

Batch Processing a Folder

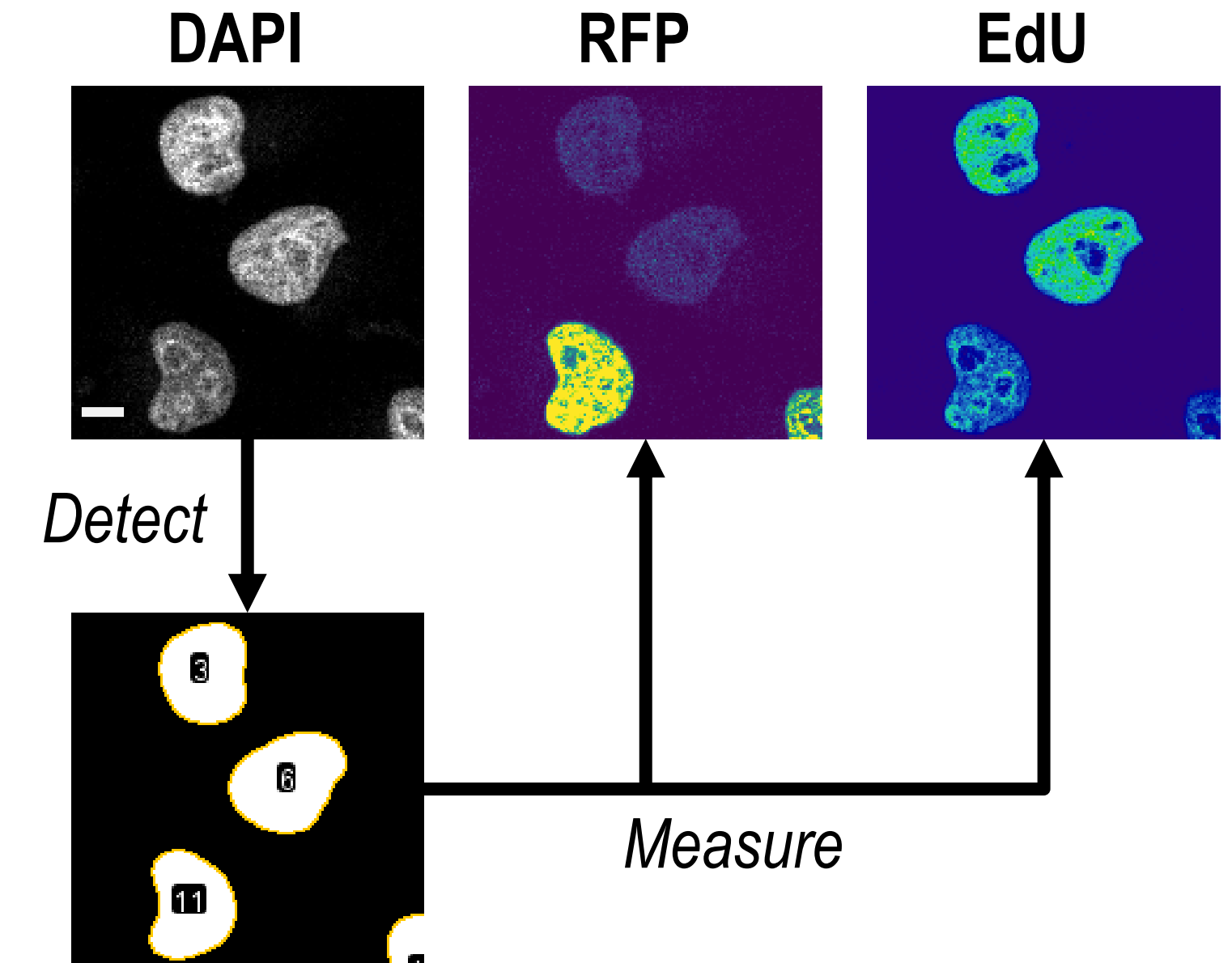
Image Processing & Analysis for Life Scientists

Olivier Burri, Romain Guiet & Arne Seitz

Batch Processing a Folder

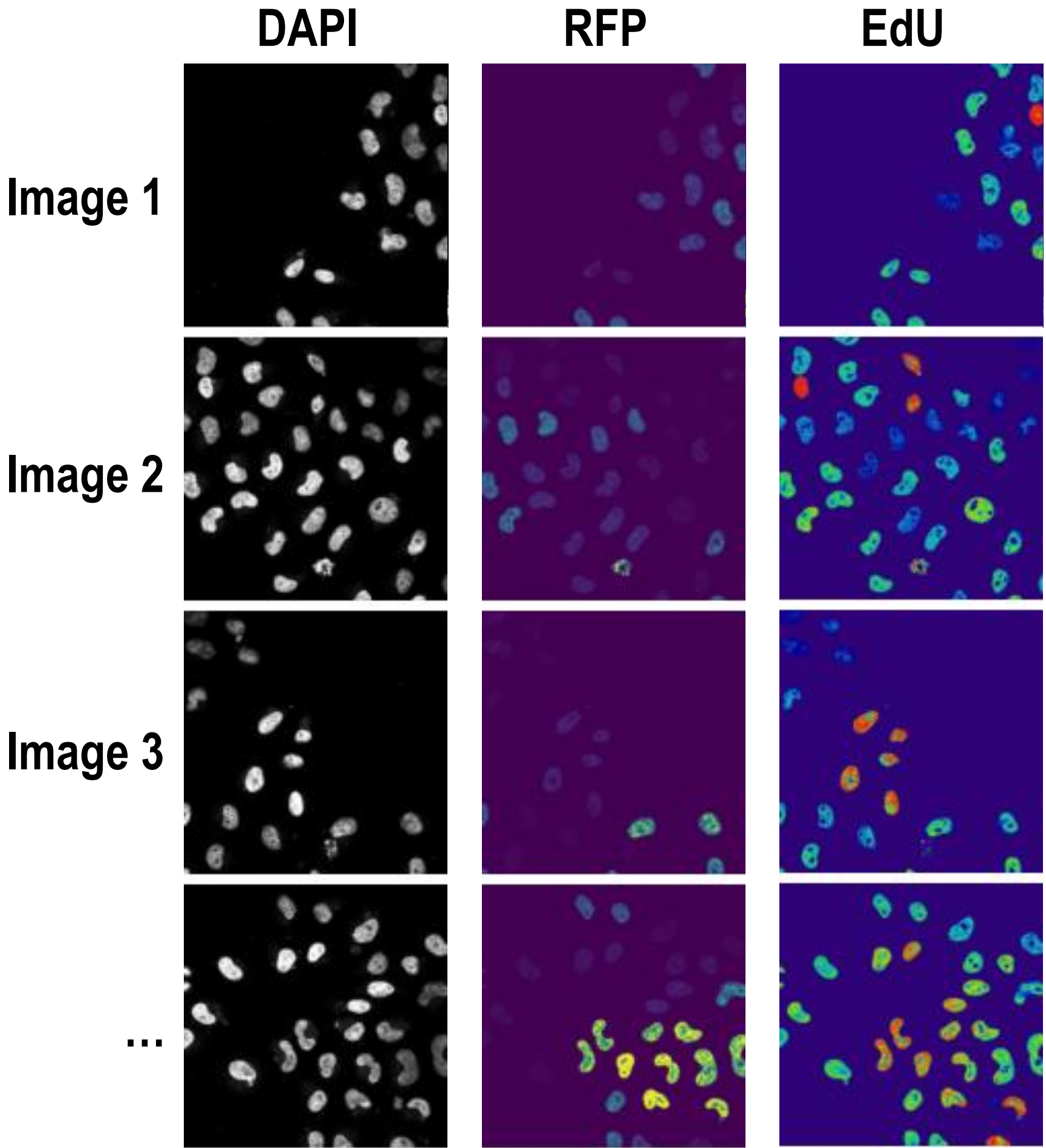
- Motivation
- The Batch Processor
- Custom Batch Process

