UNIVERSITY OF HELSINKI
FACULTY OF MEDICINE

ImageJ/Fiji Macro Language

Elnaz Fazeli, Biomedicum Imaging Unit, University of Helsinki, Finland
Anna Klemm, SciLifeLab, NBIS, Sweden

2024-09-30

@mycroscopy

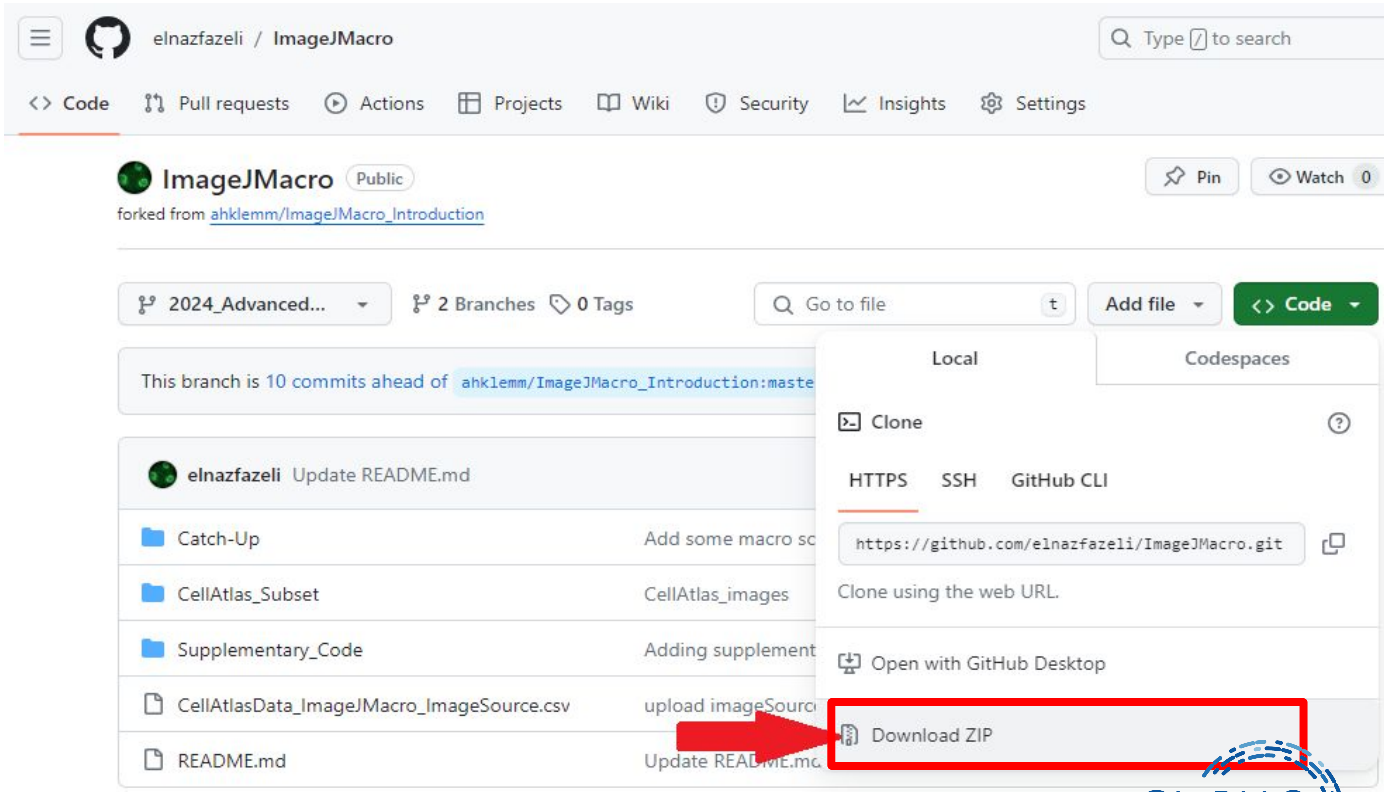
GloBIAS Global BioImage Analysts' Society

neubias network of european bioimage analysts

Preparations

Download material:
https://tinyurl.com/ImageJMacro

- Click on Code > Download ZIP
- unzip on your computer



forked from ahklemm/ImageJMacro_Introduction

This branch is 10 commits ahead of ahklemm/ImageJMacro_Introduction:maste

elnazfazeli Update README.md

Catch-Up	Add some macro sc
CellAtlas_Subset	CellAtlas_images
Supplementary_Code	Adding supplement
CellAtlasData_ImageJMacro_ImageSource.csv	upload imageSourc
README.md	Update README.mc

```
https://github.com/elnazfazeli/ImageJMacro.git
```

Clone using the web URL.

Outline of This Session

- Biological Data Set and Image Analysis Problem
- How can we ”talk” to Fiji? - Macro Recorder, Built-in Macro Function
- Step-by-Step Workflow

The Cell Atlas/Human Protein Atlas

Cell Atlas Aim:

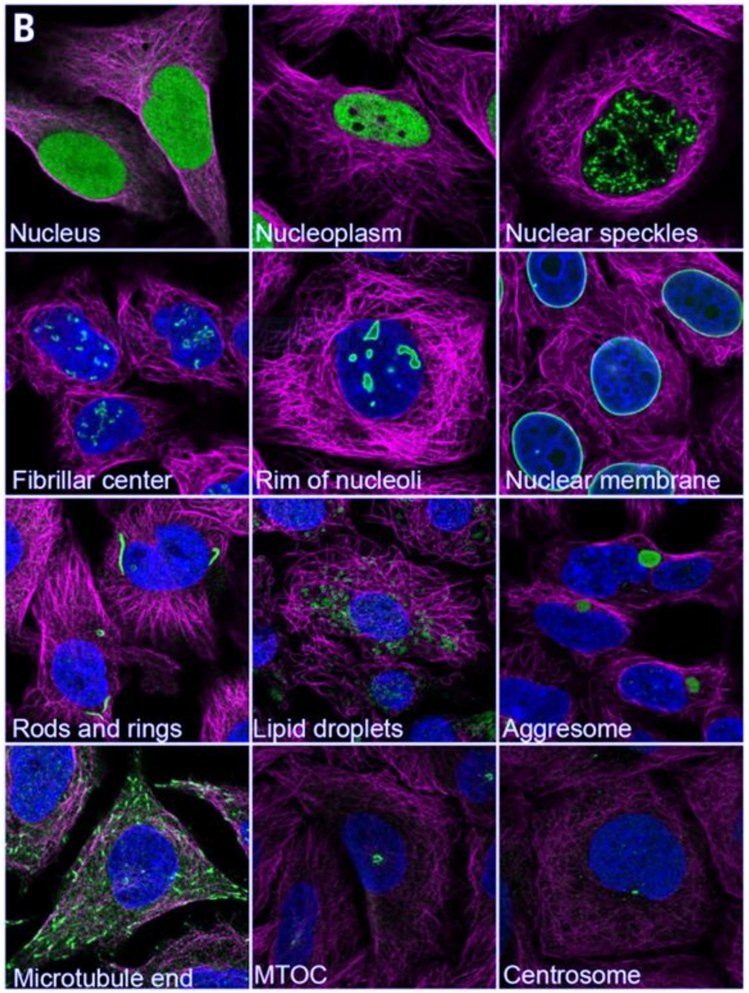
Determine the subcelluar location of all cellular proteins.

Experimental Methods:

- Antibody generation against 12.000 human proteins
- Immunostaining, 22 cell lines
- Automated confocal microscopy
- □ 82.152 images

Image Analysis Aim:

- Mapping 12.000 human proteins to 30 subcellular structures

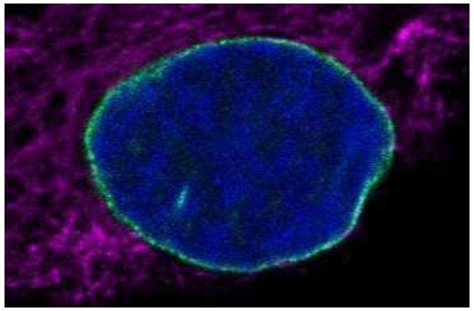


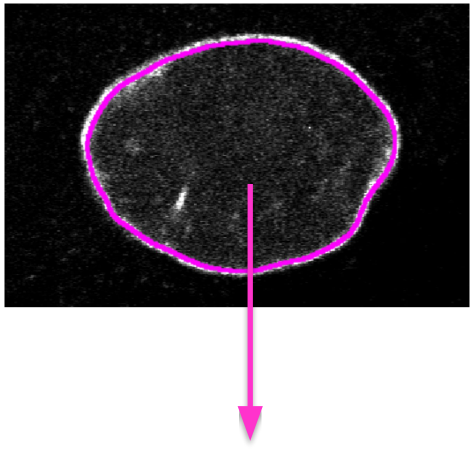
Adapted from Thul, P.J. et al. (2017). A subcellular map of the human proteome. Science *356*.

The Aim: Quantify Signal Accumulation Inside Nuclei Region

Image source: Human Protein Atlas

v19.proteinatlas.org/ENSG00000113368-LMNB1





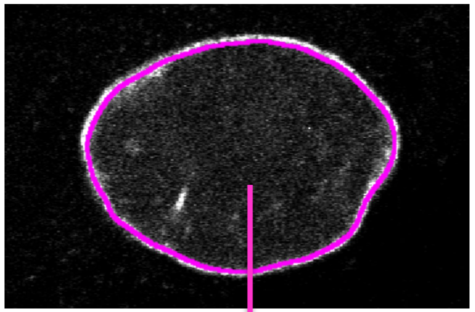
Mean intensity of green inside nucleus region

Dataset:

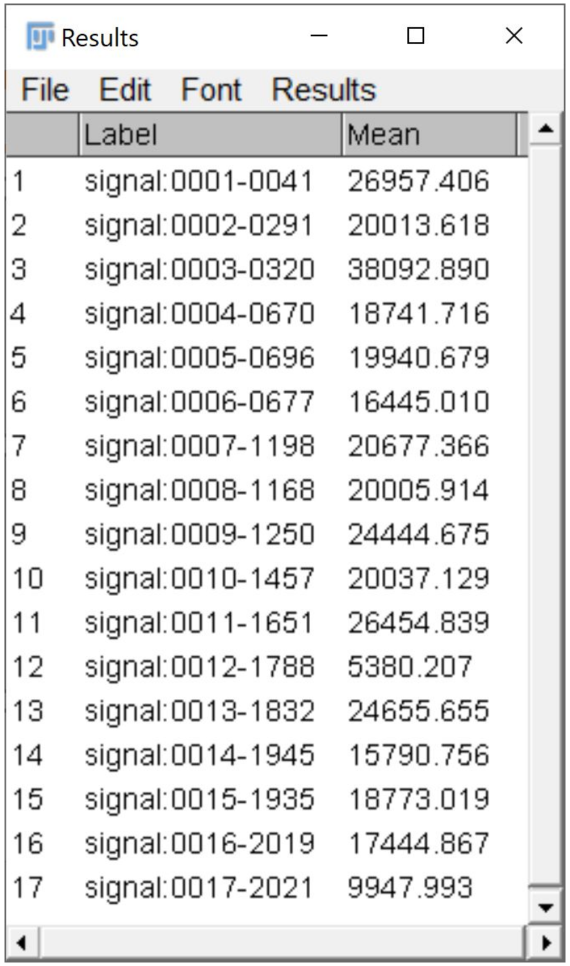
- Subset of The Cell Atlas (Human Protein Atlas)
- 3 color stack: microtubules (magenta), protein detected by antibody (green), nuclei (blue)

The Aim: Quantify Signal Accumulation Inside Nuclei Region

Image source: Human Protein Atlas
v19.proteinatlas.org/ENSG00000113368-LMNB1



Mean intensity of green inside nucleus region



	Label	Mean
1	signal:0001-0041	26957.406
2	signal:0002-0291	20013.618
3	signal:0003-0320	38092.890
4	signal:0004-0670	18741.716
5	signal:0005-0696	19940.679
6	signal:0006-0677	16445.010
7	signal:0007-1198	20677.366
8	signal:0008-1168	20005.914
9	signal:0009-1250	24444.675
10	signal:0010-1457	20037.129
11	signal:0011-1651	26454.839
12	signal:0012-1788	5380.207
13	signal:0013-1832	24655.655
14	signal:0014-1945	15790.756
15	signal:0015-1935	18773.019
16	signal:0016-2019	17444.867
17	signal:0017-2021	9947.993

