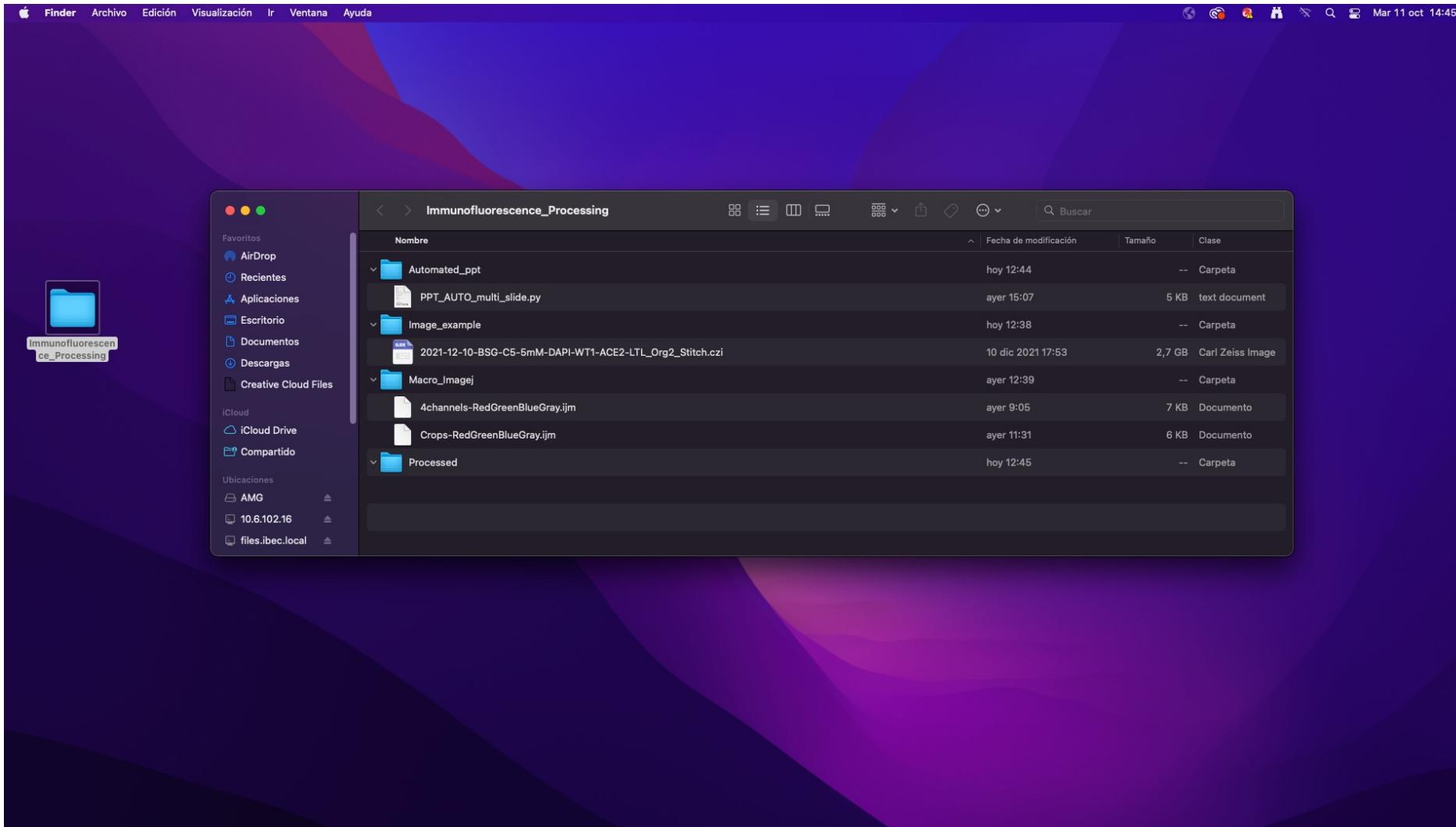
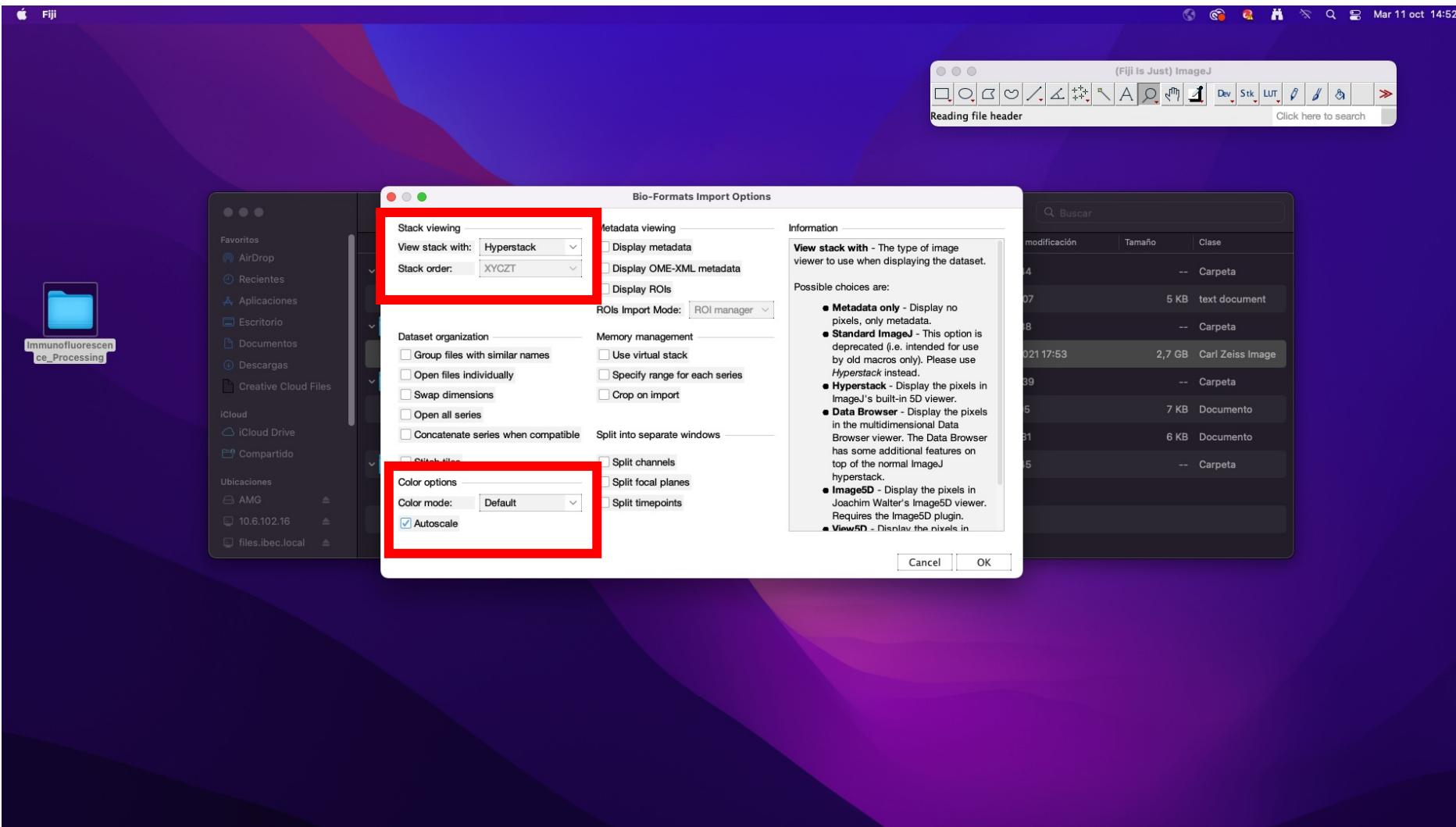


Files Organization



Full Image Processing

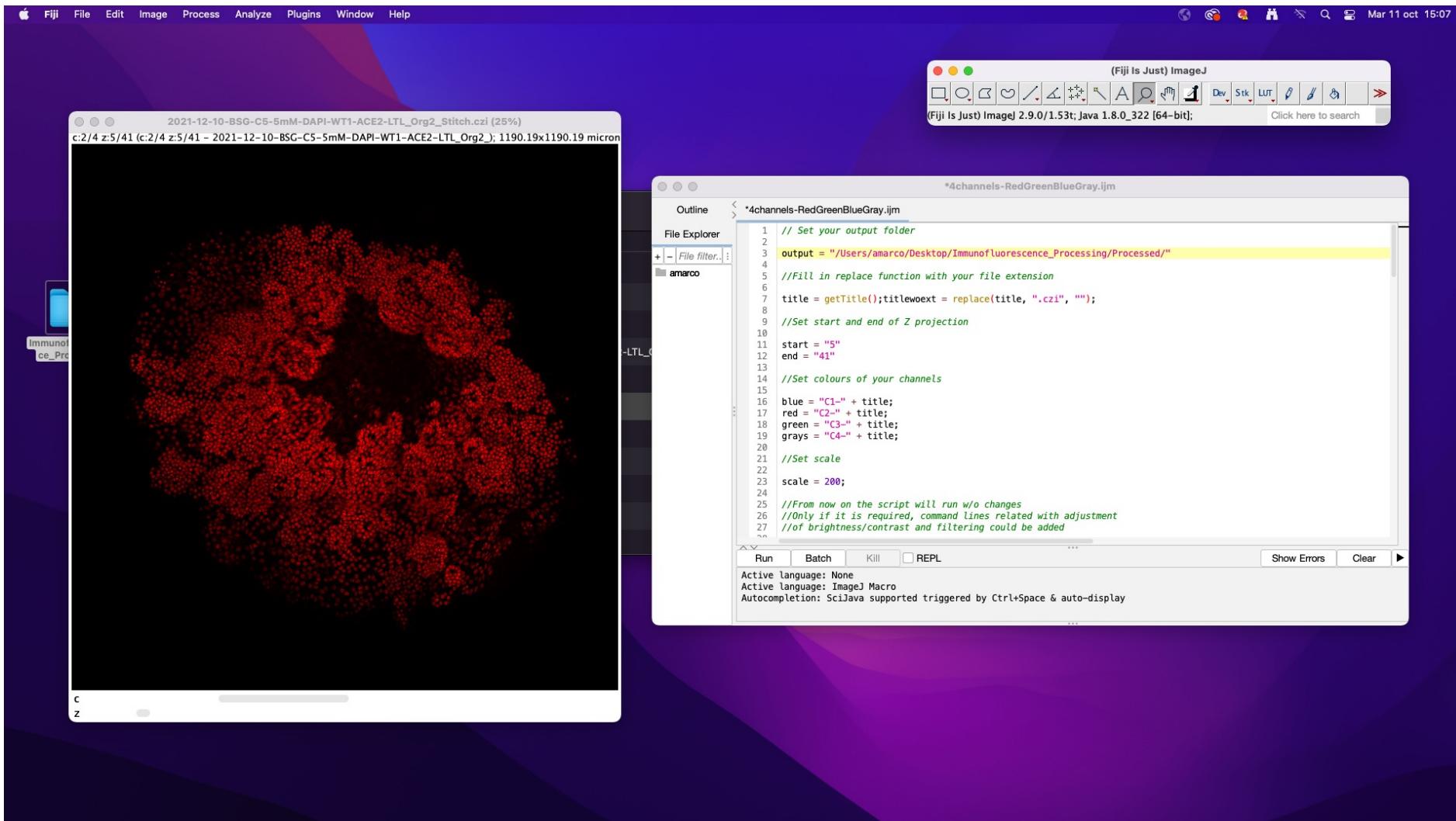
1. Open image example file by drag and drop in Fiji.



