



Introduction to ImageJ/Fiji for Image Analysis and Macros Scripting

1. Introduction to FIJI
2. Image J Macro
3. Practice A Illumination Correction
4. Practice B Object Tracking

<https://go.epfl.ch/imagej>



Daniel Sage
Biomedical Imaging Group



ImageJ and Fiji



ImageJ 1.x

NIH Vanilla



W. Rasband

Retired NIH



General-purpose image-processing software package

- Open-source, public domain
- Handling multidimensional images (science)
- Including basic operations on images
- GUI (old-style)
- Extension up to 5D images
- Open architecture → Many plugins
- Scripting in macro or Java
- Macro recorder



ImageJ 2

- New data structure (large image)
- Same GUI
- Integration in scijava

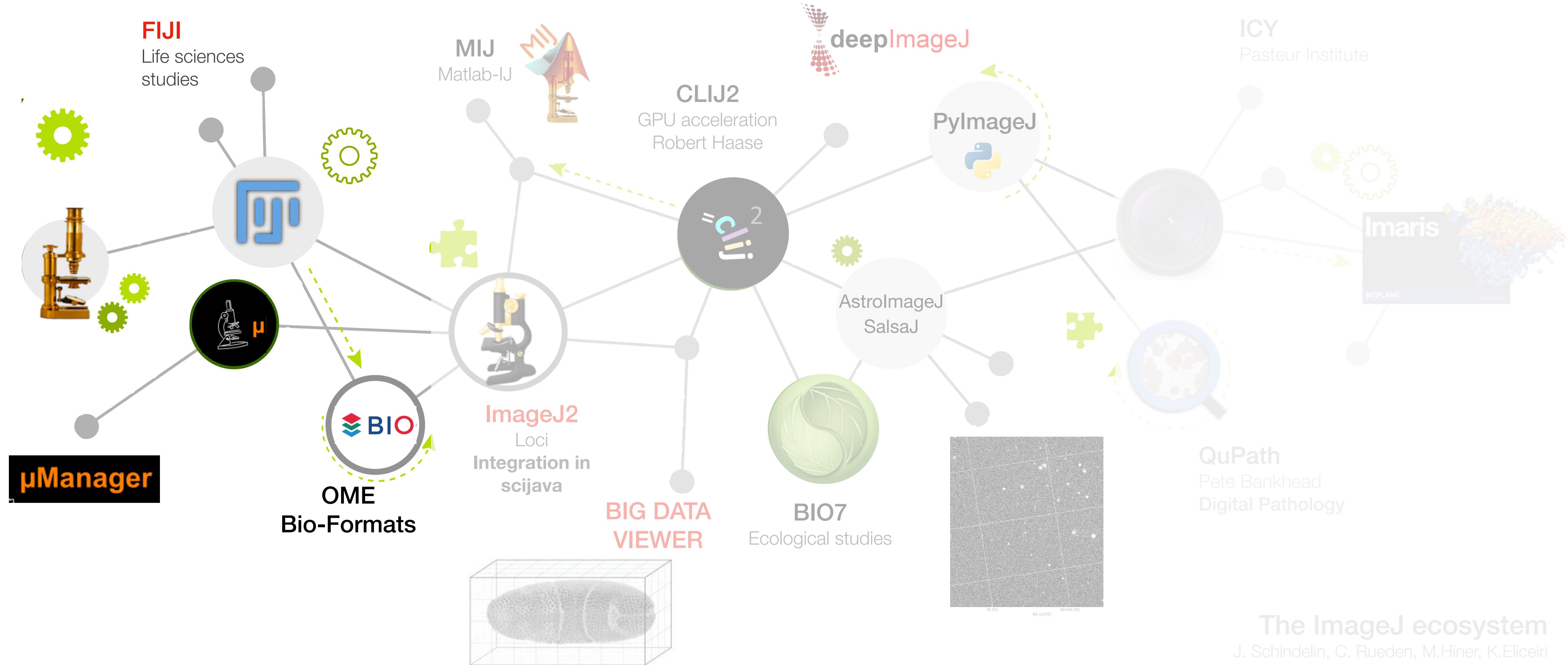


A "batteries-included" distribution of [ImageJ](#)

- + Many useful plugins
- + Access to advanced research tools
- + Manager of plugins and dependencies
- + Better scripting tools (more languages)



The ImageJ Ecosystem



The ImageJ ecosystem

J. Schindelin, C. Rueden, M. Hiner, K. Eliceiri
Mol Reprod Dev, 2015



Handling Images FIJI

Setting-up Fiji

- Install fiji.sc
- Memory
- Benchmark
- Update
- Help → Search
- Install plugins (LabKit)

Open and save images

- File → Open
File → Import → Import Sequence
Plugins → Bio-Formats → Importer
File → Save

Exploring images

Image → Adjust → B&C (hrct)

Pixel Value, Zoom (lake)

Set Scale

Image → Color → Split

File format (sky)

Plot profile, surface

Histogram

Image nD (mitosis)

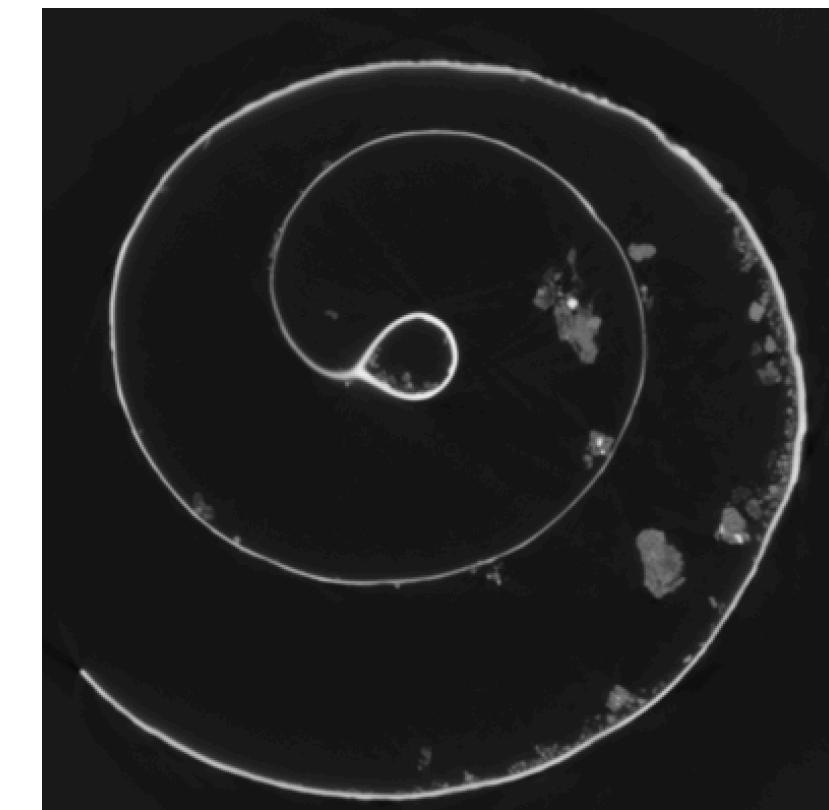
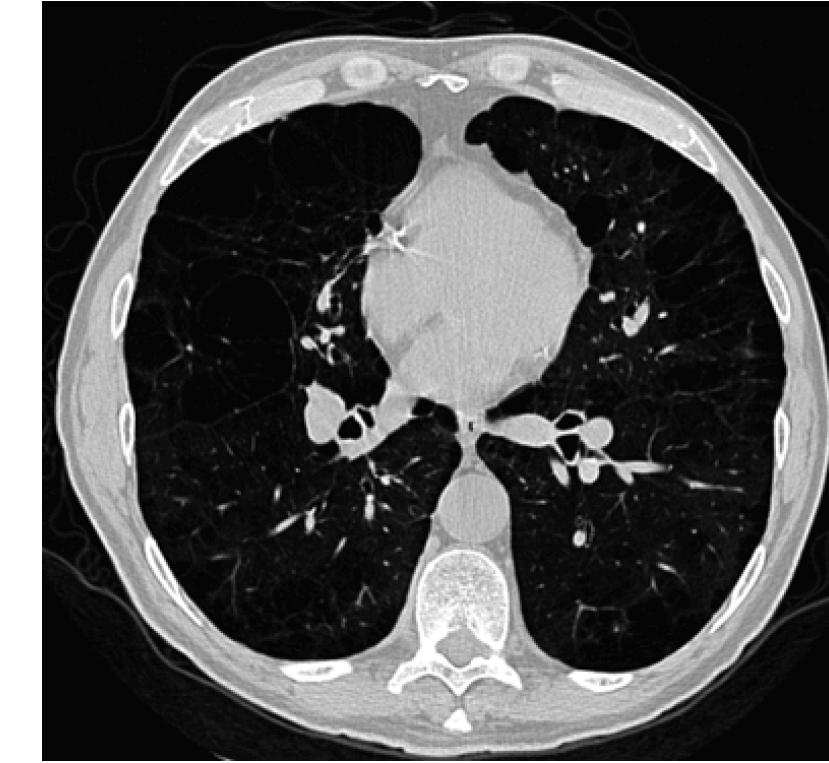
Image → Stack → Hyperstack