Exercise: Find a Workflow

Task: Create selections (ROIs) around nuclei Create selections for each nucleus

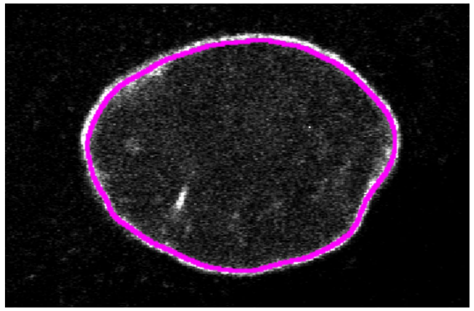
- Try to do some preprocessing first to get a smooth segmentation
- Add ROIs to ROI manager and analyze

Image:

- CellAtlas_Subset/711_D6_1.tif

Image source: Human Protein Atlas

v19.proteinatlas.org/ENSG00000113368-LMNB1



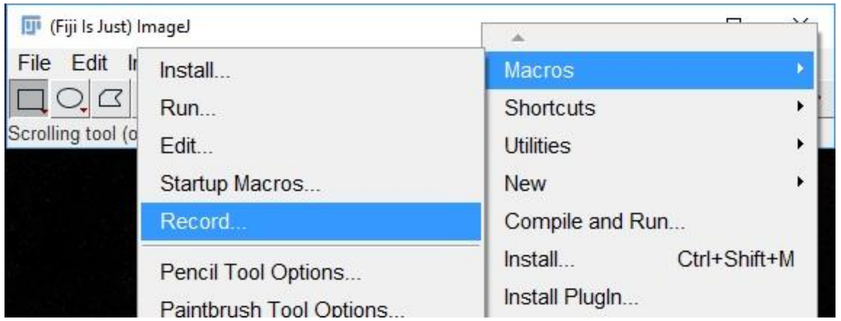
We now want to automate this process

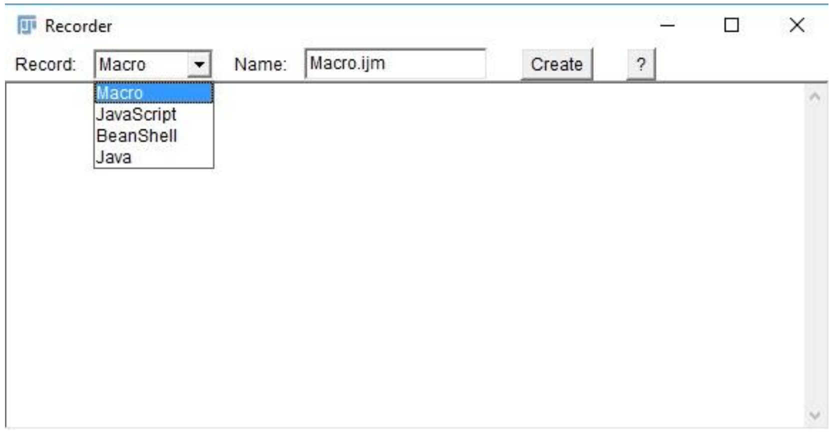
IMAGEJ MACRO LANGUAGE

Recorder

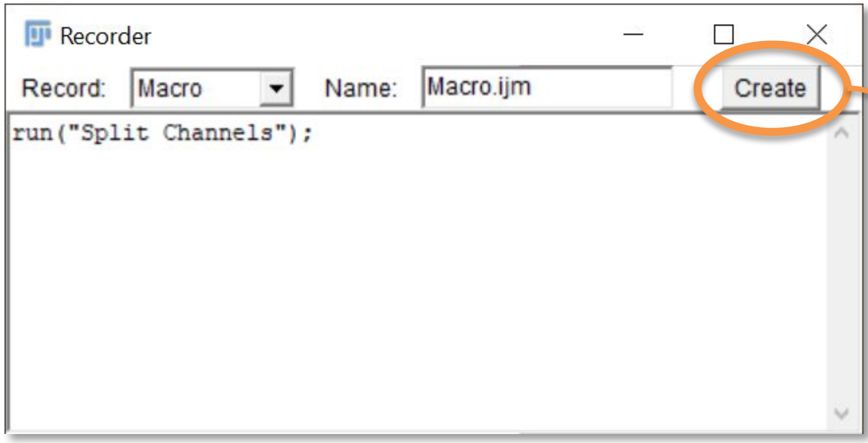
Open record window:

- Plugins > Macros > Record...
- Choose language – if needed



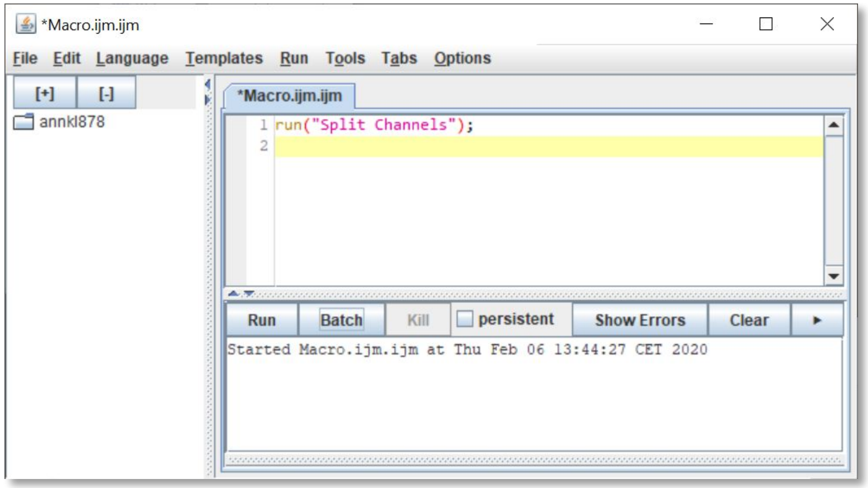


Let’s try with only one command: Split Channels



```
run("Split Channels");
```

Recorder


*Macro.ijm.ijm
File Edit Language Templates Run Tools Tabs Options
[+] [-]
annkl878
*Macro.ijm.ijm
Run Batch Kill persistent Show Errors Clear

```
run("Split Channels");
```

```
Started Macro.ijm.ijm at Thu Feb 06 13:44:27 CET 2020
```

Script Editor

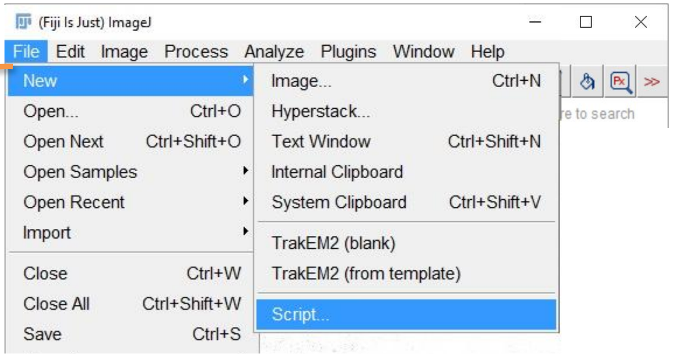
Recorder:

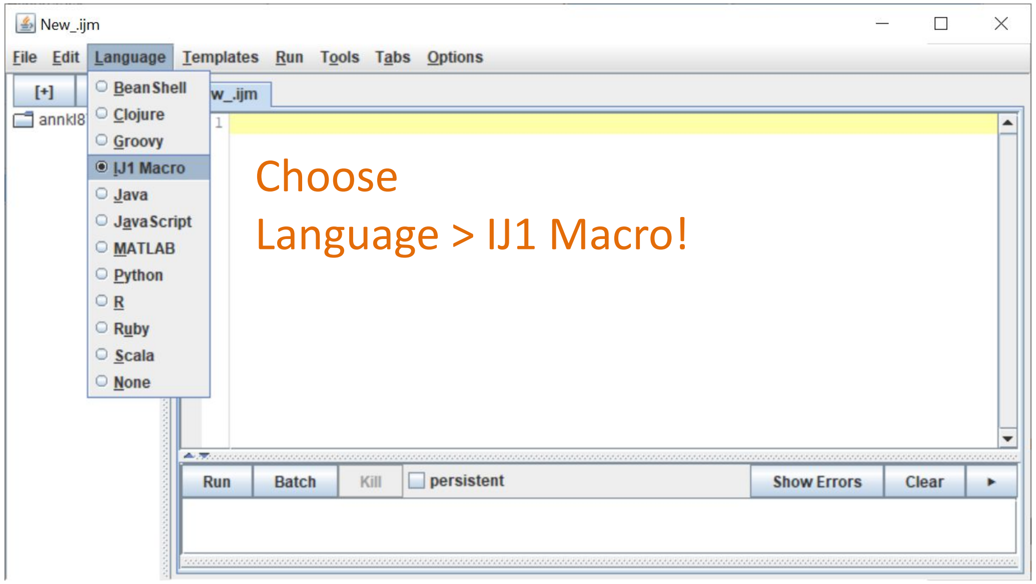
- Discover commands
- Window can be edited, copied, pasted, cut etc.

Script Editor:

- For building a script
- Has color-coding, code-completion, run-option etc.

How to start a new Script




New_.ijm
File Edit Language Templates Run Tools Tabs Options
BeanShell
Clojure
Groovy
IJ1 Macro
Java
JavaScript
MATLAB
Python
R
Ruby
Scala
None
Run Batch Kill persistent Show Errors Clear

Choose
Language > IJ1 Macro!

Importance of recorder

- Discover commands
- Get arguments for specific functions

But also:

- Record and save your workflow – for documentation and reproducibility!

Exercise: Record the Workflow

Task:

- Open the recorder and record the discussed workflow.
- Discuss open questions within the group.

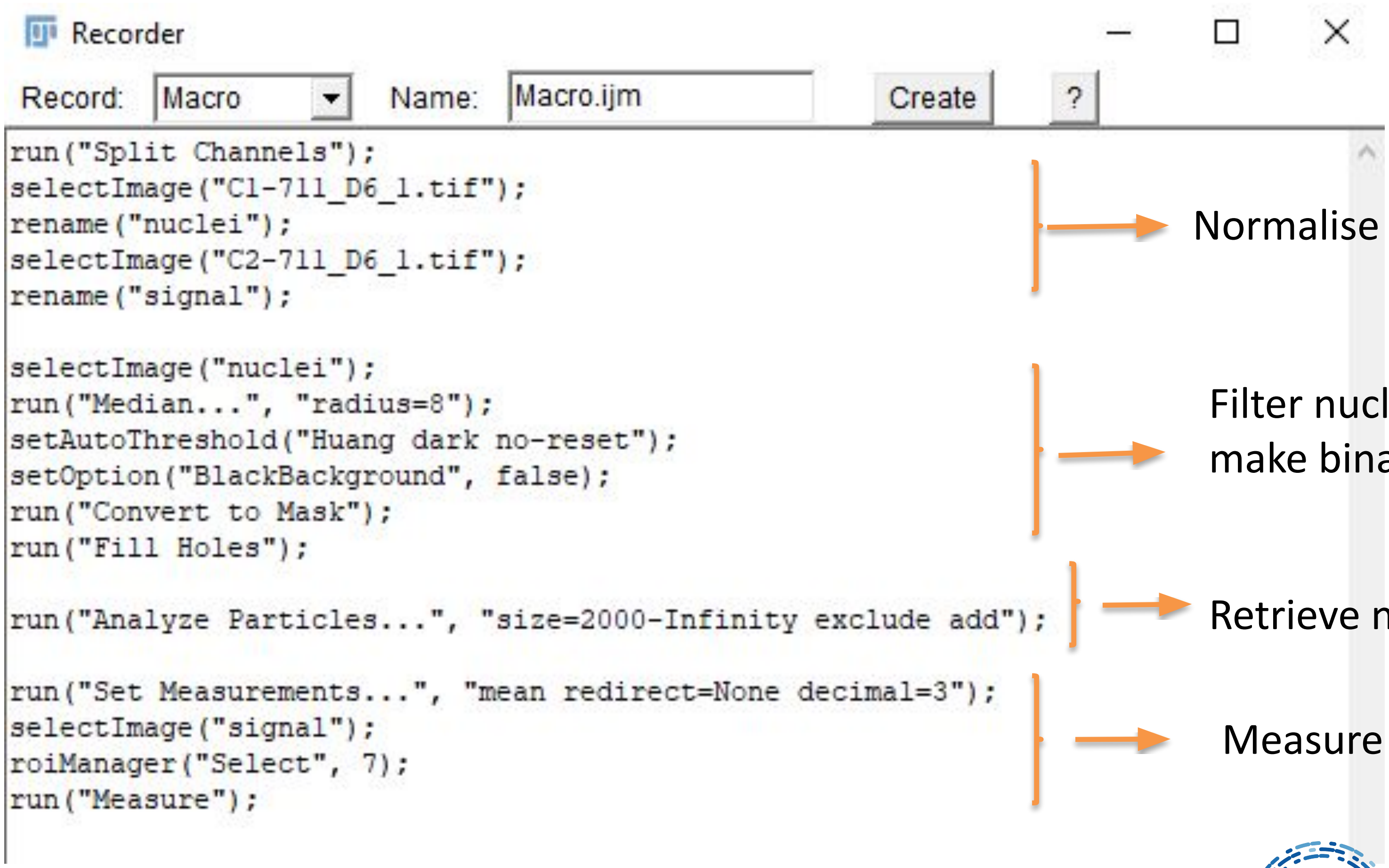
Image:

- Any image from CellAtlas_Subset

Exercise: Record the Workflow - Solution

Cleaned-up recorder:

Recorded.ijm



```
run("Split Channels");
selectImage("C1-711_D6_1.tif");
rename("nuclei");
selectImage("C2-711_D6_1.tif");
rename("signal");

selectImage("nuclei");
run("Median...", "radius=8");
setAutoThreshold("Huang dark no-reset");
setOption("BlackBackground", false);
run("Convert to Mask");
run("Fill Holes");

run("Analyze Particles...", "size=2000-Infinity exclude add");

run("Set Measurements...", "mean redirect=None decimal=3");
selectImage("signal");
roiManager("Select", 7);
run("Measure");
```

Normalise the data name

Filter nuclear image and make binary image

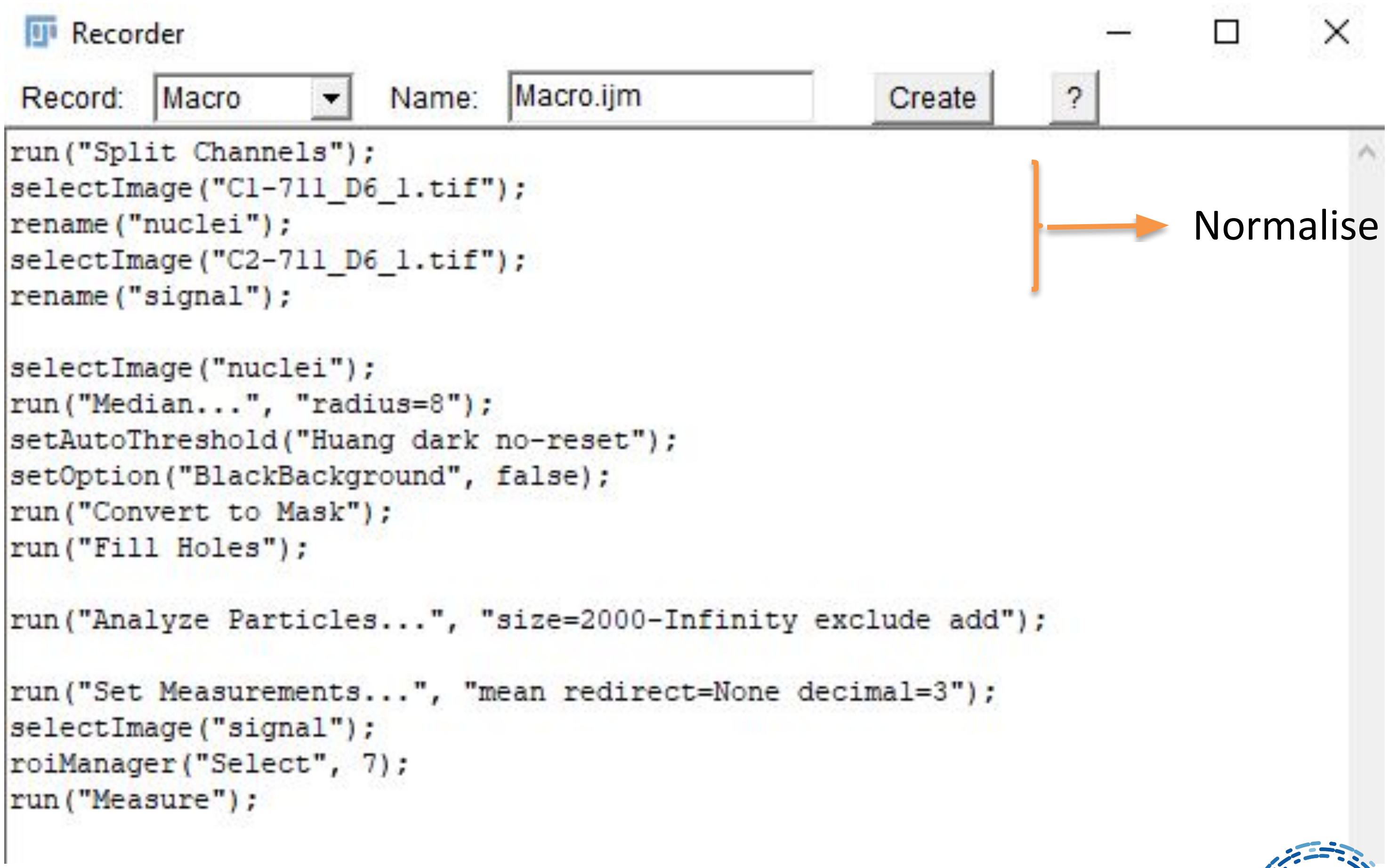
Retrieve nuclei boundaries

Measure

Exercise: Record the Workflow - Solution

Cleaned-up recorder:

Recorded.ijm



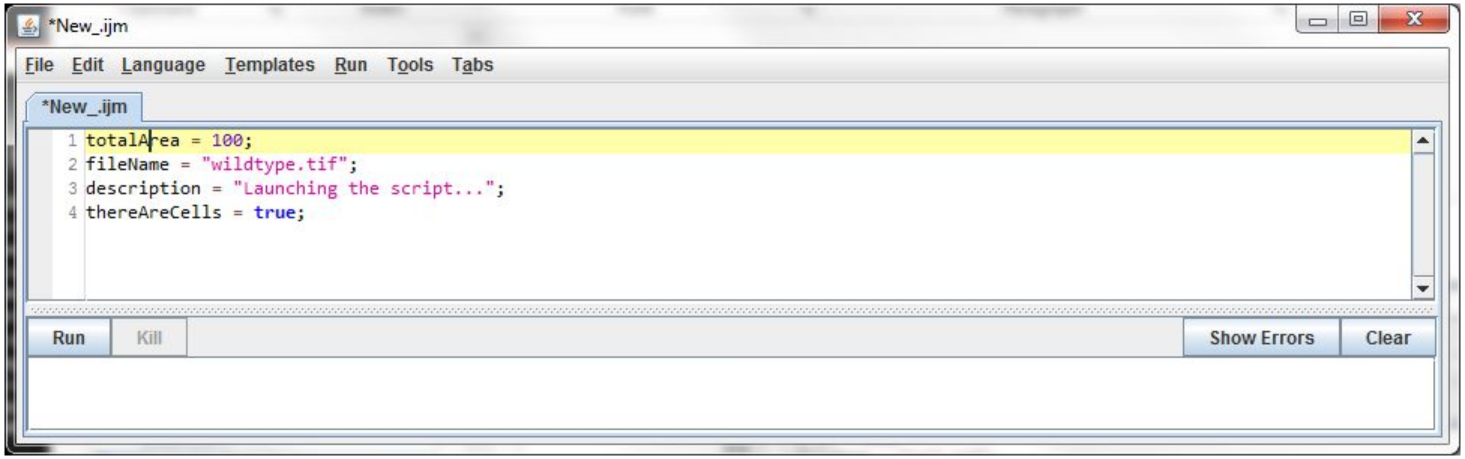
```
run("Split Channels");
selectImage("C1-711_D6_1.tif");
rename("nuclei");
selectImage("C2-711_D6_1.tif");
rename("signal");

selectImage("nuclei");
run("Median...", "radius=8");
setAutoThreshold("Huang dark no-reset");
setOption("BlackBackground", false);
run("Convert to Mask");
run("Fill Holes");

run("Analyze Particles...", "size=2000-Infinity exclude add");

run("Set Measurements...", "mean redirect=None decimal=3");
selectImage("signal");
roiManager("Select", 7);
run("Measure");
```

Normalise the data name

Programming Basics I

VARIABLES

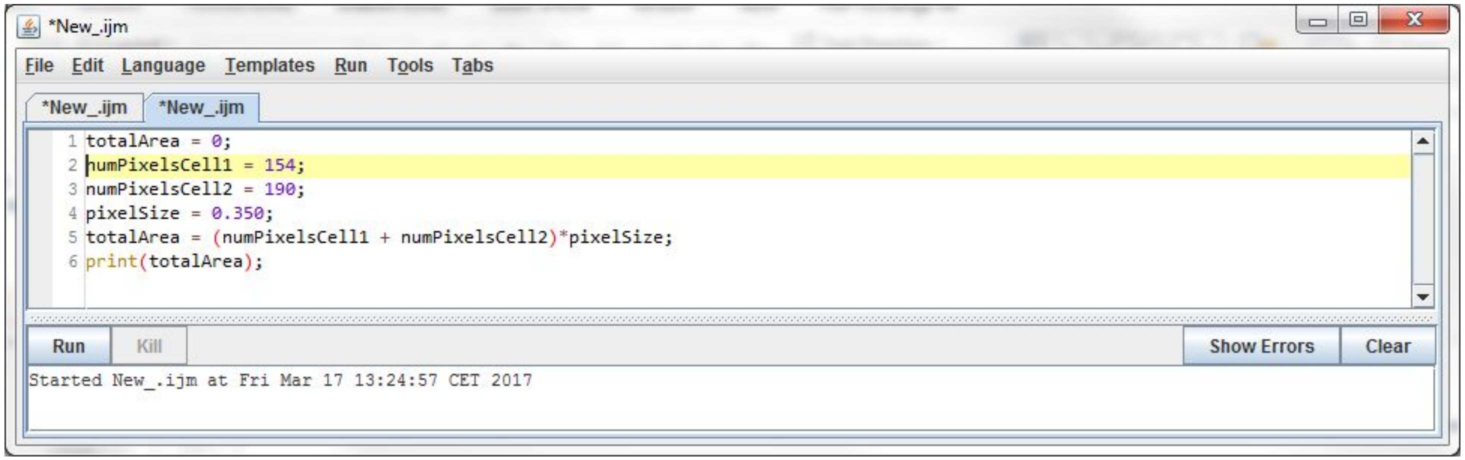
Variables: definition



```
totalArea = 100;
fileName = "wildtype.tif";
description = "Launching the script...";
thereAreCells = true;
```

- Can hold numbers or phrases/strings, but only one at a time
- Used whenever a value is used many times inside the script
- You define a variable by assigning it some content
- Variable name is on the left followed by an equal sign followed by the item (or items) being assigned
- Variable names can only start with characters

Numeric Variables: assignment



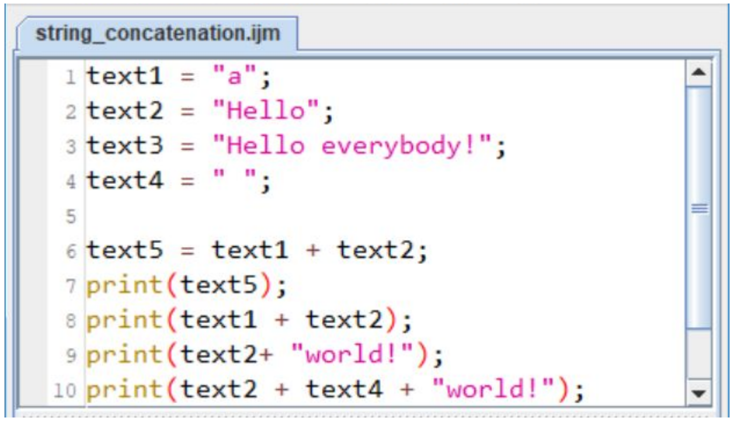
```
totalArea = 0;
numPixelsCell1 = 154;
numPixelsCell2 = 190;
pixelSize = 0.350;
totalArea = (numPixelsCell1 + numPixelsCell2)*pixelSize;
print(totalArea);
```

```
Started New_.ijm at Fri Mar 17 13:24:57 CET 2017
```

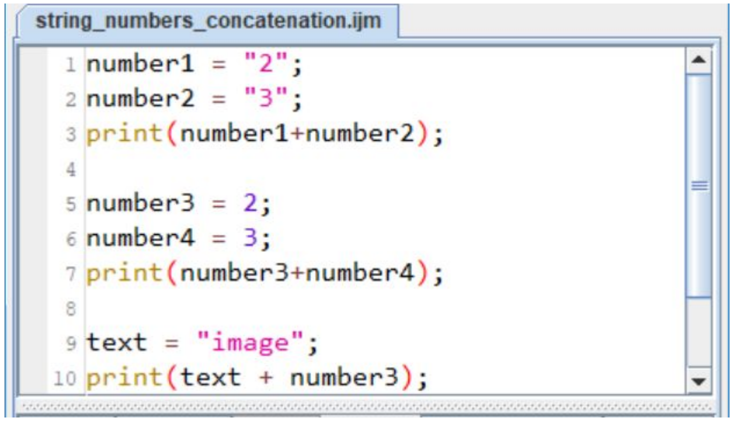
- Content of numeric variables can be modified using mathematical operations
- After an assignment, the previous content (if any) is forgotten
- Good practice is to use ***d2s*** (decimal to string) when printing numbers

    e.g. print(d2s(totalArea));