Full Image Processing

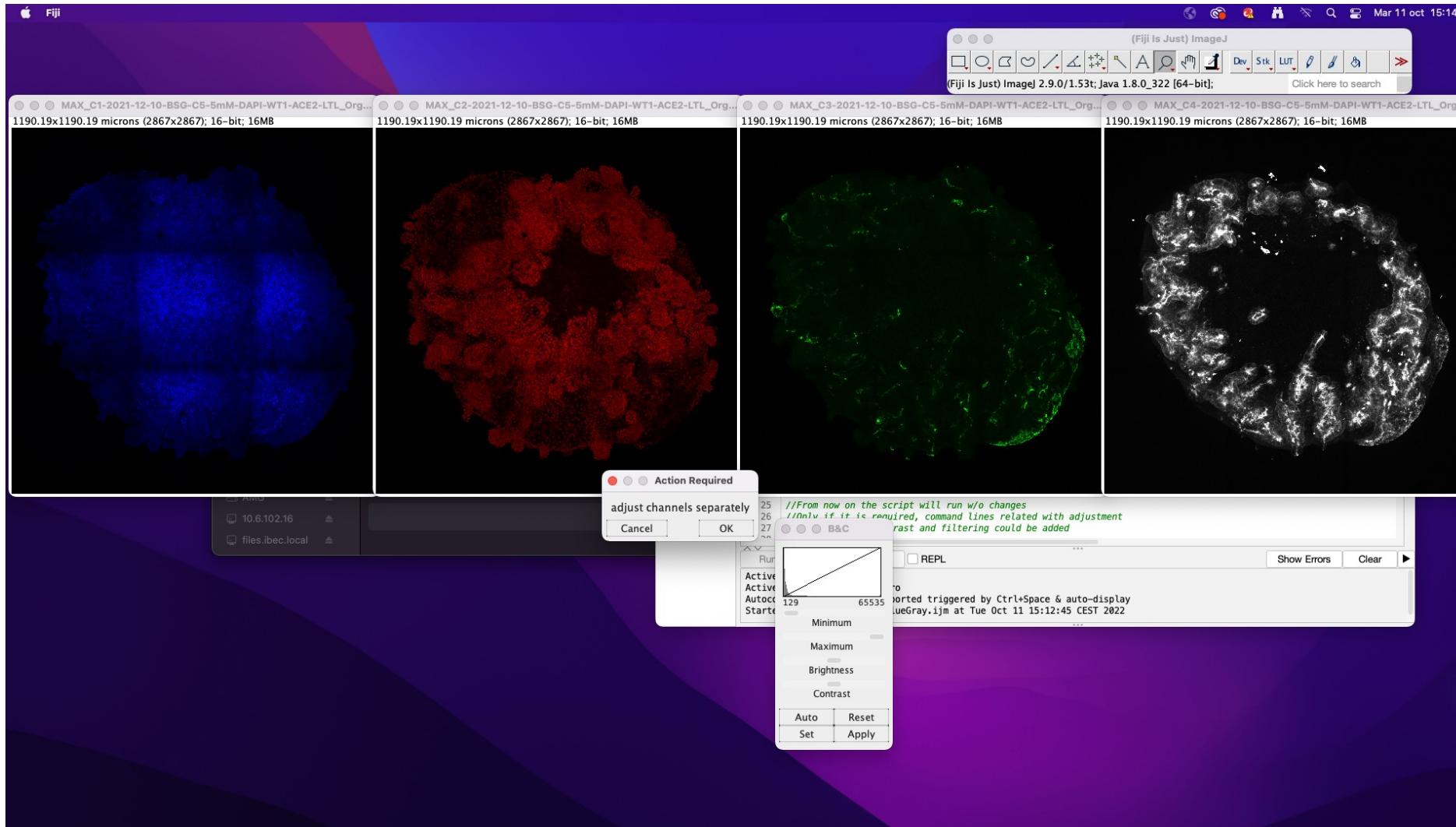
2. Open the **4channels-RedGreenBlueGray.ijm** by drag and drop in Fiji and fill it by following instructions (plotted in green)

Channel 1 ("C1-") Will be plotted in blue
Channel 2 ("C2-") Will be plotted in red
Channel 3 ("C3-") Will be plotted in green
Channel 3 ("C4-") Will be plotted in grays



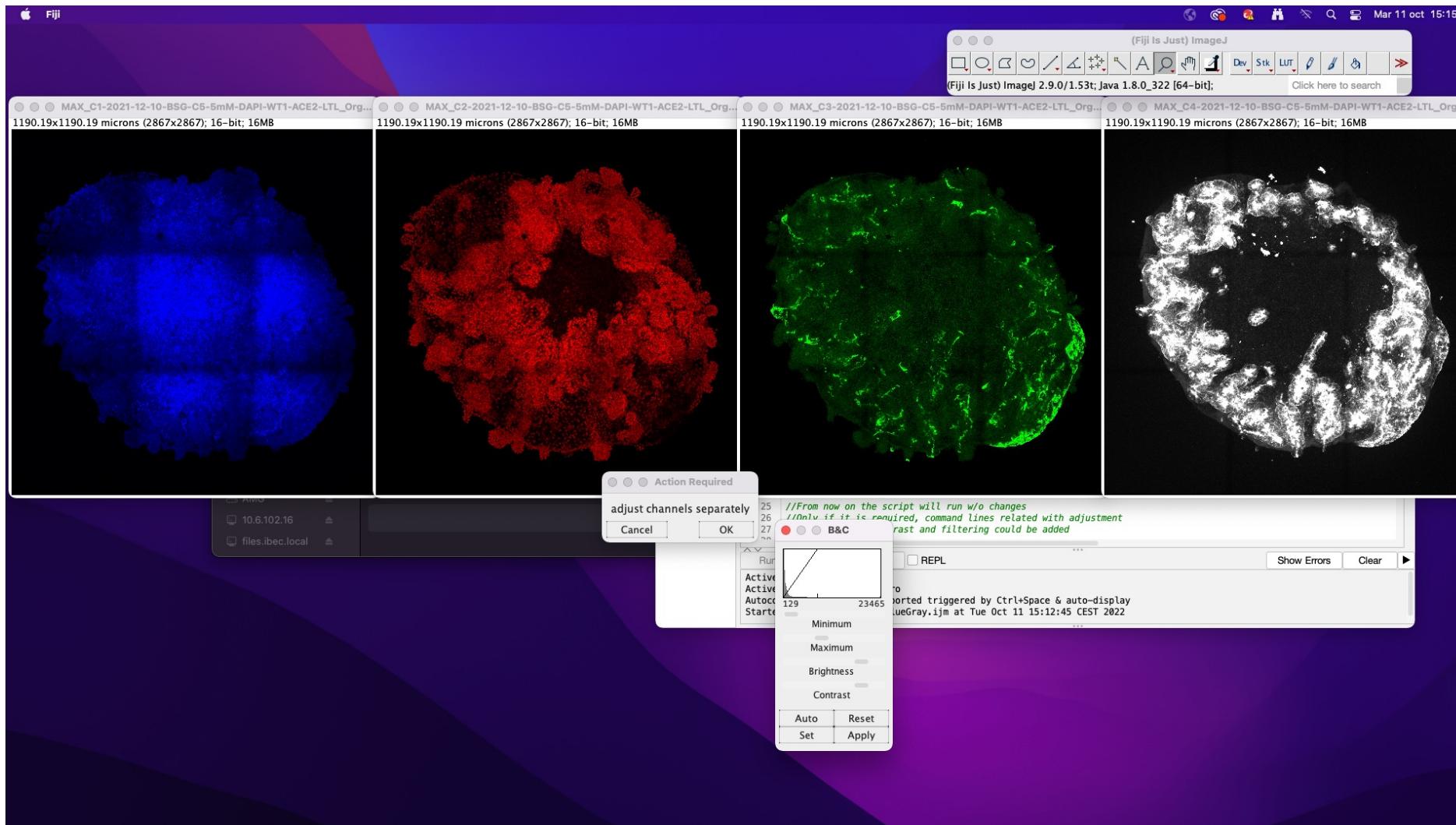
Full Image Processing

3. Run the **4channels-RedGreenBlueGray.ijm** and wait for actions asked to the user: **1st Break to adjust individual channels**



