Ratio` column: this is the final data

2 Analyzing Algorithm

1. Subtracting Background to enhance contrast from channel-2 image

2. Next, enhance contrast until saturating 0.3 % of pixels

3. Apply Gaussian Blur, sigma equals to 4

4. Apply threshold by Minimum method to detect cell body without nucleus (Filter 1)

5. Select the area with 50 - 500 μm^2

6. Thresholded image of step 4 is duplicated

7. Apply threshold by Otsu method to detect cell body including nucleus (Filter 2)

8. Select the area with 50- 800 μm^2

9. By multiplying Filter 1 and Filter 2, and generate nuclear mask (Filter 3)

10. Measuring the average intensity of channel-1 and channel-2 by detecting Filter 1 (data calculating cell body including nucleus) (Data 1)

11. Measuring the average intensity of channel-1 and channel-2 by detecting Filter 3 (data calculating only nucleus) (Data 2)

12. Calculate `((Area of Data 1) * (Mean Intensity of Data 1) - ((Area of Data 2) * (Mean Intensity of Data 2))) / ((Area of Data 1) - (Area of Data 2))` for each channel

13. Calculate `(Mean Intensity of Channel-2) / (Mean Intensity of Channel-1)`