

# Introduction to ImageJ

2017-04-10

Xavier Draye and Guillaume Lobet

# Macros in ImageJ

# What are macros?

Simply put, an ImageJ macro is a **succession of existing ImageJ commands**.

Once you have defined exactly the type of analysis you would like to perform on your images, it is worth creating a macro with the different steps.

# Advantage 1: Track the steps in your analysis

First, it help you keep track of all the steps you used in your analysis. If you need to do the same analysis a year later, or ou want to check which algorithm you used when writing the paper, macro are perfect for that.

They are a nice way to **keep track of eveything manipulation and analysis** you did on your images.

#openscience #reproducibility

## Advantage 2: Batch processing

1. Define your pipeline on several test images. If possible pick them such as they represent the variability in your dataset.
2. Record the step to create a macro
3. Apply it on the whole dataset.

It will save you a lot of time and avoid errors.

# Record your macro

ImageJ has a very handy utility to record command and create a macro out of them.

1. Open the recorder

```
> Plugins > Macro > Record...
```

2. Start your different operations

3. Click `Create` and save it with the `.ijm` file extension

4. Run your macro

# Use your macro in batch

The main advantage of the macros in ImageJ is that they allow you to process hundreds of images at once.

ImageJ has an utility to launch a specific macro on a whole folder

```
> Process > Batch > Macro...
```

# ImageJ macro language is a simplified version of Java

The ImageJ macro language is a **modified version of Java**. Therefore, they share some similarities.

# ImageJ macro $\approx$ Java - 1

Commented lines start with `//`

```
// This is a commented line
```

Or you can comment several lines using `/***/`

```
/*  
* This is several  
* commented lines  
*/
```

# ImageJ macro $\approx$ Java - 2

Each line need to finish with an ;

```
// Each line need to end with ";"  
run("Convert to Mask");
```

But not for loops start and end

```
// Not for loop start  
for(k = 0 ;k < 10 ; k++){  
    // But do not forget inside the loop  
    print(k);  
} // Or end
```

# ImageJ macro $\approx$ Java - 3

The first element in a vector as the index 0

```
dir=getDirectory("Where are your images");
list=getFileList(dir);
num=list.length;

for(k = 0 ;k < num ; k++){
    open(dir+list[k]);
    close();
}
```

# Going further with the macros

Macros are extremely versatile and can be used to do a lot of different tasks

Let's build a more complicated one.

More about macros:

<https://imagej.nih.gov/ij/developer/macro/macros.html>

# Advanced macro - 1

Setup the initial parameters

```
// Initial parameters

// We do not want ImageJ to open the images
setBatchMode(true);

// We define the measurements we want to make
run("Set Measurements...", "area centroid center
redirect=None decimal=2");
```

## Advanced macro - 2

Open user defined folder

```
// Define the directories for the analysis  
dir = getDirectory("Where are your raw images");
```

This will trigger a pop-window and ask the user to choose a folder.

Then, get the list of file inside the folder

```
// Get the file list  
list = getFileList(dir);  
num = list.length;
```

# Advanced macro - 3

Navigate the list of files

```
// Loop over the file list to analyse all the images
for(k = 0 ; k < num ; k++){

    // Get the file and open it
    t = dir + list[k];
    open(t);

    // Get the file name
    ti=getTitle();

    [...]
}
```

# Advanced macro - 4

Threshold the image

```
// Threshold the images  
  
setAutoThreshold("Default dark");  
run("Convert to Mask");
```

And analyse the particules

```
// Analyse the particules  
  
run("Analyze Particles...", "size=50-Infinity  
circularity=0.00-1.00 show=Masks add  
display clear exclude");
```

# Advanced macro - 5

Save the resulting image

```
// Close the old image
selectWindow(ti);
close();

// Save the new one
selectWindow("Mask of " + ti);
saveAs("Tiff", dir+"new-", list[k]);
close();
```

# Advanced macro

```
// Initial parameters
setBatchMode(true);
run("Set Measurements...", "area centroid center redirect:

// Define the directories for the analysis
dir = getDirectory("Where are your raw images");

// Get the file list
list = getFileList(dir);
num = list.length;

for(k = 0 ; k < num ; k++){

    // Get the file and open it
    t = dir + list[k];
    print(list[k]+" analysis started");
    open(t);

    // Get the file name
```

```

// Get the file name
ti=getTitle();

setAutoThreshold("Default dark");
run("Convert to Mask");

run("Analyze Particles...", "size=50-Infinity circ

// Close the old image
selectWindow(ti);
close();

// Save the new one
selectWindow("Mask of " + ti);
saveAs("Tiff", dir+"new-", list[k]);
close();
}
print("Analysis done on "+num+ " images");

```