BDWorkshop_Fiji-DividingCells

Learn how to detect dividing cells using Fiji and ImageJ macro

A - Overview

During cell division, the chromosomes contained in the nuclei are compacted before being segregated between the two daughter cells. This phenomenon is easily observed under fluorescence microscopy as the DNA staining (e.g. DAPI) appears brighter in dividing cells.

In this tutorial, we'll use Fiji's basic commands and the ***ImageJ Macro*** language to automatically detect dividing cells in a collection of images. We will first manually execute the procedure and then use the ***macro recorder*** to retrieve the commands needed to assemble a macro.

1. Create mask

First, we will create a mask using an automatic thresholding procedure.

During this process we will keep track of the commands using the ***macro recorder***:

- Plugins > Macros > Record...

Open `image_01.tif` from the `data` directory:

- File > Open...

Duplicate the image:

- Image > Duplicate...

Filter image using a Gaussian kernel:

- Process > Filters > Gaussian Blur...
    - Select ***Sigma*** = 2

Apply threshold:

- Image > Adjust > Threshold...
    - Select ***Otsu*** method
    - Click the ***Apply*** button

We obtain a binary black (0) and white (255) mask.

2. Binary operations

We will now apply some binary operations to improve our mask.

Fill holes in the segemented objects:

- Process > Binary > Fill Holes

Separate touching objects using the ***Watershed*** function:

- Process > Binary > Watershed

3. Objects selection

Using the ***Analyse Particles*** menu we will next select objects based on their size and position:

- Analyse > Analyse Particles...
    - Select ***Size*** = 300-Infinity
    - Select ***Exclude on edges***
    - Select ***Add to Manager***

Our objects are now stored as regions of interests in the ***ROI manager***.

4. Fluorescence intensities measurments

The last step will consist of measuring the mean fluorescence intensity on the original image.

Setup Fiji to measure mean intensities:

- Analyse > Set Measurments...
    - Select only ***Mean gray value***

Select the original image and do the following on the ROI manager:

- Display ROIs by selecting ***Show All*** and ***Labels***
- Make sure that no ROIs are selected by clicking ***Deselect***
- Quantify fluorescence by clicking ***Measure***

This should open a ***Results*** window with fluorescence mean intensity for every ROI.
You can compare values for normal and dividing cells.

B - Automation

We will now use ***ImageJ Macro*** language to automate the analysis and also add some more advanced features.

Exercice 1 : Automate the basic procedure

Open the Fiji ***IDE*** (Integrated Development Environment):

- Plugins > New > Macro

Gather commands from the macro recorder and recapitulate the above procedure.

- The macro should:
    1. Open the image
    2. Create the mask
    3. Perform binary operations
    4. Select objects
    5. Make the measurments

 When writing paths within the macro be sure to use slash `/` and not backslash `\`.
 Add this to the of your code to close all windows that could interfere with the execution.

```
runMacro(".../BDWorkshop_Fiji-DividingCells/CloseAll.ijm");
```

Exercice 2 : Detect dividing cells

As we have seen above, nuclei in dividing cell are brighter due to the compaction of their chromosomes. We will take advantage of this to property to automatically detect dividing cells.

We will use a ***for*** loop combined to an ***if*** statement to check the mean fluorescence intensity for all segmented objects. If the measured intensity is above a given threshold we will draw a rectangle around the object centroid on the original image.

Here are some of the code snippets you will need to perform this task:

- Modify your `Set Measurements...` statement to extract object ***centroid*** coordinates:

```
run("Set Measurements...", "mean centroid redirect=None decimal=3");
```

    - Add ***X*** and ***Y*** columns to the result table

- Get the number of segmented objects by measuring the length of the result table:

```
n = nResults;
```

- Retreive values in the result table:

```
value = getResult(column, idx);
```

    - ***column*** - column name (e.g. "Mean")
    - ***idx*** - row index (start at 0)

- Make a rectangle selection:

```
makeRectangle(x, y, width, height);
```

    - ***x*** and ***y*** - top left pixel coordinates
    - ***width*** and ***height*** - rectangle size (pixels)

- Draw a selection:

```
run("Draw", "slice");
```

- Use an ***if*** statement within a ***for*** loop:

```
for (i = 0; i < nResults ; i++) {
    if (value > threshold) {
    }
}
```

- Clean the original image at the end:

```
run("Remove Overlay");
run("Select None");
```

Exercice 3 : Batch processing

The next step will consists of running the analysis on all images contained in the `data` directory.