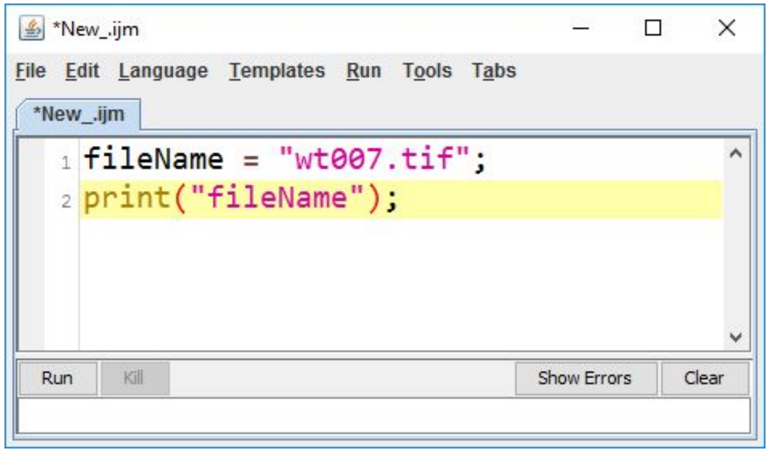
String Variables: concatenation



```
text1 = "a";
text2 = "Hello";
text3 = "Hello everybody!";
text4 = " ";

text5 = text1 + text2;
print(text5);
print(text1 + text2);
print(text2+ "world!");
print(text2 + text4 + "world!");
```



```
number1 = "2";
number2 = "3";
print(number1+number2);

number3 = 2;
number4 = 3;
print(number3+number4);

text = "image";
print(text + number3);
```



```
fileName = "wt007.tif";
print("fileName");
```

What about this?

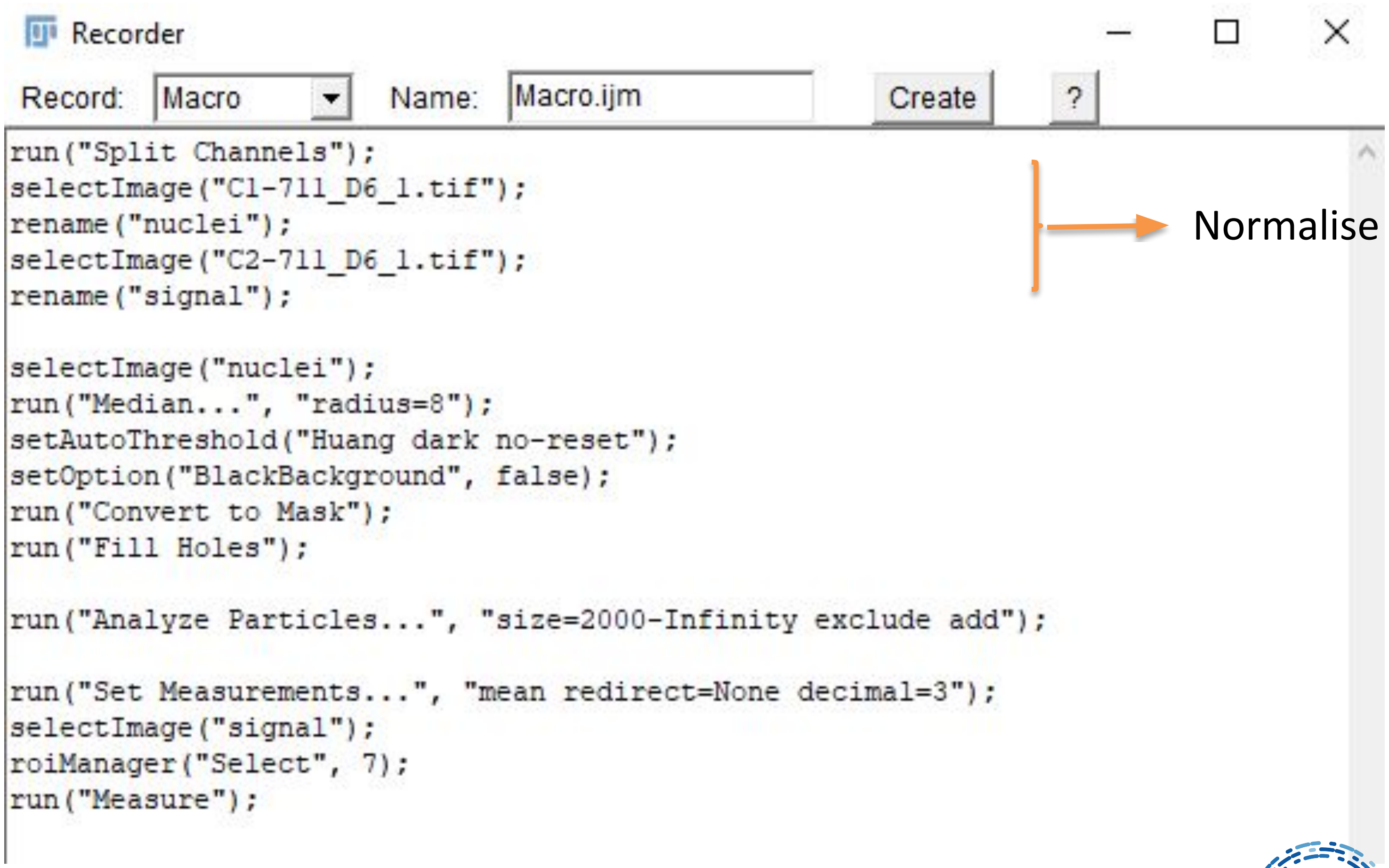
GloBIAS Global BioImage Analysts' Society

neubias network of european bioimage analysts

Exercise: Record the Workflow - Solution

Cleaned-up recorder:

Recorded.ijm



```
run("Split Channels");
selectImage("C1-711_D6_1.tif");
rename("nuclei");
selectImage("C2-711_D6_1.tif");
rename("signal");

selectImage("nuclei");
run("Median...", "radius=8");
setAutoThreshold("Huang dark no-reset");
setOption("BlackBackground", false);
run("Convert to Mask");
run("Fill Holes");

run("Analyze Particles...", "size=2000-Infinity exclude add");

run("Set Measurements...", "mean redirect=None decimal=3");
selectImage("signal");
roiManager("Select", 7);
run("Measure");
```

Normalise the data name

Introduction of the next Exercise

- Define a string variable with the name of the image
- Build `selectImage()` using the variables

```
run("Split Channels");

selectImage("C1-711_D6_1.tif");
rename("nuclei");

selectImage("C1-711_D6_1.tif");
rename("signal");
```

Exercise: Usage of Variables

- Replace the highlighted text using the variable *title.*
- Check out slide 19 (String Variables: concatenation) for help.
- **start with file:** Step_00_UsingVariables.ijm

```
title = "711_D6_1.tif";

run("Split Channels");
selectImage("C1-711_D6_1.tif");
rename("nuclei");

selectImage("C2-711_D6_1.tif");
rename("signal");
```

Solution: Usage of Variables

```
title = "711_D6_1.tif";

run("Split Channels");
selectImage("C1-" + title);
rename("nuclei");

selectImage("C2-" + title);
rename("signal");
```

Step_00_UsingVariables_Solution.ijm

Structuring the Code

Technical point:

Structuring the code using comments

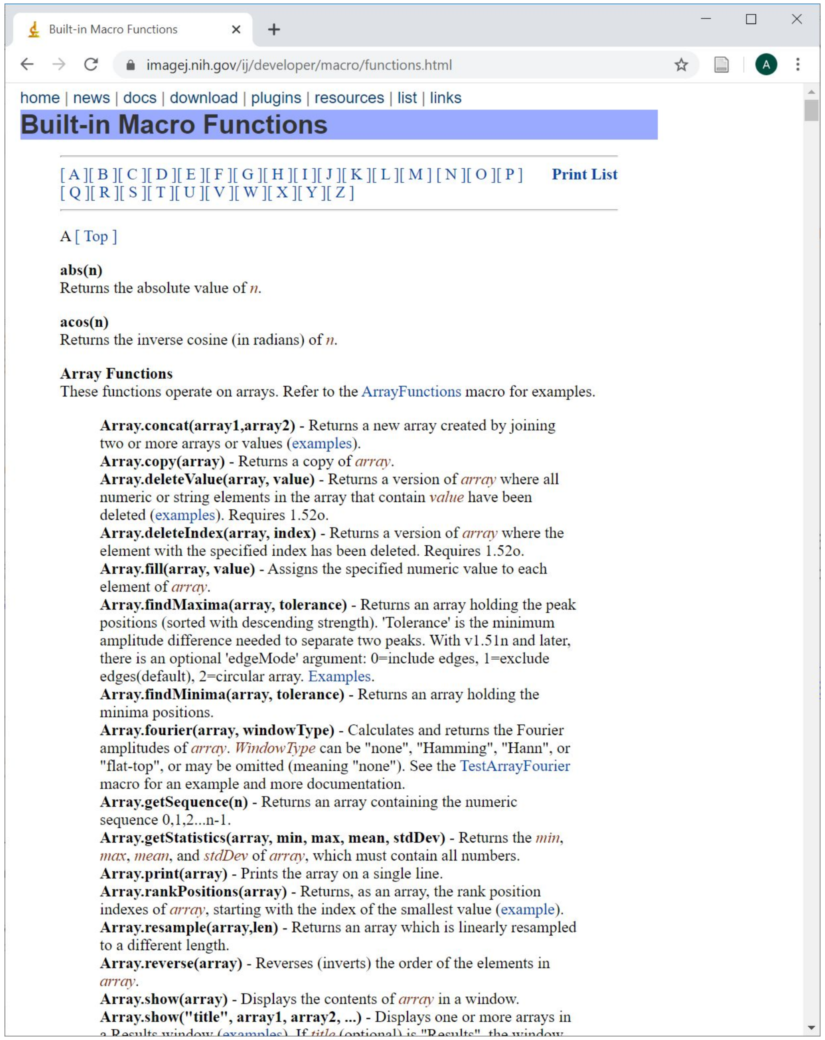
- Comments are non-interpreted elements of code
- They help structure the code
- They help collaborators interpret the original analyst's intentions
- Comments are introduced either by // of surrounded by /* */:

```
//This is a short comment

/*
	This is a very long comment, spanning over
multiple lines, allowing line breaks
*/
```

We now know what variables are but..

HOW TO GET THE NAME OF AN IMAGE AUTOMATICALLY?

Built-in Macro Functions



Built-in Macro Functions

imagej.nih.gov/ij/developer/macro/functions.html

home | news | docs | download | plugins | resources | list | links

Built-in Macro Functions

[A][B][C][D][E][F][G][H][I][J][K][L][M] [N][O][P] **Print List**
[Q][R][S][T][U][V][W][X][Y][Z]

A [Top]

abs(n)
Returns the absolute value of *n*.

acos(n)
Returns the inverse cosine (in radians) of *n*.