Full Image Processing

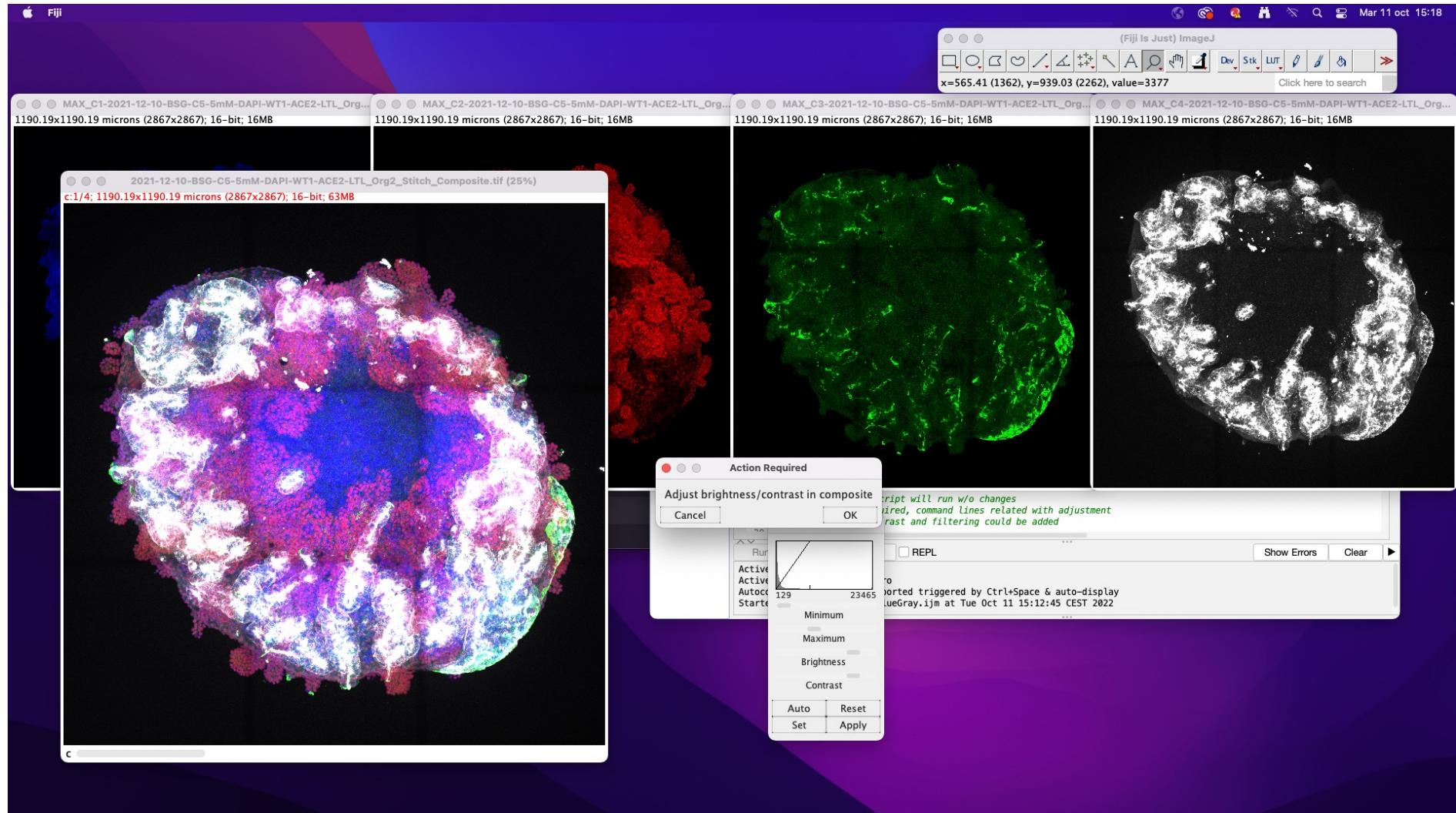
3. Run the **4channels-RedGreenBlueGray.ijm** and wait for actions asked to the user: **1st Break to adjust individual channels**



Full Image Processing

3. Run the **4channels-RedGreenBlueGray.ijm** and wait for actions asked to the user: **2nd Break to adjust channels in composite**

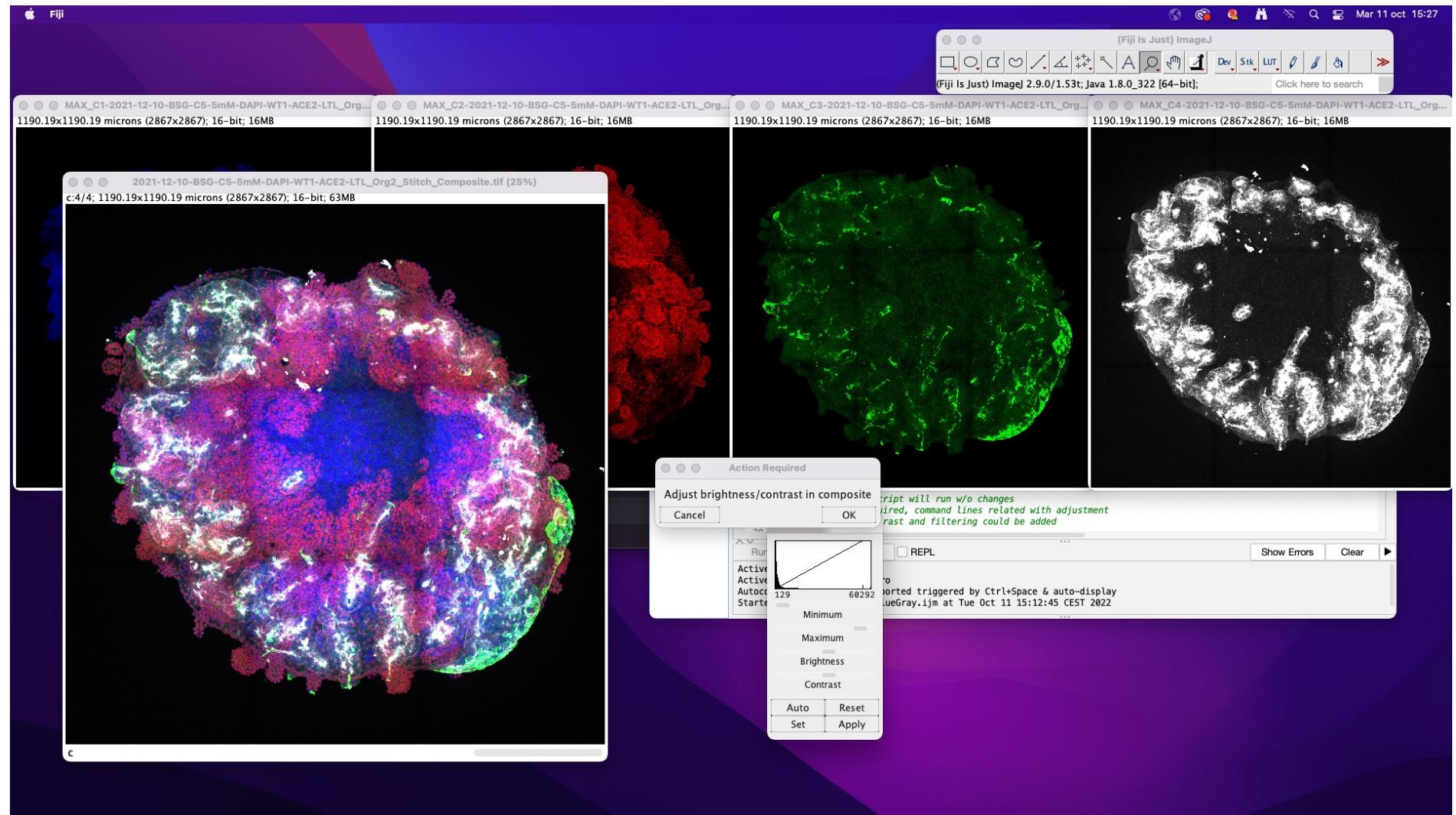
For example gray channel is a bit saturated. You have the chance in the composite mode to readjust each channel.



Full Image Processing

3. Run the **4channels-RedGreenBlueGray.ijm** and wait for actions asked to the user: **2nd Break to adjust channels in composite**

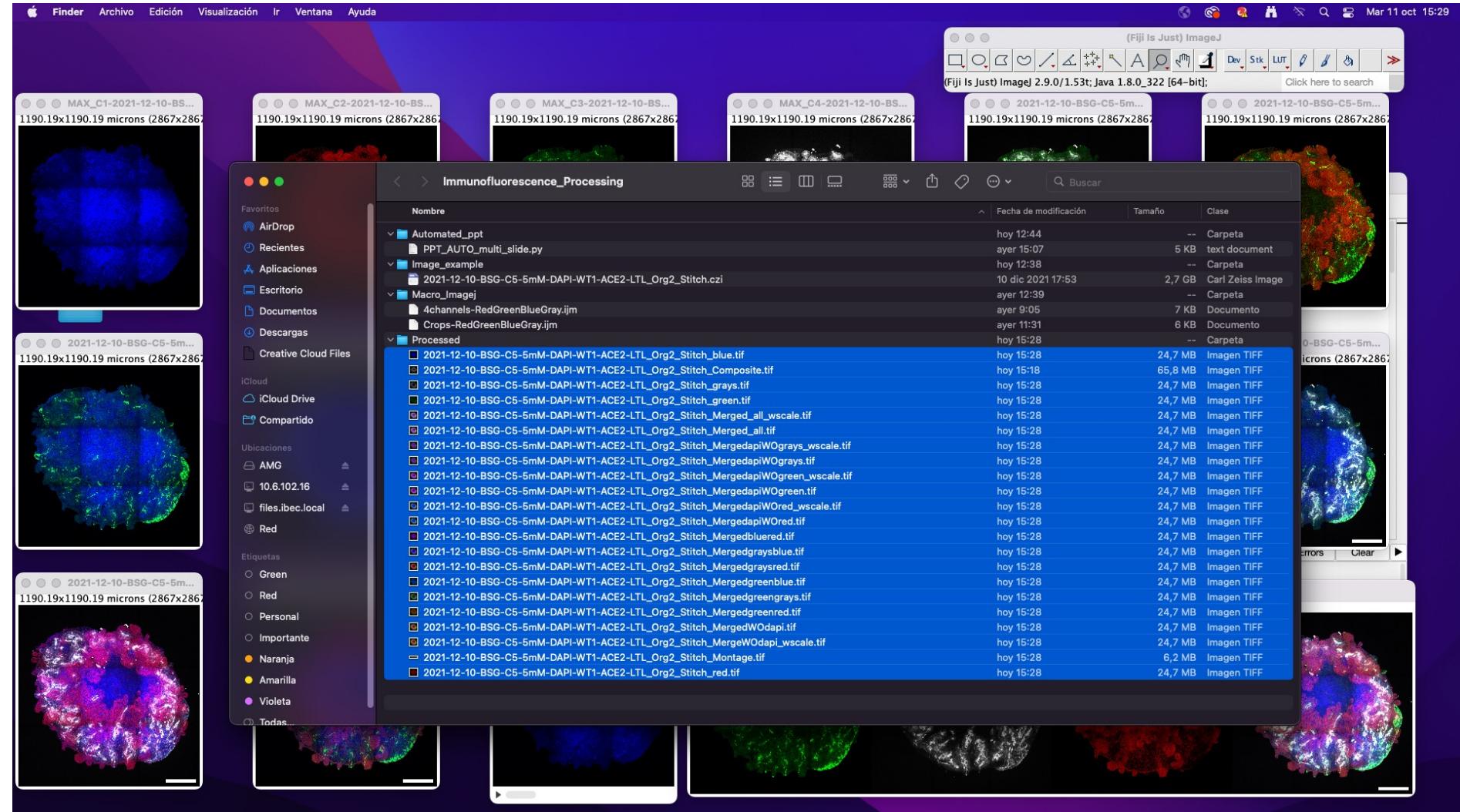
For example gray channel is a bit saturated. You have the chance in the composite mode to readjust each channel.