It's Never Just One Image

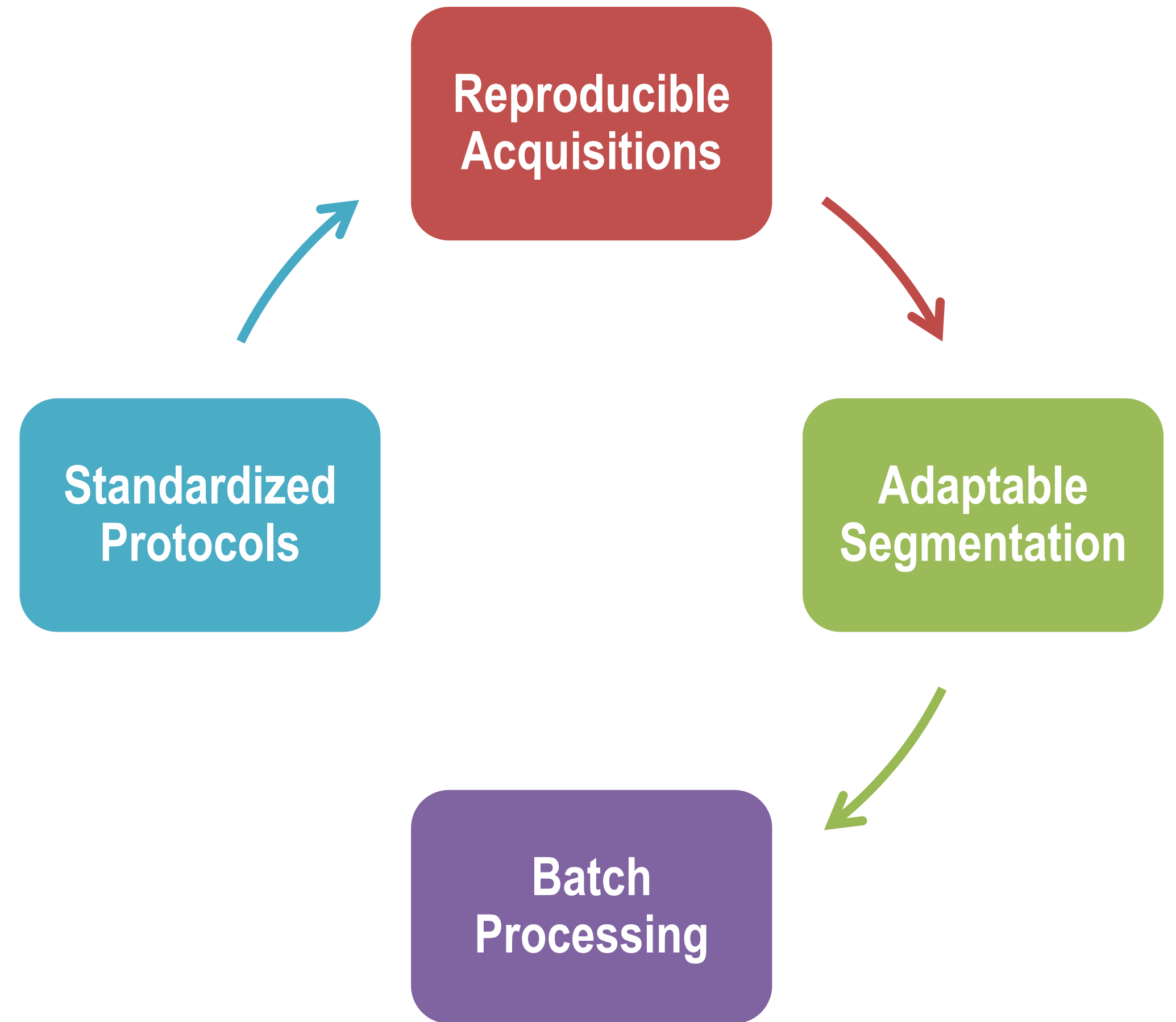


```
// Run the segmentation.  
// Duplicate the first channel (DAPI)  
// Filter the image with a median filter and reduce inte  
// Set an auto-threshold and get a mask  
// Clean the mask with a median filter and a watershed  
run("Duplicate...", "title=DAPI duplicate channels=1");  
run("Median...", "radius=4");  
run("32-bit");  
run("Square Root");  
run("Enhance Contrast", "saturated=0.35");  
setAutoThreshold("Otsu dark no-reset");  
setOption("BlackBackground", true);  
run("Convert to Mask");  
run("Median...", "radius=2");  
run("Watershed");  
  
// End of segmentation
```

