DESKTOP IMAGE ANALYSIS in CellProfiler

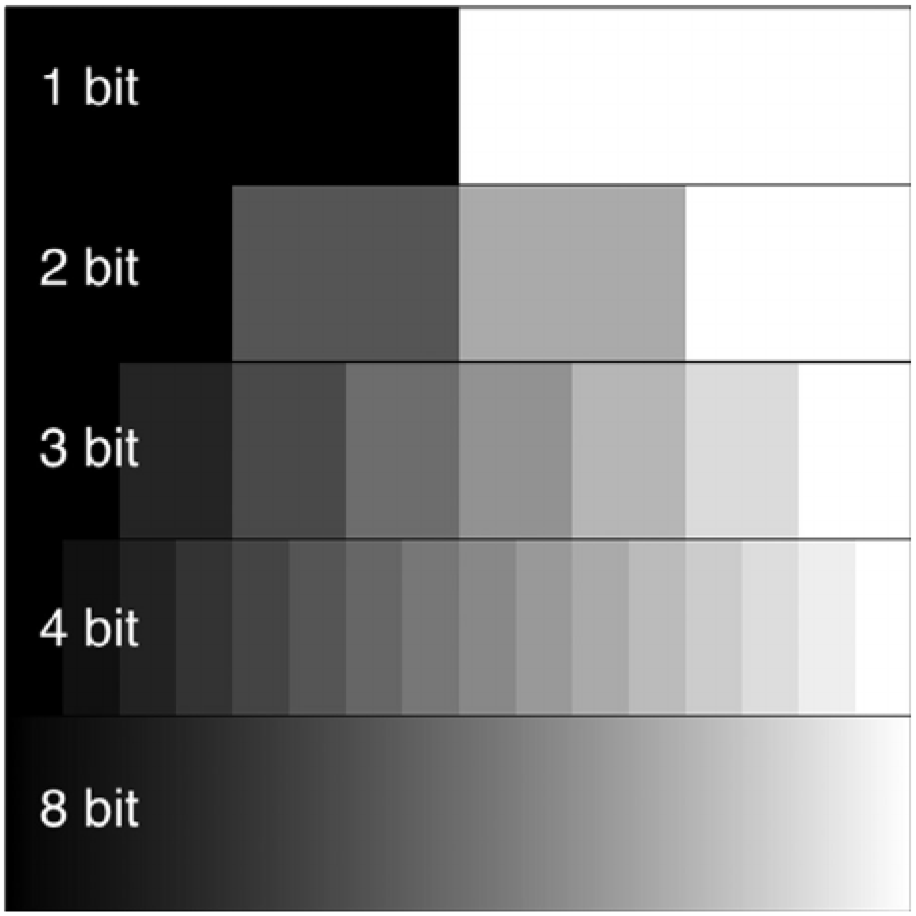
CellProfiler Installation:

If you have not done so yet, go to https://cellprofiler.org/releases and download the release (either Windows or MacOS) for your laptop. Accept the license agreement and install.

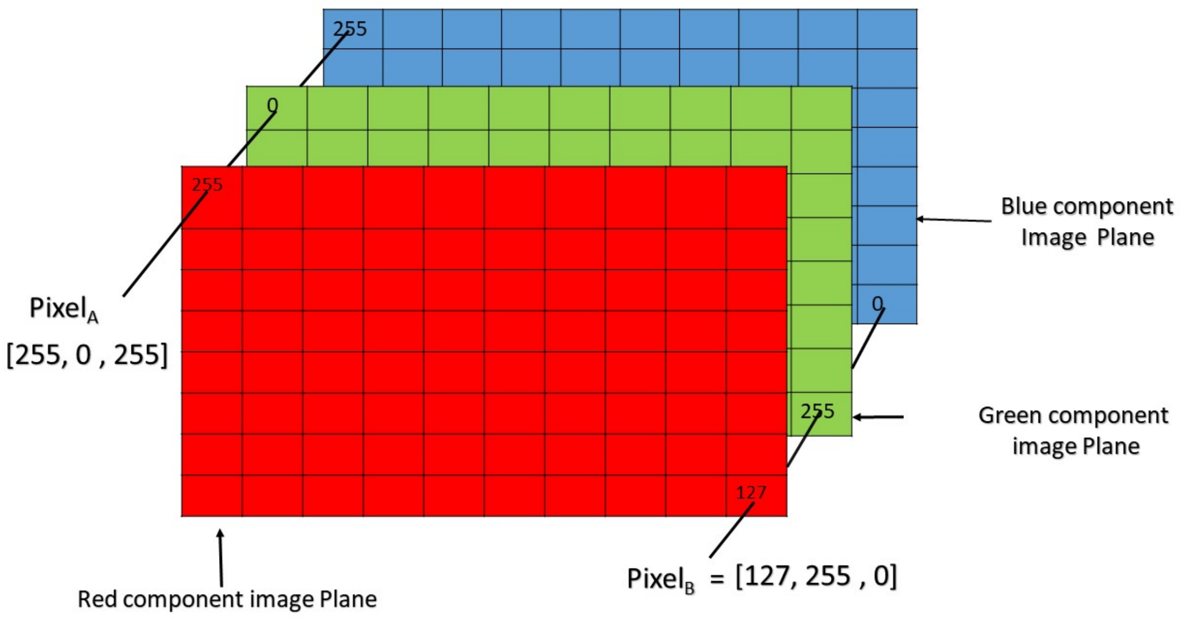
WHAT IS IN AN IMAGE?

Images are just arrays of information in the two dimensions and are defined by two components: location and intensity. A pixel is the smallest unit of a digital image and, for color images, each pixel contains a representative matrix of the intesity levels of the colors red, green, and blue.

The most common types of color images are 8-bit or 16-bit images. Let's break down the meaning of an 8-bit color image to understand a little bit more about it. You may know that a bit holds only two possible values, either 1 or 0. So an 8-bit piece of information is represented by a set of eight 1s or 0s in any combination (ex. 01101110) which is a total of 256 different combinations! Now each of these 8-bit combinations represents the intesity of a single color in our pixel.



As we learned previously, a single pixel in a color image is a matrix of red, green, and blue. This means that for each pixel there are 256 intesities of red which could combine with 256 intensities of greem which could combine with 256 intensities of blue. This is how we generate a large array of different colors within an image.


255
0
255
Blue component Image Plane
Pixel$_A$ [255, 0 , 255]
0
255
Green component image Plane
127
Red component image Plane
Pixel$_B$ = [127, 255 , 0]

Pixel of an RGB image are formed from the corresponding pixel of the three component images

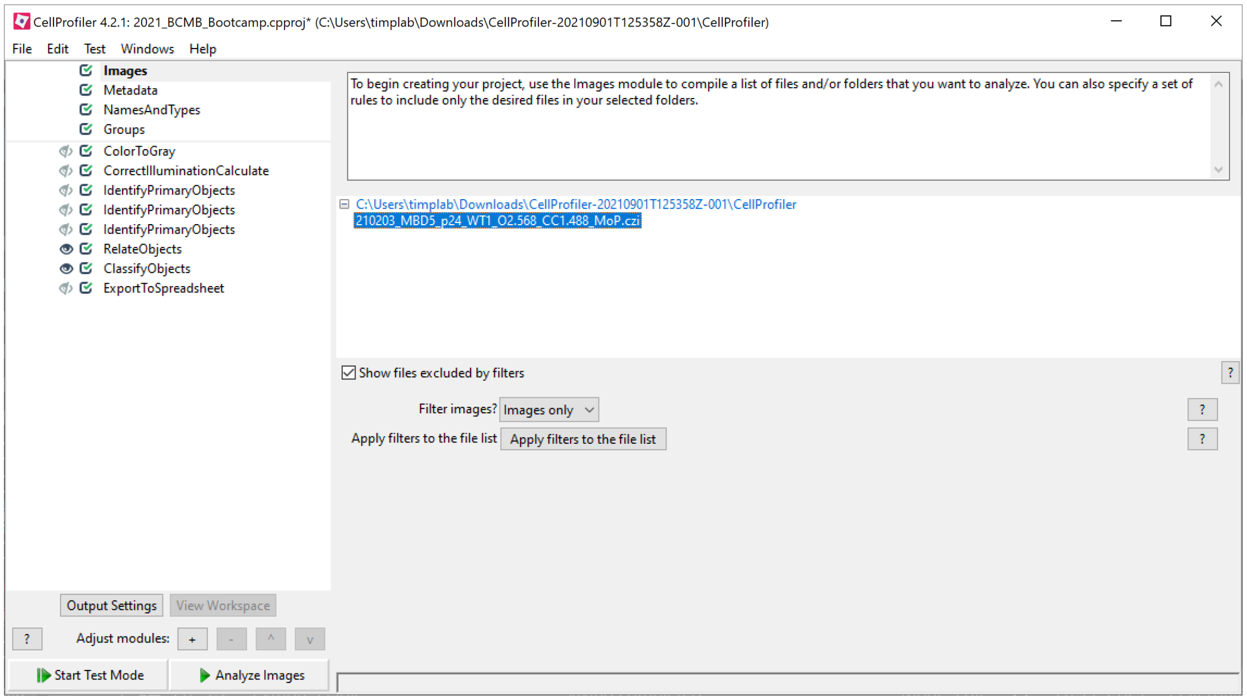
Now that we know the makeup of an image (basically lots of numbers in a matrix), we can start to understand how a lot of image processing and editing software is able to analyze or manipulate this information (i.e. reading or changing the integer values in the matrix).

WHY CELLPROFILER?

The CellProfiler project originally started in 2003 and, as such, it has a long history of development and refinement. It has many built in modules for image processing and analysis which can be formatted into a reproducible and executable pipeline. Additionally, CellProfiler can be utilized from the command line or from the graphical user interface (GUI). For today we will utilize the GUI for ease of installation, but if you would like to explore command line options you can check out CellProfiler's interface with R or with python.

Let's Get Started

When you first open up CellProfiler you will see a screen that offers a user manual, example data and analysis, tutorials, etc. Feel free to close out this screen for now, but you can go back and explore more of the information at your convenience. Next you should be in the images section where you can drag and drop the image data we have provided for you. Start by just selecting a single image to load.



To begin creating your project, use the Images module to compile a list of files and/or folders that you want to analyze. You can also specify a set of rules to include only the desired files in your selected folders.