Full Image Processing

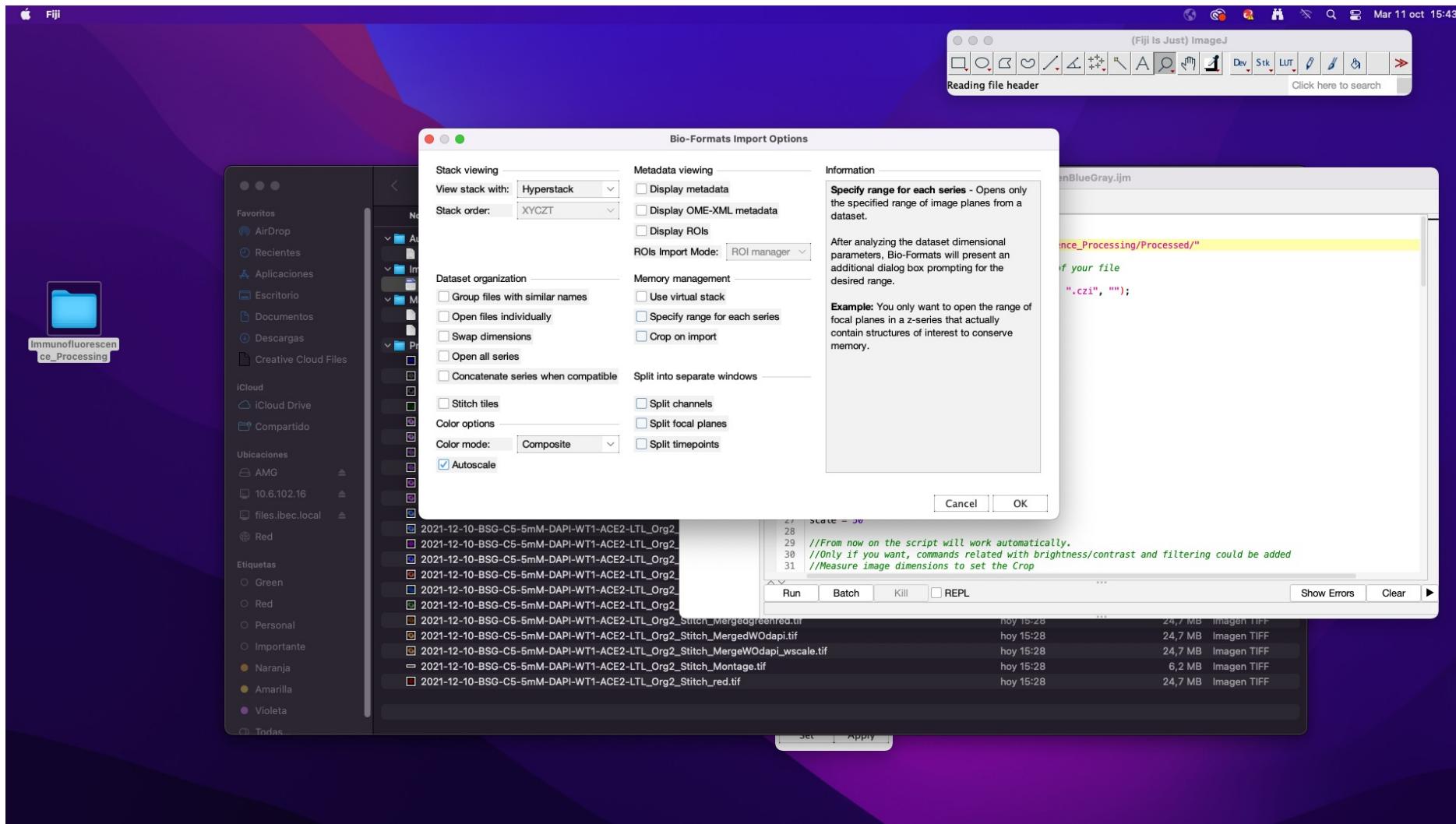
4. Look at saved files in your output folder:

You will have all possible combination of channels in merged tif files that have or not the scale bar.



Processing of crops for detailed views

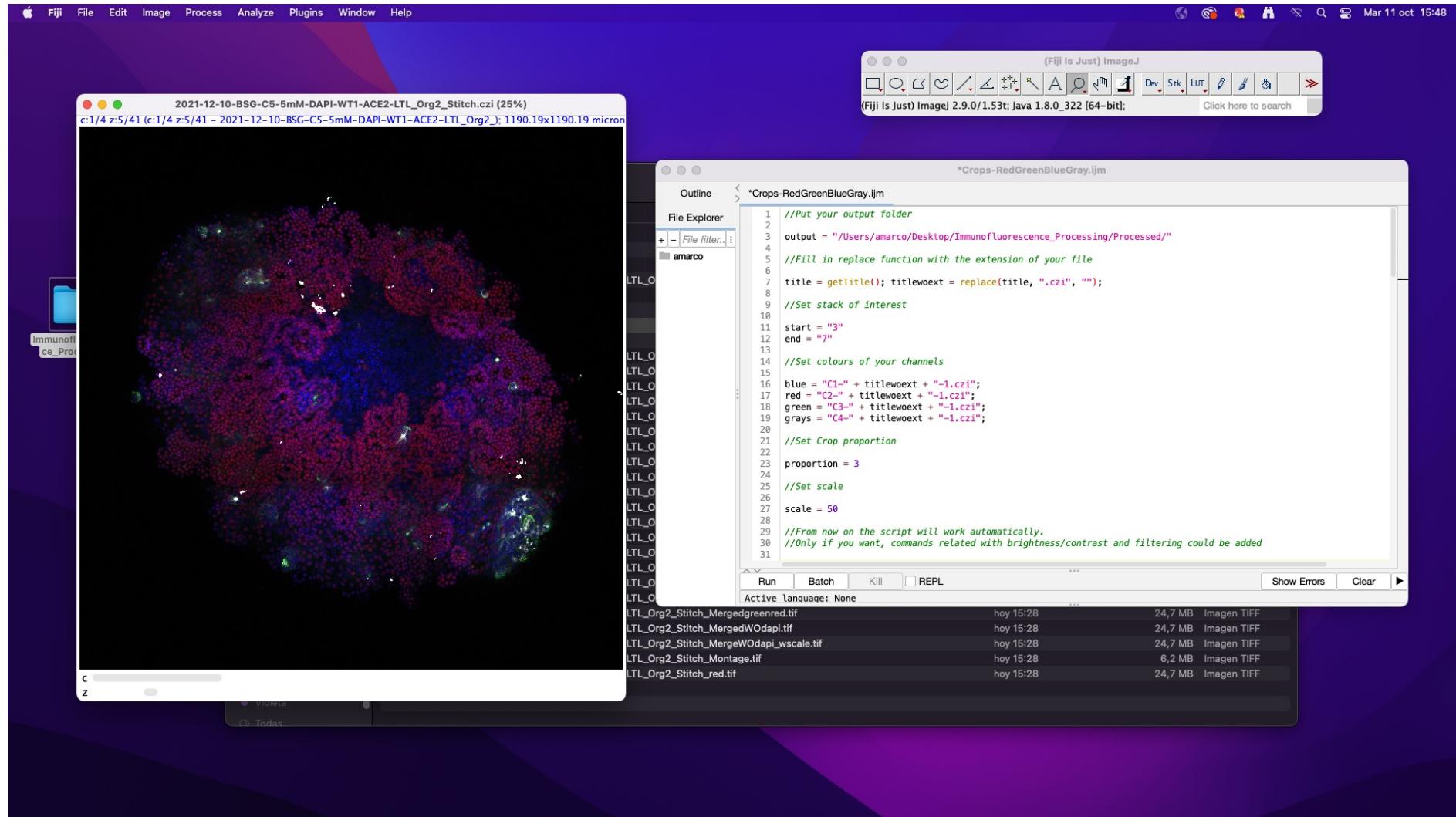
1. Open the image example file and the macro ***Crops-RedGreenBlueGray.ijm*** by drag and drop.



Processing of crops for detailed views

2. Run the macro **Crops-RedGreenBlueGray.ijm** and wait for actions asked to the user.

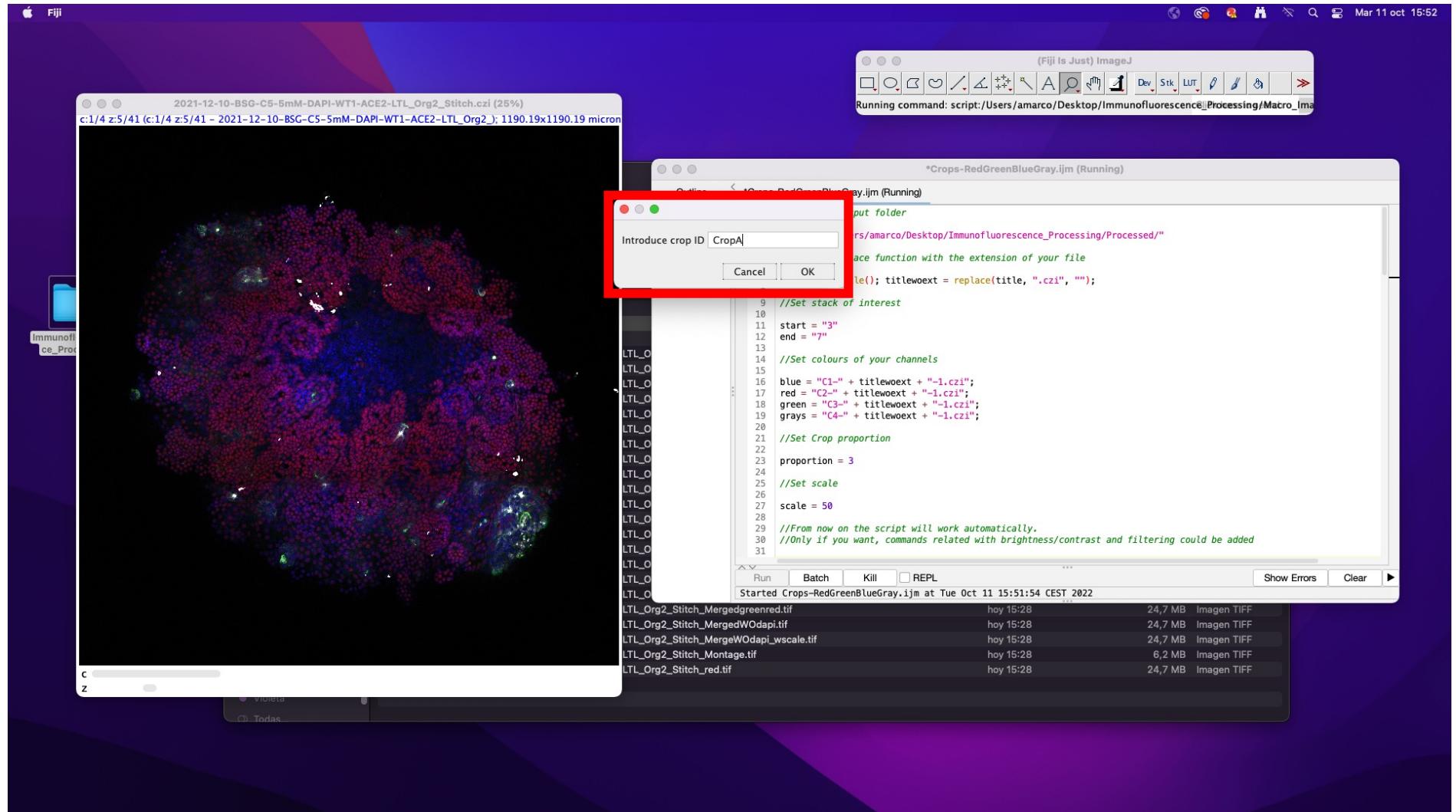
The proportion parameter fixed in 3 allows to set a cropped area 3 times smaller than the full image