Crop, subset

Volume viewer (escargot)

Image → Stack → Reslice

Image → Stack → Z-Projection





Basic Image Analysis FIJI

Processing

Image Calculator (street)

Background estimation (mouse)

Filtering, median, Gaussian (worm-0000)

Remove background (worm-0000)

Pipeline of Image Analysis (blob)

Record macro

Image → Histogram (live)

Process → Filter → Median / Gaussian

Image → Adjust → Threshold (blob)

Process → Binary → Fill Hole

Analyze → Analyze Particle (outline)

Process → Binary → Watershed

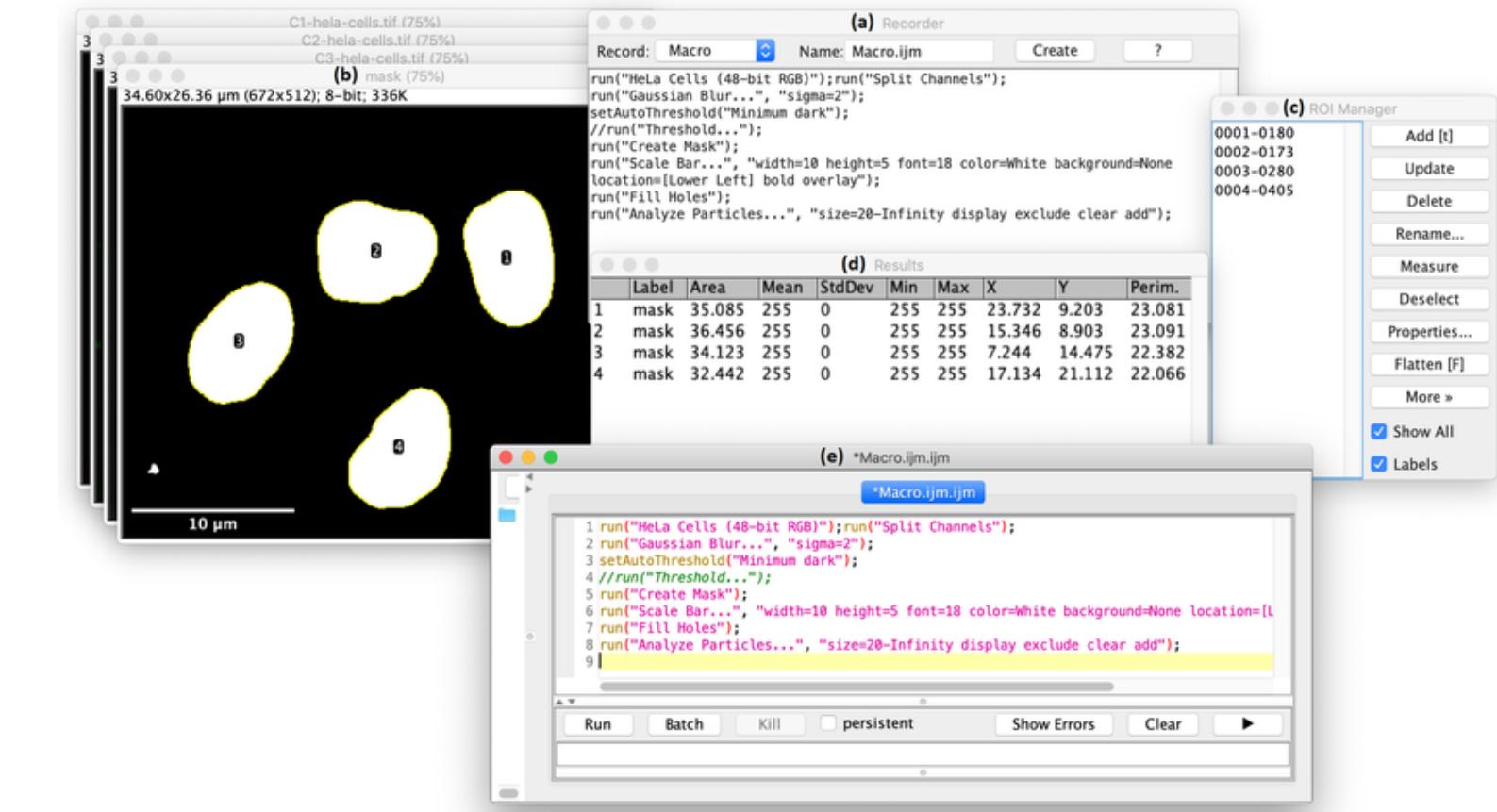


Why Use ImageJ Macro?

- ✓ **Productivity:** Automate repetitive tasks.
- ✓ **User experience:** Customize ImageJ to specific needs
- ✓ **Reproducibility:** Share procedures for consistent outcomes.
- ✓ **Distribution:** Disseminate methods for publication
- ✓ **Accessibility:** Cross-platform, free, open access
- ✓ **Steep learning curve:** Simplify with the Recorder.
- ✓ **Documentation:** Report protocols



- **Ready-to-use:** Start right away with built-in tools and plugins.
- **Long-term continuity:** A 25-year-old macro is still operational!



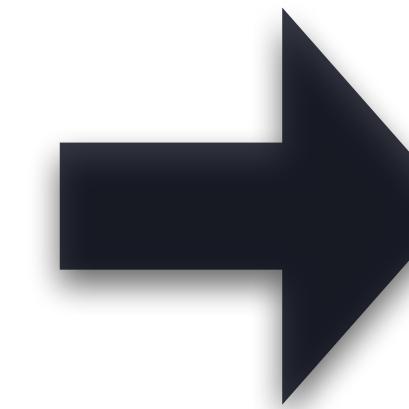
Why Don't Use ImageJ Macro?

- ✗ **Productivity:** Automate repetitive tasks.
- ✗ **Limited performance:** Slow, display issues.
- ✗ **Lack access to resources** GPU, HPC, RAM, Web.
- ✗ **Lack of user interface facilities**
- ⚠ **Strong dependency on ImageJ ecosystem**
- ⚠ **Maintenance:** Difficult to debug and to update.
- ⚠ **Poor programming practices:** Risk of bugs.



