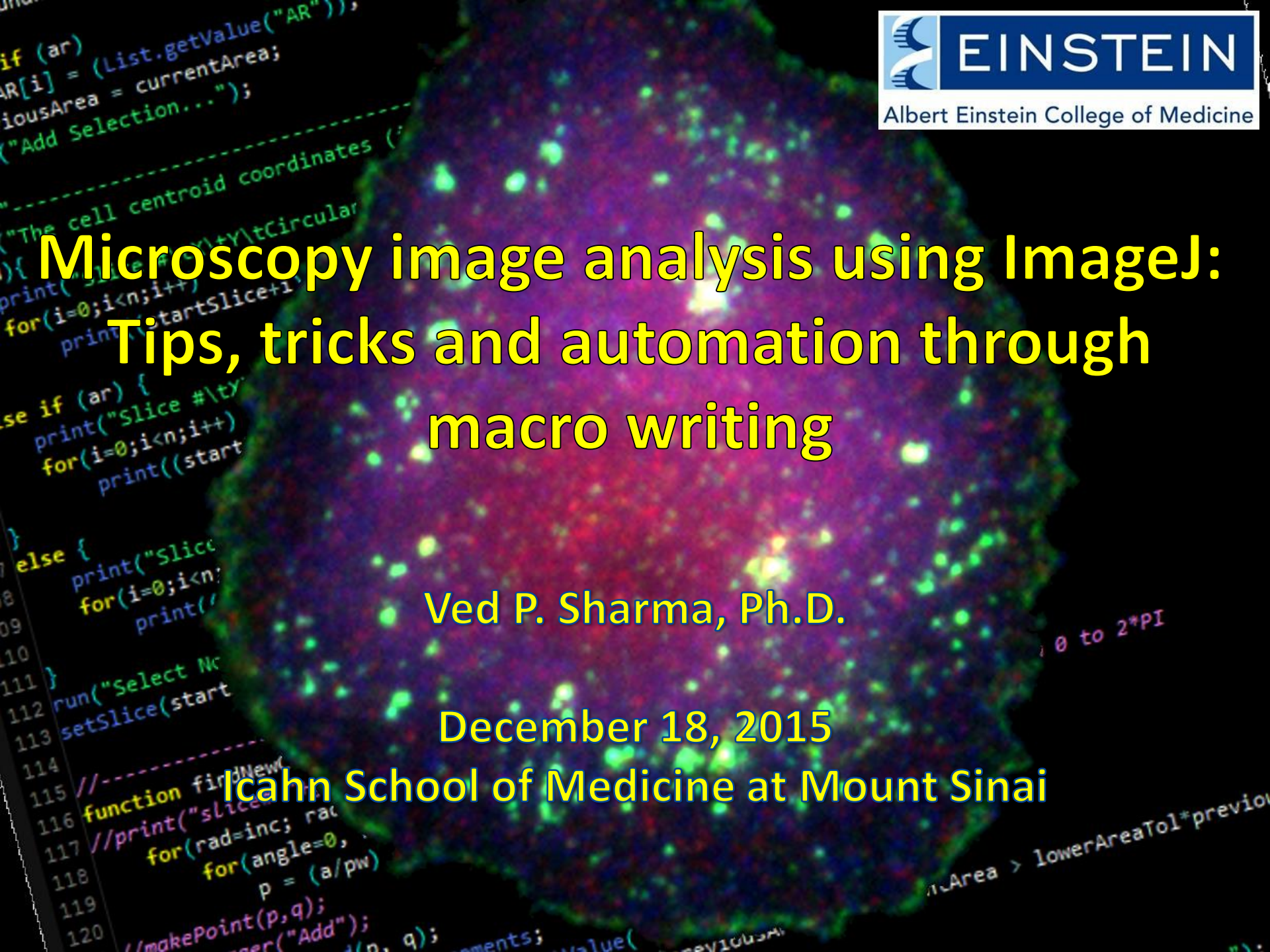


Microscopy image analysis using ImageJ: Tips, tricks and automation through macro writing

Ved P. Sharma, Ph.D.

