Writing Macros with Fiji

http://imagej.net/Presentations

Why are Macros useful?

Reproducible science

- document your work
- automate your analysis
- share with the world
- identify plugins of interest

http://imagej.net/Macros

Recording Macros (1/3)

Exercise: Record a Macro

Many ways to start recording:

- use the Command Finder! *(Ctrl+L)*
- click the Dev icon, then *Record*...
- *Plugins>Macros>Record*...

http://imagej.net/Macros

Recording Macros (2/3)

Exercise: Record a Macro

Suggested workflow:

1. open the Blobs sample image
2. apply a threshold
3. create a Mask
4. dilate
5. invert
6. watershed
7. analyze particles

Tip: when in doubt, use the
Command Finder!
(ctrl + L)

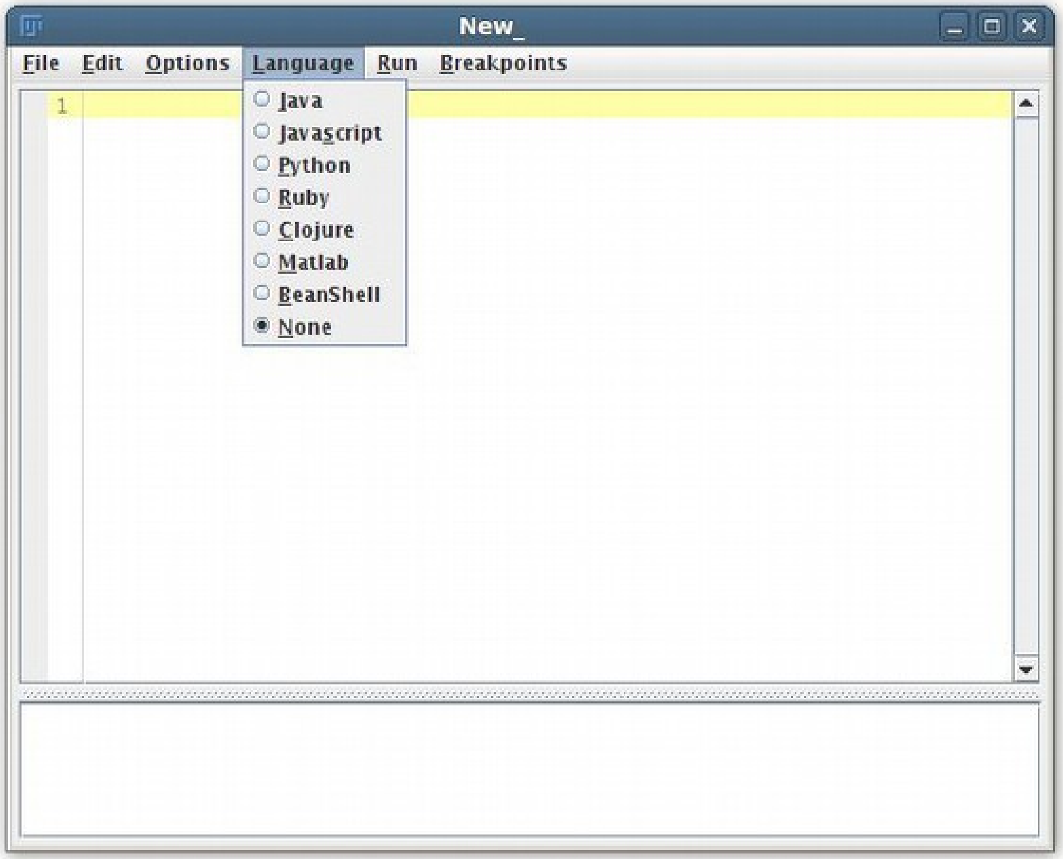
http://imagej.net/Macros

Recording Macros (3/3)

Make your macro better:

- Batch mode
- Use Image IDs
- Store in *plugins/* (with underscores, to tell ImageJ to make a menu item)

Script Editor: beyond "Record"



- Comments
- Variables
- Functions
- String manipulation
- Conditionals
- Loops

http://imagej.net/Script_Editor

Macros: comments

```
// Comments allow you to put human-readable thoughts
// into your code.

// The goal of this “macro” is simply to teach you about
comments!

// Comments help you to remember why you did something:
// Set the value to “2” because my boss said so!
value = 2; // Comments can be added to any line!

// Code can be disabled by commenting it out:
// x = y * 2;
```

http://imagej.net/Macros

Macros: variables (1/2)

```
intensity = 255;

a = exp(x * sin(y)) + atan(x * y – a);

title = “Hello, World!”;

text = “title”;

text = title;
```

http://imagej.net/Macros

Macros: variables (2/2)

```
// after this, y will have the same value as x
y = x;

// now, x will be assigned a new value, but y will stay the same
x = y * y – 2 * y + 3;

// the variable is assigned after the expression is evaluated
intensity = intensity * 2;
```

http://imagej.net/Macros

Macros: functions

```
print(“Hello, world!”);

// functions can return values
number = getNumber(“Type in a number!”);

// the “run” function is the most important one
run(“Duplicate...”, “title=New”);

run(“Duplicate...”, “title=[with spaces]”);

// Try Tools>Help on Macro Functions...
// then select a function name, such as “print” and try again
```

http://imagej.net/Macros

Macros: strings

```
number = 1;

// you can concatenate strings, and strings and numbers
text = “The number is “ + number;

// what happens when we run this?
run(“My plugin”, “does_not_work=number”);

// what's different with this line?
run(“My plugin”, “this_works=” + number);
```

http://imagej.net/Macros

Macros: conditionals

```
if (getBoolean("Is Curtis going too fast?")) {
        hint = "Tell him!";
} else {
        hint = "Try to modify the code, play with it...";
}

showMessage(hint);
```

http://imagej.net/Macros

Macros: loops

```
for (i = 1; i <= 10; i++) {
        print(“Counter: ” + i);
}

while (getBoolean(“Are you sick of my questions yet?”)) {
        print(“You know, I really have all day to keep asking...”);
}
```

http://imagej.net/Macros

Macros: tying it together

```
// this example makes a stack of blurred versions of the
// current slicewith a range of radii.

radius = getNumber(“Maximal radius?”);

title = “Blurred stack of “ + getTitle();
run(“Duplicate...”, “title=[” + title + “]”);
run(“Select All”);
run(“Copy”);
for (i = 1; i <= radius; i++) {
        run(“Add Slice”);
        run(“Paste”);
        run(“Gaussian Blur...”, “radius=” + radius);
}
```

Further reading

Help from the community—ImageJ mailing list! ~2000 members:

http://imagej.net/Help

Scripting guide:

http://imagej.net/Scripting

Additional workshops and presentations:

http://imagej.net/Presentations