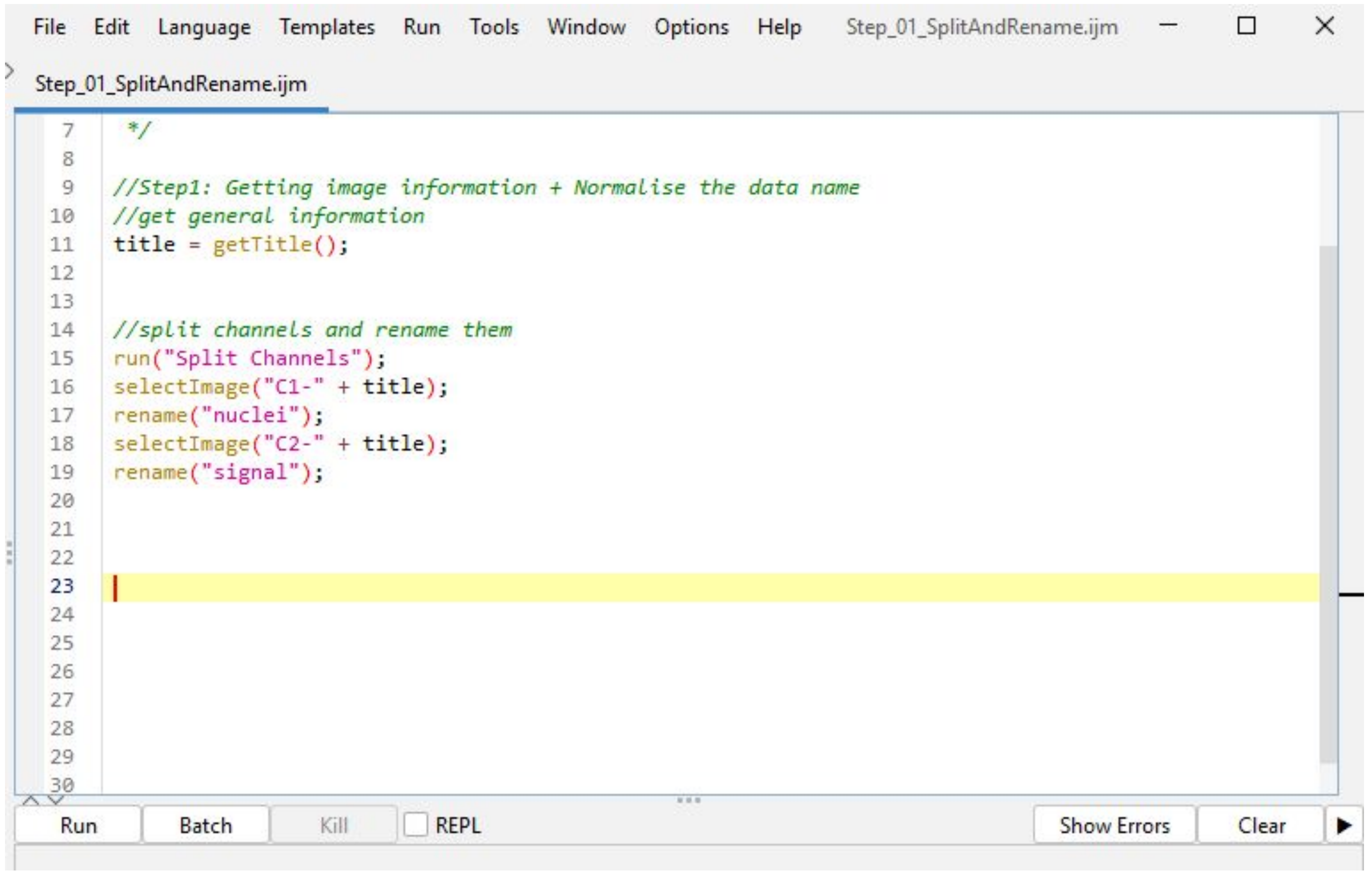
Array Functions
These functions operate on arrays. Refer to the ArrayFunctions macro for examples.

Array.concat(array1,array2) - Returns a new array created by joining two or more arrays or values (examples).
Array.copy(array) - Returns a copy of *array*.
Array.deleteValue(array, value) - Returns a version of *array* where all numeric or string elements in the array that contain *value* have been deleted (examples). Requires 1.52o.
Array.deleteIndex(array, index) - Returns a version of *array* where the element with the specified index has been deleted. Requires 1.52o.
Array.fill(array, value) - Assigns the specified numeric value to each element of *array*.
Array.findMaxima(array, tolerance) - Returns an array holding the peak positions (sorted with descending strength). 'Tolerance' is the minimum amplitude difference needed to separate two peaks. With v1.51n and later, there is an optional 'edgeMode' argument: 0=include edges, 1=exclude edges(default), 2=circular array. Examples.
Array.findMinima(array, tolerance) - Returns an array holding the minima positions.
Array.fourier(array, windowType) - Calculates and returns the Fourier amplitudes of *array*. *WindowType* can be "none", "Hamming", "Hann", or "flat-top", or may be omitted (meaning "none"). See the TestArrayFourier macro for an example and more documentation.
Array.getSequence(n) - Returns an array containing the numeric sequence 0,1,2...n-1.
Array.getStatistics(array, min, max, mean, stdDev) - Returns the *min*, *max*, *mean*, and *stdDev* of *array*, which must contain all numbers.
Array.print(array) - Prints the array on a single line.
Array.rankPositions(array) - Returns, as an array, the rank position indexes of *array*, starting with the index of the smallest value (example).
Array.resample(array,len) - Returns an array which is linearly resampled to a different length.
Array.reverse(array) - Reverses (inverts) the order of the elements in *array*.
Array.show(array) - Displays the contents of *array* in a window.
Array.show("title", array1, array2, ...) - Displays one or more arrays in

- Not everything is recorded. Much more functions can be found at: https://imagej.nih.gov/ij/developer/macro/functions.html

- Tip: do a page-search (CTRL+F)

Step 1:
Normalise the data name


File Edit Language Templates Run Tools Window Options Help Step_01_SplitAndRename.ijm
Step_01_SplitAndRename.ijm
Run Batch Kill REPL Show Errors Clear

```
 */

//Step1: Getting image information + Normalise the data name
//get general information
title = getTitle();


//split channels and rename them
run("Split Channels");
selectImage("C1-" + title);
rename("nuclei");
selectImage("C2-" + title);
rename("signal");
```

Step_01_SplitAndRename.ijm

More about the Built-in Macro Functions

Stack.setChannel(1);	Function with input
getTitle();	Function with output;
nameOfMyImage = getTitle();	output is assigned to a variable

getDimensions(width, height, channels, slices, frames);
Output is assigned to variables within the brackets

GloBIAS Global BioImage Analysts' Society

neubias network of european bioimage analysts

Exercise: Built-in Macro Functions

Task 1: Catch-up with the script

- include the getTitle() function.
- **start with file:** Step_00_UsingVariables_Solution.ijm

Task 2: explore the built-in macro functions.

- open a new script, set language to IJ1 macro
- What happens when you run **`getDimensions(channels, height, width, slices, frames)`**?
- Use the `print()` function to explore the content of the variables channels and width.
- Check the usage of the getDimensions function either using code autocompletion or on the "built-in macro function" website.

Solution: Built-in Macro Functions

Task 1: Catch-up with the script

- title = getTitle()
- solution file: Step_01_SplitAndRename.ijm

Task 2: explore the built-in macro functions.

- What happens when you run

```
getDimensions(channels, height, width, slices, frames)?
```

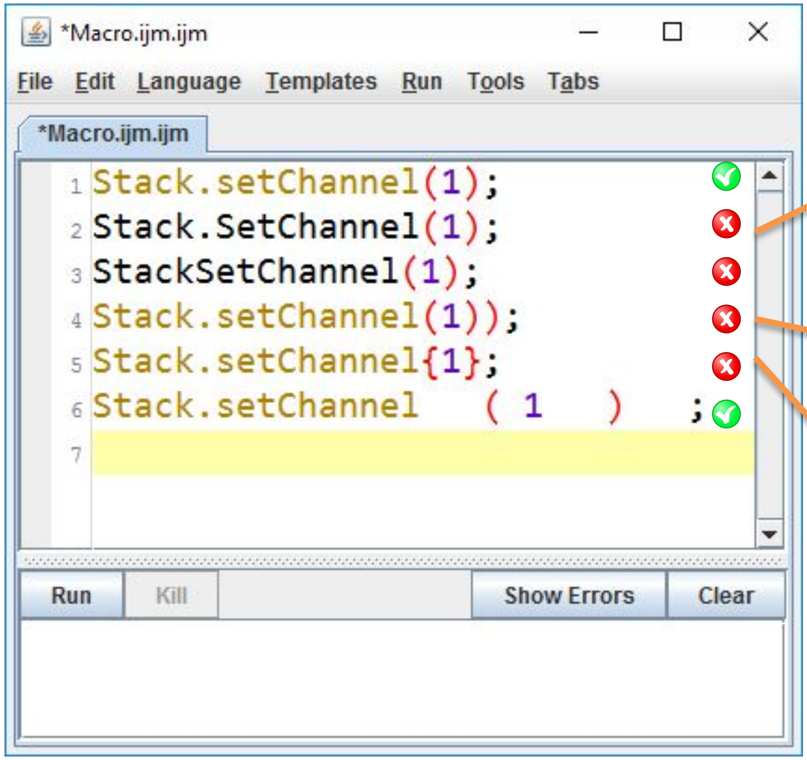
- correct usage:

  getDimensions(width, height, channels, slices, frames);

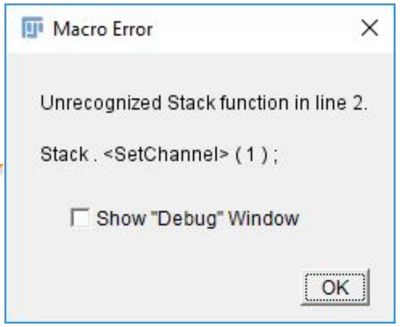
-

The script editor supports you with colors and error messages.

Color-coding in Script-Editor



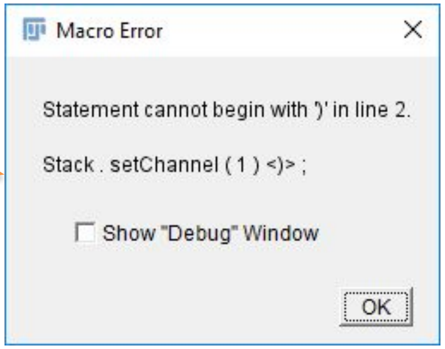
```
Stack.setChannel(1);
Stack.SetChannel(1);
StackSetChannel(1);
Stack.setChannel(1));
Stack.setChannel{1};
Stack.setChannel    ( 1    )    ;
```



Macro Error

Unrecognized Stack function in line 2.

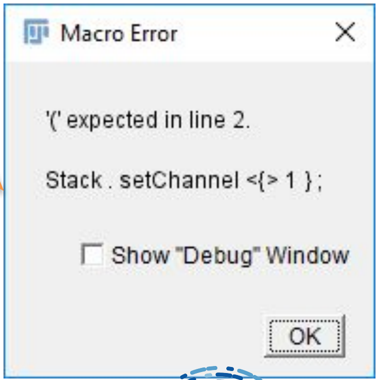
Stack . <SetChannel> (1) ;



Macro Error

Statement cannot begin with ')' in line 2.

Stack . setChannel (1) <)> ;



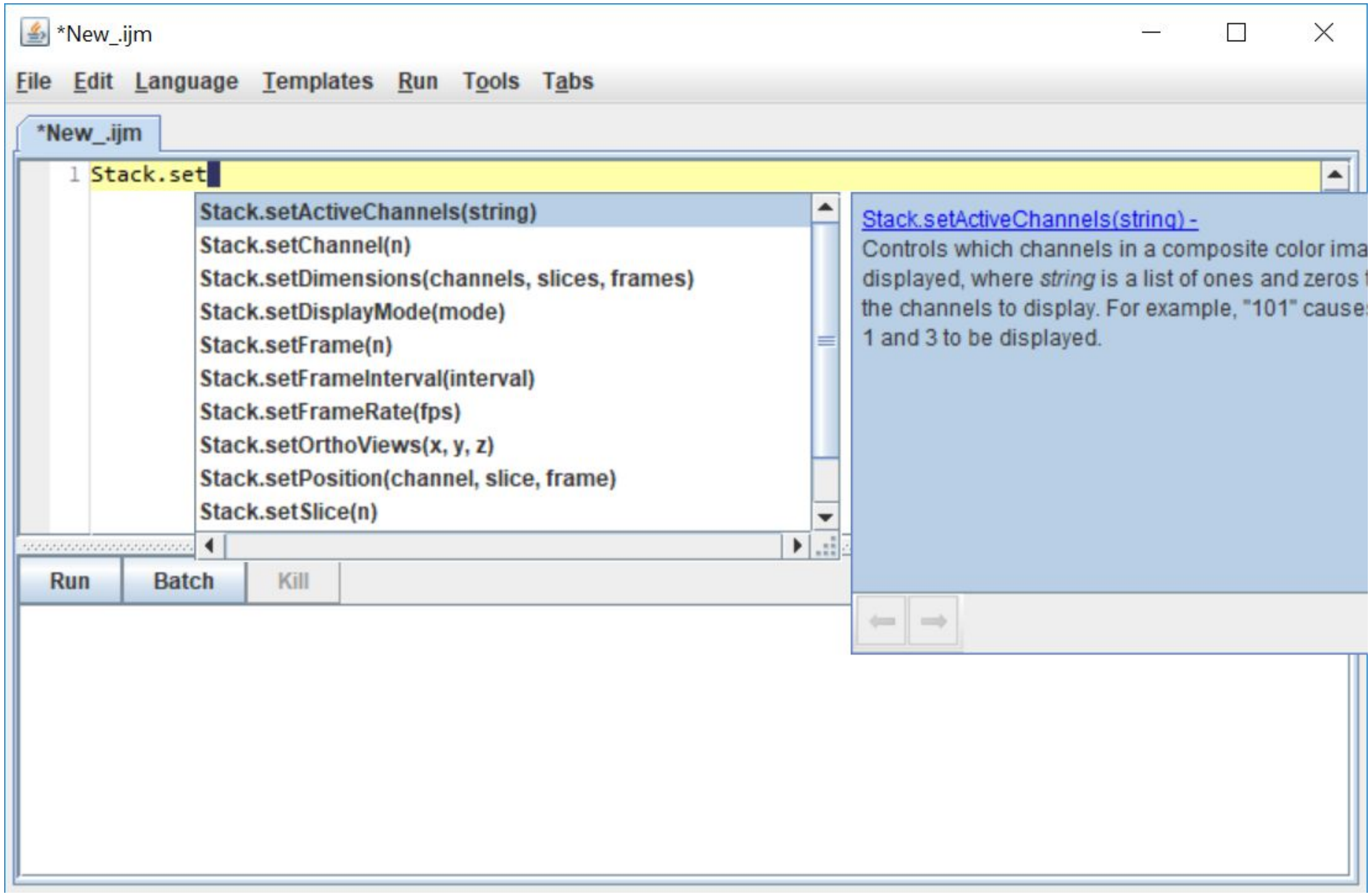
Macro Error

'(' expected in line 2.