Albert Einstein College of Medicine

December 18, 2015

Icahn School of Medicine at Mount Sinai

Outline

Navigating through ImageJ commands

Building blocks of macro writing
- for loop and if/else statement
- Variables and Operators

Macro language

Macro examples
1. Split channels, color, merge and save
2. Extract Best Focus Slices
3. Pipette assay macro
4. Outline leading edge segment
5. Cell Tracking

Navigating ImageJ commands – tips and tricks

Getting started with a few commands…

1. Make Substack
2. Synchronize Windows
3. Wand tool
4. Measurements and List.setMeasurements

Navigating ImageJ commands – tips and tricks

Accessibility

1. Keyboard Shortcuts (e.g. Ctrl+Shift+C)
2. Custom Shortcuts
3. Control Panel
4. Action Bar link
5. Startupmacros.txt and right click options
6. Command Finder (Ctrl + L)

Why writing macros is the way to go for Image analysis

1. To repeat a set of ImageJ commands, e.g. on all the images in a folder
   - tiring and boring!!

2. Consistency and precision in analysis

3. Speed

4. Fun!!

Example:

split a 2 channel time-lapse movie into 2 stacks, color 2nd channel green, merge two channels and save

Example macro 1

split file into 2 channels, color 2nd channel, merge and save

```
// This macro splits a delta vision .dv file into 2 channels, colors the 2nd channel green,
// merges both the channels and saves the mergred file in the user-specified directory.

Dir = "d:\\Users\\xx\\2015.12.18 Image analysis seminar at Mount Sinai\\data1\\";

filenames = getFileList(Dir);

pattern = ".*R3D.dv";

for(i = 0; i< filenames.length; i++) {
        if(matches(filenames[i], pattern)) {
                open(Dir+filenames[i]);
                n = nSlices;
                title = replace(getTitle, "R3D.dv", "merge");
                run("Make Substack...", "delete slices=1-"+n+"-2");
                rename("phase");
                setMinAndMax(0, 2000);
                run("Put Behind [tab]");
                rename("green");
                setMinAndMax(300, 3000);
                run("Merge Channels...", "c2=green c4=phase");
                saveAs("Tiff", Dir+title);
                close();
        }
}
```

Demo

Getting started with macro writing…

Macro recorder

Plugins > Macros > Record…

- Create button will make an .ijm file
- If you write your macro in a .txt file, include an underscore in the file name e.g. image_analysis.txt
- Save Macro file under plugins folder in ImageJ

Limitations

Macro Recorder does not record everything
e.g. wait() command, process whole folder

If/else and *for* loops

if and if/else statement are used for checking a condition
e.g. checking if the file name ends with ".tif"

```
if (condition is true) {
    do job;
}
```

```
if (condition is true) {
    do job1;
}
else {
    do job2;
}
```

for loop is used for repeating a set of commands "n" number of times
e.g. repeat certain commands on all the images in a folder

```
for (starting value; condition; increment) {
                do something;
}
```

```
for (i=0; i<10; i++) {
    print(i);
}
```

```
for (i=0; i<10; i++) {
    if(i%2 == 0) {
        print(i);
    }
}
```

Variables

Macro language has three types of variables:

1. Number
2. String
3. Array

Boolean variable (true/false) is represented as numbers (1/0)

Variable names are case-sensitive. "Name" and "name" are different variables.

Practice

```
a = 1.23;
print(a);
b = "a string";
print(b);
c = newArray(10, 20, 50);
for (i=0; i<c.length; i++)
   print(c[i]);
```

Operators

Operators	Description
++	Increment
--	Decrement
=	Assignment
==, !=	Equal, not equal
&&	boolean AND
||	boolean OR
+	Addition, or joining strings
-	Subtraction
*, /, %	Multiplication, division, remainder

```
i++ means i=i+1
i=3;
i++;
print(i);

b=10/2;
b==5;
c = (b==5);
print(c);

if(a==5 && b>3) {
  do something
}
```