C:\Users\timplab\Downloads\CellProfiler-20210901T125358Z-001\CellProfiler
210203_MBD5_p24_WT1_O2.568_CC1.488_MoP.czi

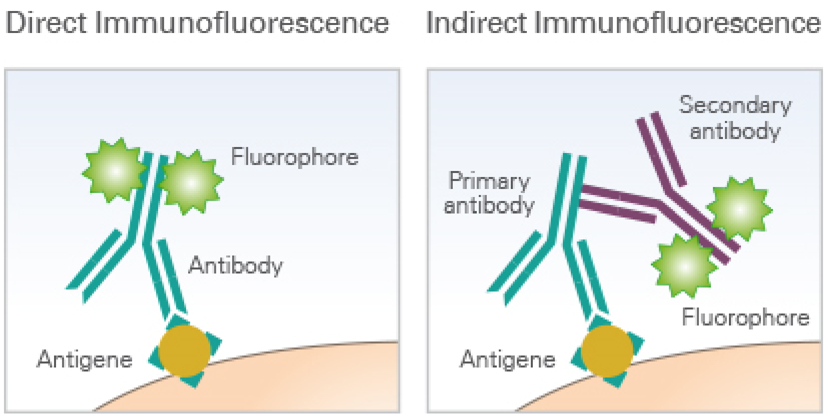
☑ Show files excluded by filters

Filter images? Images only

Apply filters to the file list: Apply filters to the file list

A little backgroud on this data

MBD5 is a methyl-CpG-binding domain protein that when mutated results in severe cognitive development (ex. speech imparement, intellectual disabilities, etc.) due to happloinsufficiency of the MBD5 protein. Abberant meylination has been implicated in cognitive disorders (ex. Alzheimers and autism) and this dataset is an small test pilot to see if there are differing numbers of mature oligodendrocytes (OLs) in the primary motor region of wild type and mutant MBD5 mice. So essentially we are just counting cells which would be extremely boring to do for each image that we capture. These images for this training dataset are 50 micron brain slices that are immunostained with a pan imature and mature OL marker (Olig2 or O2) and a mature OL marker (CC1).

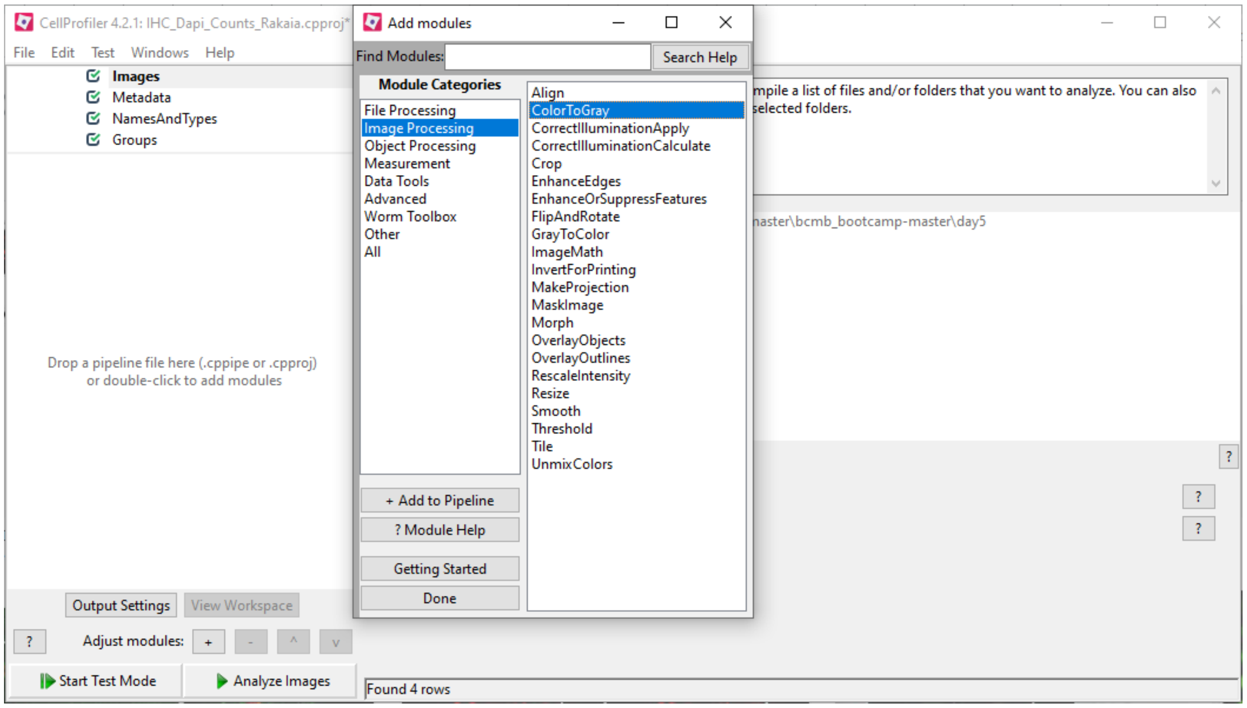


Now you will notice that these images have a lot of information already in the title: date the image was captured, animal family (MBD5), age (P24), designation as wild-type (WT) or heterzygous (het) followed by the animal number (1 or 2), the Olig2 channel (fluorophore emits in red), the CC1 channel (fluorophore emits in green), and the brain region (MoP for primary motor area). One piece of information not noted in this title is DAPI staining (fluorescence emission in blue).

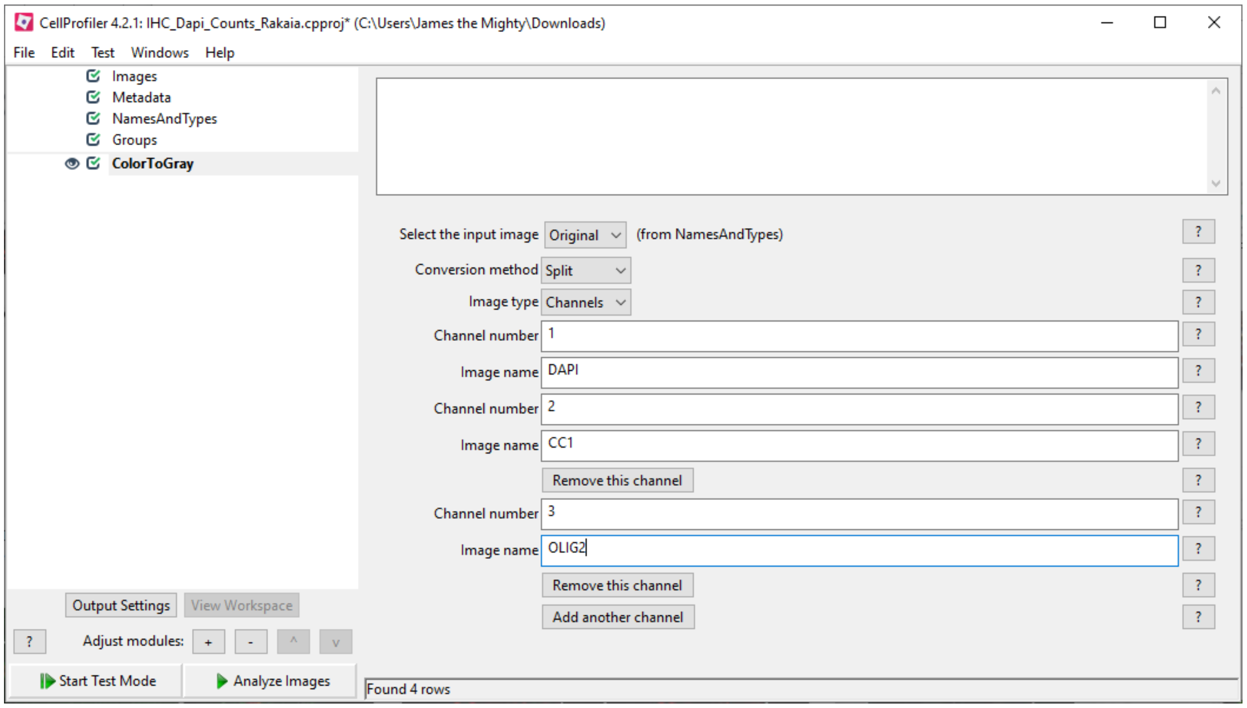
Building a Pipeline

STEP 1

Now that the image file is loaded, the first thing we need to do is split and name our three different channels with our three different markers. The way to do this will be to add the ColorToGray module to our pipeline. To do this click the "+" sign next to "Adjust Modules" Then select Image Processing > ColorToGray > Add to Pipeline > Done.

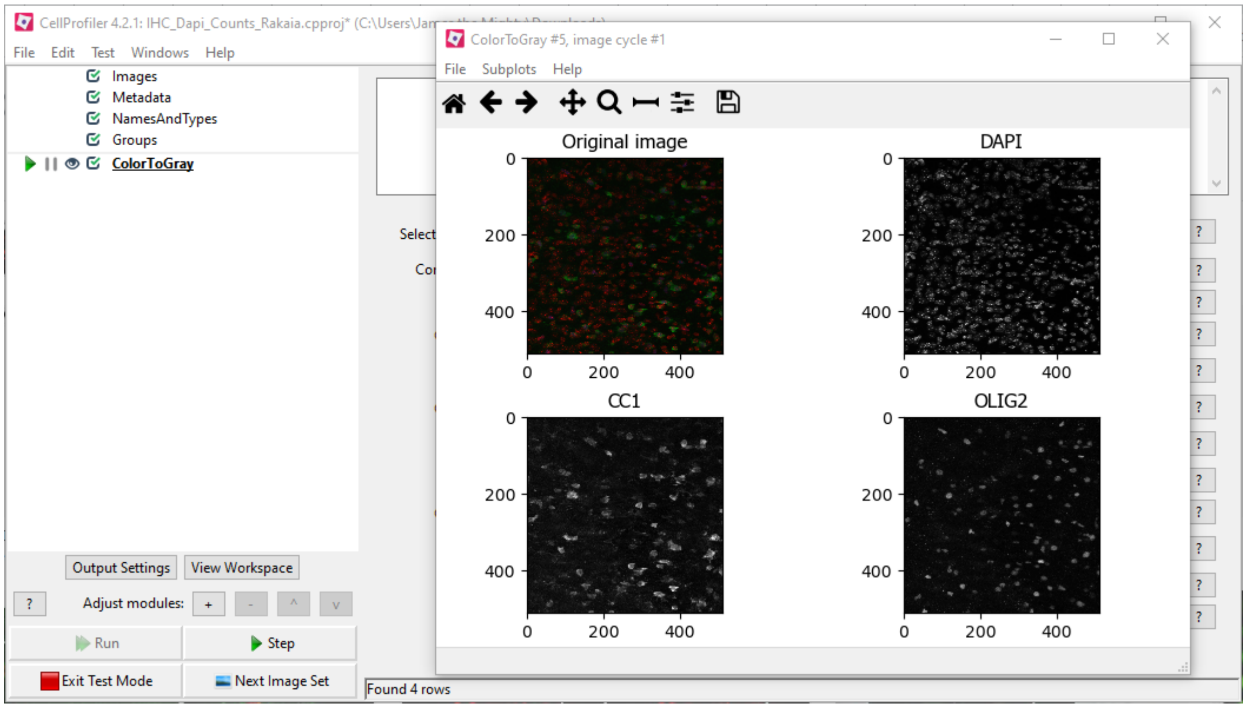


Now split the original image by channel and add the following three channels: 1:DAPI, 2:CC1, and 3:OLIG2. Note that you would know which channels were used in the settings that were established when capturing the image.