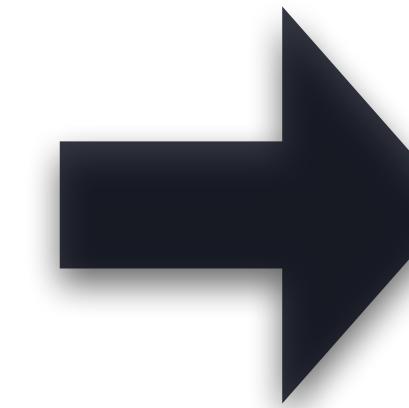
Usage of Macros

Simple image tasks: explore images, manipulation, quantification on thresholded images



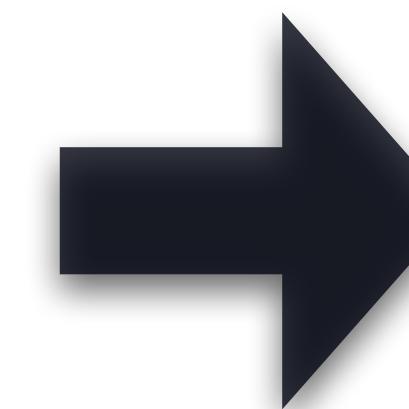
Built-in functions

Repetitive tasks: automatization, sequential simple operations, measurements



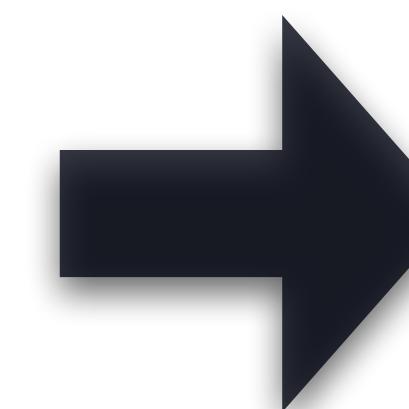
Macros
Scripting languages

Standard image-analysis tasks: e.g. neuro tracing, colocalization, localization microscopy



Third-party plugins

Specific image-analysis tasks: interactions, high density, complex organization, multi-channels, multithreading



Development plugins



Macro as Programming Language

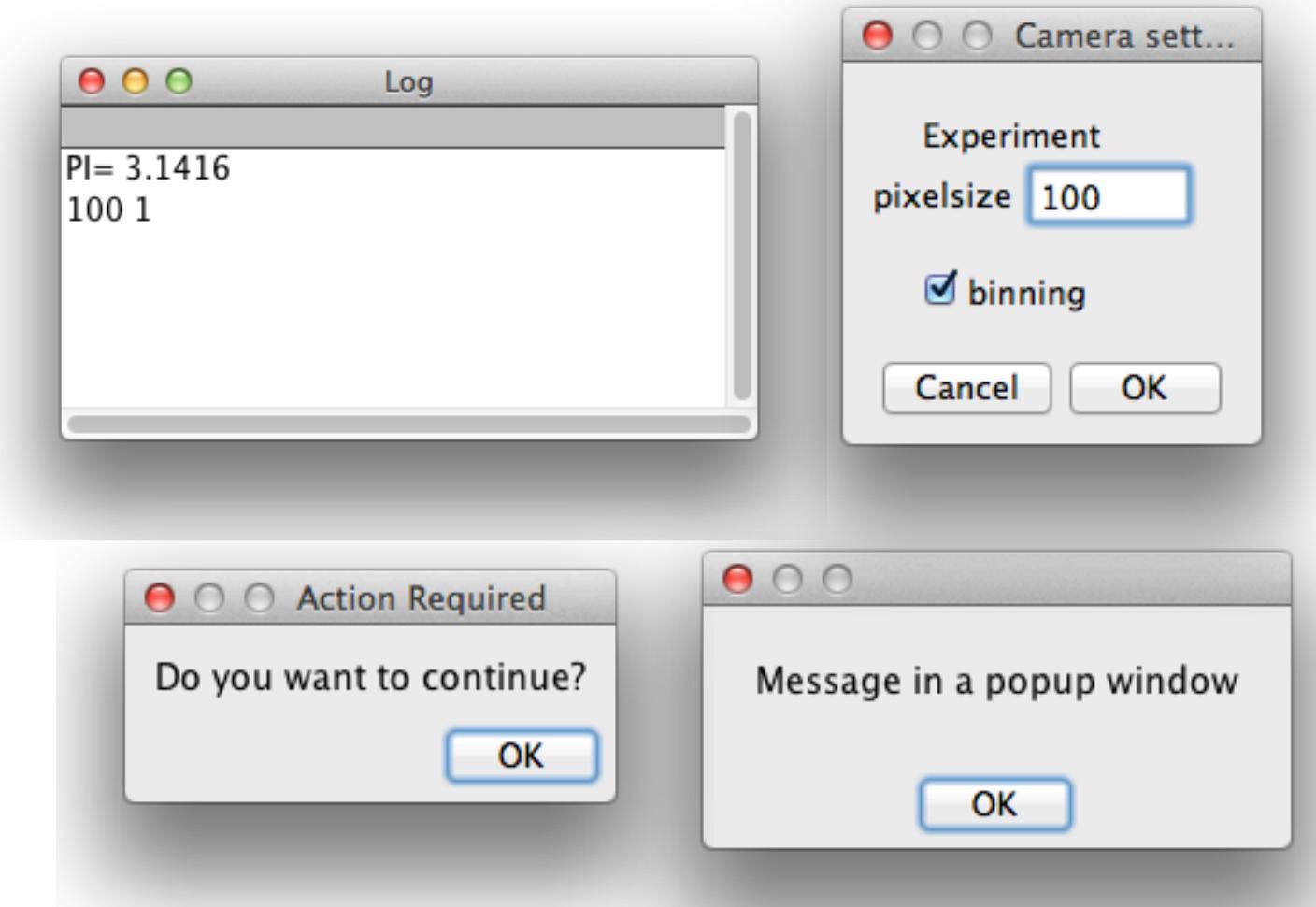
```
// Numerical variable  
a = 1024;  
radius = 20;  
thresholdFluorescence = 30;  
arr = newArray(20, 11, 14);  
radius = arr[0];  
/* Computation, usual arithmetics,  
mathematical functions */  
area = radius*radius * PI;  
volume = 4/3 * PI * pow(radius, 3);  
radius++;  
radius--;  
// String variable  
print("Area=" + area);  
print("Volume=" + volume);  
imageName = "Nucleus";  
channel = "Phase " + imageName;  
print(channel);
```

```
rename("A");  
selectWindow("A (blue)");  
close();
```

Difficult to
handle several
images

```
a = getNumber("Pixel size in nm", 150);  
getSelectionCoordinates(x, y);  
d = 0;  
for (i=1; i<x.length; i++) {  
    d = d + dist(x[i],y[i],x[i-1],y[i-1]);  
}  
d = d * a;  
if (d > 10000)  
    print("Distance=" + (d/1000) + " um");  
else  
    print("Distance=" + d + " nm");  
  
// Returns the distance  
function dist(x1, y1, x2, y2) {  
    dx = x1-x2;  
    dy = y1-y2;  
    return sqrt(dx*dx+dy*dy);  
}
```

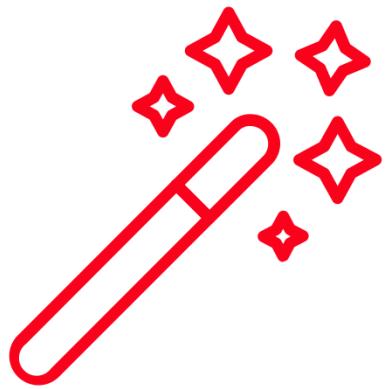
```
print("PI=", PI);  
showProgress(10, 100);  
showStatus("Message in the toolbar");  
showMessage("Message in a popup window");  
a = getNumber("pixelsize", 100);  
waitForUser("Do you want to continue?");  
Dialog.create("Camera settings");  
Dialog.addMessage("Experiment");  
Dialog.addNumber("pixelsize", 100);  
Dialog.addCheckbox("binning", true);  
Dialog.show;  
a = Dialog.getNumber();  
b = Dialog.getCheckbox();  
print(a, b);
```