Processing of crops for detailed views

3. Run the macro **Crops-RedGreenBlueGray.ijm** and wait for actions asked to the user: **1st input your CropID**

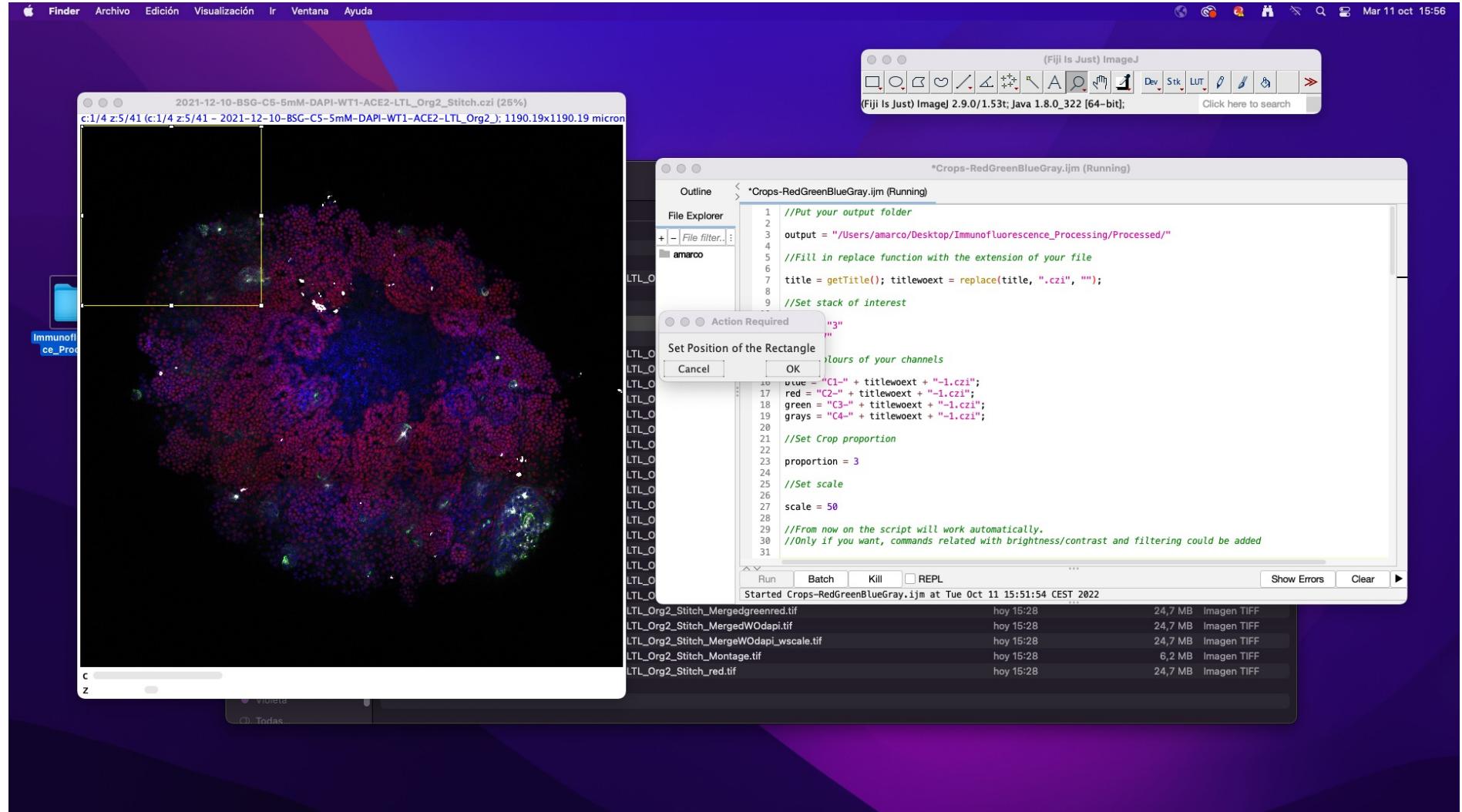
The Crop ID is asked.
This value allow to
save different cropped
areas from the same
image with different
file names.



Processing of crops for detailed views

4. Run the macro **Crops-RedGreenBlueGray.ijm** and wait for actions asked to the user: **2nd set your area of interest**

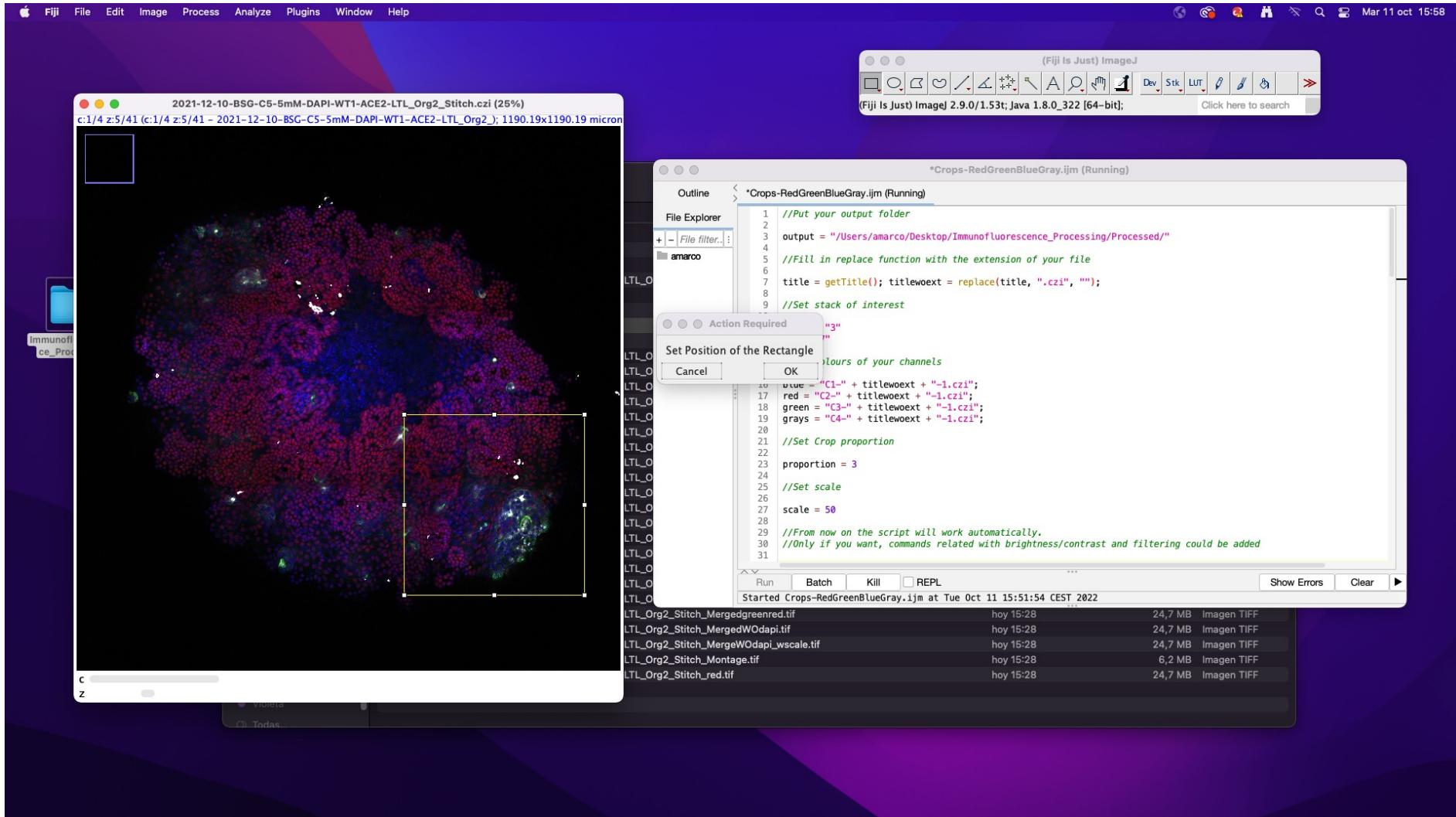
A rectangle 3 times smaller than the full image is plotted to be moves until the area of interest.