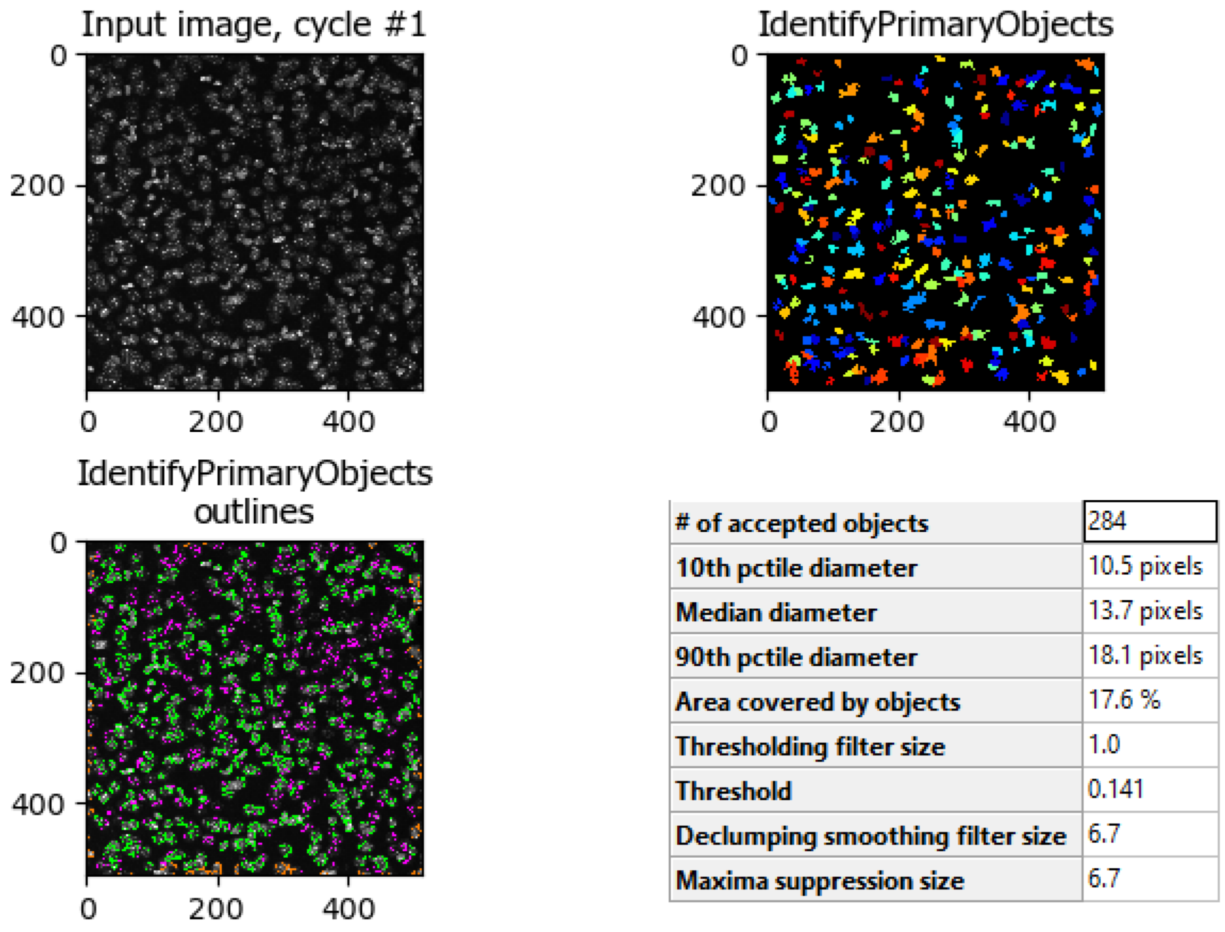
Select the input image: Original (from NamesAndTypes)
Conversion method: Split
Image type: Channels
Channel number: 1
Image name: DAPI
Channel number: 2
Image name: CC1
Channel number: 3
Image name: OLIG2

Let's check how our channels are split! Click Start Test Mode > Run. A new box will pop up that will show you the original image and the division of each of the color channels now converted to gray scale. You can close the popup window and click Exit Test Mode.



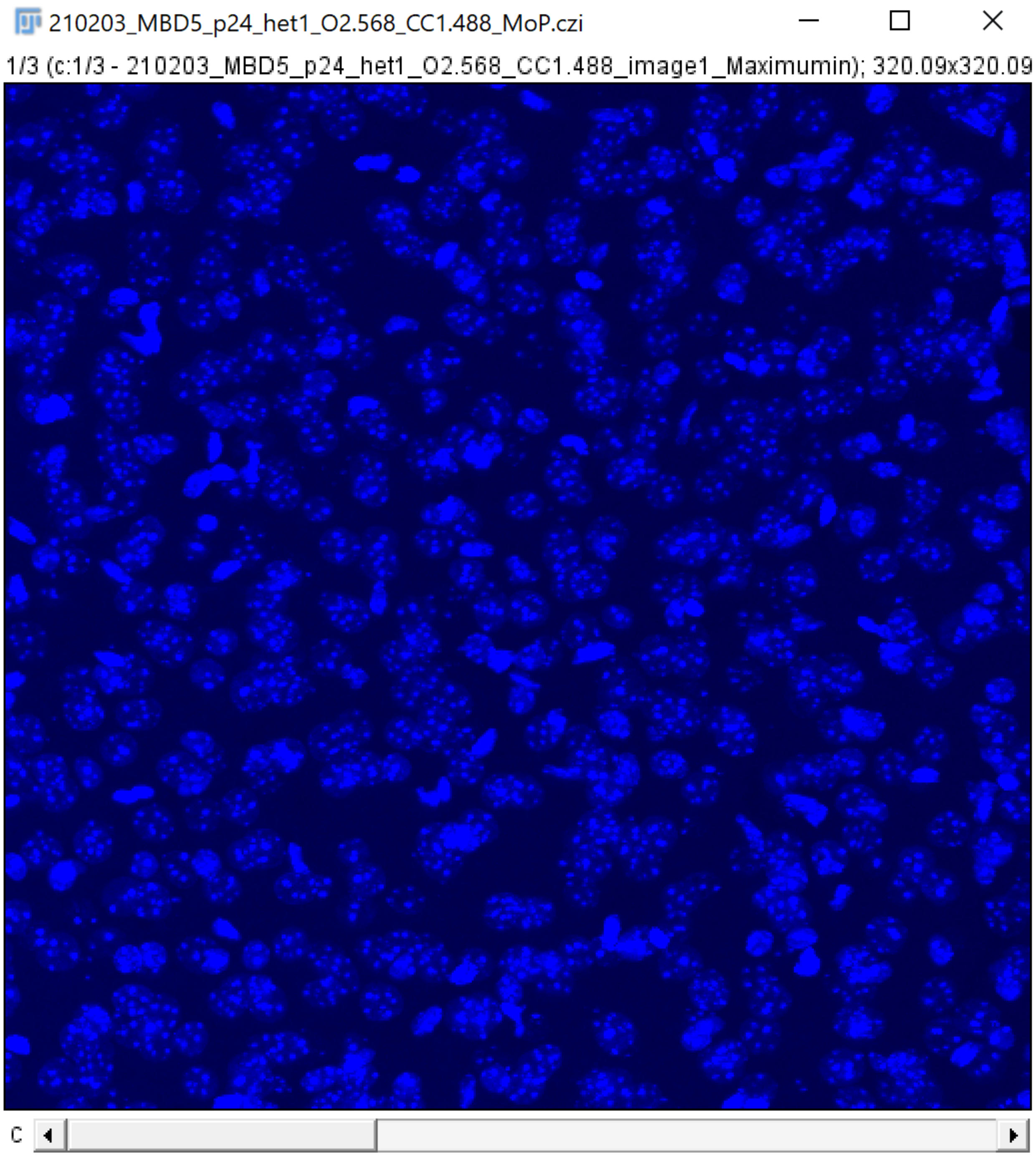
STEP 2

If we eventually want to count the number of our cells that are positive for our specific marker, we first need to identify our cells. There is a built in function for cell profiler known as IdentifyPrimaryObjects. Let's add that to our pipeline after splitting the cells by channel, and just use the default settings of the DAPI channel for now (select input image as DAPI). Enter test mode and run to this step. One thing you might notice right away (and we can zoom in on a region) is that the cell counts appear off. Notice there appear to be a lot of small puncta which the software is excluding because they do not meet the lower threshold for what we set as the exclusion for recognizing a true object.