It's Never Just One Image

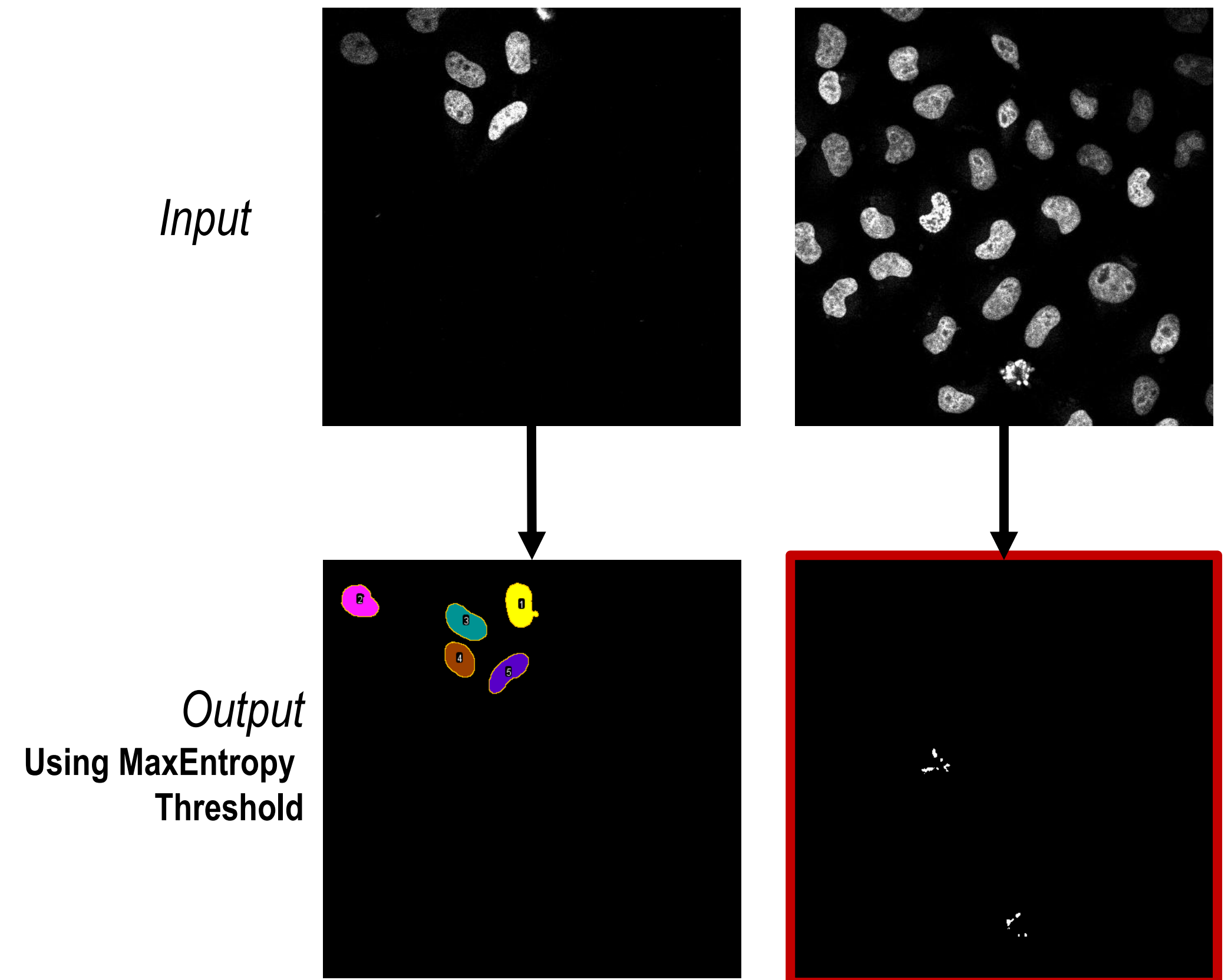


The Need For Standardization



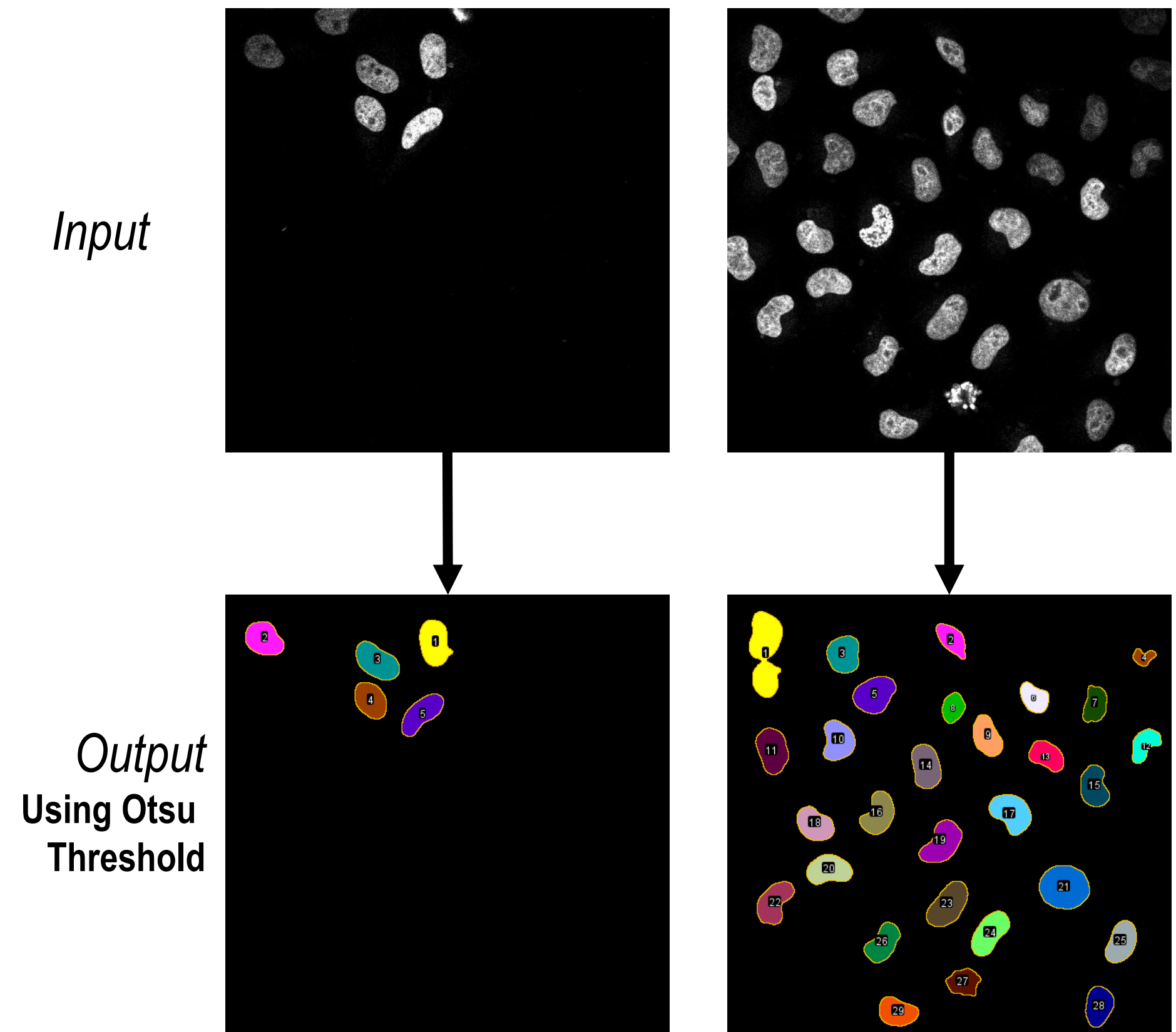
Batch: What do we Need?

