

## Introduction

In the [original Bodenmiller](#) or [our modified Bodenmiller](#) segmentation pipeline, nuclei and cytoplasmic markers are used to determine probabilities for the a) nucleus, b) cytoplasm, and c) background using Ilastik. When it comes to the actual segmentation in CellProfiler, the Primary Objects are identified (i.e. nuclei) and then the cell border is expanded out through the cytoplasm (secondary objects) until they conflict with neighbouring cell borders. This approach has limitations, as it a) often does not accurately segment based on the edge of the cell, and b) is difficult to visually assess until the data is examined in HistoCat. An alternative approach is to use cytoplasmic markers to define cell boundaries in Ilastik, which can then be used to create cell masks. These masks can then be filtered/refined based on size or other features. This approach is potentially advantageous where nuclear staining is sub-optimal, or in cases where the density of cells in the image is extremely high.

## Software installation

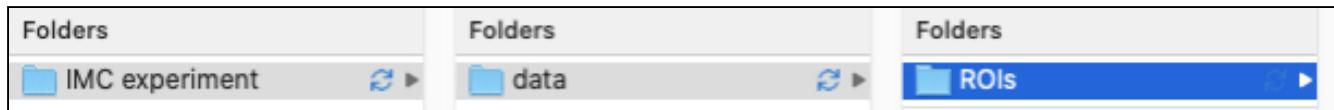
To use this workflow, you will need to install [R/RStudio](#), the [spatial branch of Spectre](#), as well as [Ilastik](#).

- If you are unfamiliar with using R, RStudio, or Spectre, check out our '[getting started](#)' page.

## Setup

### Experiment folder

Create a folder for your experiment, and then create a subfolder called 'data', another subfolder under that one called 'ROIs'.



### Download CellProfiler template

Go to: <https://github.com/ImmuneDynamics/Spectre/tree/spatial> and download the repository.

This branch is 9 commits ahead of master.

**tomashhurst** Update README.md

- R Updates
- data Spatial fun...
- docker Merge pull...
- image Add files via upload

7 months ago

**About**

A computational toolkit in R for the integration, exploration, and analysis of high-dimensional single-cell cytometry and imaging data.

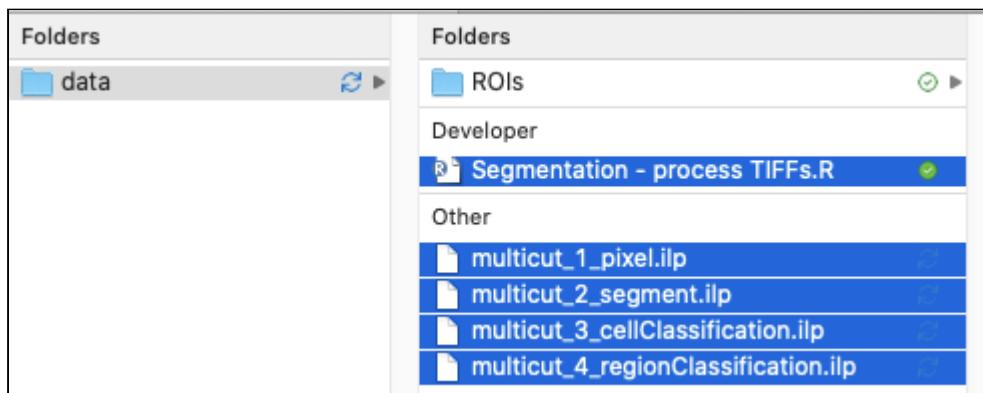
<https://github.com/ImmuneDynamics/Spectre>

[Clone](#) [HTTPS](#) [SSH](#) [GitHub CLI](#)

[Open with GitHub Desktop](#)

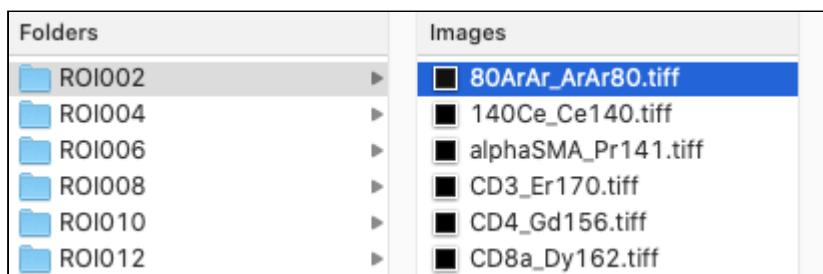
[Download ZIP](#)

Unzip the file, find the llastik template files (under 'segmentation') and add them to your experiment folder under 'data'.