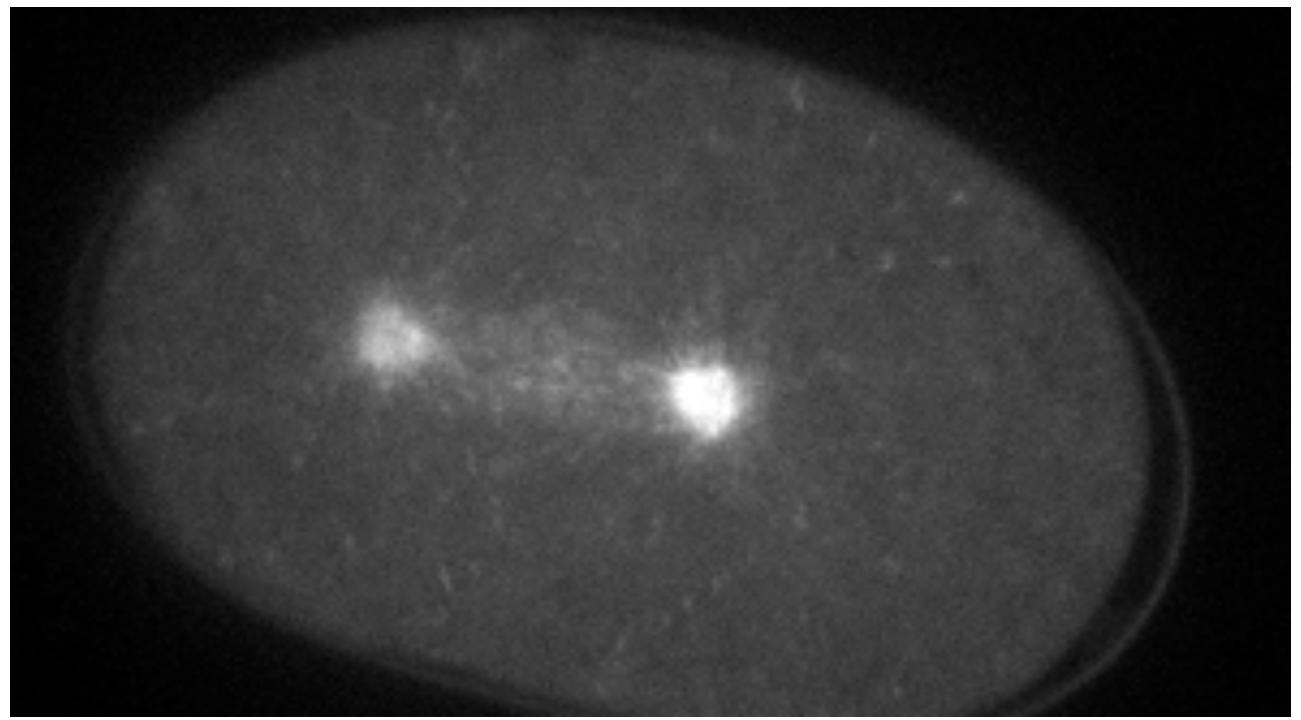


The Magic Tools

RECORDER



```
run("Enhance Contrast");
run("Make Binary");
run("Median...", "radius=2");
run("Fill Holes");
doWand(141, 153);
run("Measure");
```



AUGMENT FOR PERSONALIZED USE

StartupMacro.txt

```
macro "Measure Area [y]" {
    run("Enhance Contrast");
}

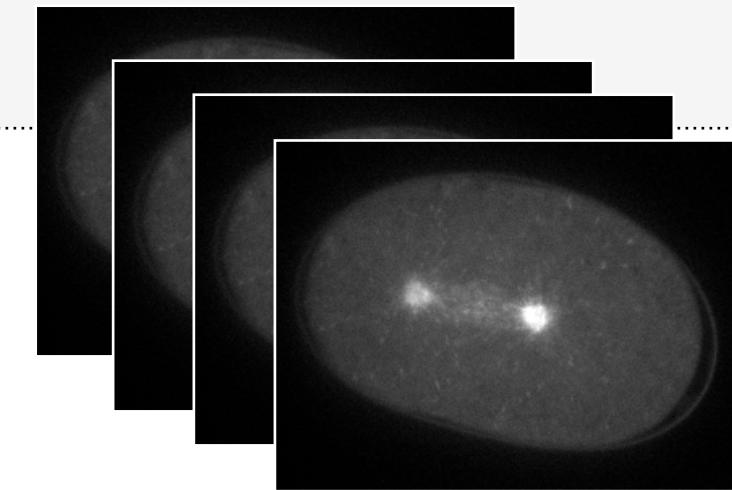
macro "Measure Action Tool -C669-F3366-F9966" {
    open("F0000.tif");
}

macro "AutoRun" {
    open("F0000.tif");
}
```

BATCH PROCESSING



```
dir = getDirectory("Choose Source
Directory ");
list = getFileList(dir);
setBatchMode(true);
for (i=0; i<list.length; i++) {
    showProgress(i+1, list.length);
    open(dir+list[i]);
    beep();
    run("Measure");
    close();
}
```



Collective Decision in Ants

Macros of Nathalie Stroeymeyt, University of Lausanne

Results										
	Area	X	Y	XM	YM	Perim.	BX	BY	Width	Height
16	1151	164.918	942.904	164.918	942.904	137.296	149	918	32	50
17	835	100.835	969.894	100.835	969.894	109.154	86	951	28	38
18	911	1710.367	1081.131	1710.367	1081.131	112.912	1692	1066	38	29
19	1034	641.624	1140.078	641.624	1140.078	118.569	625	1122	34	37
20	637	422.219	1136.886	422.219	1136.886	95.740	406	1124	33	26
21	1784	2831.681	1152.511	2831.681	1152.511	163.924	2805	1130	56	43
22	1221	1288.516	1193.661	1288.516	1193.661	130.225	1268	1176	41	37
23	763	1754.192	1234.564	1754.192	1234.564	102.326	1740	1219	30	32
24	1033	2852.057	1311.844	2852.057	1311.844	127.397	2832	1295	41	34
25	1411	2699.805	1317.814	2699.805	1317.814	140.811	2681	1297	39	44
26	1036	2962.709	1346.156	2962.709	1346.156	118.468	2944	1328	37	36
27	285	3533.581	1362.114	3533.581	1362.114	61.113	3524	1353	19	18
28	278	164.453	1490.313	164.453	1490.313	60.284	155	1482	19	17
29	674	3379.033	1603.755	3379.033	1603.755	95.497	3365	1589	28	30
30	878	724.743	1736.984	724.743	1736.984	110.569	709	1719	31	36
31	1558	2315.763	1756.660	2315.763	1756.660	150.711	2295	1736	45	45
32	139	1296.183	1783.227	1296.183	1783.227	42.627	1290	1777	13	13
33	2673	2029.134	1813.012	2029.134	1813.012	195.823	1995	1788	67	52
34	1003	1285.581	1818.153	1285.581	1818.153	123.054	1269	1796	32	43
35	1199	2805.465	1884.197	2805.465	1884.197	129.640	2784	1866	42	36
36	785	2051.933	1907.622	2051.933	1907.622	113.154	2031	1895	41	26
37	568	2371.641	2140.840	2371.641	2140.840	87.255	2359	2127	26	27
38	489	1482.044	2147.966	1482.044	2147.966	81.255	1469	2136	26	24
39	1267	2751.078	2200.776	2751.078	2200.776	138.610	2730	2179	45	41
40	869	3567.285	2261.775	3567.285	2261.775	112.326	3553	2242	28	39
41	1420	2771.046	2376.998	2771.046	2376.998	151.338	2746	2353	49	46
42	1184	2430.950	2471.502	2430.950	2471.502	127.640	2413	2450	35	42
43	968	1385.677	2484.770	1385.677	2484.770	117.054	1367	2467	37	35

10'000 lines of macro!