Stack . setChannel <{> 1 } ;

Read the error messages!

Auto-completion helps to avoid common mistakes



```
Stack.set
```

Stack.setActiveChannels(string)
Stack.setChannel(n)
Stack.setDimensions(channels, slices, frames)
Stack.setDisplayMode(mode)
Stack.setFrame(n)
Stack.setFrameInterval(interval)
Stack.setFrameRate(fps)
Stack.setOrthoViews(x, y, z)
Stack.setPosition(channel, slice, frame)
Stack.setSlice(n)

Stack.setActiveChannels(string) -
Controls which channels in a composite color ima
displayed, where *string* is a list of ones and zeros t
the channels to display. For example, "101" causes
1 and 3 to be displayed.

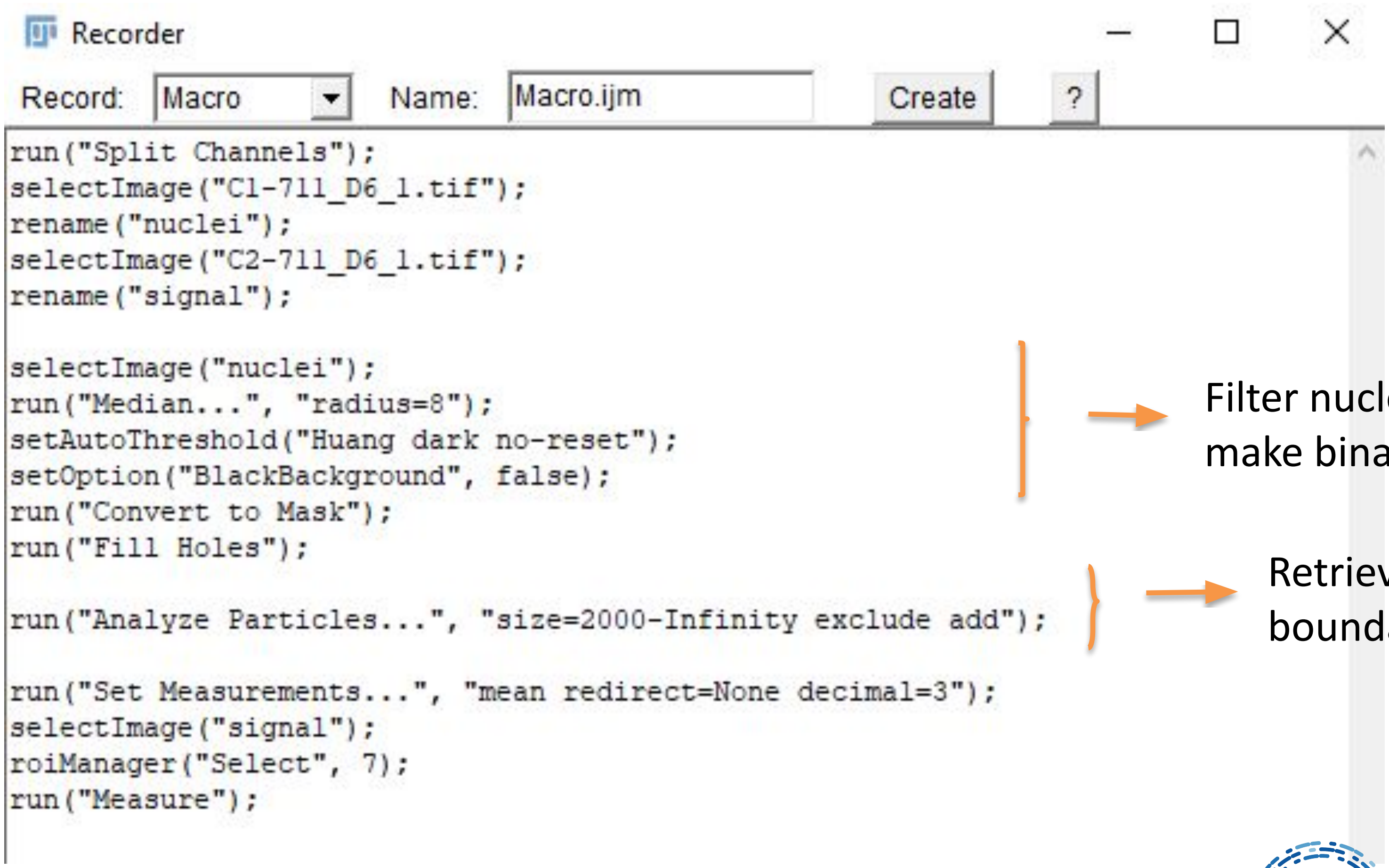
Workflow: How will we tackle the problem ?

Step 1	**Getting image information + Normalise the data name**	*Structuring the code, Recording basic operations, IJ macro functions' structure, Using simple variables*
Step 2	**Prefilter nuclear image and make binary image**	*Some useful shortcuts*
Step 3	**Retrieve the nuclei's boundaries**	*Using Analyze Particles*
Step 4	**Measure mean intensity and save the result**	*Saving of data; Extracting paths*
Step 5	**Do this for all images in the folders**	*For-Loops, templates*

The recorded workflow

Cleaned-up recorder:

Exercises/Recorded.ijm



```
run("Split Channels");
selectImage("C1-711_D6_1.tif");
rename("nuclei");
selectImage("C2-711_D6_1.tif");
rename("signal");

selectImage("nuclei");
run("Median...", "radius=8");
setAutoThreshold("Huang dark no-reset");
setOption("BlackBackground", false);
run("Convert to Mask");
run("Fill Holes");

run("Analyze Particles...", "size=2000-Infinity exclude add");

run("Set Measurements...", "mean redirect=None decimal=3");
selectImage("signal");
roiManager("Select", 7);
run("Measure");
```

Filter nuclear image and make binary image

Retrieve nuclei boundaries

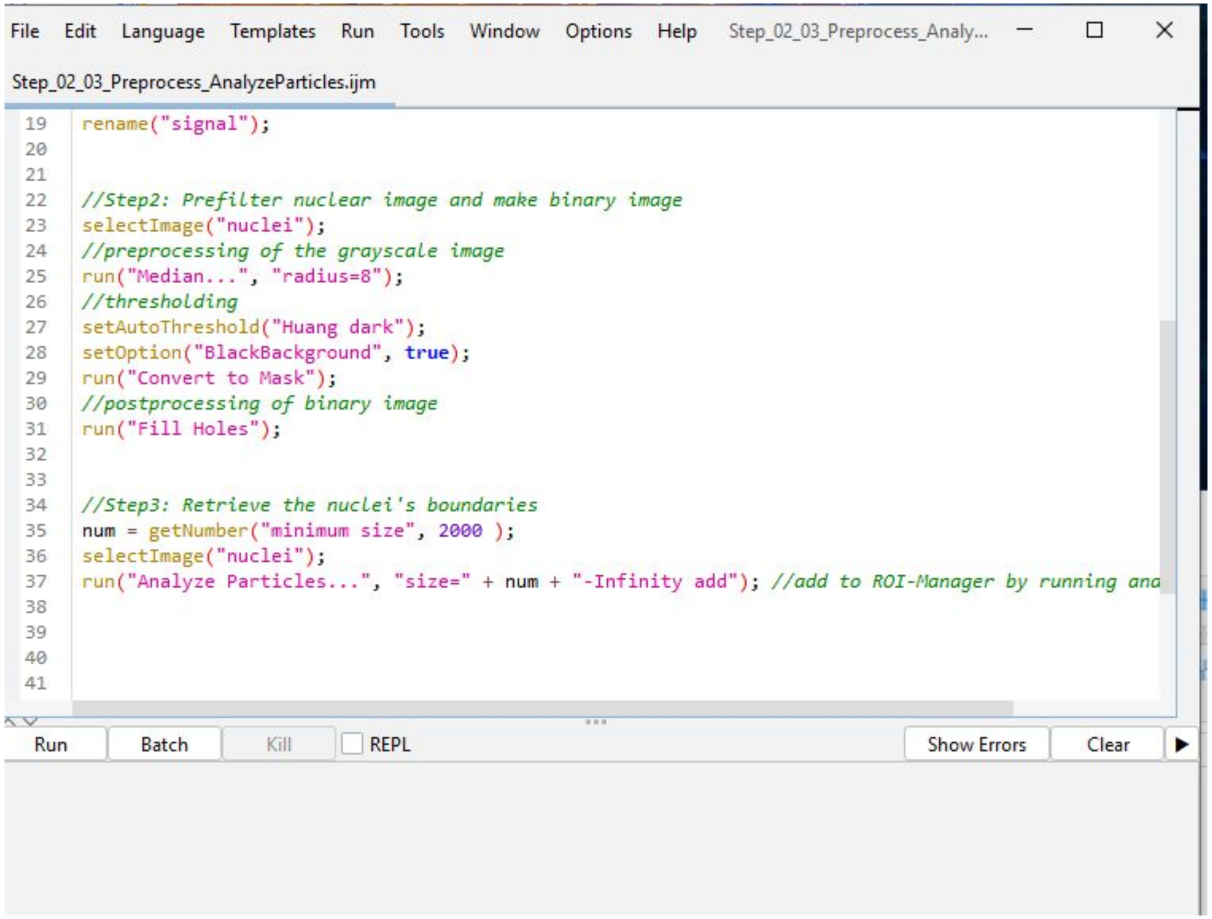
Exercise: Preprocessing, Asking for User-Input

Task:

- Insert the preprocessing and segmentation steps to your code (median filter, thresholding, fill holes)
- Use the `getNumber()` function to ask the user for a minimum size of the nuclei in pixel

- **start with file:** Step_01_SplitAndRename.ijm
- use recorded commands from Recorded.ijm

Solution: Preprocessing, Asking for User-Input


File Edit Language Templates Run Tools Window Options Help Step_02_03_Preprocess_Analy...
Step_02_03_Preprocess_AnalyzeParticles.ijm
Run
Batch
Kill
REPL
Show Errors
Clear

```
rename("signal");


//Step2: Prefilter nuclear image and make binary image
selectImage("nuclei");
//preprocessing of the grayscale image
run("Median...", "radius=8");
//thresholding
setAutoThreshold("Huang dark");
setOption("BlackBackground", true);
run("Convert to Mask");
//postprocessing of binary image
run("Fill Holes");


//Step3: Retrieve the nuclei's boundaries
num = getNumber("minimum size", 2000 );
selectImage("nuclei");
run("Analyze Particles...", "size=" + num + "-Infinity add"); //add to ROI-Manager by running ana
```