Processing of crops for detailed views

4. Run the macro **Crops-RedGreenBlueGray.ijm** and wait for actions asked to the user: **2nd set your area of interest**

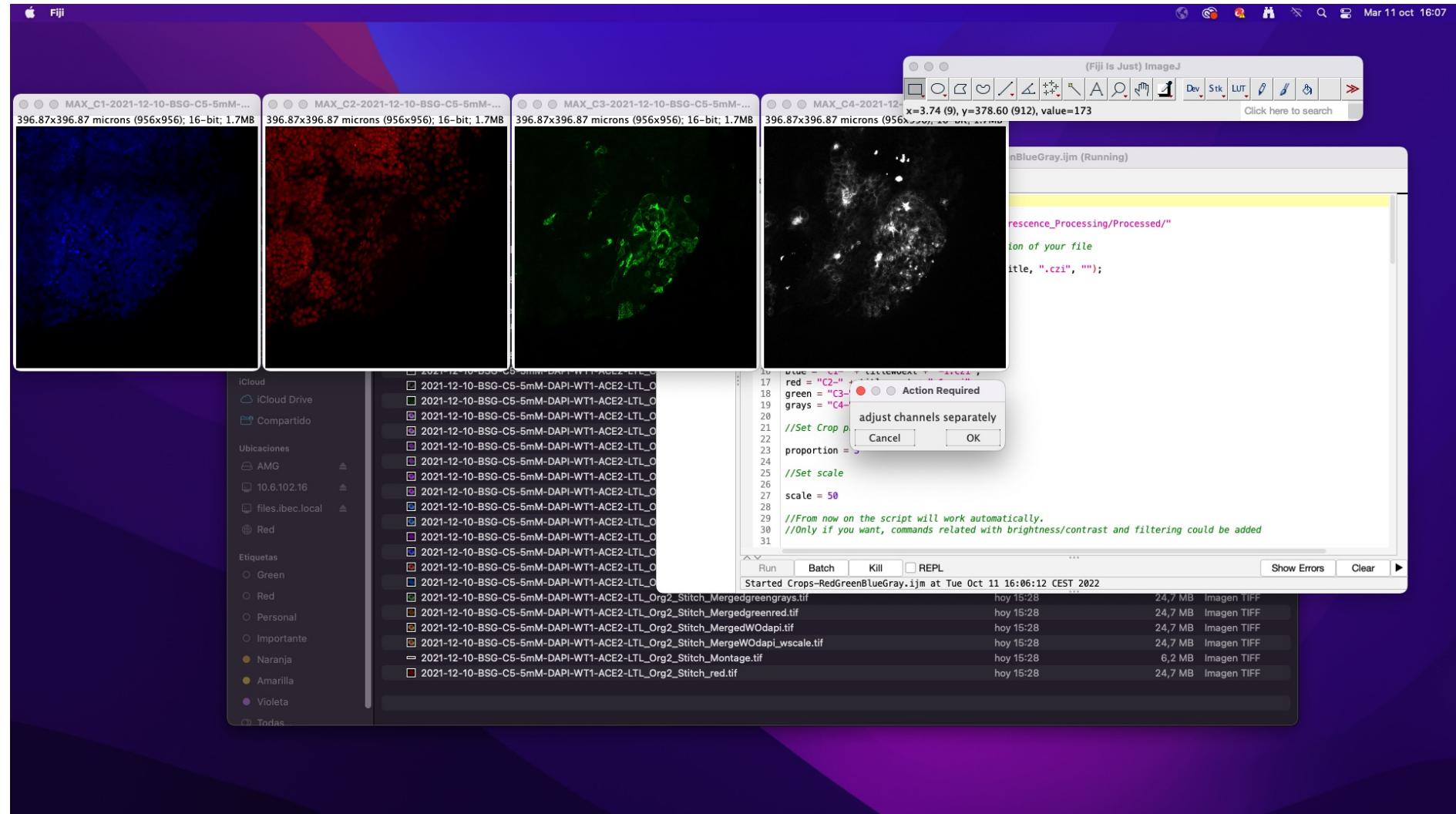
A rectangle 3 times smaller than the full image is plotted to be moves until the area of interest.



Processing of crops for detailed views

5. Run the macro **Crops-RedGreenBlueGray.ijm** and wait for actions asked to the user: **3rd adjust individual channels**

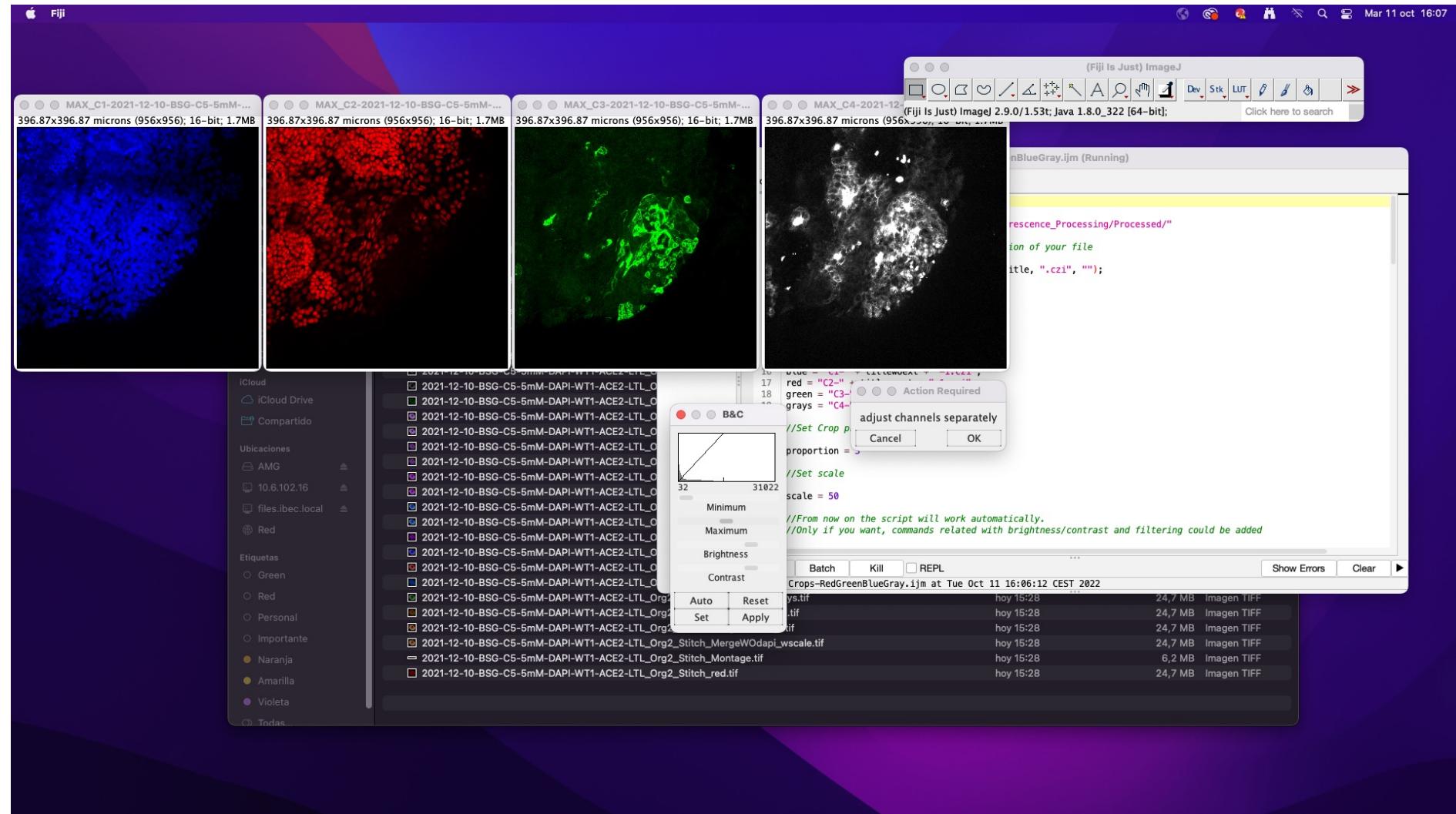
Then the macro runs as the one for full images processing



Processing of crops for detailed views

5. Run the macro **Crops-RedGreenBlueGray.ijm** and wait for actions asked to the user: **3rd adjust individual channels**

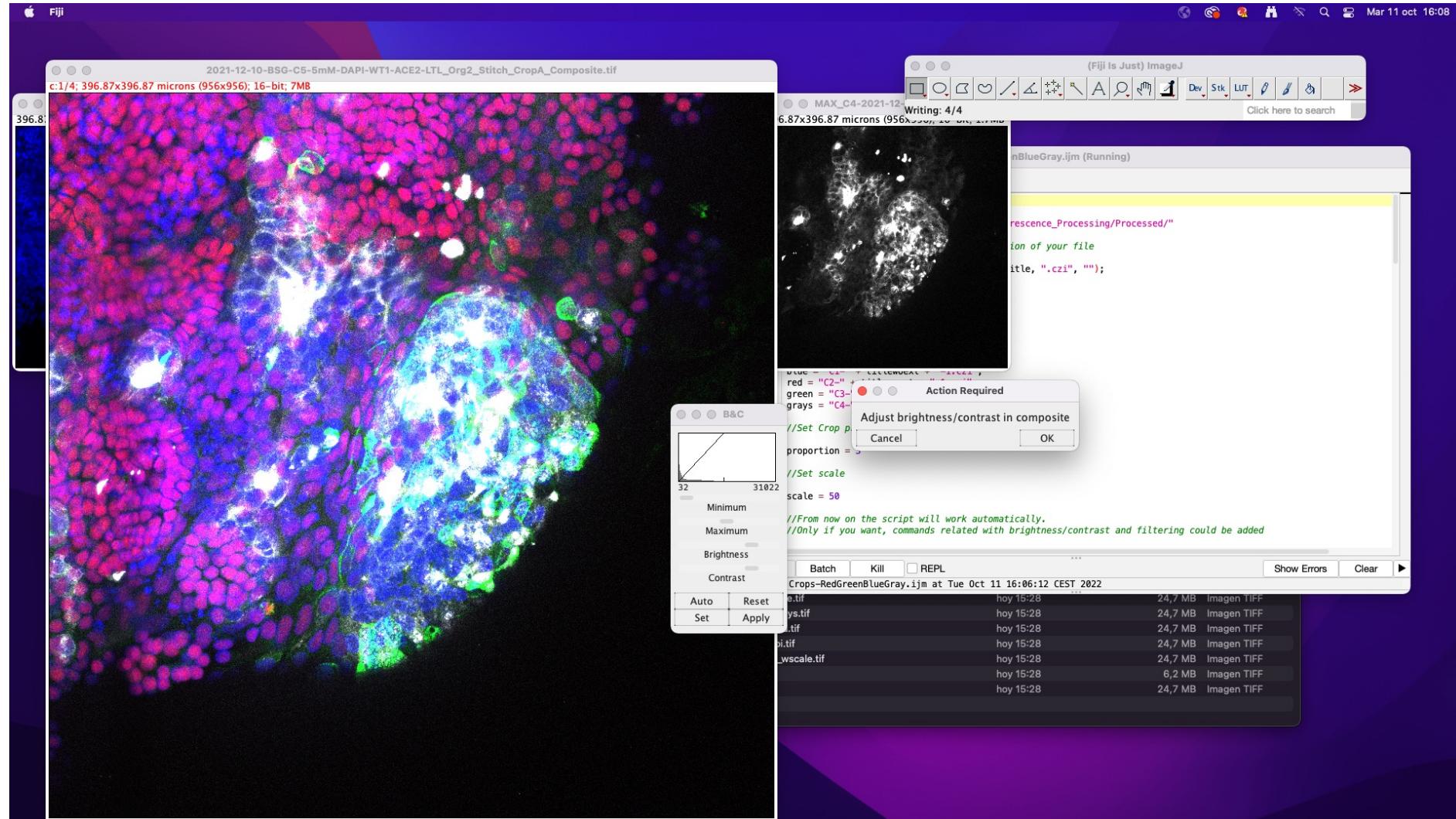
Then the macro runs as the one for full images processing