01_worker_filter_onecolony.ijm
02_blurring_all_cluster_forDaniel.ijm
03_a_detection_of_darkblue.ijm
03_b_detection_of_green_forDaniel.ijm
03_c_detection_of_orange.ijm
Darkblue_threshold.ijm
Green_threshold.txt



ImageJ Macro

Adipocyte 2:3, 160–164; July/August/September 2013; © 2013 Landes Bioscience

A novel automated image analysis method for accurate adipocyte quantification

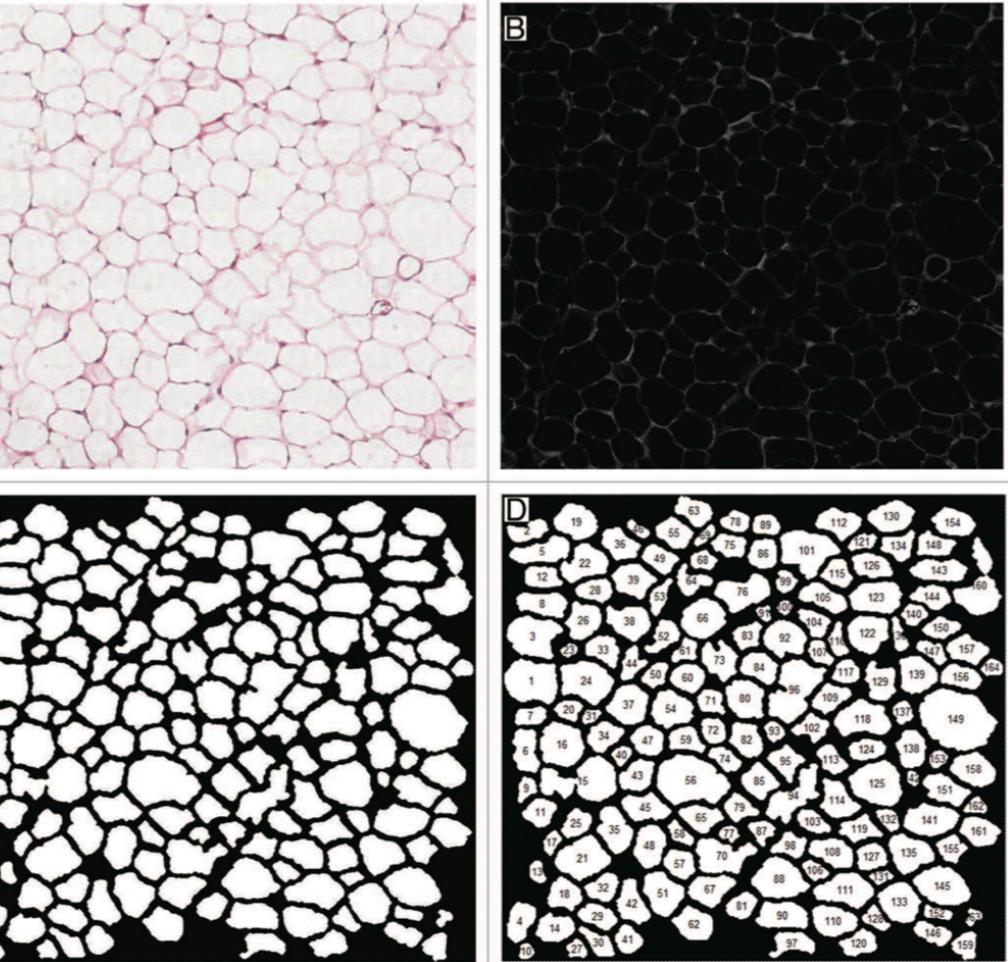
Osman S Osman,^{1,2} Joanne L Selway,^{1,*} Małgorzata A Kępczyńska,¹ Claire J Stocker,¹ Jacqueline F O'Dowd,¹ Michael A Cawthorne,¹ Jonathan RS Arch,¹ Sabah Jassim,² and Kenneth Langlands¹

¹The Clore Laboratory; University of Buckingham; Buckingham, UK; ²Department of Applied Computing; University of Buckingham; Buckingham, UK

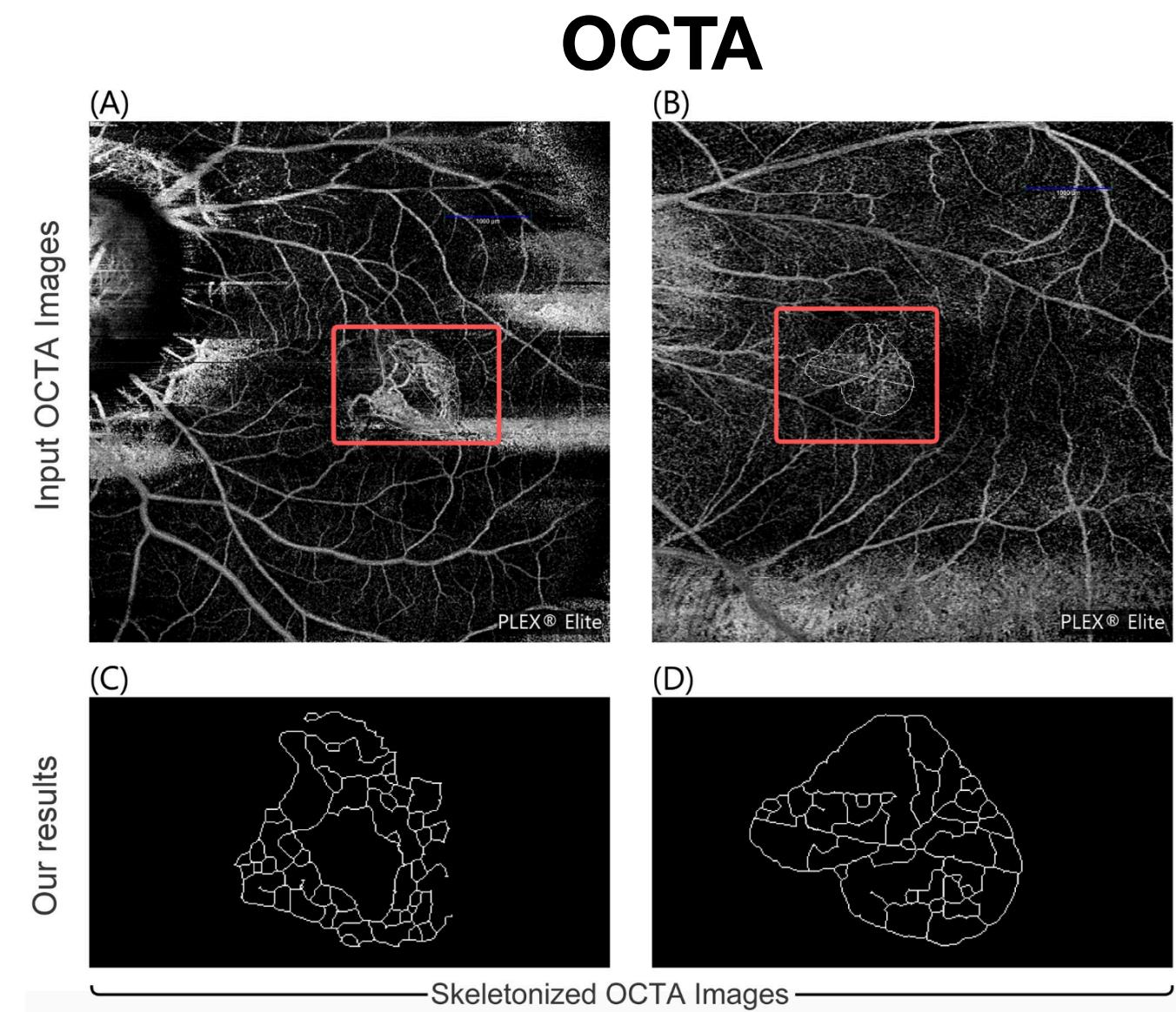
Keywords: adipocyte, histology, paraffin embedding, diet, high fat, automation, histomorphometry, image analysis