Example macro 1

split file into 2 channels, color 2nd channel, merge and save

```
// This macro splits a delta vision .dv file into 2 channels, colors the 2nd channel green,
// merges both the channels and saves the mergred file in the user-specified directory.

Dir = "d:\\Users\\xx\\2015.12.18 Image analysis seminar at Mount Sinai\\data1\\";

filenames = getFileList(Dir);

pattern = ".*R3D.dv";

for(i = 0; i< filenames.length; i++) {
        if(matches(filenames[i], pattern)) {
                open(Dir+filenames[i]);
                n = nSlices;
                title = replace(getTitle, "R3D.dv", "merge");
                run("Make Substack...", "delete slices=1-"+n+"-2");
                rename("phase");
                setMinAndMax(0, 2000);
                run("Put Behind [tab]");
                rename("green");
                setMinAndMax(300, 3000);
                run("Merge Channels...", "c2=green c4=phase");
                saveAs("Tiff", Dir+title);
                close();
        }
}
```

read all the filenames into an array:
filenames[0] : 1st file
filenames[1] : 2nd file
....

these are the files we want to process

Built-in functions

Highlighted in yellow!

```
// This macro splits a delta vision .dv file into 2 channels, colors the 2nd channel green,
// merges both the channels and saves the mergred file in the user-specified directory.

Dir = "d:\\Users\\vsharma1\\Condeelis Lab\\Meetings and Presentations\\2015.12.xx Image analysis seminar at Mount Sinai\\data1\\";

filenames = getFileList(Dir);

pattern = ".*R3D.dv";

for(i = 0; i< filenames.length; i++) {
		if(matches(filenames[i], pattern)) {
				open(Dir+filenames[i]);
				n = nSlices;
				title = replace(getTitle, "R3D.dv", "merge");
				run("Make Substack...", "delete slices=1-"+n+"-2");
				rename("phase");
				setMinAndMax(0, 2000);
				run("Put Behind [tab]");
				rename("green");
				setMinAndMax(300, 3000);
				run("Merge Channels...", "c2=green c4=phase");
				saveAs("Tiff", Dir+title);
				close();
		}
}
```

Built-in functions

nImages, getTitle, getWidth, getHeight
nSlices, getSliceNumber(), setSlice(n)
getStatistics(area, mean, min, max, std, histogram)
getDirectory() - Displays a "choose directory" dialog and returns the selected directory

endsWith(string, suffix) - Returns true if string ends with suffix.
matches(string, regex) - Returns true if string matches the specified regular expression.

getFileList(directory) - Returns an array containing the names of the files in the specified directory path.

List.setMeasurements - Measures the current image or selection. All parameters listed in the Analyze>Set Measurements dialog box are measured. Use List.getValue() to retrieve the values.

roiManager("count"); roiManager("select", index)

print()
showMessage("message")
exit, waitForUser

and many more ... link

Macro #2: Extract_best focus slices

Goal: A folder full of single channel z-stacks, extract best focus slice for each z-stack and save it in a folder

Macro contains: if statement, for loop, function, local vs global variables, setBatchMode

Idea: As the image comes in focus, its edges become sharp

Process > Find Edges

Demo:

Batch Processing…

Process > Batch > Macro…

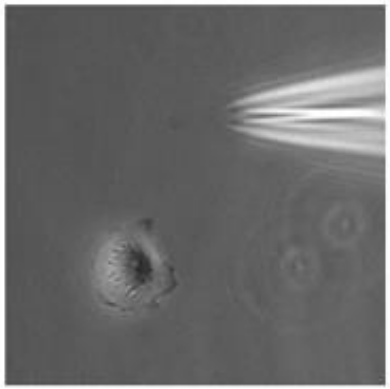
Example:
Scale images 1.5 times
Apply Red LUT
Save as jpeg

Macro code

```
scale=0.5;
w=getWidth*scale; h=getHeight*scale;
run("Size...", "width=w height=h interpolation=Bilinear");
run("Red");
```