# of accepted objects	284
10th pctile diameter	10.5 pixels
Median diameter	13.7 pixels
90th pctile diameter	18.1 pixels
Area covered by objects	17.6 %
Thresholding filter size	1.0
Threshold	0.141
Declumping smoothing filter size	6.7
Maxima suppression size	6.7

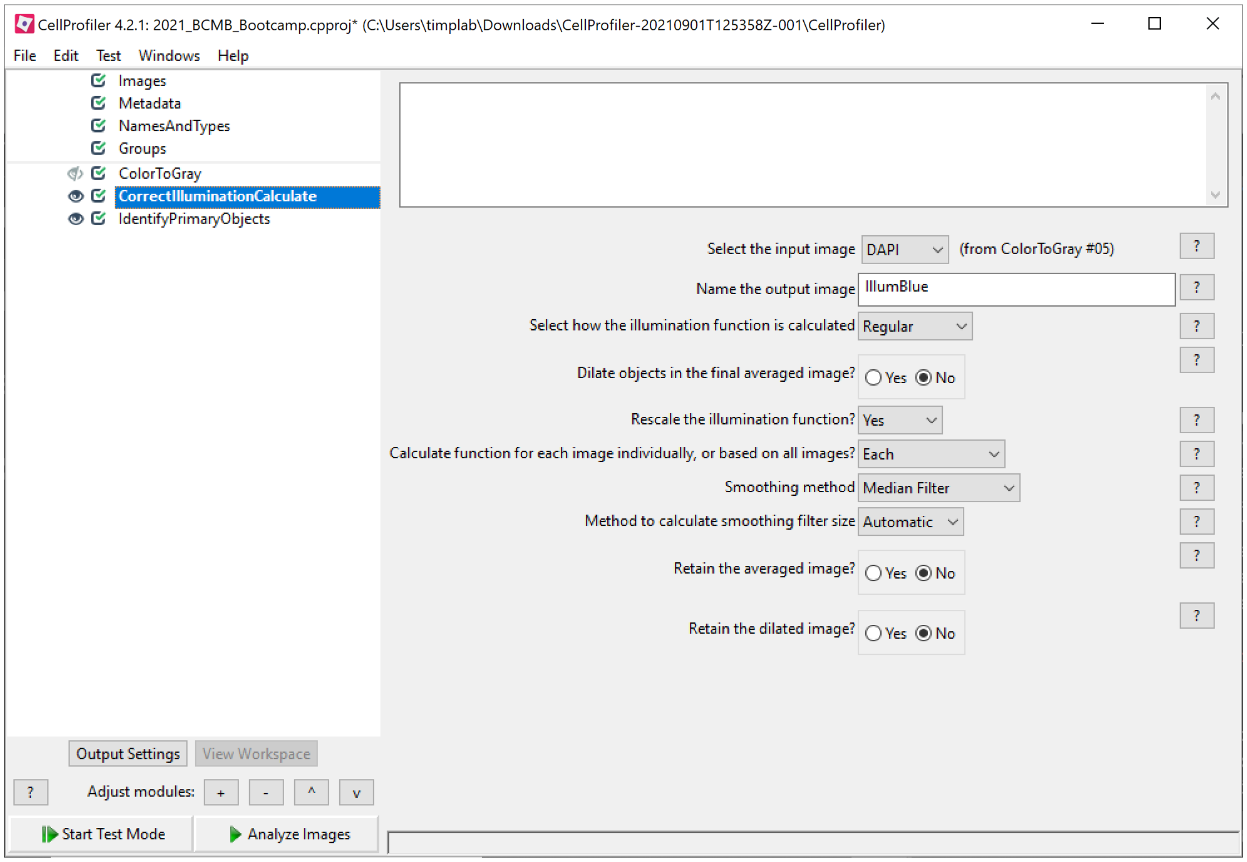
Tissue samples and certain types of immuno staining can be finicky depending on the epitope of the antibody or the type of dye. In this case DAPI is a stain that binds in -AT rich regions of DNA, which means that the signal will be stronger in dense heterchromatic regions. These "speckles" are likely true cells that are being lost due to the intesity of the DNA puncta from the heterochromatic regions. We can sort of see if we increase the brightness:


210203_MBD5_p24_het1_O2.568_CC1.488_MoP.czi
1/3 (c:1/3 - 210203_MBD5_p24_het1_O2.568_CC1.488_image1_Maximumin); 320.09x320.09
c

So we could try to adapt our object identity settings or we can try to smooth the image to help us count our DAPI cells. Let's try smoothing.

STEP 3

Smoothing will help to unify the intensities across the nuclear region (where DAPI and Olig2 are located) or the cell body which will make it easier to identify the cellular objects in the next step. To do this we will add the module CorrectIlluminationCalculate. Selecting DAPI as the input, naming the output image a name of your choice, selecting Regular as the illumination function, smoothing method as Median Filter, and selecting the method to filter as automatic.

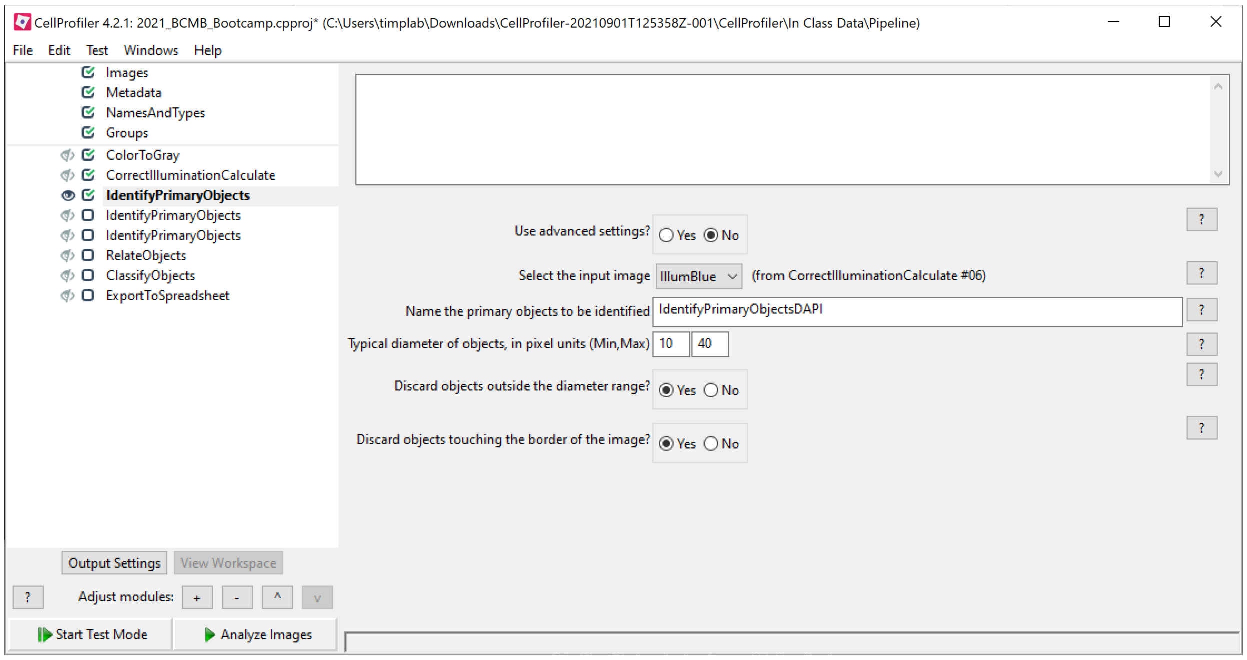

CellProfiler 4.2.1: 2021_BCMB_Bootcamp.cpproj* (C:\Users\timplab\Downloads\CellProfiler-20210901T125358Z-001\CellProfiler)
File Edit Test Windows Help
Images
Metadata
NamesAndTypes
Groups
ColorToGray
CorrectIlluminationCalculate
IdentifyPrimaryObjects
Output Settings
View Workspace
Adjust modules: + - ^ v
Start Test Mode
Analyze Images

Select the input image: DAPI (from ColorToGray #05)
Name the output image: IllumBlue
Select how the illumination function is calculated: Regular
Dilate objects in the final averaged image? Yes [✓ No]
Rescale the illumination function? Yes
Calculate function for each image individually, or based on all images? Each
Smoothing method: Median Filter
Method to calculate smoothing filter size: Automatic
Retain the averaged image? Yes [✓ No]
Retain the dilated image? Yes [✓ No]

Essentially what we are doing here is trying to smooth the signal intensity of DAPI across the surface of the nuclei which will generate more uniform objects that we can use in our next step to identify each of our DAPI positive cells.

STEP 4

Next we can try to identify each of our DAPI positive cells. To do this move our IdentifyPrimaryObjects after our CorrectIlluminationCalculate module in the pipeline. Now enter our dialated image which you saved as the output of your smoothing function from your CorrectIlluminationCalculate module.

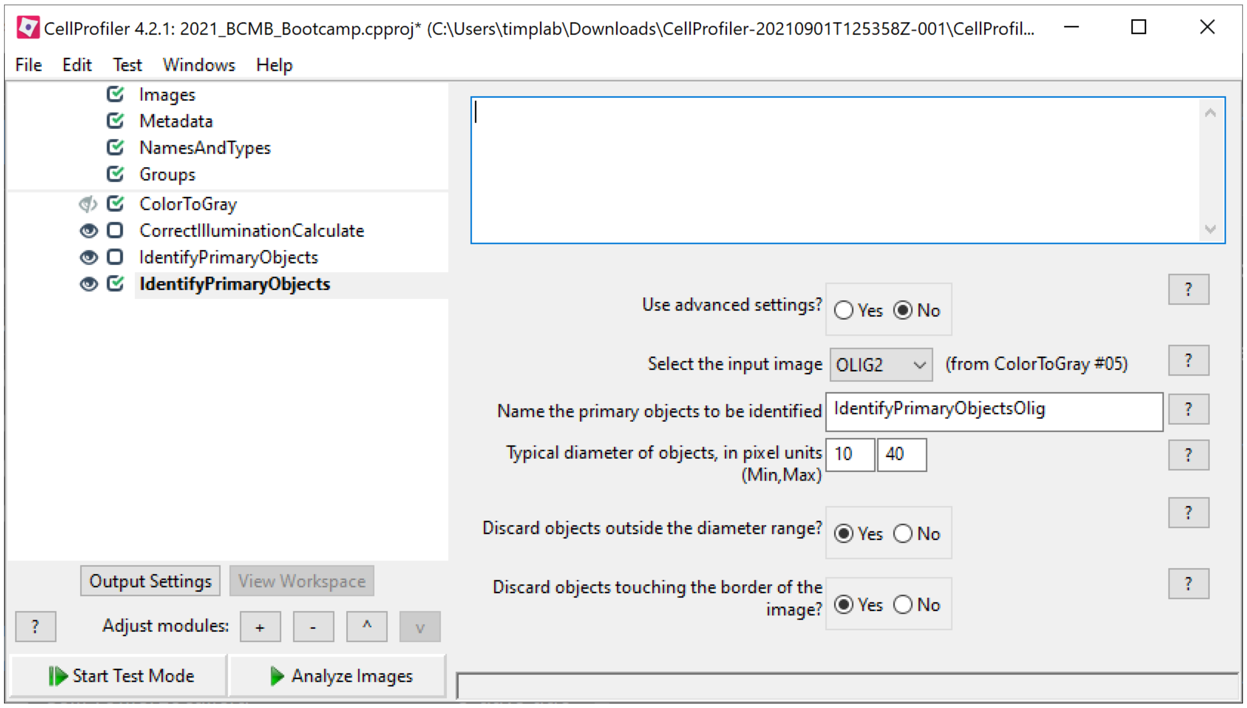


Use advanced settings? Yes [✓ No]
Select the input image IllumBlue (from CorrectIlluminationCalculate #06)
Name the primary objects to be identified IdentifyPrimaryObjectsDAPI
Typical diameter of objects, in pixel units (Min,Max) 10 40
Discard objects outside the diameter range? [✓ Yes] No
Discard objects touching the border of the image? [✓ Yes] No

Now just for fun, lets test out all of these steps we just did. Select Start Test Mode and Click the step button one step at a time to see the output of the settings we just used (a new window will pop up for each step). Once you get through each step for this first image you can select Exit Test Mode on the main programming page.