Abbreviations: HSV, hue saturation and value

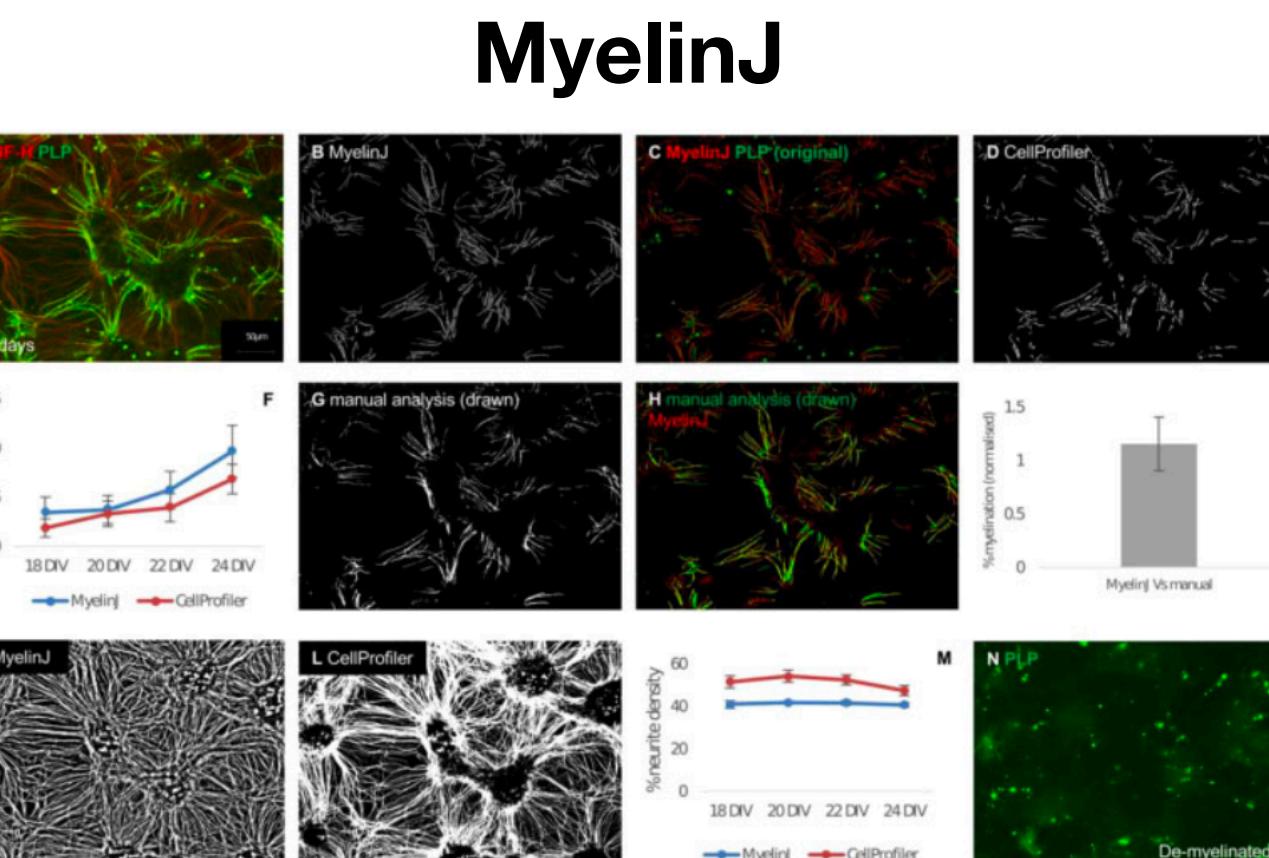
Increased adipocyte size and number are associated with many of the adverse effects observed in metabolic disease states. While methods to quantify such changes in the adipocyte are of scientific and clinical interest, manual methods to determine adipocyte size are both laborious and intractable to large scale investigations. Moreover, existing computational methods are not fully automated. We, therefore, developed a novel automatic method to provide accurate measurements of the cross-sectional area of adipocytes in histological sections, allowing rapid high-throughput quantification of fat cell size and number. Photomicrographs of H&E-stained paraffin sections of murine gonadal adipose were transformed using standard image processing/analysis algorithms to reduce background and enhance edge-detection. This allowed the isolation of individual adipocytes from which their area could be calculated. Performance was compared with manual measurements made from the same images, in which adipocyte area was calculated from estimates of the major and minor axes of individual adipocytes. Both methods identified an increase in mean adipocyte size in a murine model of obesity, with good concordance, although the calculation used to identify cell area from manual measurements was found to consistently over-estimate cell size. Here we report an accurate method to determine adipocyte area in histological sections that provides a considerable time saving over manual methods.



Adipocyte



OCTA



MyelinJ

Bioimage informatics

MyelinJ: an ImageJ macro for high throughput analysis of myelinating cultures

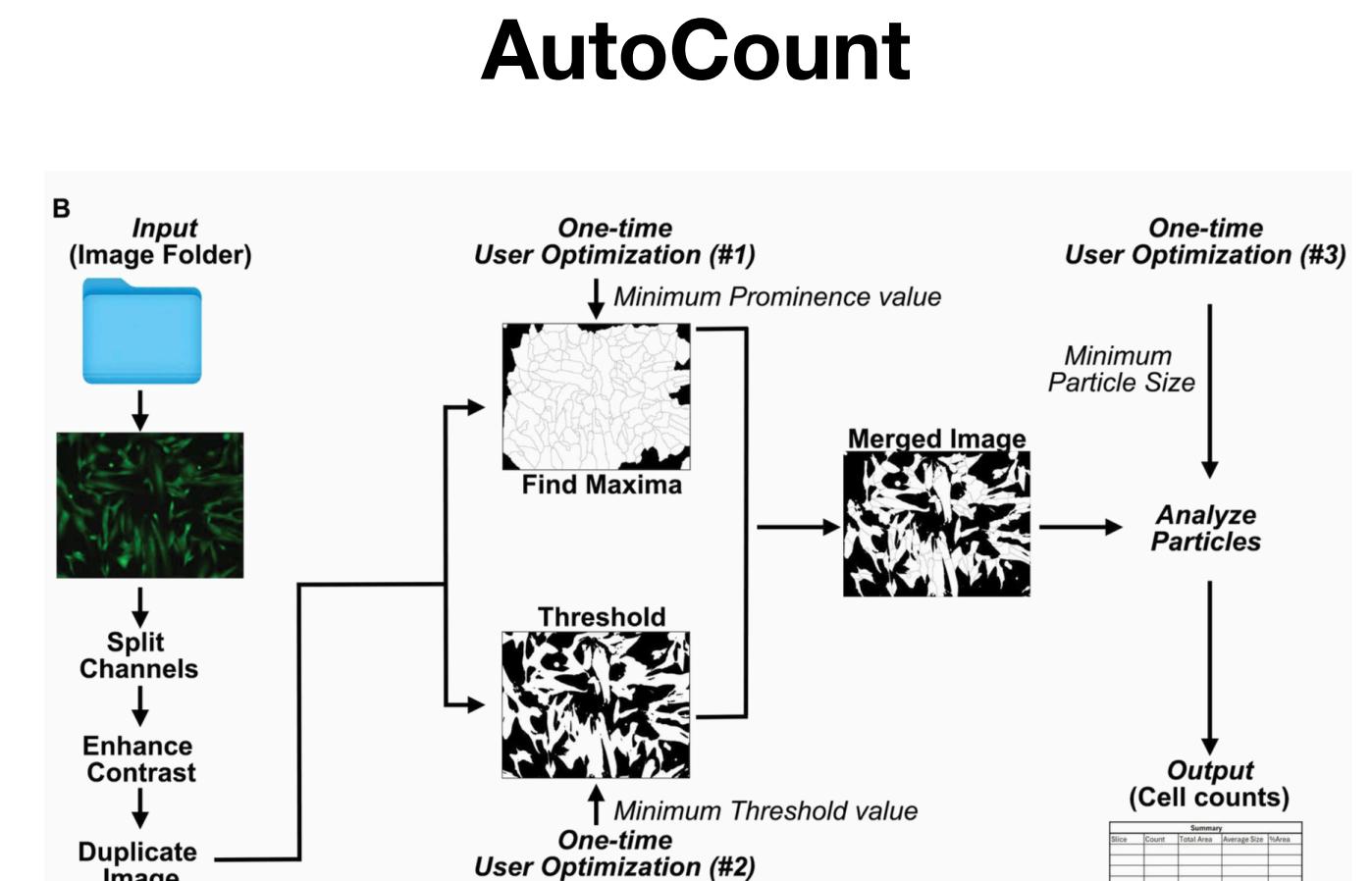
Michael J. Whitehead, George A. McCanney, Hugh J. Willison and Susan C. Barnett*