- Output for Quality Control



Batch: What do we Need?

- Output for Quality Control



Batch: What do we Need?

- Image Name
- Image Channel
- Measurements

Label	Channel	Area	Mean
Image 1: ROI 1	1	23.2	210.2
Image 1: ROI 1	2	23.2	158.3
Image 1: ROI 1	3	23.2	53.7
Image 1: ROI 2	1	34.5	123.2
Image 1: ROI 2	2	34.5	173.0
...

Batch: What do we Need?

- Image Name
- Image Channel
- Measurements

Label	Channel	
Image 1: ROI 1	1	2
Image 1: ROI 1	2	2
Image 1: ROI 1	3	2
Image 1: ROI 2	1	3
Image 1: ROI 2	2	3
...

Set Measurements

☒Area

☐Standard deviation

☐Min & max gray value

☐Center of mass

☐Bounding rectangle

☒Shape descriptors

☐Integrated density

☐Skewness

☐Area fraction

☒Mean gray value

☐Modal gray value

☐Centroid

☐Perimeter

☐Fit ellipse

☐Feret's diameter

☐Median

☐Kurtosis

☒Stack position

☐Limit to threshold

☒Display label

☐Invert Y coordinates

☐Scientific notation

☐Add to overlay

☐NaN empty cells

Redirect to:

None

Decimal places (0-9):

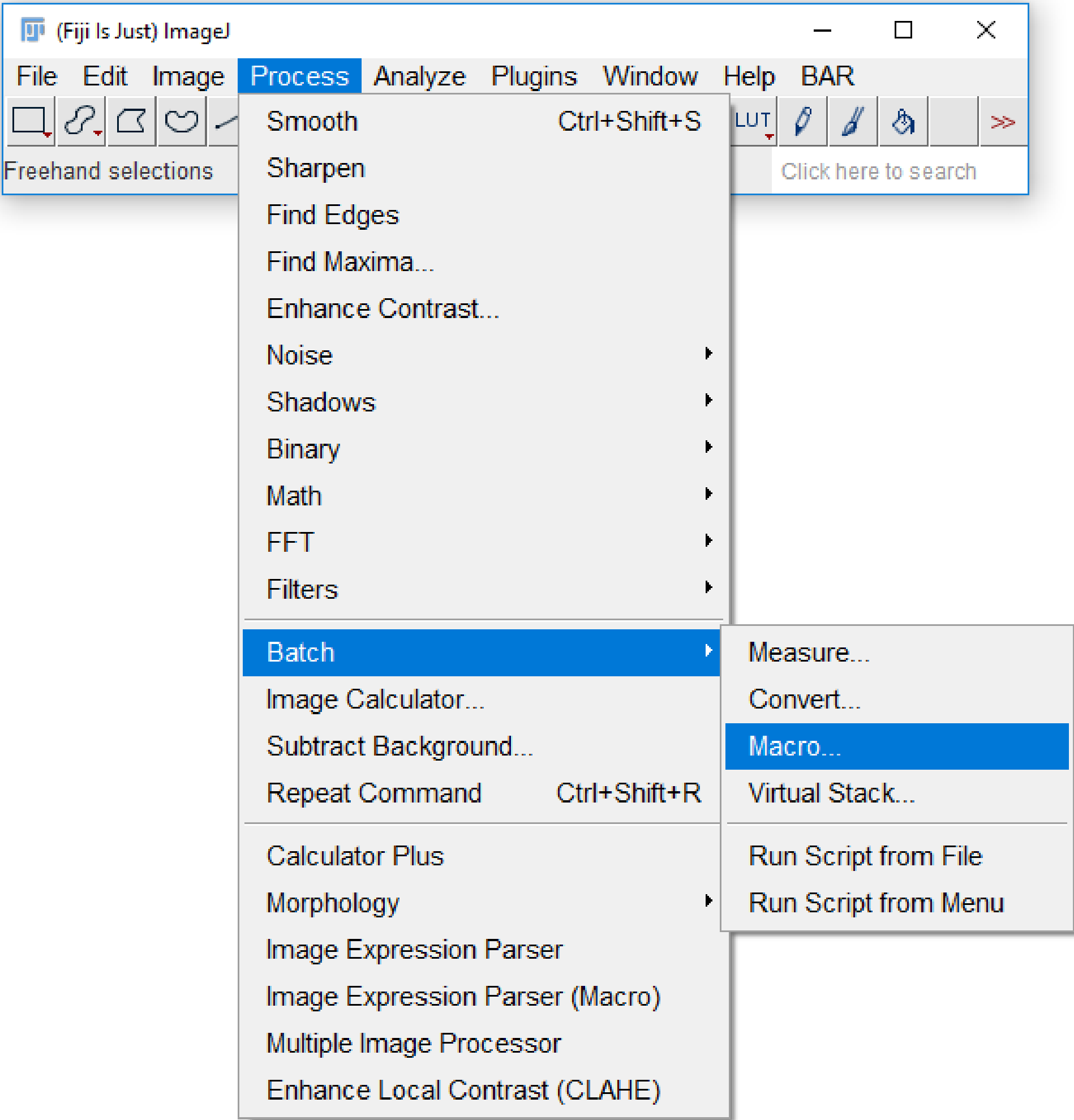
3

OK


Cancel

Help

Batch Process Macro



Batch Process Macro

 Batch Process

Input...