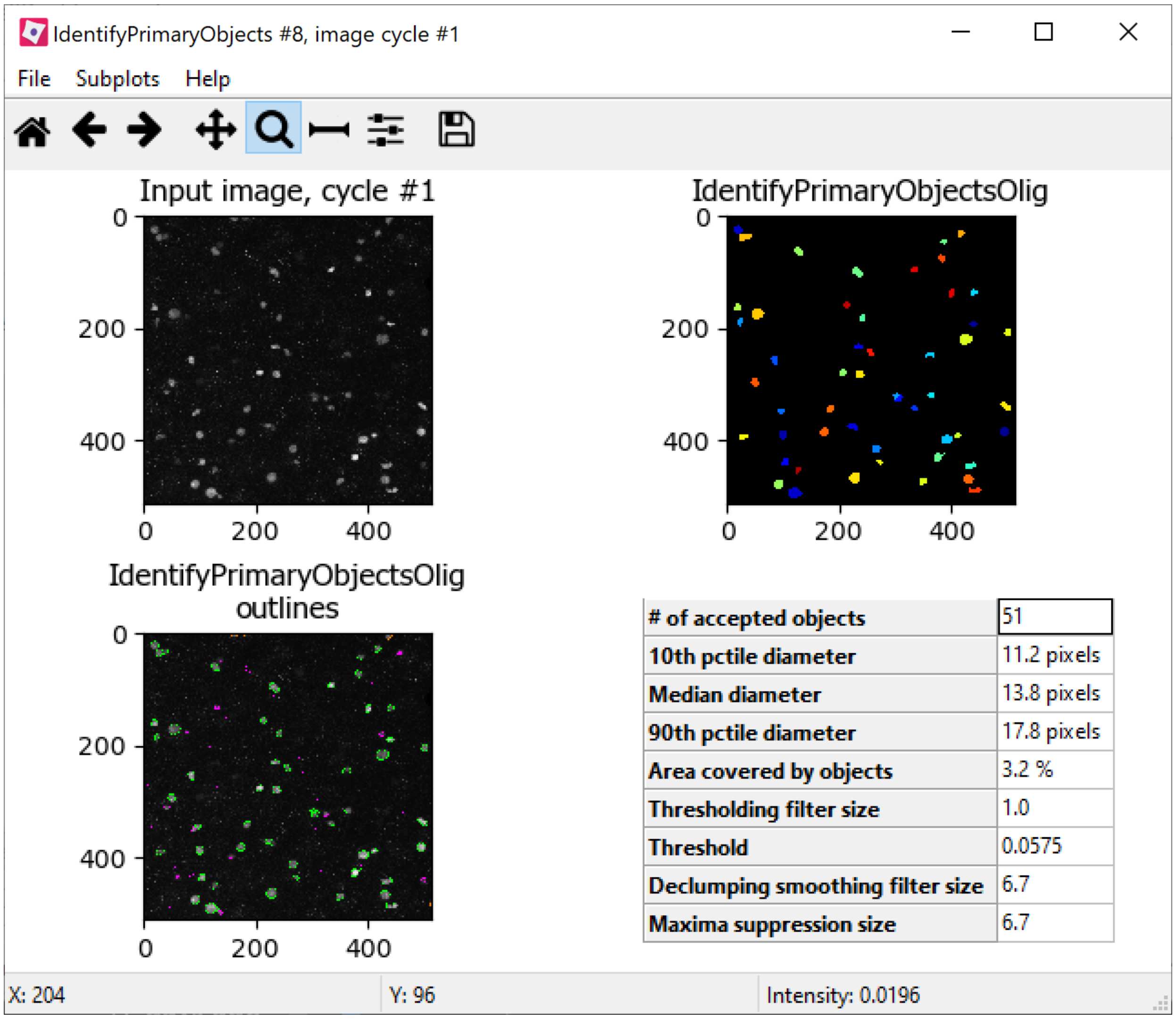
STEP 5

Now let's try the same thing starting with IdentifyPrimaryObjects but for our Olig2 channel. Olig2 is a transcription factor that will be localized to the nucleus. We can test to see if we will have to do the same illumination correction or if this nuclear stain will be more uniform.


CellProfiler 4.2.1: 2021_BCMB_Bootcamp.cpproj* (C:\Users\timplab\Downloads\CellProfiler-20210901T125358Z-001\CellProfil...
File Edit Test Windows Help
Images
Metadata
NamesAndTypes
Groups
ColorToGray
CorrectIlluminationCalculate
IdentifyPrimaryObjects
IdentifyPrimaryObjects
Output Settings
View Workspace
Adjust modules: + - ^ v
Start Test Mode
Analyze Images

Use advanced settings? Yes [✓ No]
Select the input image OLIG2 (from ColorToGray #05)
Name the primary objects to be identified IdentifyPrimaryObjectsOlig
Typical diameter of objects, in pixel units (Min,Max) 10 40
Discard objects outside the diameter range? [✓ Yes] No
Discard objects touching the border of the image? [✓ Yes] No

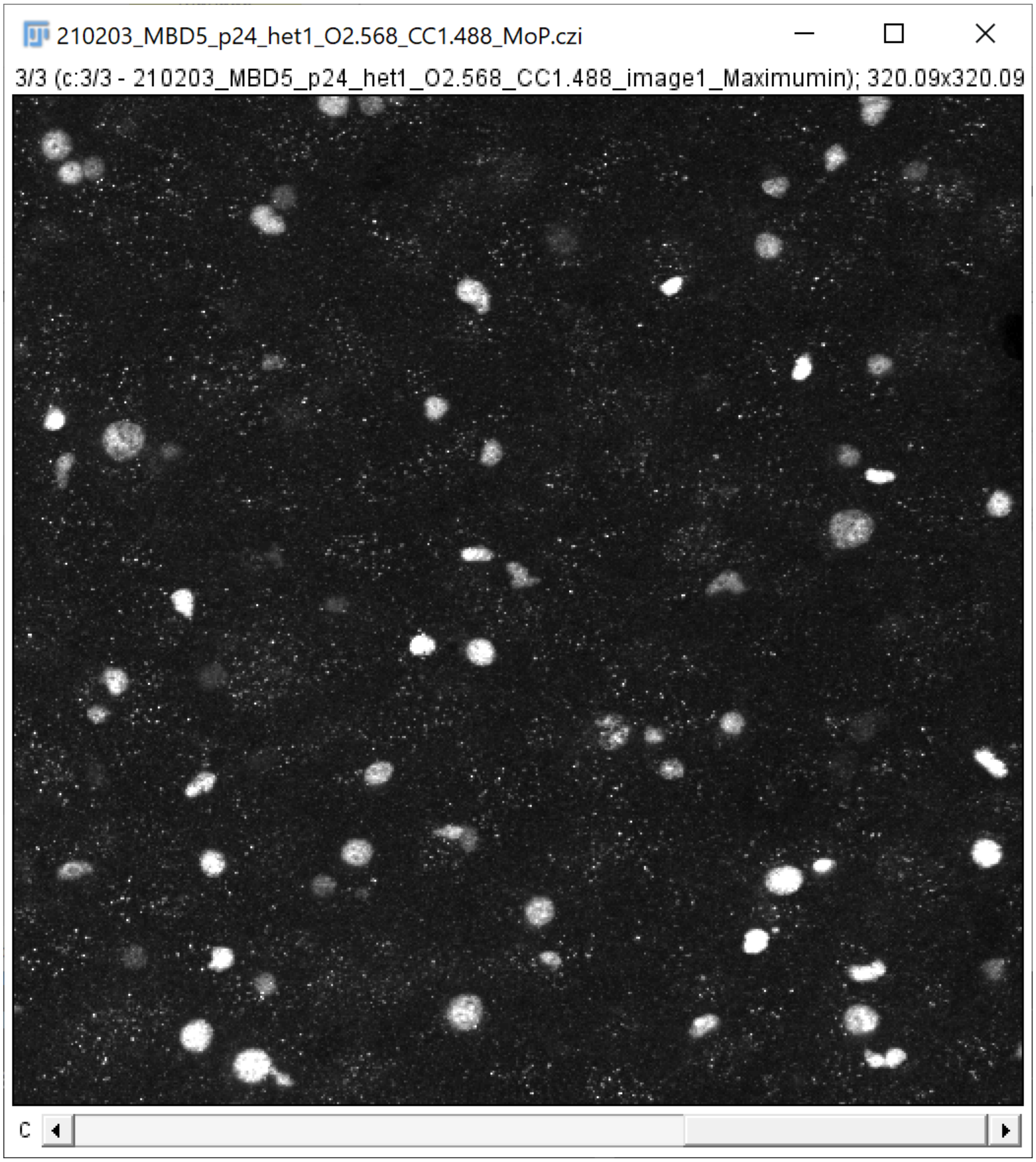
This looks pretty good!