Step 1

We will proceed step by step and first enhance the flexibility of our code by managing image names dynamically, using ***variables***, instead of hardcoding them directly into our script.

To understand why this is critical, we will open the second image by simply modifying the opening path.

- Modify image path as follow:

```
open("../BDWorkshop_Fiji-DividingCells/data/image_02.tif");
```

After running the macro we should get the following error : `No window with the title "image_01.tif" found`
This error arises from the fact that deeper in the code we are using a hardcoded command to select the original image window : `selectWindow("image_01.tif");`

To get around this problem, we need to store the name of imported images in a variable that we will later use to refer to the image.

- Retrieve and store the image name in the `image_name` variable:

```
open(".../BDWorkshop_Fiji-DividingCells/data/image_01.tif");
image_name = getTitle();
```

- Later in the code you can replace:

```
selectWindow("image_01.tif");
```

by:

```
selectWindow(image_name);
```

- You can now modify the image path to open the second image:

```
open(".../BDWorkshop_Fiji-DividingCells/data/image_02.tif");
```

Step 2

We will now modify our code to process all images in a single execution. The basic principle involves listing the files contained in the `data` directory and iterating over this list to successively open and process each image. Since, this will imply some new concepts, particularly in terms of file path handling, it would be a good practice to start with a new, empty macro.

Create a new macro from Fiji ***IDE***:

- File > New

Create a ***for*** loop to iterate through the `data` directory content to sucessively open all images.

Here are some of the code snippets you will need to perform this task:

- Define `data` directory path:

```
dir_path = ".../BDWorkshop_Fiji-DividingCells/data/";
```

- List `data` directory files:

```
dir_list = getFileList(dir_path);
```

- Retreive the number of files:

```
nFiles = dir_list.length;
```

- Create the first image path and display it:

```
image_01_path = dir_path + dir_list[0];
print(image_01_path);
```

Step 3

Finally, we will now merge our two macros by inserting the detection procedure within the for loop we just created. By doing so we will ensure that outputs from previous iterations do not interfere with the current analysis.

Here are some of the code snippets you will need to perform this task:

- Clear ROI manager:

```
roiManager("Deselect");
roiManager("Delete");
```

- Clear Results:

```
run("Clear Results");
```

 Indexing variable (e.g. `i`) in nested loops must be unique.

Macro_01_BasicProcedure.ijm

```
runMacro(".../BDWorkshop_Fiji-DividingCells/CloseAll.ijm"); // close all windows

// Open image:
open(".../BDWorkshop_Fiji-DividingCells/data/image_01.tif");

// Segmentation mask:
run("Duplicate...", " ");
run("Gaussian Blur...", "sigma=2");
setAutoThreshold("Otsu dark no-reset");
setOption("BlackBackground", true);
run("Convert to Mask");

// Binary operations:
run("Fill Holes");
run("Watershed");

// Objects selection:
run("Analyze Particles...", "size=300-Infinity exclude add");

// Fluorescence intensities measurements:
run("Set Measurements...", "mean redirect=None decimal=3");
selectWindow("image_01.tif");
roiManager("Show All with labels");
roiManager("Deselect");
roiManager("Measure");
```