Input Folder (files only)

Output...

Empty Output Folder

Output format:

TIFF

Add macro code:

[Select from list]

File name contains:

Macro Code Here

Test

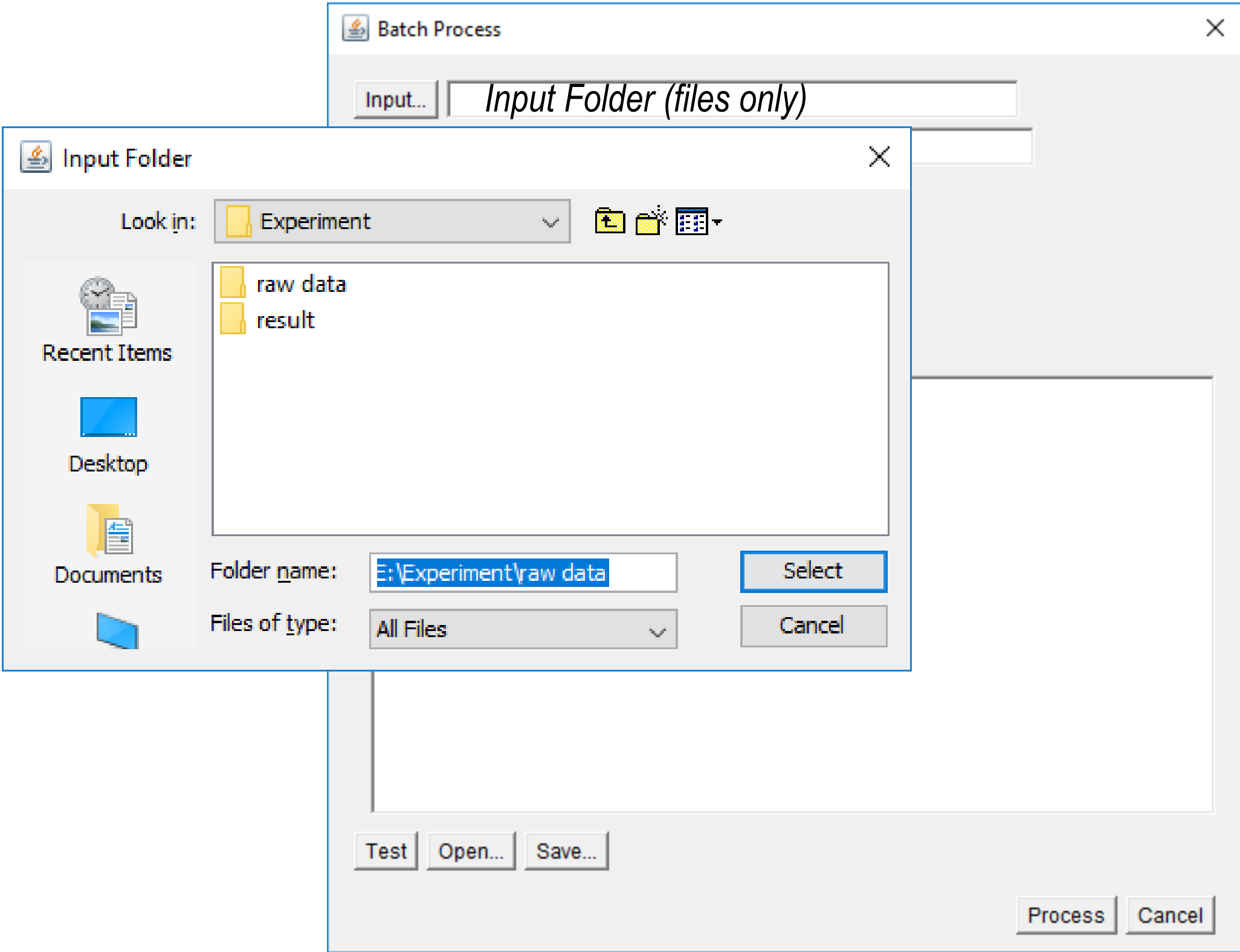
Open...

Save...

Process

Cancel

Batch Process Macro



Batch Process Macro

Batch Process

×

Input...

E:\Experiment\raw data\

Output...

E:\Experiment\result\

Output format:

TIFF

▼

Add macro code:

[Select from list]

▼

File name contains:

Macro Code Here

Test

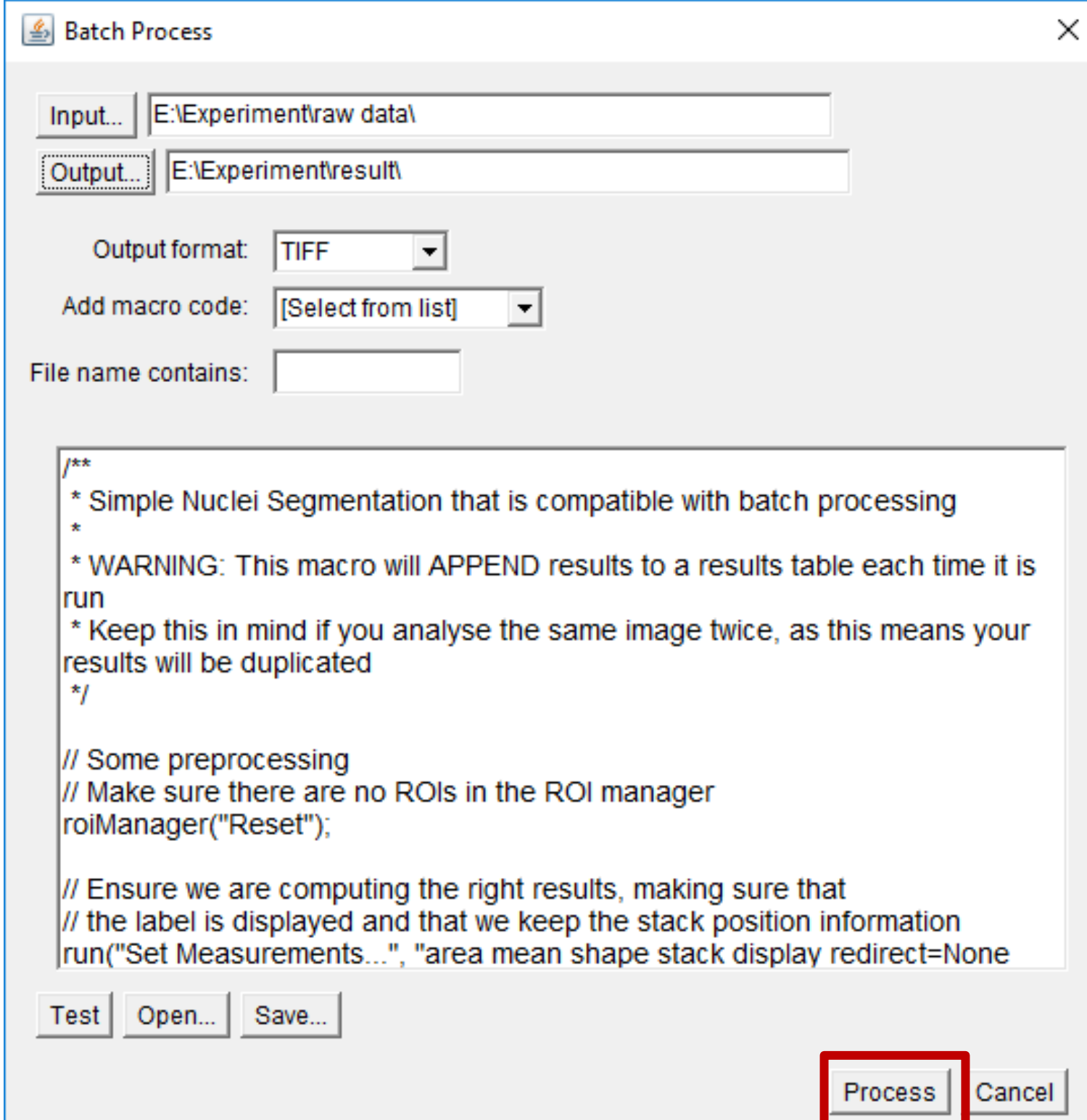
Open...

Save...

Process

Cancel

Batch Process Macro

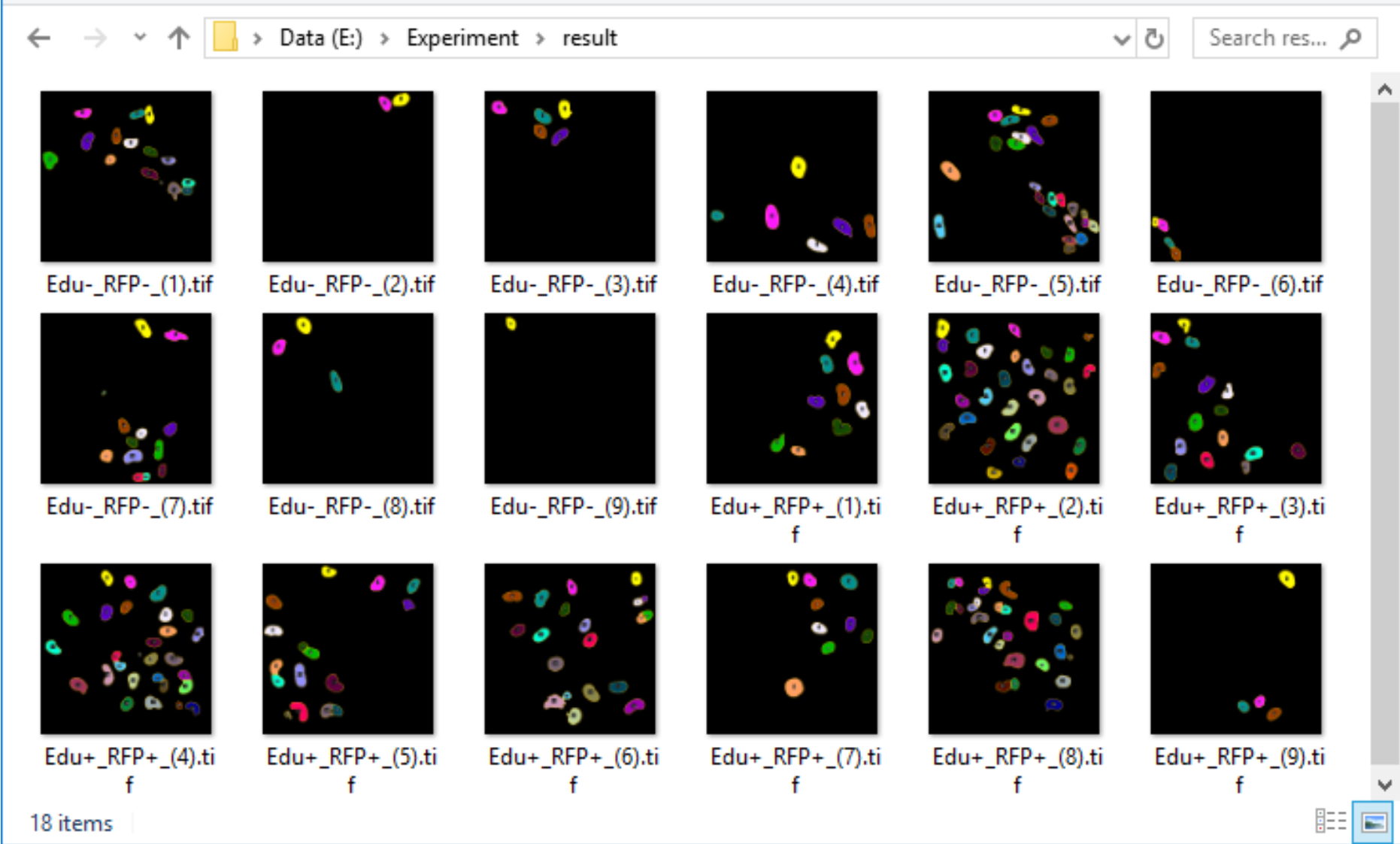


The image shows a 'Batch Process' dialog box with the following fields and controls:

- Input...** button and text field containing 'E:\Experiment\raw data\'
- Output...** button and text field containing 'E:\Experiment\result\'
- Output format:** dropdown menu set to 'TIFF'
- Add macro code:** dropdown menu set to '[Select from list]'
- File name contains:** empty text field
- A large text area containing the following macro code:

```
/**  
 * Simple Nuclei Segmentation that is compatible with batch processing  
 *  
 * WARNING: This macro will APPEND results to a results table each time it is  
 run  
 * Keep this in mind if you analyse the same image twice, as this means your  
 results will be duplicated  
 */  
  
// Some preprocessing  
// Make sure there are no ROIs in the ROI manager  
roiManager("Reset");  
  
// Ensure we are computing the right results, making sure that  
// the label is displayed and that we keep the stack position information  
run("Set Measurements...", "area mean shape stack display redirect=None
```
- Buttons at the bottom: 'Test', 'Open...', 'Save...', 'Process' (highlighted with a red box), and 'Cancel'.

Batch Process Macro

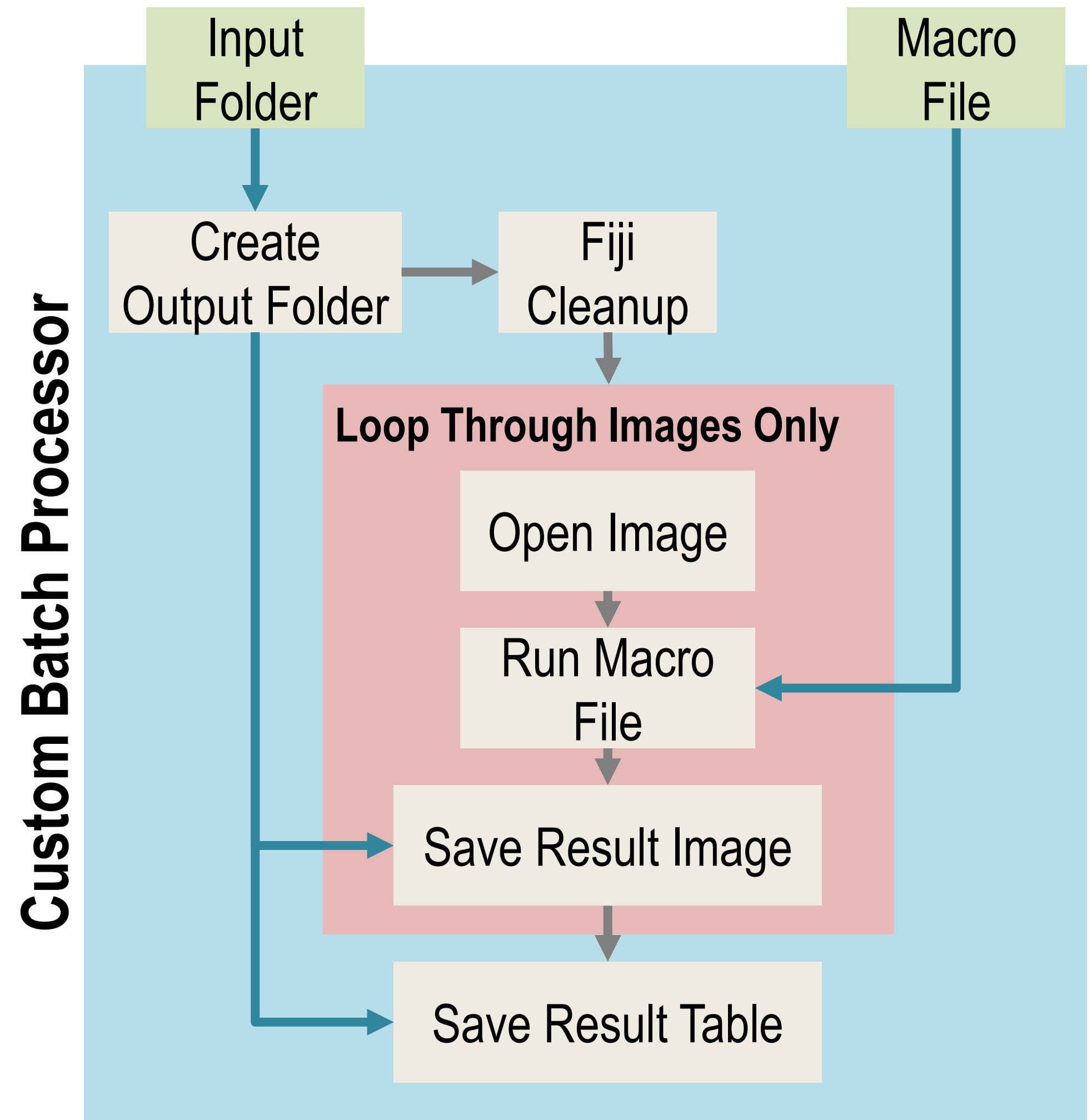


Results									
File	Edit	Font	Results						
	Label		Area	Mean	Circ.	Ch	AR	Round	Solidity
1	Edu-_RFP-_(1).tif.0001-0071:1		165.787	24.484	0.679	1	2.395	0.418	0.930
2	Edu-_RFP-_(1).tif.0002-0067:1		172.852	14.224	0.721	1	2.055	0.487	0.936
3	Edu-_RFP-_(1).tif.0003-0069:1		146.830	29.273	0.720	1	2.253	0.444	0.948
4	Edu-_RFP-_(1).tif.0004-0135:1		158.549	28.899	0.780	1	1.927	0.519	0.955
5	Edu-_RFP-_(1).tif.0005-0151:1		230.412	20.789	0.797	1	1.752	0.571	0.947
6	Edu-_RFP-_(1).tif.0006-0157:1		171.991	32.691	0.822	1	1.630	0.613	0.956
7	Edu-_RFP-_(1).tif.0007-0182:1		142.694	41.935	0.785	1	1.764	0.567	0.953
8	Edu-_RFP-_(1).tif.0008-0208:1		288.145	14.140	0.824	1	1.222	0.819	0.947
9	Edu-_RFP-_(1).tif.0009-0206:1		120.118	15.254	0.917	1	1.211	0.826	0.959
10	Edu-_RFP-_(1).tif.0010-0210:1		125.805	35.329	0.741	1	1.968	0.508	0.958
11	Edu-_RFP-_(1).tif.0011-0248:1		175.265	50.037	0.698	1	2.252	0.444	0.933
12	Edu-_RFP-_(1).tif.0012-0275:1		141.832	50.187	0.820	1	1.612	0.621	0.946

Batch Process Macro Limitations

- Will not save your results table
- Will open ALL files in the input directory.
 - Might cause bugs
- Can overwrite your result images if input and output folders are the same!!