IdentifyPrimaryObjects #8, image cycle #1
File
Subplots
Help
Input image, cycle #1
IdentifyPrimaryObjectsOlig
IdentifyPrimaryObjectsOlig outlines
0
200
400

# of accepted objects	51
10th pctile diameter	11.2 pixels
Median diameter	13.8 pixels
90th pctile diameter	17.8 pixels
Area covered by objects	3.2 %
Thresholding filter size	1.0
Threshold	0.0575
Declumping smoothing filter size	6.7
Maxima suppression size	6.7

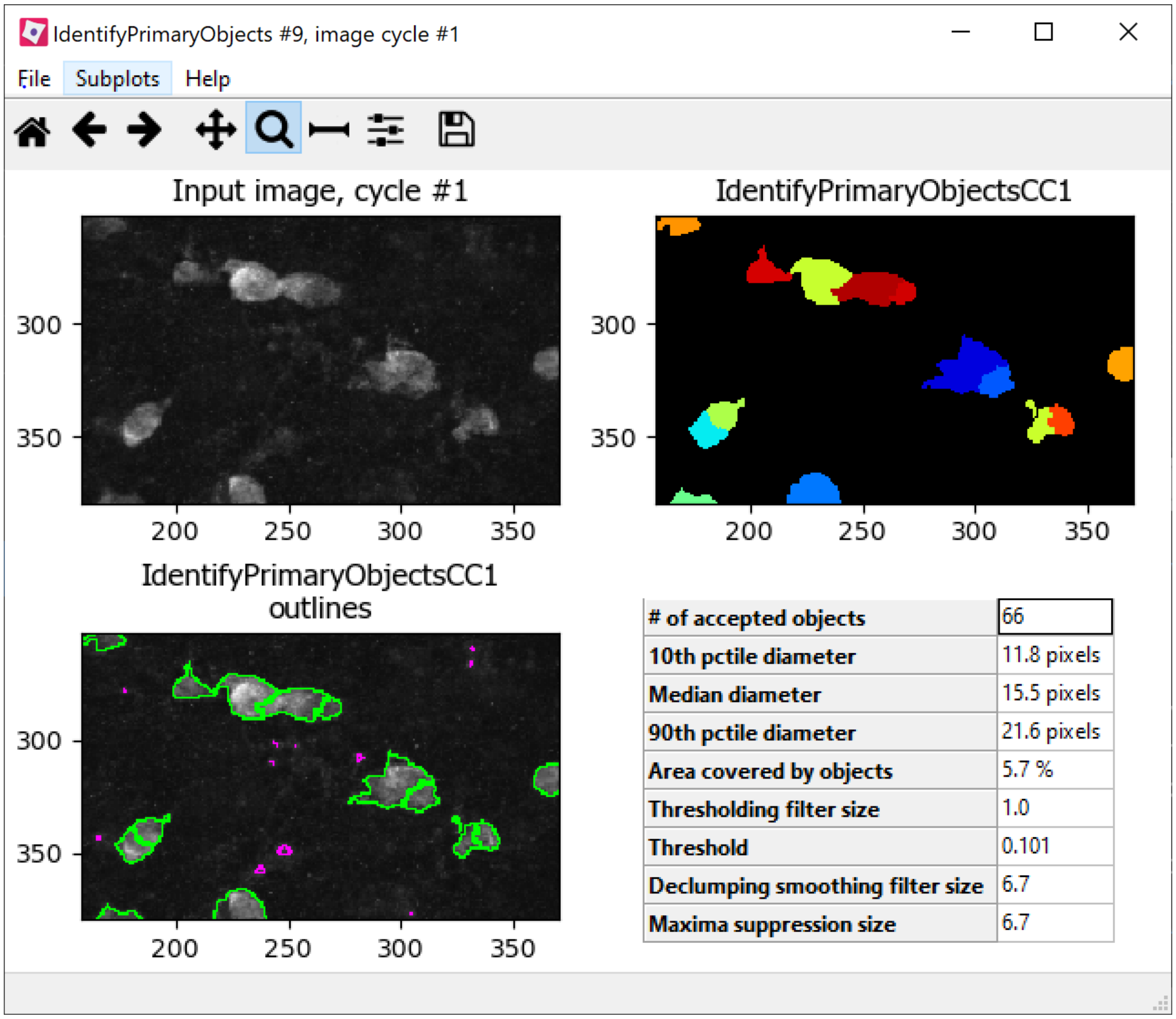
X: 204 Y: 96 Intensity: 0.0196

There are a few small speckles, but if we look at the channel and increase the brightness, these appears to widely dispersed and are likely just background noise.


210203_MBD5_p24_het1_O2.568_CC1.488_MoP.czi
3/3 (c:3/3 - 210203_MBD5_p24_het1_O2.568_CC1.488_image1_Maximumin); 320.09x320.09
c

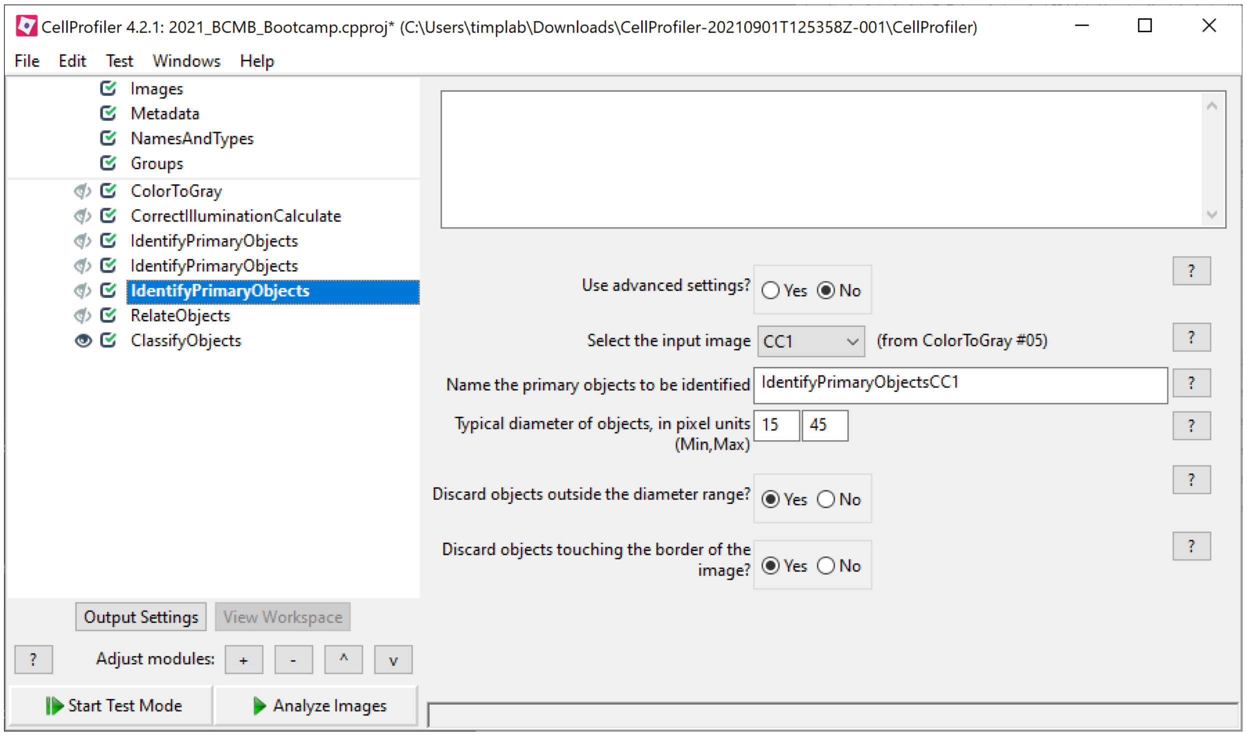
STEP 6

Now try to IdentifyPrimaryObjects for the CC1 channel. CC1, which actually stands for monoclonal antibody anti-adenomatous polyposis coli (APC) clone CC1, is a non-nuclear (primarily cytosolic/cytoskeletal/membrane) marker for mature OLs. As such we may have to adjust the diameter variables to capture the whole cell. For example, if we stick with the standard diameter variables we see that some of these cells appear to be prematurely split:


IdentifyPrimaryObjects #9, image cycle #1
File
Subplots
Help
Input image, cycle #1
IdentifyPrimaryObjectsCC1
IdentifyPrimaryObjectsCC1 outlines
300
350
200
250
300
350

# of accepted objects	66
10th pctile diameter	11.8 pixels
Median diameter	15.5 pixels
90th pctile diameter	21.6 pixels
Area covered by objects	5.7 %
Thresholding filter size	1.0
Threshold	0.101
Declumping smoothing filter size	6.7
Maxima suppression size	6.7

To keep things simple (there are definitely better and more sophisticated ways to do this), lets adjust the values of the Min/Max diameter up to avoid this splitting problem.



Use advanced settings? Yes [✓ No]

Select the input image: CC1 (from ColorToGray #05)

Name the primary objects to be identified: IdentifyPrimaryObjectsCC1

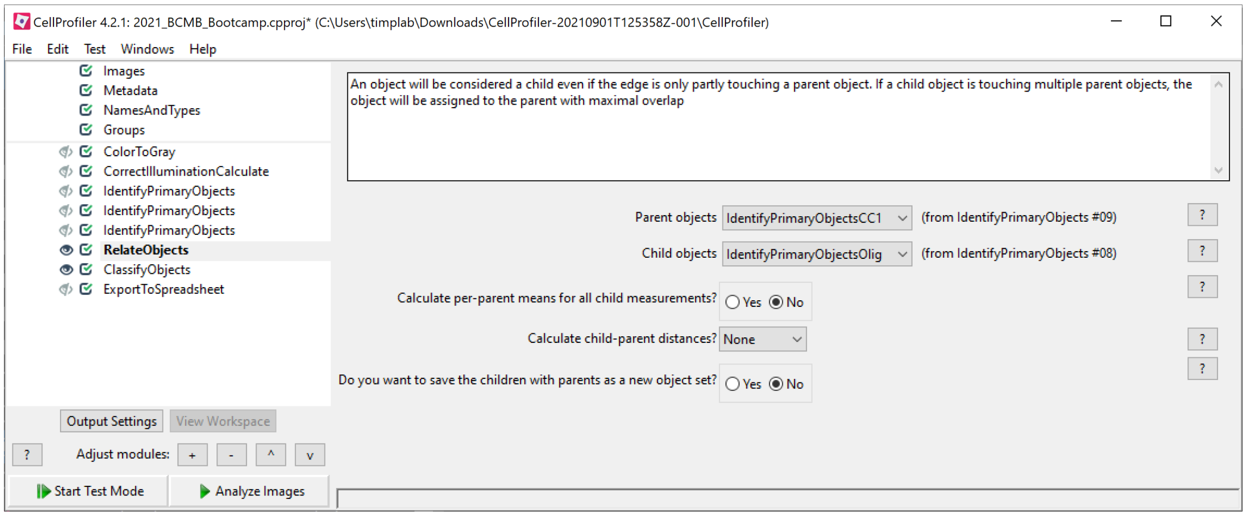
Typical diameter of objects, in pixel units (Min,Max): 15 45

Discard objects outside the diameter range? [✓ Yes] No

Discard objects touching the border of the image? [✓ Yes] No

STEP 7

Great! Now we have identified all of our components and we can start to get at our questions. We know how many CC1 positive cells were identified, but how manu dual positive (Olig2 and CC1) cells are there versus how many cells are only Olig2 positive (which could be another olig2 positive cell in the oligodendrocyte cell lineage)? First we have to establish a relationship between the Olig2 objects and the CC1 objects. We can do this by establishing a "parent-child" using the RelateObjects function. Our parent will be the CC1 objects and our child will be the Olig2 objects.