Institute of Infection, Immunity and Inflammation, College of Medical, Veterinary and Life Sciences, University of Glasgow, Glasgow G12 8TA, UK

*To whom correspondence should be addressed.

Associate Editor: Robert Murphy

Received on February 6, 2019; revised on April 8, 2019; editorial decision on May 3, 2019; accepted on May 7, 2019



AutoCount



Learn ImageJ Macro

The ImageJ Macro, Programmer's Reference Guide by J. Mutterer and W. Rasban.

Macro Functions webpage: <http://imagej.nih.gov/ij/developer/macro/functions.html>

Programming macro: http://fiji.sc/wiki/index.php/Introduction_into_Macro_Programming



ImageJ

Recorder

Examples



Web

Stackoverflow

ChatGPT

EPFL

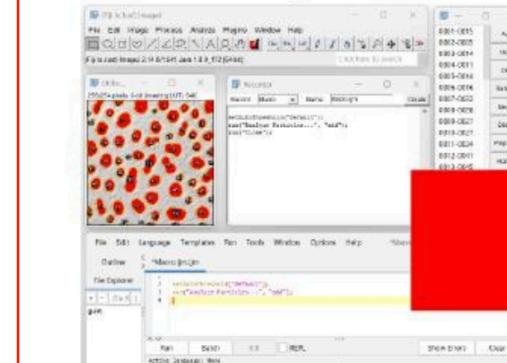
BIOP

Practical Course

Moocs Coursera
IPA for Life Scientists

Practical Course by BIOP : Fiji for Everyday Tasks

PCB: Fiji for Everyday Tasks



Tuesday, Oct. 7th 2024
13h to 17h



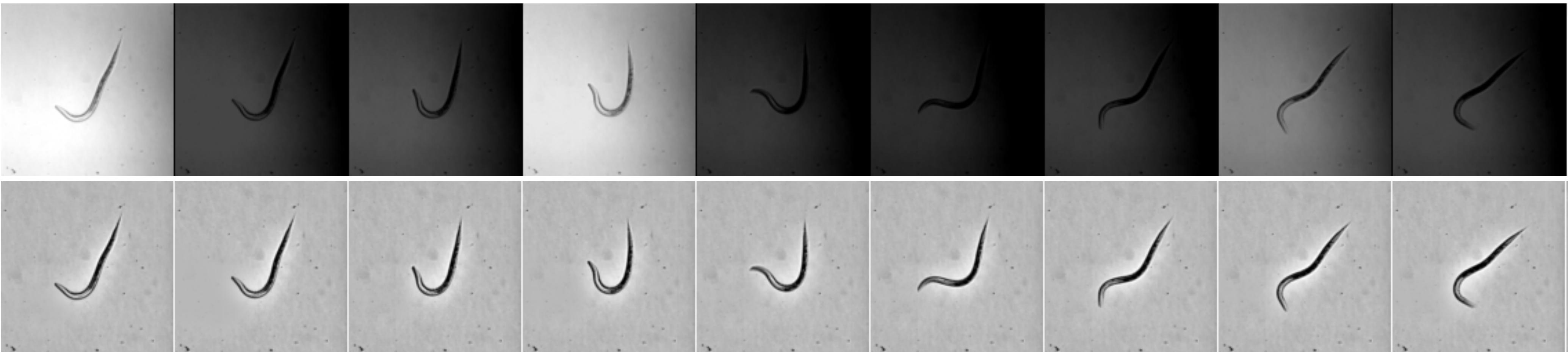
BioImaging and Optics Platform

Event details

Date	07/10/2025
Hour	13:00 > 17:00
Speaker	Romain Guiet, Nicolas Chiaruttini, Rémy Dornier
Location	SG 0213
Category	Internal trainings
Event Language	English



Practice A Illumination Correction



```
// This macro is applying a min-max normalization to all images
run("Close All");
dir_in = getDirectory("Choose Source Directory ");
dir_out= getDirectory("Choose Output Directory ");
list = getFileList(dir_in);
for (i=0; i<list.length; i++) {
    if (endsWith(list[i], ".tif") || endsWith(list[i], ".TIF")) {
        open(dir_in+list[i]);
        rename("input");
        run("32-bit");
        getStatistics(area, mean, min, max, std, histogram);
        print(i, area, mean, min, max, std);
        run("Subtract...", "value=" + min + " slice");
        run("Divide...", "value=" + (max-min) + " slice");
        run("Multiply...", "value=255 slice");
        save(dir_out+list[i]);
        close();
    }
}
```