### Export TIFF files from the original MCD files

You can do this using either [MCD Viewer \(on windows\)](#) or [HistoCat++ \(on Mac\)](#). Each ROI should be a folder, containing single-page TIFFs (one per channel for that ROI).



Under 'ROIs', place your ROI folders

Folders	Folders	Images
ROIs	ROI02	80ArAr_ArAr80.tiff
Developer	ROI04	140Ce_Ce140.tiff
Segmentation - process TIFFs.R	ROI06	alphaSMA_Pr141.tiff
Other	ROI08	CD3_Er170.tiff
multicut_1_pixel.ilp	ROI10	CD4_Gd156.tiff
multicut_2_segment.ilp	ROI12	CD8a_Dy162.tiff
multicut_3_cellClassification.ilp		CD11b_Sm149.tiff
multicut_4_regionClassification.ilp		CD20_Dy161.tiff
		CD45_Sm152.tiff
		CD68_Tb159.tiff
		CollagenI_Tm169.tiff
		DNA1_Ir191.tiff
		DNA3_Ir193.tiff
		HistoneH3_Yb176.tiff
		VIM_Nd143.tiff

## 1. Spectre (R) - process raw TIFF files

Open the 'Spectre - process TIFFs.R' file in **RStudio**.

 If you are unfamiliar with using R, RStudio, or Spectre, check out our '[getting started](#)' page.

### Setup packages and directories

First, load the required packages.

```
### Load packages

library(Spectre)

package.check(type = 'spatial')
package.load(type = 'spatial')
```

Set some directories

```
### Set PrimaryDirectory

dirname(rstudioapi::getActiveDocumentContext()$path) # Finds the directory where
this script is located
setwd(dirname(rstudioapi::getActiveDocumentContext()$path)) # Sets the working directory to
where the script is located
getwd()
PrimaryDirectory <- getwd()
PrimaryDirectory
```

```
### Set InputDirectory

setwd(PrimaryDirectory)
setwd("ROIs/")
InputDirectory <- getwd()
InputDirectory
```

```
### Create directory for Ilastik HDF5 files
```

```

setwd(PrimaryDirectory)
dir.create("masks")
setwd("masks")
MaskDirectory <- getwd()
MaskDirectory

### Create directory for CROPPED Ilastik HDF5 files

setwd(PrimaryDirectory)
dir.create("cropped")
setwd("cropped")
CroppedDirectory <- getwd()
CroppedDirectory

```

## Check ROIs and TIFFs

First, we will create a list of the ROI (i.e. folder) names.

```

### Initialise the spatial data object with channel TIFF files

setwd(InputDirectory)

rois <- list(dirs(full.names = FALSE, recursive = FALSE))
as.matrix(rois)

```

The result will look something like this.

```

[,1]
[1,] "ROI002"
[2,] "ROI004"
[3,] "ROI006"
[4,] "ROI008"
[5,] "ROI010"
[6,] "ROI012"

```

Now we are going to create a list of the channels (i.e. TIFFs) within each ROI

```

tiff.list <- list()

for(i in rois){
  setwd(InputDirectory)
  setwd(i)
  tiff.list[[i]] <- list.files(getwd())
}

```

We can view this as a table.

```
t(as.data.frame(tiff.list))
```

The results should look similar to this. If the names are jumbled, it means that the channels are not consistent between ROIs.

[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[,7]	[,8]				
ROI002	"CD11b_Sm149.tiff"	"CD20_Dy161.tiff"	"CD3_Er170.tiff"	"CD4_Gd156.tiff"	"CD45_Sm152.tiff"
"CD8a_Dy162.tiff"	"DNA1_Ir191.tiff"	"DNA3_Ir193.tiff"			

```
ROI004 "CD11b_Sm149.tif" "CD20_Dy161.tif" "CD3_Er170.tif" "CD4_Gd156.tif" "CD45_Sm152.tif"
"CD8a_Dy162.tif" "DNA1_Ir191.tif" "DNA3_Ir193.tif"
ROI006 "CD11b_Sm149.tif" "CD20_Dy161.tif" "CD3_Er170.tif" "CD4_Gd156.tif" "CD45_Sm152.tif"
"CD8a_Dy162.tif" "DNA1_Ir191.tif" "DNA3_Ir193.tif"
ROI008 "CD11b_Sm149.tif" "CD20_Dy161.tif" "CD3_Er170.tif" "CD4_Gd156.tif" "CD45_Sm152.tif"
"CD8a_Dy162.tif" "DNA1_Ir191.tif" "DNA3_Ir193.tif"
ROI010 "CD11b_Sm149.tif" "CD20_Dy161.tif" "CD3_Er170.tif" "CD4_Gd156.tif" "CD45_Sm152.tif"
"CD8a_Dy162.tif" "DNA1_Ir191.tif" "DNA3_Ir193.tif"
ROI012 "CD11b_Sm149.tif" "CD20_Dy161.tif" "CD3_Er170.tif" "CD4_Gd156.tif" "CD45_Sm152.tif"
"CD8a_Dy162.tif" "DNA1_Ir191.tif" "DNA3_Ir193.tif"
```