Macro_02_DividingCells.ijm

```
runMacro(".../BDWorkshop_Fiji-DividingCells/CloseAll.ijm");

// Open image:
open(".../BDWorkshop_Fiji-DividingCells/data/image_01.tif");

// Create mask:
run("Duplicate...", " ");
run("Gaussian Blur...", "sigma=2");
setAutoThreshold("Otsu dark no-reset");
setOption("BlackBackground", true);
run("Convert to Mask");

// Binary operations:
run("Fill Holes");
run("Watershed");

// Objects selection:
run("Analyze Particles...", "size=300-Infinity exclude add");

// Fluorescence intensities measurements:
run("Set Measurements...", "mean centroid redirect=None decimal=3"); // add centroid
selectWindow("image_01.tif");
roiManager("Show All with labels");
roiManager("Deselect");
roiManager("Measure");

// Detect dividing cells:
for (i = 0; i < nResults; i++) {

 mean = getResult("Mean", i);
 x = getResult("X", i);
 y = getResult("Y", i);

 // Draw rectangle:
 if (mean > 90) {
  makeRectangle(x-50, y-50, 100, 100);
  run("Draw", "slice");

 }

}
```

```
// Clean display image:
run("Remove Overlay");
run("Select None");
```

Macro_03_BatchProcessing_Step1.ijm

```
runMacro(".../BDWorkshop_Fiji-DividingCells/CloseAll.ijm");

// Open image:
open(".../BDWorkshop_Fiji-DividingCells/data/image_01.tif");
image_name = getTitle(); // Store the image name in a variable

// Create mask:
run("Duplicate...", " ");
run("Gaussian Blur...", "sigma=2");
setAutoThreshold("Otsu dark no-reset");
setOption("BlackBackground", true);
run("Convert to Mask");

// Binary operations:
run("Fill Holes");
run("Watershed");

// Objects selection:
run("Analyze Particles...", "size=300-Infinity exclude add");

// Fluorescence intensities measurements:
run("Set Measurements...", "mean centroid redirect=None decimal=3");
selectWindow(image_name); // Use image_name variable instead of the hard-coded name
roiManager("Show All with labels");
roiManager("Deselect");
roiManager("Measure");

// Detect dividing cells:
for (i = 0; i < nResults; i++) {

 mean = getResult("Mean", i);
 x = getResult("X", i);
 y = getResult("Y", i);

 // Draw rectangle:
 if (mean > 90) {
  makeRectangle(x-50, y-50, 100, 100);
  run("Draw", "slice");

 }

}

// Clean display image:
run("Remove Overlay");
run("Select None");
```

Macro_03_BatchProcessing_Step2.ijm

```
// Data directory path and content
dir_path = "C:/Users/bdeha/Projects/BDWorkshop_Fiji-DividingCells/data/";
dir_list = getFileList(dir_path);

for (i = 0; i < dir_list.length; i++) {

 image_path = dir_path + dir_list[i];
 print(image_path); // Display created paths
 open(image_path); // Open images using created paths

}
```

Macro_03_BatchProcessing_Step3.ijm

```
runMacro("C:/Users/bdeha/Projects/BDWorkshop_Fiji-DividingCells/CloseAll.ijm");

// Define directory path and list content
dir_path = "C:/Users/bdeha/Projects/BDWorkshop_Fiji-DividingCells/data/";
dir_list = getFileList(dir_path);

// Process all images
for (i = 0; i < dir_list.length; i++) {
```

```
 // Open image:
 open(dir_path + dir_list[i]);
 image_name = getTitle();

 // Create mask:
 run("Duplicate...", " ");
 run("Gaussian Blur...", "sigma=2");
 setAutoThreshold("Otsu dark no-reset");
 setOption("BlackBackground", true);
 run("Convert to Mask");

 // Binary operations:
 run("Fill Holes");
 run("Watershed");

 // Objects selection:
 run("Analyze Particles...", "size=300-Infinity exclude add");

 // Fluorescence intensities measurements:
 run("Set Measurements...", "mean centroid redirect=None decimal=3");
 selectWindow(image_name);
 roiManager("Show All with labels");
 roiManager("Deselect");
 roiManager("Measure");

 // Detect dividing cells:
 for (j = 0; j < nResults; j++) {

  mean = getResult("Mean", j);
  x = getResult("X", j);
  y = getResult("Y", j);

  // Draw rectangle:
  if (mean > 90) {
   makeRectangle(x-50, y-50, 100, 100);
   run("Draw", "slice");

  }

 }

 // Clean display image:
 run("Remove Overlay");
 run("Select None");

 // Clear ROI manager and Results
 roiManager("Deselect");
 roiManager("Delete");
 run("Clear Results");

}
```