Automating and Extending FIJI with Macros

Aaron Taylor, PhD
Managing Director
BRCF Microscopy Core
microscopy@umich.edu
Purpose:

- Give a broad overview of what the IJ macro language can do.
- Get some hands-on coding experience with real examples.

What is a macro language?

A macro language is a simple programming syntax designed to automate an application’s existing functionalities.
Why Write Macros?

1. Automate your mouse clicking

   Imagine if you have to perform this series of functions:
   - Process -> Contrast Enhancement
   - Process -> Gaussian Blur
   - Process -> Threshold
   - Analyze -> Count Objects
   - Log data ...

   And you have 50 folders of images!....

2. Generate new capabilities not directly available in the menu by re-combining existing commands in new ways.
How is a Macro Different From a Plugin?

A macro is code written in a simplified, high-level language that is only understood by the application that the macro runs within, FIJI in this case. The code only has access to functions supplied by the application.

A plugin is code written in an established, lower-level language that extends an application. The code can do anything and can access any function written in the language. FIJI plugins are written in Java, since FIJI is written in Java.

This workshop only covers the basics of macros, which are sufficient for 99% of end user needs.
Getting Started

Macros are accessed through the Plugins -> Macros menu.

- **Install...** Installs a macro, so it will appear as a menu option here...
- **Run...** Runs selected macro from a file
- **Edit...** Editor to write macros – very useful!!
- **Startup Macros...**
- **Interactive Interpreter...** Ctrl+J
- **Record...** Records corresponding macro code when you click – very useful!!
- **Pencil Tool Options...**
- **Paintbrush Tool Options...**
- **Flood Fill Tool Options...**
- **Set Drawing Color...**
- **About Startup Macros...**
- **Save As JPEG... [j]**
- **Save Inverted FITS**

By convention, macros are stored in the \macros folder of the FIJI installation, but this is not necessary.
The IJ macro language is very well documented! A complete description of the syntax can be found here:


Let's look closely at Variables, Operators, Flow Control, and Functions.

The IJ macro language also supplies a variety of built-in functions that are listed here, along with examples:


(The list is long but with experience you will start to appreciate what is available)
Built-in Functions

Highlights of what the built-in functions can do...

- Array functions
- Dialog boxes
- File handling
  - `getDirectory()`
  - `open()`, `save()`, `close()`
- Curve fitting
- Working with images
  - Get and set functions
- Overlays (graphics)
- Plot functions
- ROIs
- `run()` – Find IJ functions using Plugins -> Macros -> Record
- Stacks

*Hint: To find a certain function, search for related words that might be in its description*

*Caution: When using functions, the name case matters*
Many, many macros are already available!

Reading / modifying existing macros is a good way to quickly learn the macro language. Don’t re-invent the wheel!

There is a large collection of macros on the IJ website....
http://imagej.nih.gov/ij/macros/

The macro functions list is also cross-referenced with examples

Many more macros can be found in publications and on the web.
How to Write the Code

**Option 1:** Use the built-in macro code editor (recommended):

Plugins -> New -> Macro

Choose “IJ1 Macro” and your code will be color coded.

**Option 2:** Use any other code editor (e.g. Notepad++)
The Basics of Programming...

1. Open Macro_Examples.txt into a text editor
2. Open a new macro (Plugins->New->Macro)
3. Copy a code block of code into the macro editor
4. Let’s go through it together...

Please ask questions!
It is important to understand all of this!
Using the Command Recorder

Use the Command Recorder to get code corresponding to where you click...

Plugins -> Macros -> Record...

Let’s record these commands....

Recorder

Recorded from GUI clicks

Editor

```
newImage("noise", "8-bit Black", 300, 300, 1);
run("Salt and Pepper");
run("Gaussian Blur...", "sigma=2");
setAutoThreshold("Default dark");
//run("Threshold...");
setThreshold(8, 255);
run("Convert to Mask");
```
Collecting User Input Programmmatically

Let’s modify the code to take user input...

```plaintext
img_w = getUserNumber("width?", 300);
img_h = getUserNumber("height?", 300);
newImage("Test", "8-bit Black", img_w, img_h, 1);
run("Salt and Pepper");
sgm = getUserNumber("Sigma?", 2);
run("Gaussian Blur...", sgm);
setAutoThreshold("Default dark");
//run("Threshold...");
setThreshold(8, 255);
run("Convert to Mask");
```

These parameters are now variables
Use of For Loops to Process a Collection of Images

Very often the goal is the process all of the files in a folder. This code will do it...

```java
//User Chooses target folder...
dir = getDirectory("Choose Image Folder");
list = getFileList(dir);

//Loops through all files in the directory and opens *.nd2 file...
for (i=0; i<list.length; i++) {
    path = dir+list[i];
    if (endsWith(path, ".nd2")) {
        //Only opens nd2
        open(path);
        convert(); // A function that holds your processing code
    }
}

close("*"); //close current files before next loop
```

Tip: For complex tasks, it is often helpful to break the code up into discrete chunks.

RUN Loop_Files.ijm on files in Neuron_Slices Folder
Use of For Loops to Process a Collection of Images

This code will process all the slices in a stack...

```
3 //nSlices is special variable that holds # slices
4 for (i=1; i<=nSlices; i++) { // Note slices are indexed starting a 1
5     slicenum = i;
6     setSlice(slicenum);
7     getStatistics(area, mean);
8     print("Slice Number: "+slicenum+"; Avg Int:"+mean);
9 }
```

RUN Loop_Stack.ijm on Neuron_Vesicles.nd2
Real World Example #1

The images in the Islet_Stacks folder show nuclei in blue and islet cells in red. (Open an image) We want to count all the nuclei within the islets (don’t care about green channel).

Open MultiChannel_Process.ijm. Let’s read through it...

RUN MultiChannel_Process.ijm on the files in Islet_Stacks folder

TIP: Always save intermediate results and visually check them to be sure the code is accurate and doing what you intended it to do.
Real World Example #2

Let’s say you want to partition some region of an image into rectangles and then take measurements within each rectangle.

Open Tile_ROIs.iwm. Let’s read through it...

RUN Tile_ROIs.iwm on any image.

Can now use the Measure button of the ROI manager to take measurements. (Could use further programming to automate the measuring).
How to Get Help After Today

1. An excellent and comprehensive tutorial by Kota Miura can be found here....
   https://zenodo.org/record/30267#.VgG8P99Viko

2. Google

3. microscopy@umich.edu

Getting good at programming takes lots of practice!
   Good Luck!
Thank you!