Step_02_03_Preprocess_AnalyzeParticles .ijm

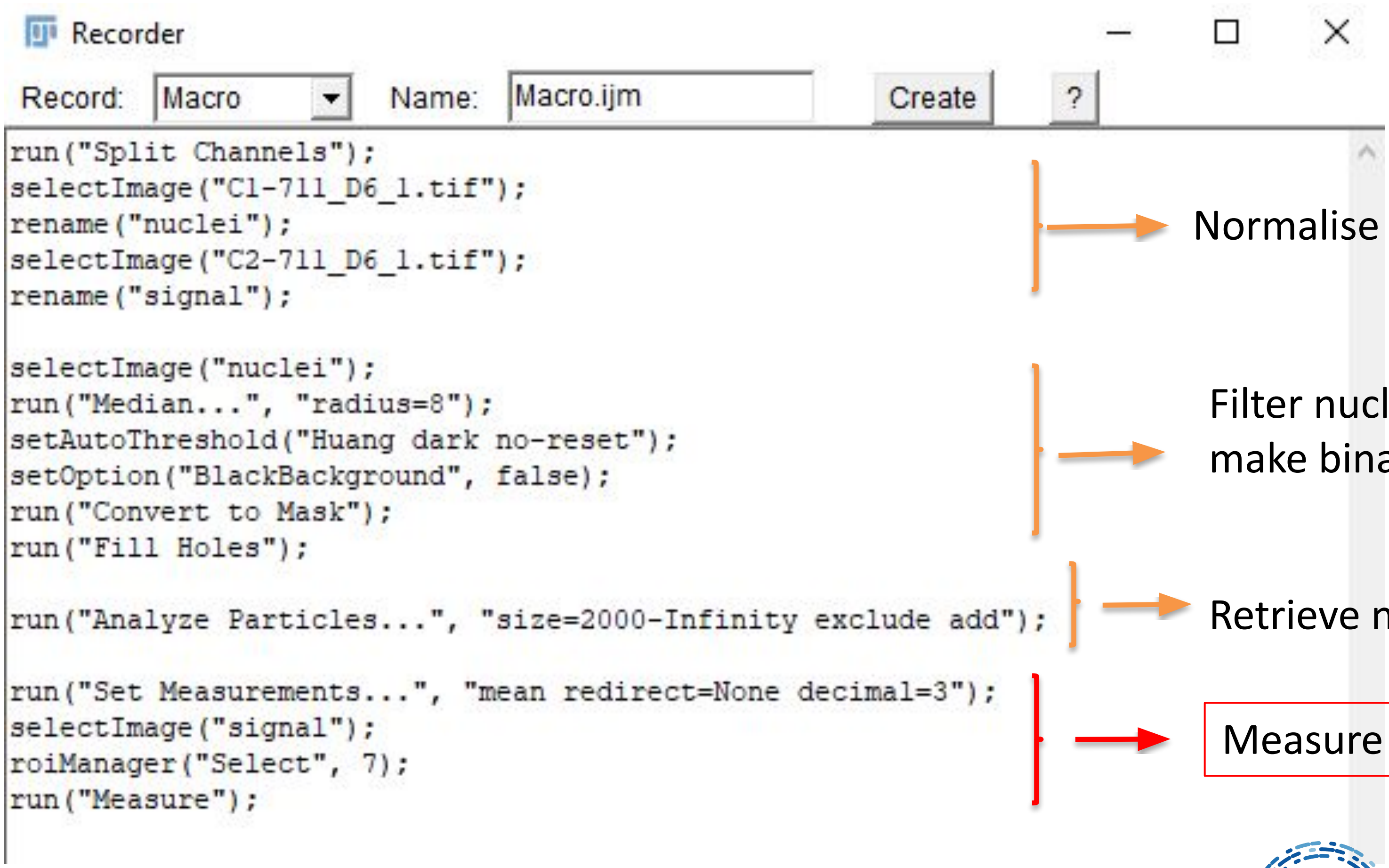
ImageJ Basics & Programming Basics

ROI MEASUREMENTS & RESULT SAVING

The recorded workflow

Cleaned-up recorder:

Recorded.ijm



```
run("Split Channels");
selectImage("C1-711_D6_1.tif");
rename("nuclei");
selectImage("C2-711_D6_1.tif");
rename("signal");

selectImage("nuclei");
run("Median...", "radius=8");
setAutoThreshold("Huang dark no-reset");
setOption("BlackBackground", false);
run("Convert to Mask");
run("Fill Holes");

run("Analyze Particles...", "size=2000-Infinity exclude add");

run("Set Measurements...", "mean redirect=None decimal=3");
selectImage("signal");
roiManager("Select", 7);
run("Measure");
```

Normalise the data name

Filter nuclear image and make binary image

Retrieve nuclei boundaries

Measure

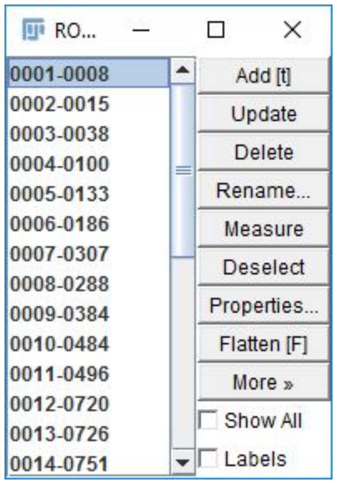
Step 5:

Make the measurement and save the result

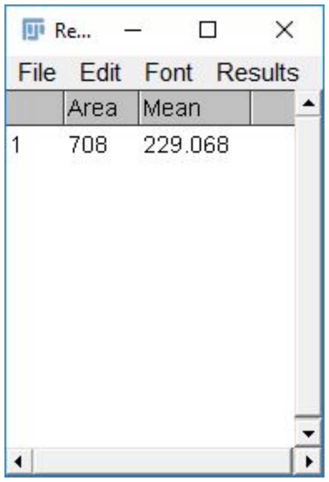
Hint:

If no ROI is selected in the ROI Manager, will measure **all ROIs**.

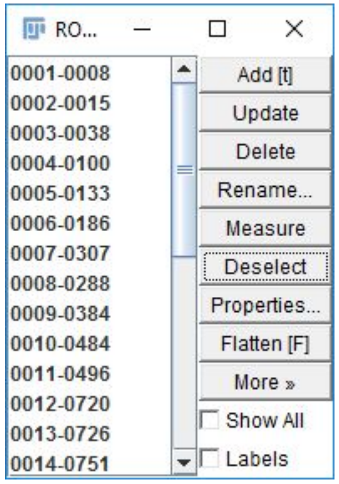
Commands: `roiManager("deselect")`and `roiManager("Measure")`



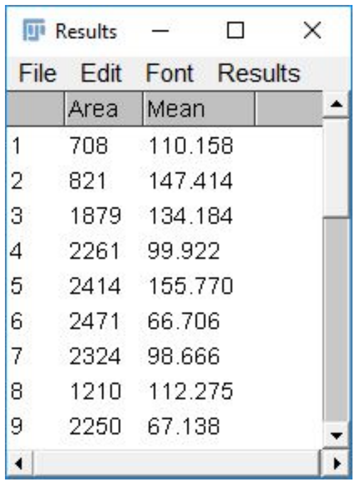
0001-0008
0002-0015
0003-0038
0004-0100
0005-0133
0006-0186
0007-0307
0008-0288
0009-0384
0010-0484
0011-0496
0012-0720
0013-0726
0014-0751



	Area	Mean
1	708	229.068



0001-0008
0002-0015
0003-0038
0004-0100
0005-0133
0006-0186
0007-0307
0008-0288
0009-0384
0010-0484
0011-0496
0012-0720
0013-0726
0014-0751

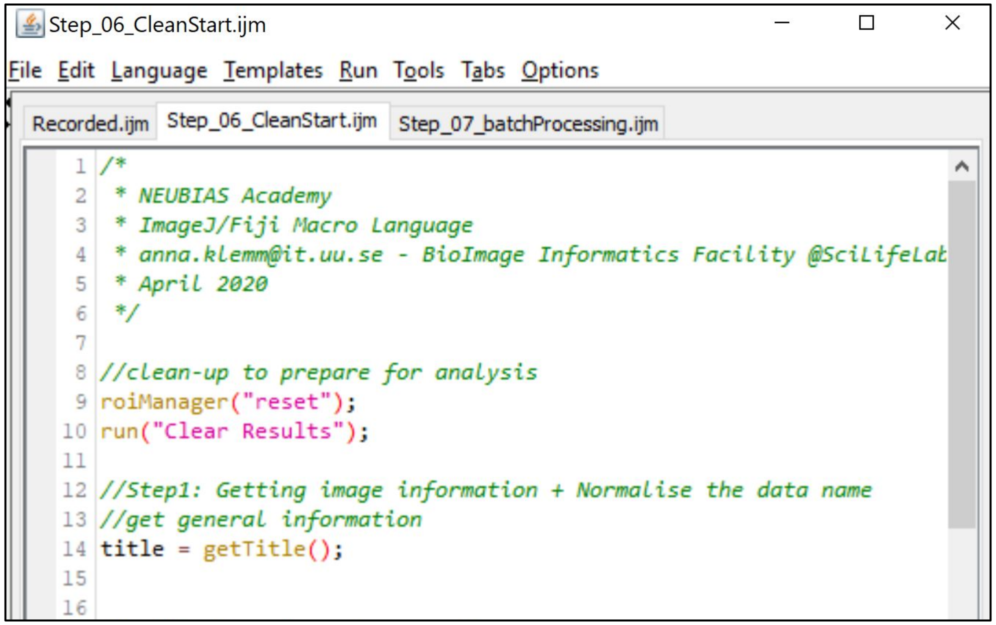


	Area	Mean
1	708	110.158
2	821	147.414
3	1879	134.184
4	2261	99.922
5	2414	155.770
6	2471	66.706
7	2324	98.666
8	1210	112.275
9	2250	67.138

Empty ROI Manager and Empty Results window

Make sure to start with an empty ROI-manager and an empty Results table!

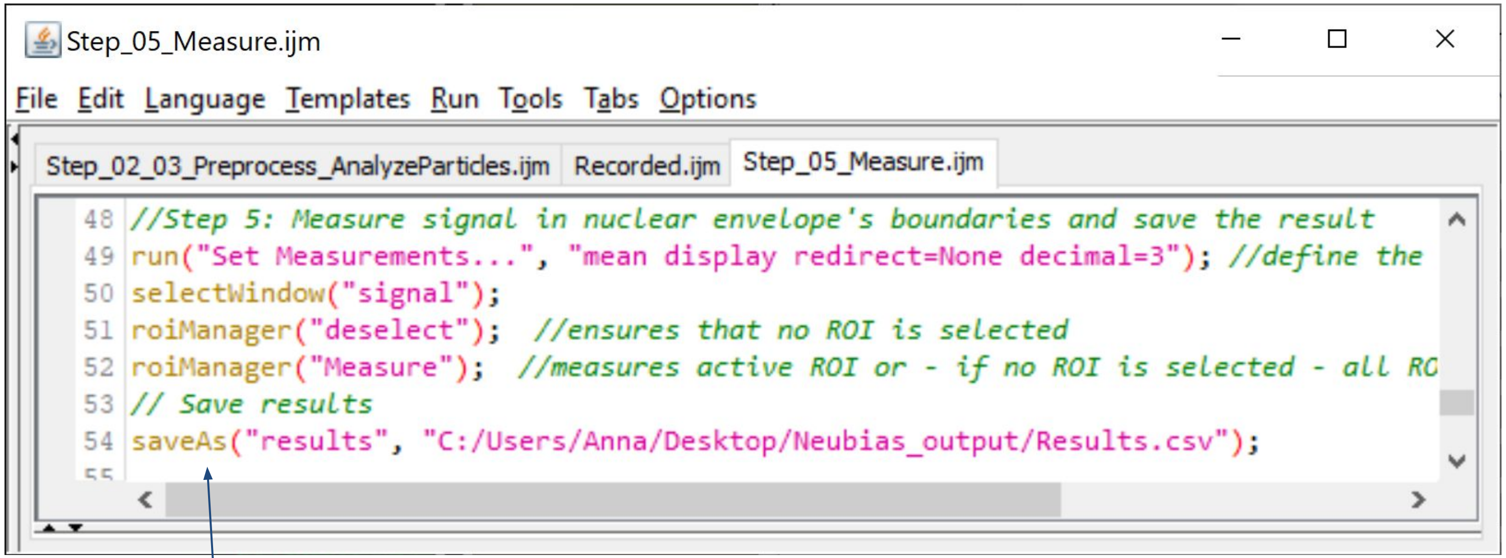
```
roiManager("reset");
run("Clear Results");
```


Step_06_CleanStart.ijm

File Edit Language Templates Run Tools Tabs Options
Recorded.ijm
Step_06_CleanStart.ijm
Step_07_batchProcessing.ijm

```
/*
 * NEUBIAS Academy
 * ImageJ/Fiji Macro Language
 * anna.klemm@it.uu.se - BioImage Informatics Facility @SciLifeLab
 * April 2020
 */

//clean-up to prepare for analysis
roiManager("reset");
run("Clear Results");

//Step1: Getting image information + Normalise the data name
//get general information
title = getTitle();


```

Result saving


Step_05_Measure.ijm
File Edit Language Templates Run Tools Tabs Options
Step_02_03_Preprocess_AnalyzeParticles.ijm
Recorded.ijm
Step_05_Measure.ijm

```
//Step 5: Measure signal in nuclear envelope's boundaries and save the result
run("Set Measurements...", "mean display redirect=None decimal=3"); //define the
selectWindow("signal");
roiManager("deselect");  //ensures that no ROI is selected
roiManager("Measure");  //measures active ROI or - if no ROI is selected - all RO
// Save results
saveAs("results", "C:/Users/Anna/Desktop/Neubias_output/Results.csv");
```

Macro record yourself saving the Results table!