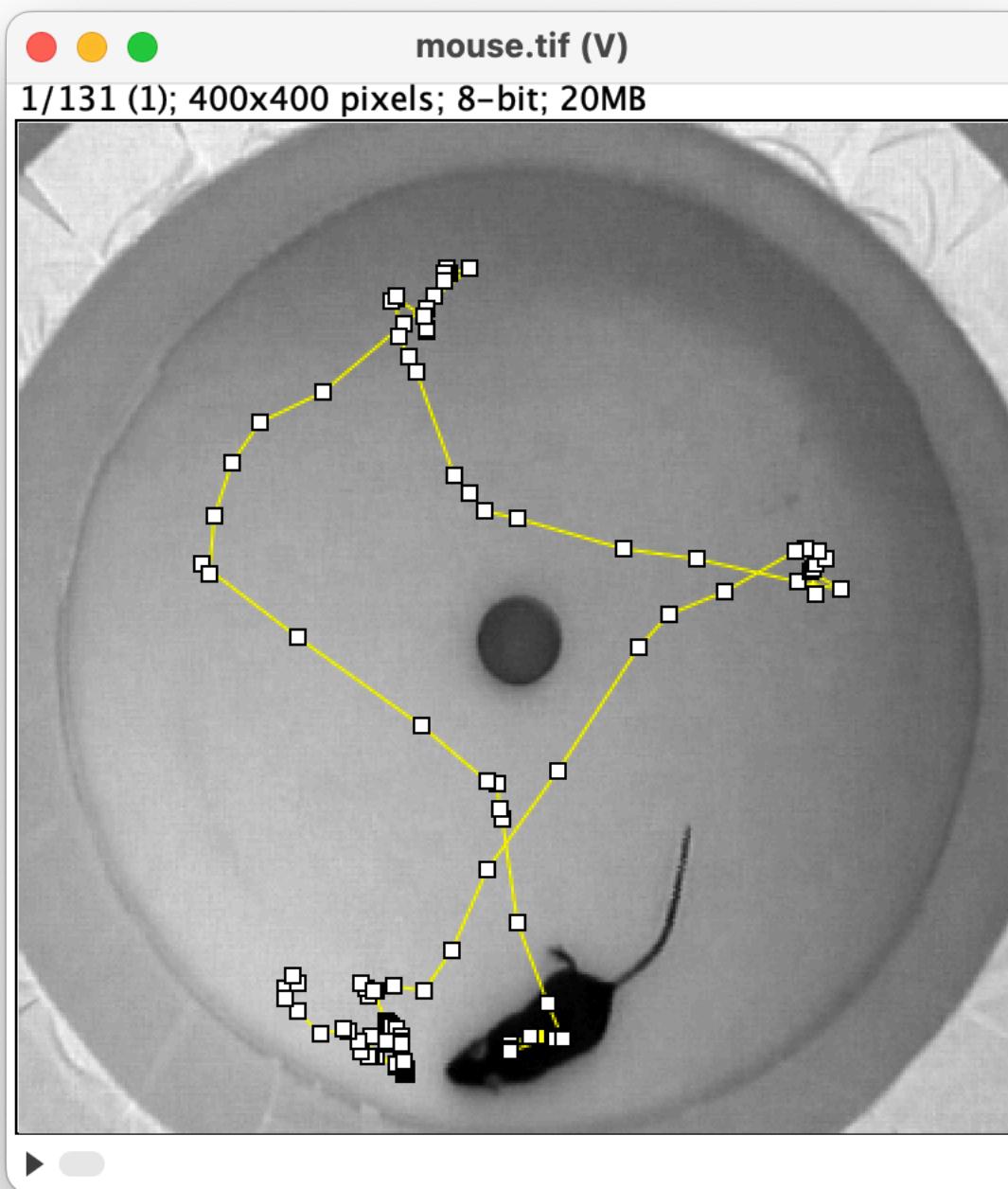
Min-max normalization

```
// This macro is removing the estimated background
run("Close All");
dir_in = getDirectory("Choose Source Directory ");
dir_out= getDirectory("Choose Output Directory ");
setBatchMode(true); // True: fast processing, no display of images
list = getFileList(dir_in);
for (i=0; i<list.length; i++) {
    if (endsWith(list[i], ".tif") || endsWith(list[i], ".TIF")) {
        showProgress(i+1, list.length);
        open(dir_in+list[i]);
        rename("input");
        run("Duplicate...", "title=gaussian");
        run("Gaussian Blur...", "sigma=30");
        imageCalculator("Subtract create 32-bit", "input","gaussian");
        save(dir_out+list[i]); close();
        selectWindow("gaussian"); close();
        selectWindow("input"); close();
    }
}
```

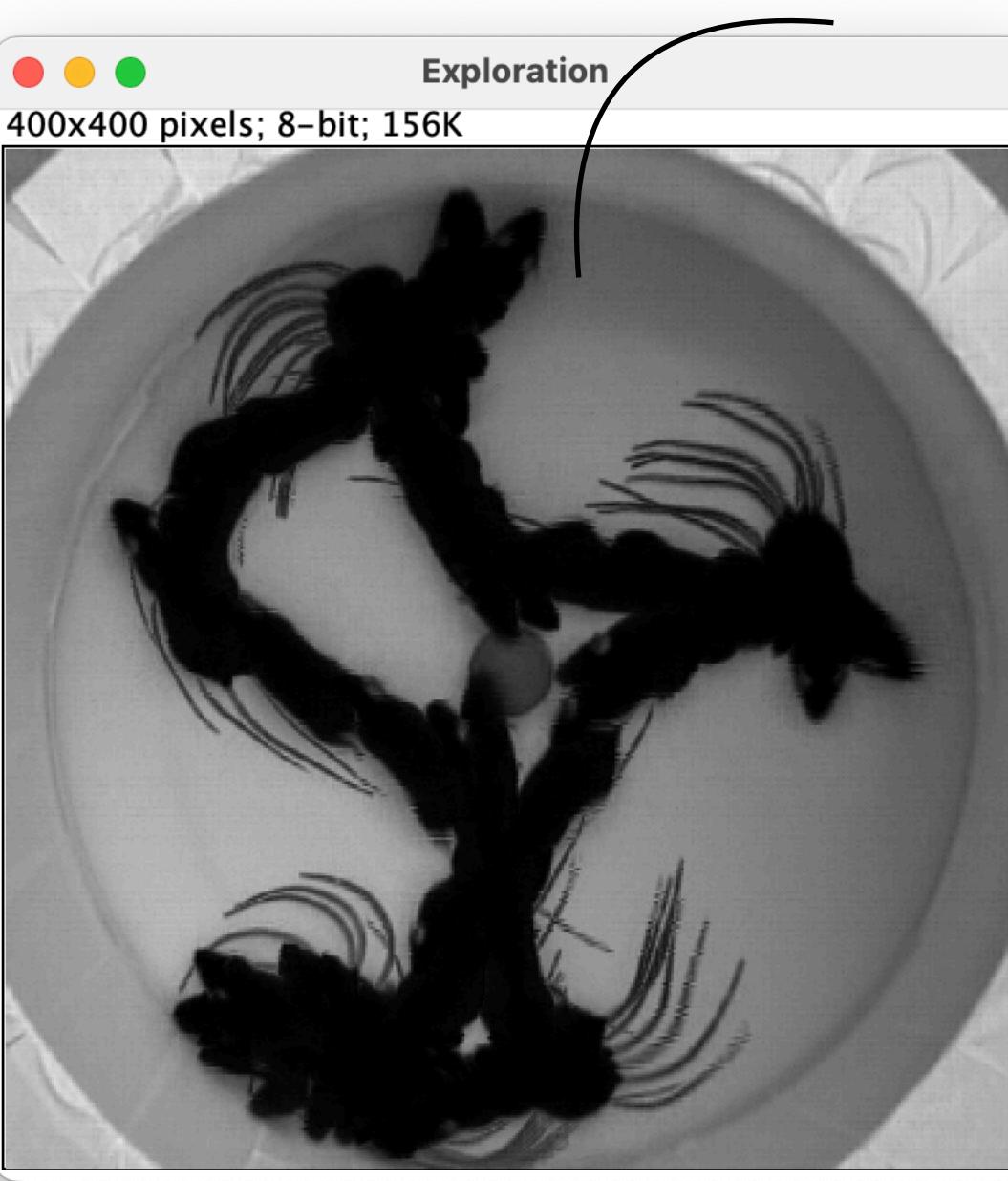
Flatten

Practice B Object Tracking

Overlay trajectory



Exploration zone



```
run("Z Project...", "projection=[Max Intensity]");
imageCalculator("Subtract create 32-bit stack",
"mouse.tif","MAX_mouse.tif");
rename("mouse-background")
run("Maximum...", "radius=2 stack");
run("Minimum...", "radius=2 stack");
run("Median...", "radius=6 stack");
run("8-bit");
run("Make Binary", "calculate");
run("Set Measurements...", "area center fit add
redirect=None decimal=2");
run("Analyze Particles...", "size=100-Infinity
show=[Overlay Outlines] display stack");

x=newArray(nResults)
y=newArray(nResults)
for (i=0; i<nResults;i++) {
  x[i]=getResult("XM", i);
  y[i]=getResult("YM", i);
}
selectWindow("mouse.tif");
makeSelection("polyline", x, y);

// Produce exploration image
selectWindow("mouse.tif");
run("Z Project...", "projection=[Min Intensity]");
rename("Exploration");
```