An object will be considered a child even if the edge is only partly touching a parent object. If a child object is touching multiple parent objects, the object will be assigned to the parent with maximal overlap

Parent objects: IdentifyPrimaryObjectsCC1 (from IdentifyPrimaryObjects #09)

Child objects: IdentifyPrimaryObjectsOlig (from IdentifyPrimaryObjects #08)

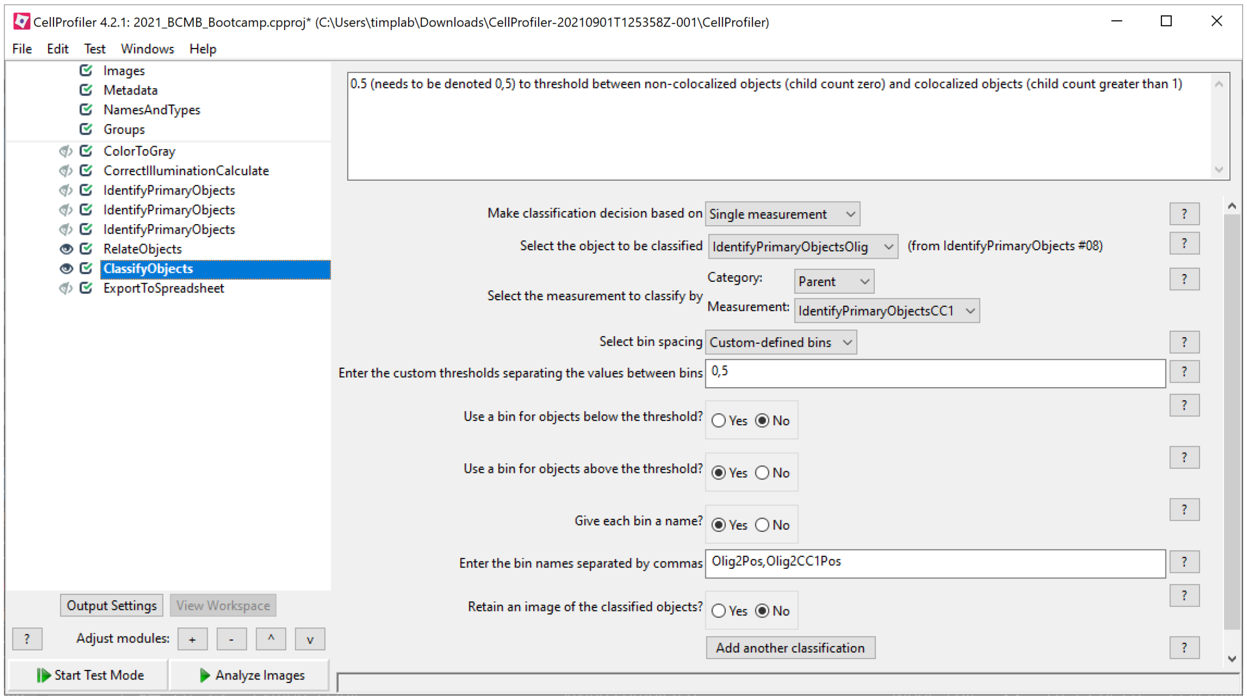
Calculate per-parent means for all child measurements? Yes [✓ No]

Calculate child-parent distances? None

Do you want to save the children with parents as a new object set? Yes [✓ No]

STEP 8

Now that we have established a relationship between these two objects we can use the ClassifyObjects module to bin each of the cells as either colocalized (Olig2+, CC1+) or not colocalized (Olig2+ only). We can set our object to classify as our Olig2 objects based on the presence or absence of its relationship with our parent object CC1. If we set our bin threshold as being 0.5 (or for some CellProfiler versions 0,5) then this will give us the cells that do not have a parent (i.e. returns 0) versus the Olig2 cells that do have a parent (i.e. return 1 or more). You can then also set these bin names to be "Olig2Pos,Olig2CC1Pos" to name your output.

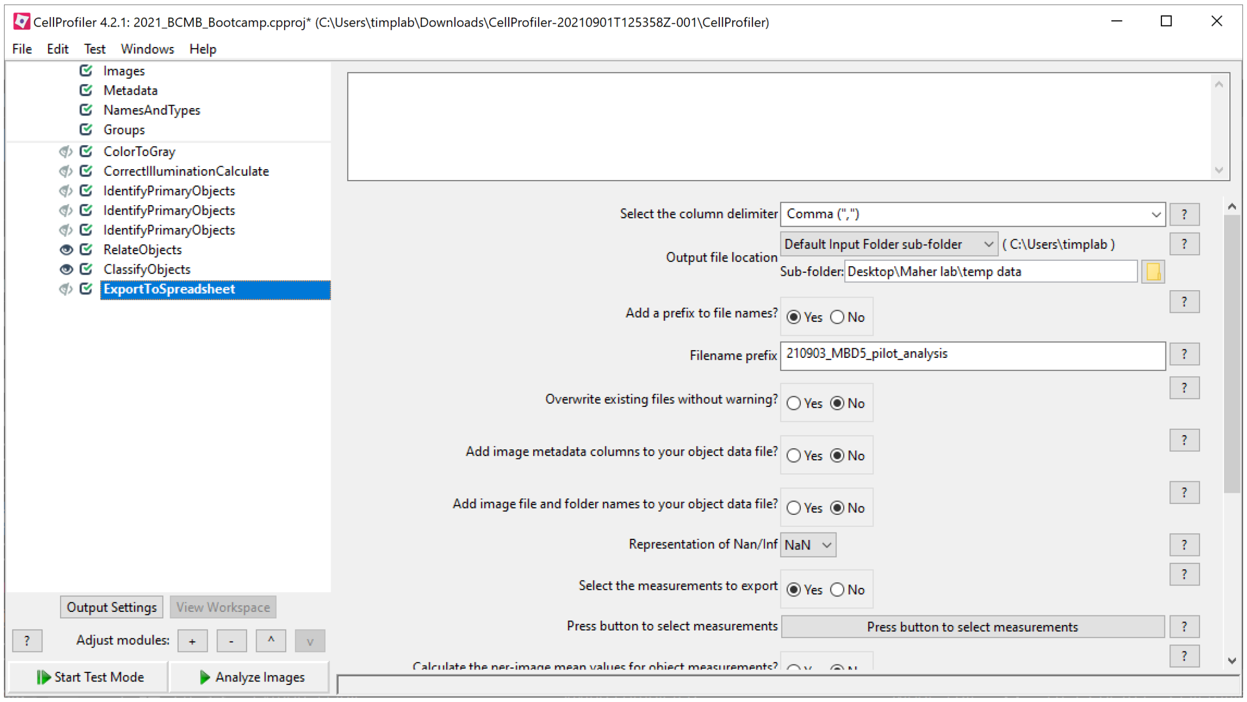

CellProfiler 4.2.1: 2021_BCMB_Bootcamp.cpproj* (C:\Users\timplab\Downloads\CellProfiler-20210901T125358Z-001\CellProfiler)

0.5 (needs to be denoted 0,5) to threshold between non-colocalized objects (child count zero) and colocalized objects (child count greater than 1)

Make classification decision based on: Single measurement
Select the object to be classified: IdentifyPrimaryObjectsOlig (from IdentifyPrimaryObjects #08)
Select the measurement to classify by: Category: Parent; Measurement: IdentifyPrimaryObjectsCC1
Select bin spacing: Custom-defined bins
Enter the custom thresholds separating the values between bins: 0,5
Use a bin for objects below the threshold? Yes [✓ No]
Use a bin for objects above the threshold? [✓ Yes] No
Give each bin a name? [✓ Yes] No
Enter the bin names separated by commas: Olig2Pos,Olig2CC1Pos
Retain an image of the classified objects? Yes [✓ No]