Exercise: Make Measurements and Save Results

```
roiManager("reset");
run("Clear Results");
```

Tasks: Add to script:
- Clean up ROI Manager at the beginning
- Make measurements
- Save measurements to a file
- **Start with file:** Step_04_ForLoop.ijm

```
roiManager("deselect");
roiManager("Measure");
```

```
saveAs("results", ... );
```

Optional Tasks:
- Make output filename reflect image name
- Make output directory a variable
- Let user choose output directory with dialog

Step 5: Measure

Solution

Step_04_Measure.ijm

EXTRA-STEPS

Programming Basics

FOR-LOOPS and Batch processing

Programming Basics: Control Structures

Technical point

Loops

Definite loop

- Known number of iterations
- 3 arguments:
  - Initialisation
  - Condition for loop entry, as a boolean
  - Iteration

```
for(i=0; i<10; i++){
        //Instruction 1

        //Instruction 2

        //Instruction 3

}
```

Indefinite loops

A priori

- Test performed **BEFORE** instructions are executed

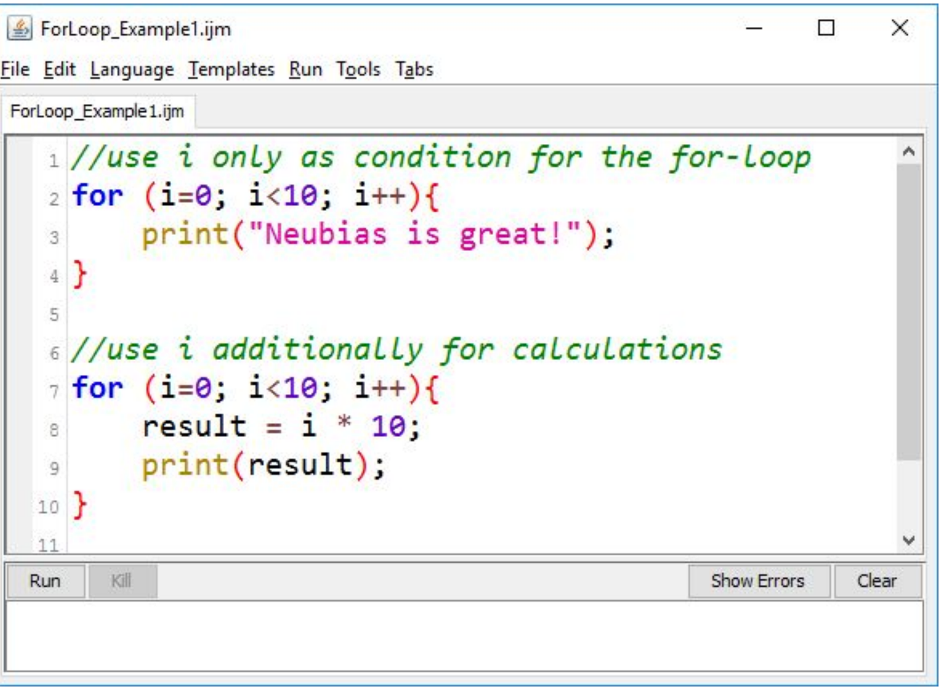
```
i=0;
while(i<10){
        //Instruction 1
        //Instruction 2
        //Instruction 3

        i++;
}
```

A posteriori

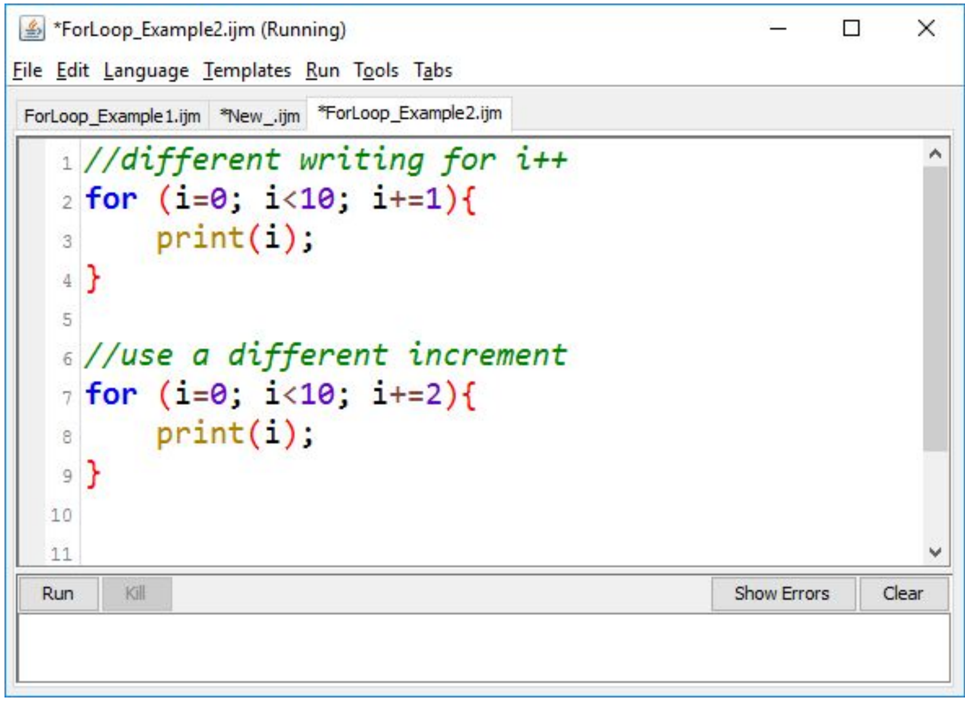
- Instructions always executed at least once
- Test performed **AFTER** instructions have been executed

```
i=0;
do{
        //Instruction 1
        //Instruction 2
        //Instruction 3

        i++;
} while(i<10)
```

Programming Basics: For-Loops


ForLoop_Example1.ijm
File Edit Language Templates Run Tools Tabs
ForLoop_Example1.ijm
Run Kill Show Errors Clear

```
//use i only as condition for the for-loop
for (i=0; i<10; i++){
    print("Neubias is great!");
}

//use i additionally for calculations
for (i=0; i<10; i++){
    result = i * 10;
    print(result);
}
```


*ForLoop_Example2.ijm (Running)
File Edit Language Templates Run Tools Tabs
ForLoop_Example1.ijm *New_.ijm *ForLoop_Example2.ijm
Run Kill Show Errors Clear

```
//different writing for i++
for (i=0; i<10; i+=1){
    print(i);
}

//use a different increment
for (i=0; i<10; i+=2){
    print(i);
}
```

ForLoop_Example1.ijm

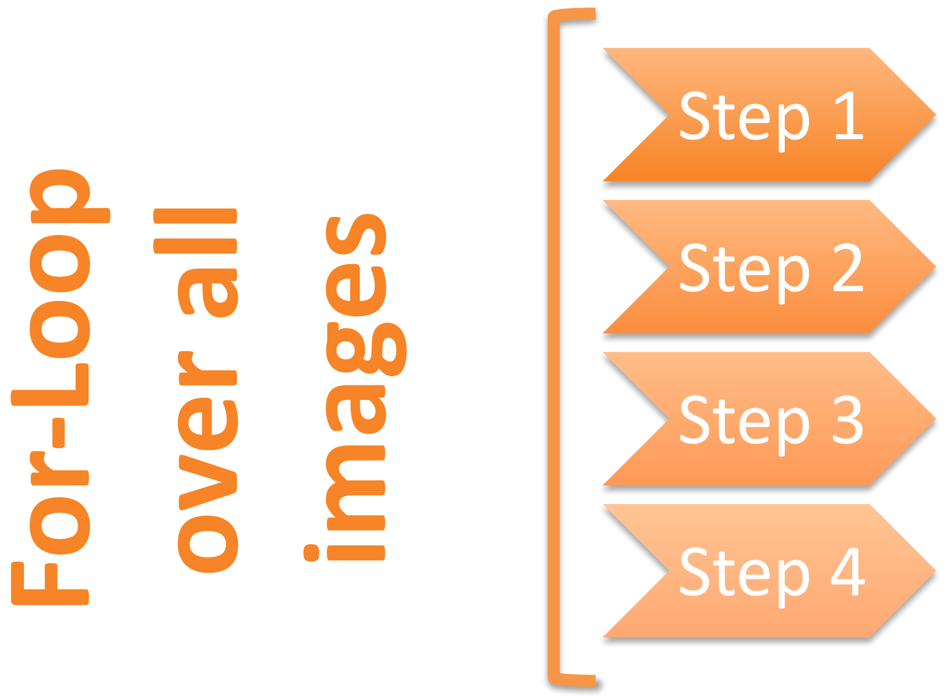
ForLoop_Example2.ijm

For-Loops Exercise (Type along)

- Get below code running in a new script
- Find (at least four) **different ways** to modify the code below to print „Hello!“ **only once** *instead of* 10 times.
- Modify below code to print 10 lines saying: “My favorite number is 0”, “My favorite number is 1”, ... , “My favorite number is 9”.

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

Extra I: Batch Processing



Exercise: Batch-processing

Task:

- Use the template below to loop your code over all files in the input folder.

```
input_path = getDirectory("input files");
fileList = getFileList(input_path);

for (f=0; f<fileList.length; f++){
     // Clean-up to prepare for next image
     roiManager("reset");
     run("Close All");
     run("Clear Results");
     // Open next image
     open(input_path + fileList[f]);
     print(input_path + fileList[f]);
     // Rest of the code
     // (…)
     saveAs("results", "C:/Users/Anna/Desktop/"+title+"_results.xls");
}
```

SOLUTION: Batch-processing

Step_05_batchProcessing.ijm

General Good Practice

Readability:

- Use meaningful variable names
- Assign variable at the top of a script if the variable is used widely, or as close to where it is used as possible
- Comment your code: for you and others

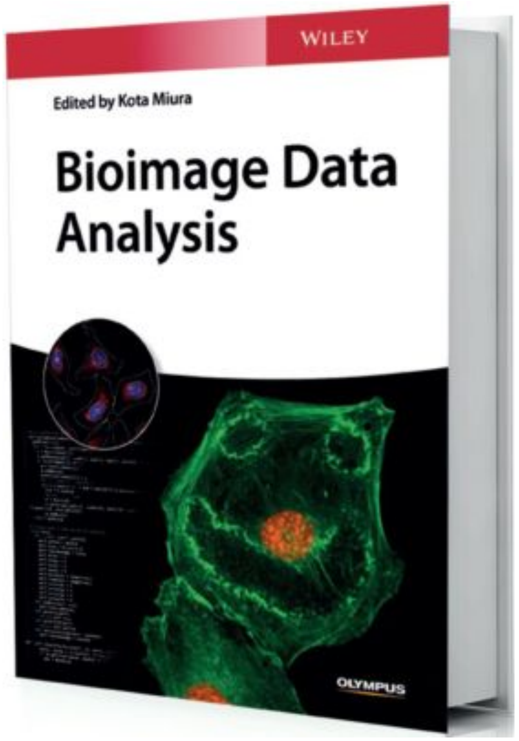
Performance:

- setBatchMode(true);

Reproducibility:

- Add Initialization code: close windows, reset roiManager, reset Results table …
- Save quality control files, e.g. save the ROI manager
- Use file names that refer to the original files
- Save the parameters used with the other results.
- Save the macro itself or document its version
- Consider sharing your macro and parameters as Supplementary Information

Where to continue



<u>Other resources:</u>

imagej.net/Introduction_into_Macro_Programming

forum.image.sc

www.springer.com/gp/book/9783030763930

Chapter 3, ImageJ Macro Language
(free download)

GloBIAS Global BioImage Analysts' Society

neubias network of european bioimage analysts

Resources connected to this course

Youtube video:
https://www.youtube.com/watch?v=o8tfkdcd3DA

Material:
https://github.com/ahklemm

Image.sc forum thread:
https://forum.image.sc/t/neubias-academy-home-interactive-course-imagej-fiji-macro-language-questions-answers/38678

Comparable workflow, but in CellProfiler:
https://www.youtube.com/watch?v=QrzHQIiIDKM

Data resource

Raw images (tif) were provided by The Human Protein Atlas.

https://www.proteinatlas.org/humanproteome/cell

Thul, P.J. et al. (2017). A subcellular map of the human proteome. Science *356*.

Acknowledgements

Material made by **Anna Klemm**
Based on NEUBIAS material,
especially Fiji Macro Writing by **Fabrice Cordelières**
with more material from **Giovanni Cardone** and **Ofra Golani**