Processing of crops for detailed views

6. Run the macro **Crops-RedGreenBlueGray.ijm** and wait for actions asked to the user: **4th adjust in composite**

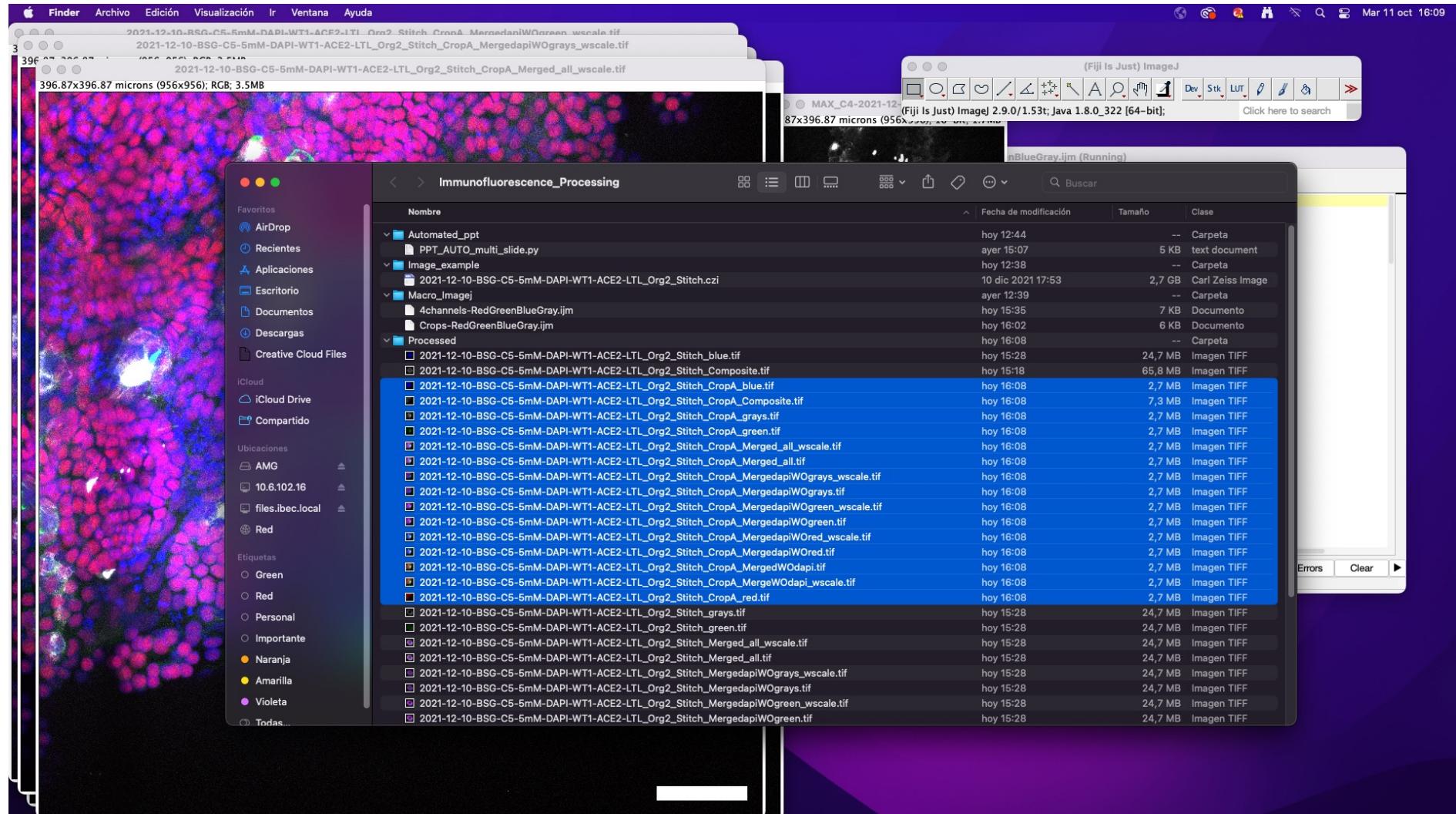
Then the macro runs as the one for full images processing