Creating Our Own Batch Processor



Custom Batch Process

```
#@File (label="Input Folder", style="directory") input_folder
#@File (label="Macro Code To Run") macro_code
#@String (label="Image File Extension", value=".tif") extension
```

```
// Generic Batch Processing Workflow
```

```
// Pre-batch processing steps
```

```
// Clean contents of Fiji
```

```
run("Close All");
run("Clear Results");
roiManager("Reset");
```

```
// Build the output folder
```

```
output_folder = input_folder+"/output";
File.makeDirectory(output_folder);
```

Custom Batch Process

```
// Get list of all the files in the input folder
files = getFileList(input_folder);

// Now make sure to process only the files with the proper extension
for(i=0; i < lengthOf(files); i++) {
    if(endsWith(files[i], extension)) {
        open(input_folder+"/"+files[i]);
        // Run the selected script
        runMacro(macro_code);

        // Save the last image
        saveAs("tif", output_folder+"/"+files[i]+"_result.tif");

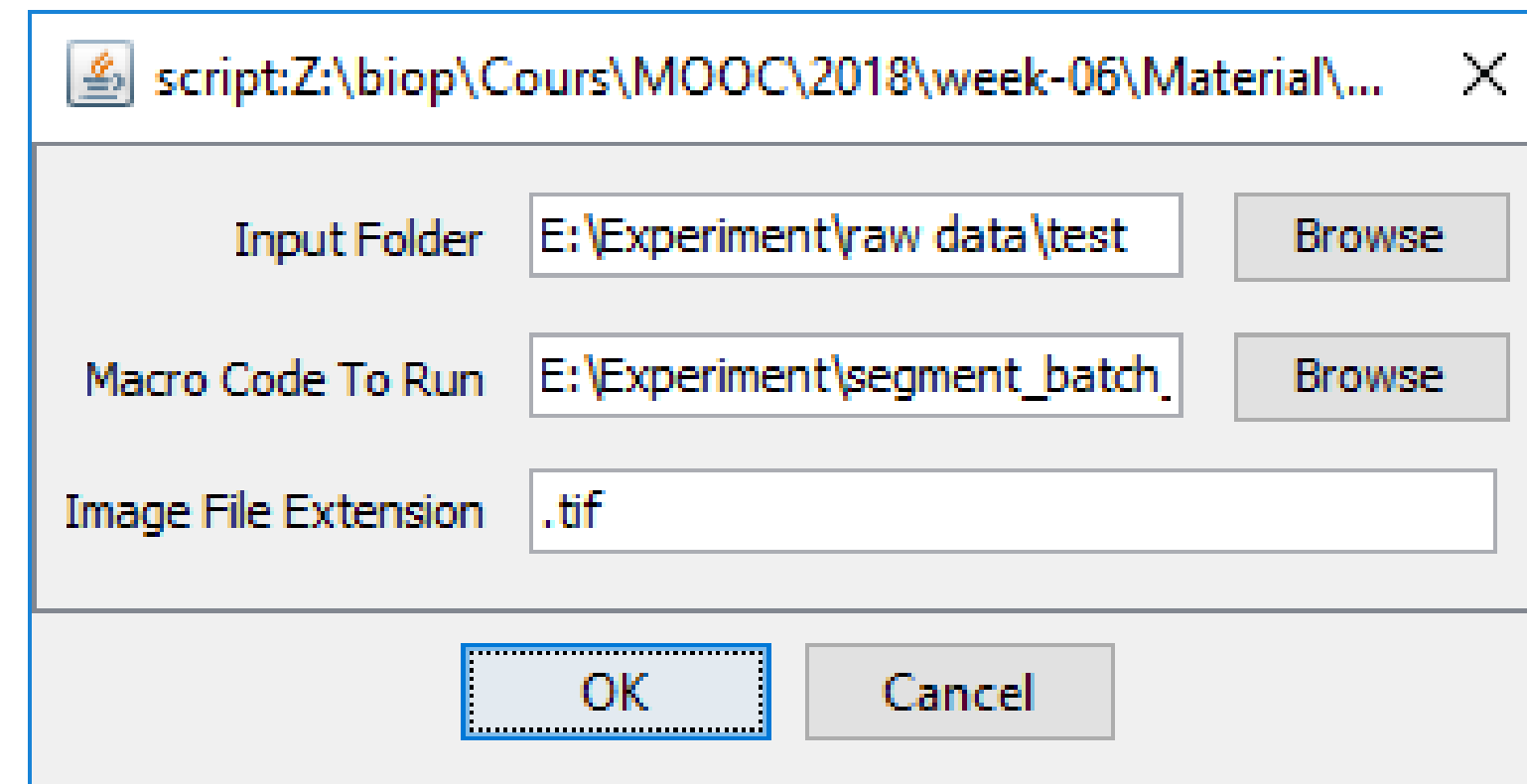
        // Cleanup for next run
        run("Close All");
    }
}
```

Custom Batch Process

```
// Save the results  
saveAs("Results", output_folder+"Results.csv");  
// Inform the user that the processing is finished  
showMessage("Batch Process Complete");
```

Custom Batch Process

```
#@File (label="Input Folder", style="directory") input_folder  
#@File (label="Macro Code To Run") macro_code  
#@String (label="Image File Extension", value=".tif") extension
```



The screenshot shows a dialog box titled "script:Z:\biop\Cours\MOOC\2018\week-06\Material\..." with a close button (X). The dialog contains three input fields and two buttons:

Field Label	Field Value	Field Type / Button
Input Folder	E:\Experiment\raw data\test	Text input with "Browse" button
Macro Code To Run	E:\Experiment\segment_batch_	Text input with "Browse" button
Image File Extension	.tif	Text input

At the bottom of the dialog are two buttons: "OK" and "Cancel".

- Motivation
 - Processing Large Amounts of Images
- The Batch Processor
 - Limitations
- Custom Batch Process
 - To Help Overcome Limitations