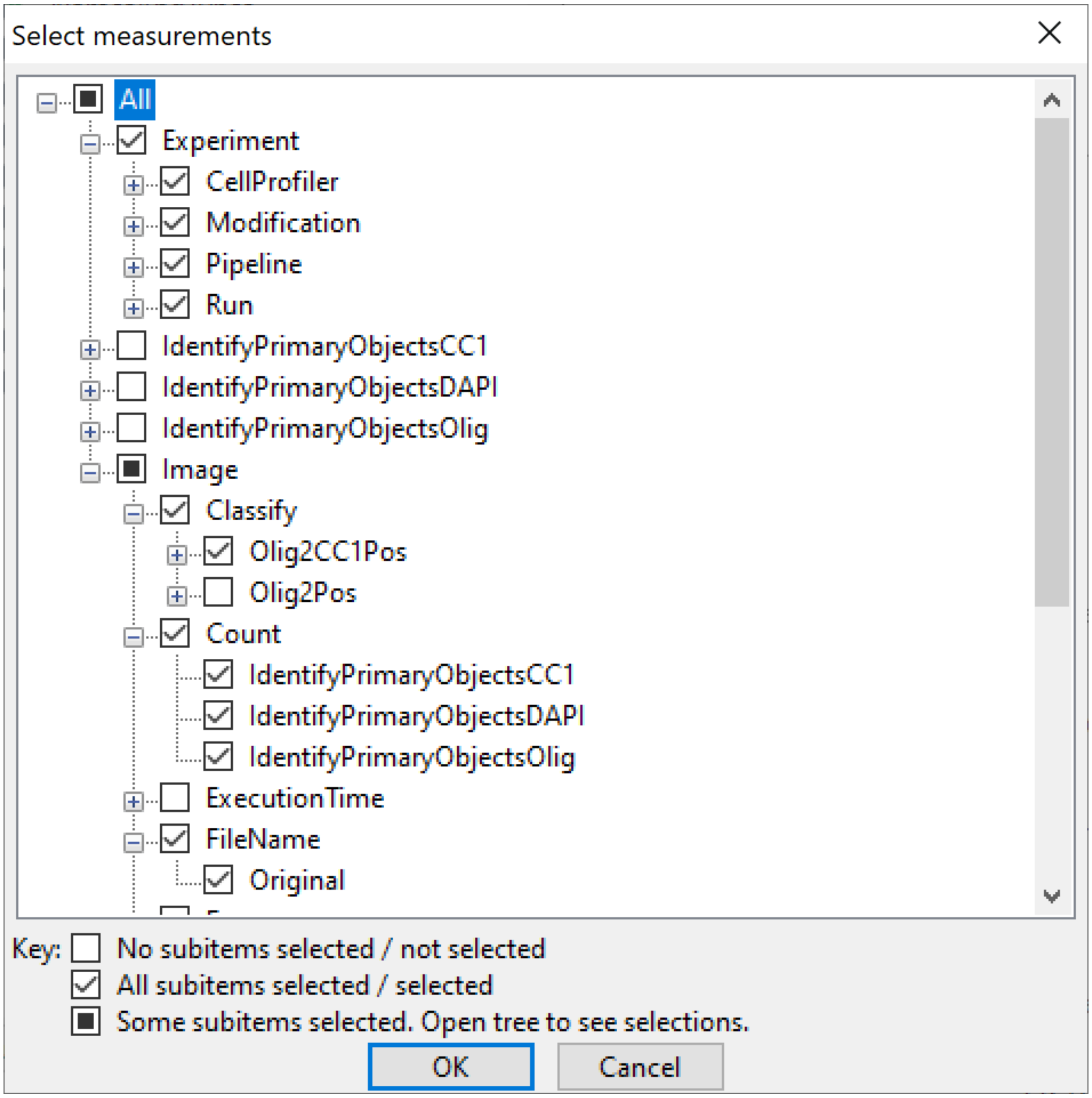
STEP 9

Good enough for now. Let's output our data into a spreadsheet so we could have a .csv file that we could use to do some analysis/tidy up/make plots! Add the ExportToSpreadsheet module to your pipeline, specify the output folder, and base name for your file outputs.


CellProfiler 4.2.1: 2021_BCMB_Bootcamp.cpproj* (C:\Users\timplab\Downloads\CellProfiler-20210901T125358Z-001\CellProfiler)

Select the column delimiter: Comma (",")
Output file location: Default Input Folder sub-folder (C:\Users\timplab); Sub-folder: Desktop\Maher lab\temp data
Add a prefix to file names? [✓ Yes] No
Filename prefix: 210903_MBD5_pilot_analysis
Overwrite existing files without warning? Yes [✓ No]
Add image metadata columns to your object data file? Yes [✓ No]
Add image file and folder names to your object data file? Yes [✓ No]
Representation of Nan/Inf: NaN
Select the measurements to export: [✓ Yes] No
Press button to select measurements: Press button to select measurements
Calculate the per-image mean values for object measurements?

We can select the information we would like to export which will be our Experiment, Classification of Olig2CC1Pos cells, each of the object counts, and the original filename.


Select measurements

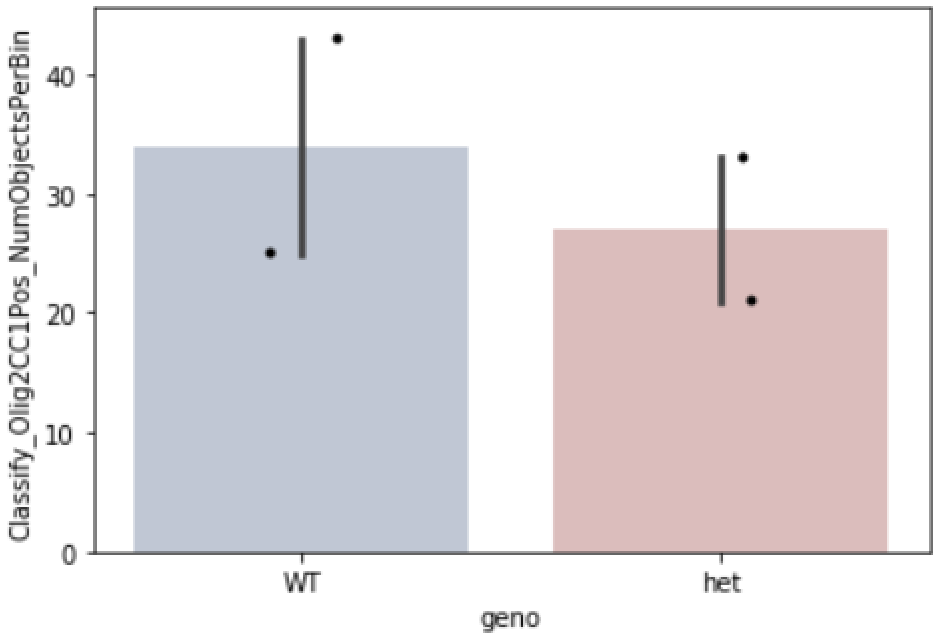
- ▣ All
  - ☑ Experiment
    - ☑ CellProfiler
    - ☑ Modification
    - ☑ Pipeline
    - ☑ Run
  - ☐ IdentifyPrimaryObjectsCC1
  - ☐ IdentifyPrimaryObjectsDAPI
  - ☐ IdentifyPrimaryObjectsOlig
  - ▣ Image
    - ☑ Classify
      - ☑ Olig2CC1Pos
      - ☐ Olig2Pos
    - ☑ Count
      - ☑ IdentifyPrimaryObjectsCC1
      - ☑ IdentifyPrimaryObjectsDAPI
      - ☑ IdentifyPrimaryObjectsOlig
    - ☐ ExecutionTime
    - ☑ FileName
      - ☑ Original

Key: ☐ No subitems selected / not selected
☑ All subitems selected / selected
▣ Some subitems selected. Open tree to see selections.

So running this will export all of our data to an excel file that is comma separated. Run this analysis to check the output.

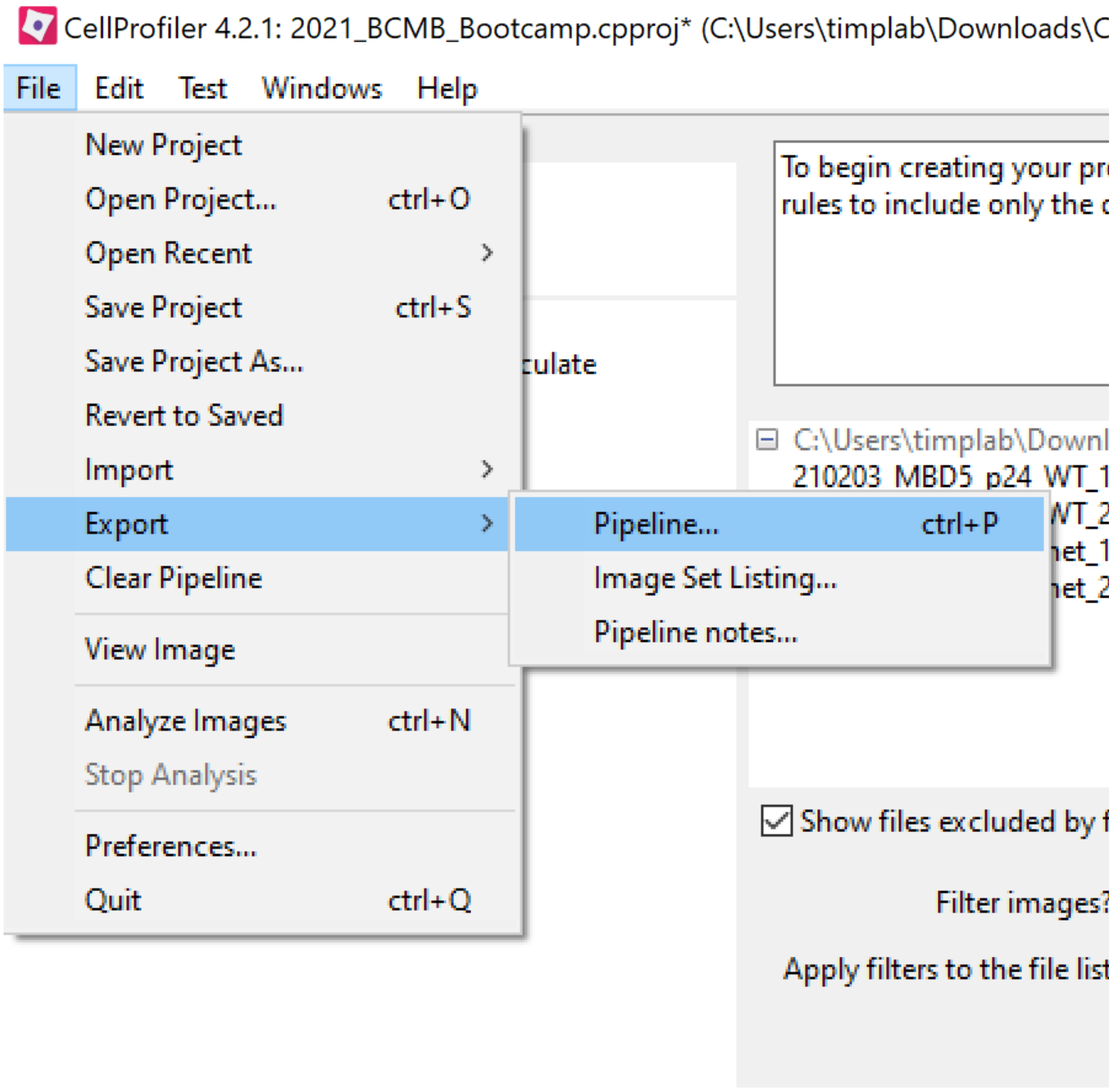
STEP 10

If your pipeline ran without error and the output file looks like it contains all of the relevant information, load in the rest of the image data (the remaining three files) and run the pipeline again! Now we can upload this to our jupyter lab and start plotting the data! This is a very small dataset, but it looks like it may be trending which is a start!



STEP 11

Finally, you can export your CellProfiler pipeline by going to File > Export > Pipeline. Save your analysis pipeline which you can use again to analyze the same type of data once you aquire more images.



To begin creating your pr
rules to include only the c

C:\Users\timplab\Downl
210203_MBD5_p24_WT_1

Exercise

Generate a pipeline for the provided dataset. These new images are primary culturs of oligodendrocyte precursor cells (OPCs) from our MBD5 mouse model. The cells were immunostained at different timepoints staining for BrdU incorporated into replicating DNA (S phase) and ki67 (nuclear mitotic marker) to assess cellular proliferation. The channels are assigned as follows: 1: DAPI 2: ki67 3: BrdU 4: Olig2

```
In [ ]:
```