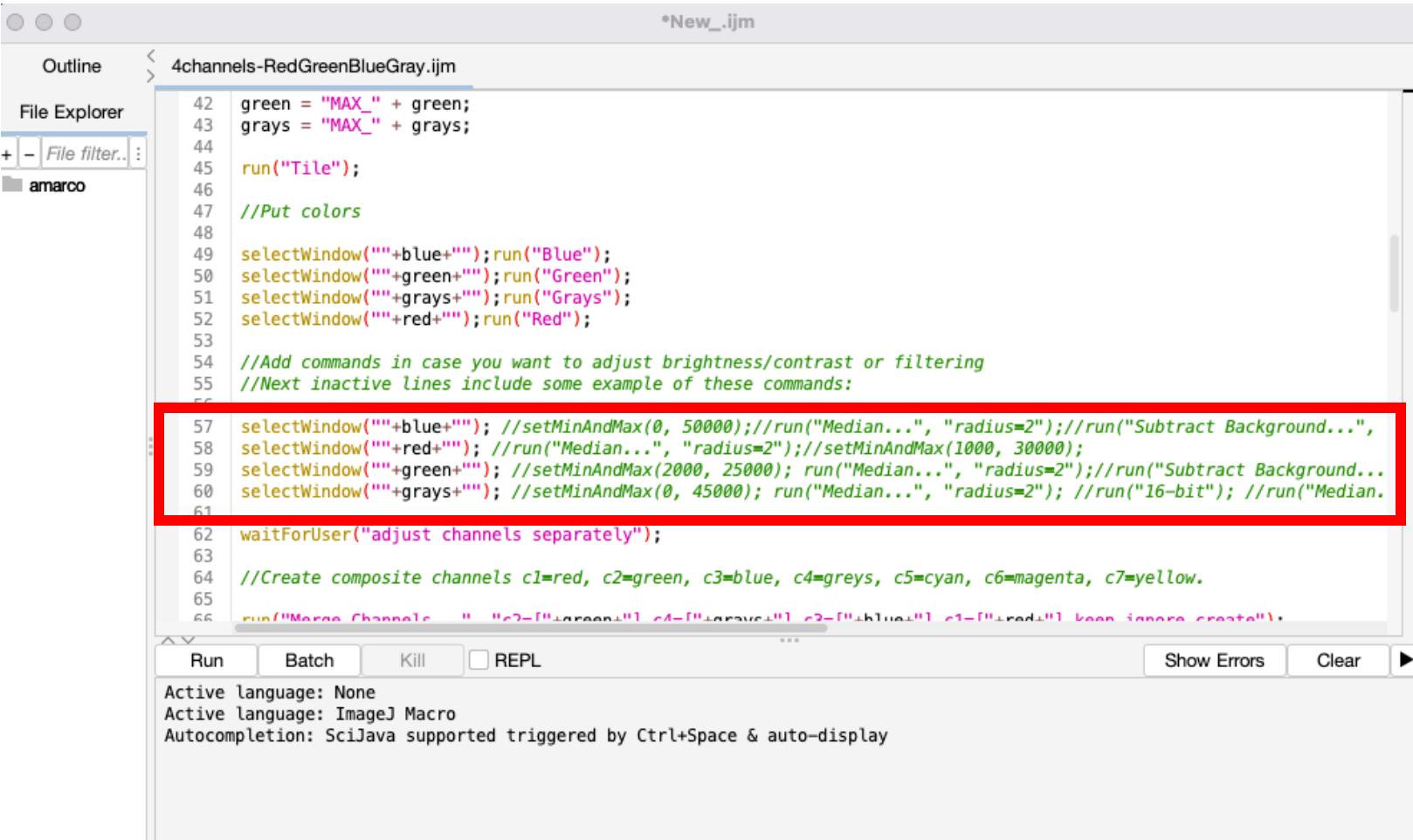
Processing of crops for detailed views

7. Look at saved files in your output folder:

Then the macro runs as the one for full images processing



Automatize adjustments if needed



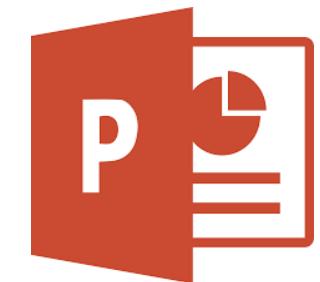
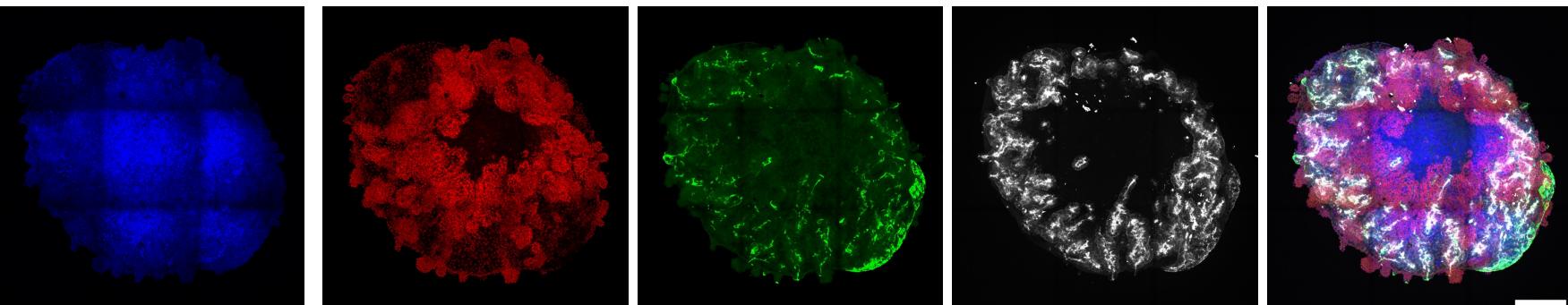
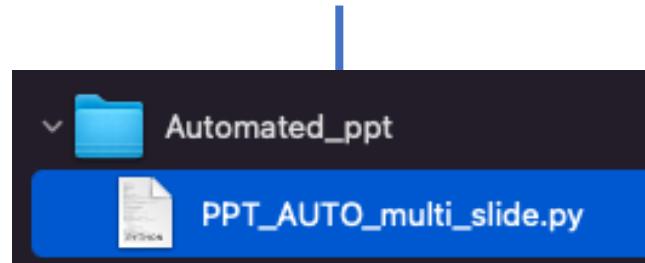
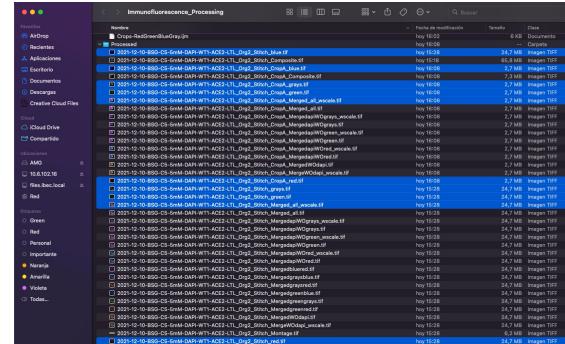
The screenshot shows the ImageJ Macro Editor interface with the file `*New_.ijm` open. The macro code is as follows:

```
42 green = "MAX_" + green;
43 grays = "MAX_" + grays;
44
45 run("Tile");
46
47 //Put colors
48
49 selectWindow(""+blue+"");run("Blue");
50 selectWindow(""+green+"");run("Green");
51 selectWindow(""+grays+"");run("Grays");
52 selectWindow(""+red+"");run("Red");
53
54 //Add commands in case you want to adjust brightness/contrast or filtering
55 //Next inactive lines include some example of these commands:
56
57 selectWindow(""+blue+""); //setMinAndMax(0, 50000); //run("Median...", "radius=2"); //run("Subtract Background...", "radius=2");
58 selectWindow(""+red+""); //run("Median...", "radius=2"); //setMinAndMax(1000, 30000);
59 selectWindow(""+green+""); //setMinAndMax(2000, 25000); run("Median...", "radius=2"); //run("Subtract Background...", "radius=2");
60 selectWindow(""+grays+""); //setMinAndMax(0, 45000); run("Median...", "radius=2"); //run("16-bit"); //run("Median...", "radius=2");
61
62 waitForUser("adjust channels separately");
63
64 //Create composite channels c1=red, c2=green, c3=blue, c4=greys, c5=cyan, c6=magenta, c7=yellow.
65
66 run("Merge Channels... -c c1+c2+c3+c4+c5+c6+c7 -keep ignore create")
```

The lines from 57 to 60 are highlighted with a red box, indicating they are examples of commands for adjusting brightness/contrast or filtering.

Automatic generation of PPTs by “PPT_AUTO_multi_slide.py”

Automated_ppt: folder with a Python script that generates a presentation with your processed images.



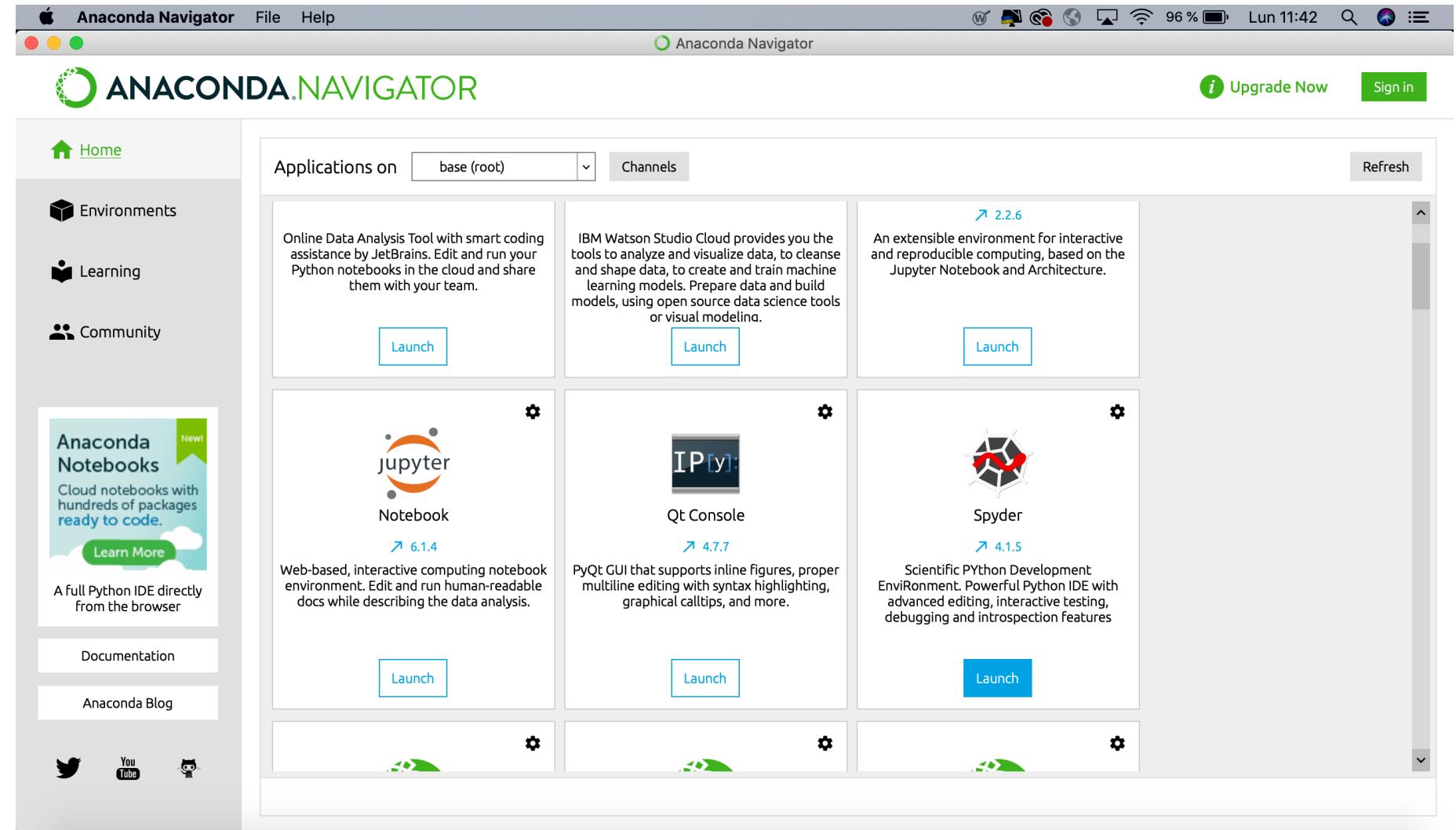
Automatic generation of PPTs by “PPT_AUTO_multi_slide.py”

You can use the spyder environment from anaconda to edit and run the “PPT_AUTO_multi_slide.py” script.

Remember to install the ppxt package from:

<https://python-pptx.readthedocs.io/en/latest/user/install.html>

`pip install python-pptx`



Automatic generation of PPTs by “PPT_AUTO_multi_slide.py”



Fill the script by following instructions in gray color starting at #Input data

The screenshot shows the Spyder IDE interface with the following details:

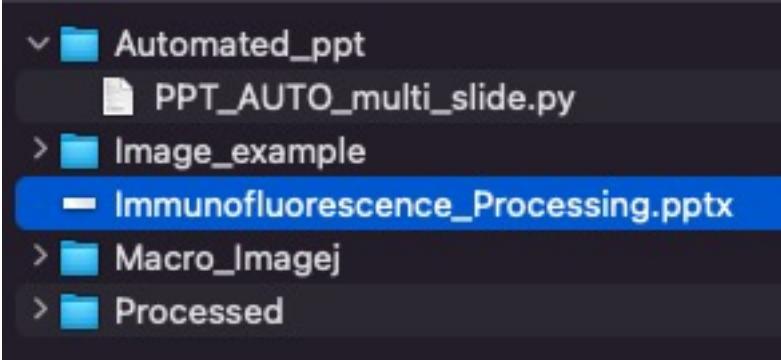
- Editor:** The script file `PPT_AUTO_multi_slide.py` is open. The code includes comments indicating steps for setting experiment name, image size, channel list, number of images per slide, margin size, and the path to processed images. A specific line, `os.chdir('/Users/amarco/Desktop/Immunofluorescence_Processing/Processed')`, is highlighted in green.
- Console:** The IPython console shows the environment setup:
 - Python 3.9.12 (main, Apr 5 2022, 01:53:17)
 - Type "copyright", "credits" or "license" for more information.
 - IPython 8.2.0 -- An enhanced Interactive Python.
- Help:** A usage dialog box is visible, providing information on how to get help for objects.

Be consistent with file names:

File names:

*_blue.tif → 'blue'
*_red.tif → 'red'
*_green.tif → 'green'
*_grays.tif → 'grays'
*_Merged_all_wscale.tif → 'Merged_all_wscale'

Automatic generation of PPTs by “PPT_AUTO_multi_slide.py”



Presentation "Immunofluorescence_Processing.pptx" created

Spyder (Python 3.9)

/Users/amarco/Desktop/Immunofluorescence_Processing

Source Console Object

Usage

Here you can get help of any object by pressing Cmd+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our tutorial

Help Variable Explorer Plots Files

Console 1/A

Python 3.9.12 (main, Apr 5 2022, 01:53:17)
Type "copyright", "credits" or "license" for more information.
IPython 8.2.0 — An enhanced Interactive Python.

In [1]: runfile('/Users/amarco/Desktop/Immunofluorescence_Processing/Automated_ppt/PPT_AUTO_multi_slide.py', wdir='/Users/amarco/Desktop/Immunofluorescence_Processing/Automated_ppt')
In [2]:

IPython console History

LSP Python: ready conda: base (Python 3.9.12) Line 35, Col 74 UTF-8 LF RW Mem 95%

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

from pptx import Presentation
from pptx.util import Cm
import glob
import os
import numpy as np

#Input data

#(1)Set the name of your experiment
Inmons_name = 'Immunofluorescence_Processing'

#(2)Set the size that you want for your images
img_size = Cm(6)

#(3)Set the size of the step that you want between images putted side by side
space_images = Cm(0.2)

#(4)Set the channels you want to plot
channels_list = ['blue','red', 'green','grays', 'Merged_all_wscale']

#(5)Set the number of images that you want to plot by slide
n_images = 1

#(6)Set the size of the margeon that you want in your ppt
margen_size = Cm(3)

#(7)Set the path were your processed images are located
os.chdir('/Users/amarco/Desktop/Immunofluorescence_Processing/Processed')

#Since this point don't modify the script
```

Finished

Automatic generation of PPTs by “PPT_AUTO_multi_slide.py”