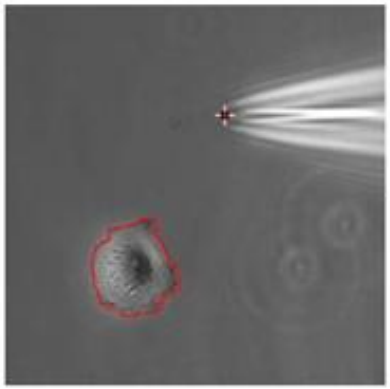
Macro #3: Pipette assay macro

Image of a cell and pipette



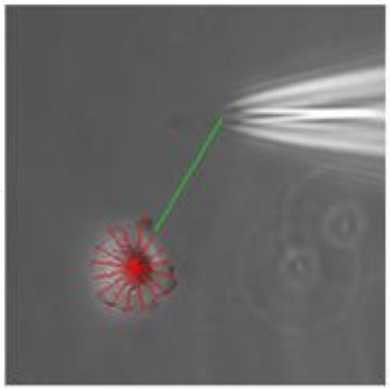
Run macro

User marks the pipette tip location and draws cell outline



Result

16 radial line (22.5° apart) are shown in red



+

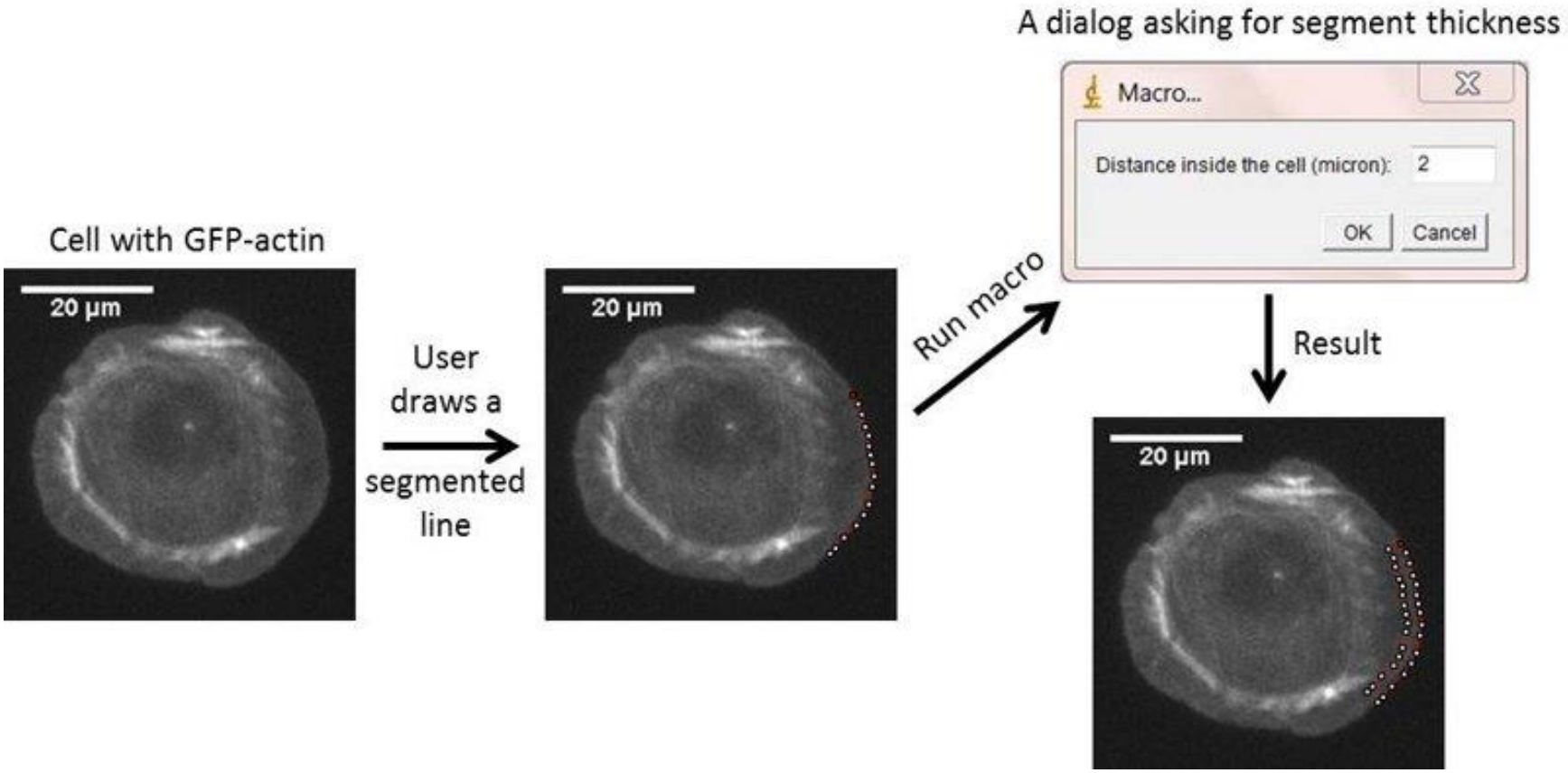
length of 16 radial lines

User input through a Dialog box

Purpose: to pass inputs (text, numbers, choices, boolean etc.) to a macro

Demo: Dialog_demo macro

Macro #4: Outline leading edge segment



Macro #5: cell_centroid tracking

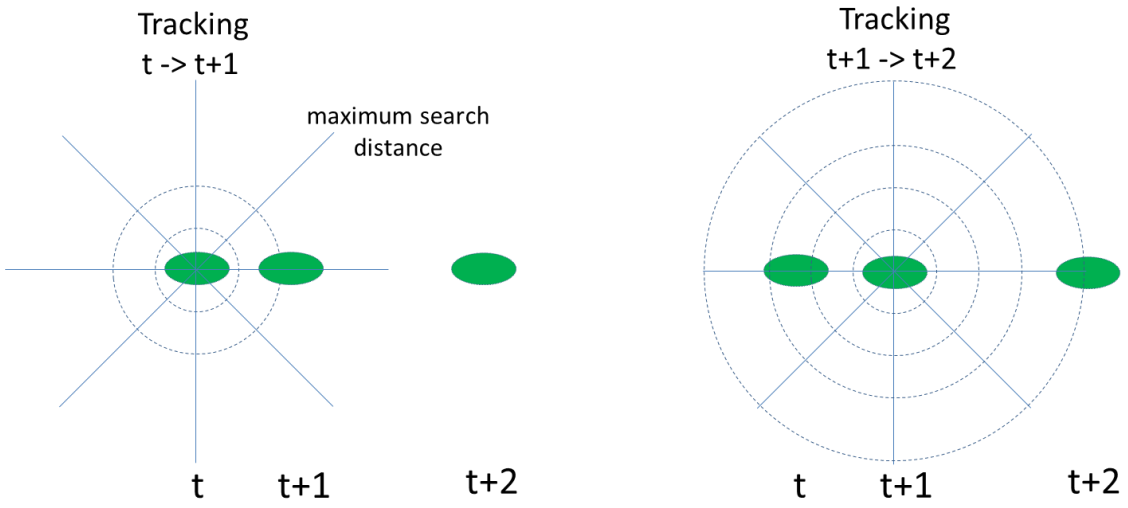
Goal: Track cells in a time-lapse movie

Macro contains: functions, local vs global variables, if/else and for loop, array, Dialog (User input), print tab-delimited values for exporting to Excel

Ideas:

- Use Wand to highlight the cell and find the cell centroid coordinate
- Draw wand at the cell centroid and see if the cell is present in the next slice
- If cell not found, search in the circular fashion around the cell until max search radius

Demo:



Interrupting a Macro: *waitForUser*

Usually for user input, e.g.

1. Adjusting B/C before saving the file

2. Drawing an ROI before running measure

Troubleshooting/Debugging a macro

```
wait(1000);

waitForUser;

print();
```

Demo: go through debugging steps for macro 1

Resources

1. ImageJ User Guide
2. ImageJ mailing list
3. Built-in Macro Functions

Acknowledgements

Wayne Rasband and the ImageJ community

Albert Einstein College of Medicine

John Condeelis
David Entenberg
Analytical Imaging Facility
Gruss-Lipper Biophotonics Center
People from the Condeelis, Segall, Cox and Hodgson labs

Contact: ved.sharma@einstein.yu.edu