## **Read in TIFF files and create spatial objects**

Now that we have performed our checks, we are going to read in the TIFF files create some spatial data objects.

```
setwd(InputDirectory)
spatial.dat <- read.spatial.files(dir = getwd())
```

Once completed, we can check the structure:

```
str(spatial.dat, 3)
```

This should return a list of the ROIs, and each will contain a raster stack.

```
List of 6
$ ROI002:Formal class 'spatial' [package "Spectre"] with 3 slots
... ..@ RASTERS:Formal class 'RasterStack' [package "raster"] with 11 slots
... ..@ MASKS : list()
... ..@ DATA   : list()
$ ROI004:Formal class 'spatial' [package "Spectre"] with 3 slots
... ..@ RASTERS:Formal class 'RasterStack' [package "raster"] with 11 slots
... ..@ MASKS : list()
... ..@ DATA   : list()
$ ROI006:Formal class 'spatial' [package "Spectre"] with 3 slots
... ..@ RASTERS:Formal class 'RasterStack' [package "raster"] with 11 slots
... ..@ MASKS : list()
... ..@ DATA   : list()
$ ROI008:Formal class 'spatial' [package "Spectre"] with 3 slots
... ..@ RASTERS:Formal class 'RasterStack' [package "raster"] with 11 slots
... ..@ MASKS : list()
... ..@ DATA   : list()
$ ROI010:Formal class 'spatial' [package "Spectre"] with 3 slots
... ..@ RASTERS:Formal class 'RasterStack' [package "raster"] with 11 slots
... ..@ MASKS : list()
... ..@ DATA   : list()
$ ROI012:Formal class 'spatial' [package "Spectre"] with 3 slots
... ..@ RASTERS:Formal class 'RasterStack' [package "raster"] with 11 slots
... ..@ MASKS : list()
... ..@ DATA   : list()
```

We can check which rasters are present in the first ROI.

```
spatial.dat[[1]]@RASTERSRS
```

```
class      : RasterStack
```

```

dimensions : 501, 500, 250500, 8 (nrow, ncol, ncell, nlayers)
resolution : 1, 1 (x, y)
extent      : 0, 500, 0, 501 (xmin, xmax, ymin, ymax)
crs        : NA
names       : CD11b_Sm149, CD20_Dy161, CD3_Er170, CD4_Gd156, CD45_Sm152, CD8a_Dy162, DNA1_Ir191,
DNA3_Ir193
min values : 0, 0, 0, 0, 0, 0, 0,
0
max values : 33, 779, 41, 40, 734, 109, 1670,
3007

```

## Create HDF5 file for Ilastik

Now we will create some two sets of HDF5 files for each ROI. The first is HDF5 files of the full ROI, and the second is some cropped areas HDF5 files. The cropped HDF5 files will be used to *train* the pixel classifier in Ilastik, and then this will be used to classify pixels in the full HDF5 files (which will allow us to create our masks).

First, we'll review the raster names (from the first ROI).

```

nms <- names(spatial.dat[[1]]$RASTERS)
as.matrix(nms)

```

```

[,1]
[1,] "CD11b_Sm149"
[2,] "CD20_Dy161"
[3,] "CD3_Er170"
[4,] "CD4_Gd156"
[5,] "CD45_Sm152"
[6,] "CD8a_Dy162"
[7,] "DNA1_Ir191"
[8,] "DNA3_Ir193

```

We can select which channels we would like to include in the HDF5 files we will use for Ilastik. Typically here we want to include markers that are helpful for segmenting cells. In this example, we are going to include channels 3 (*alphaSMA\_Pr141*) to 15 (*VIM\_Nd143*).

```

for.ilastik <- nms[c(1:8)]
as.matrix(for.ilastik)

```

```

[,1]
[1,] "CD11b_Sm149"
[2,] "CD20_Dy161"
[3,] "CD3_Er170"
[4,] "CD4_Gd156"
[5,] "CD45_Sm152"
[6,] "CD8a_Dy162"
[7,] "DNA1_Ir191"
[8,] "DNA3_Ir193

```

**Firstly, we will export HDF5 files of the full ROIs.**

```

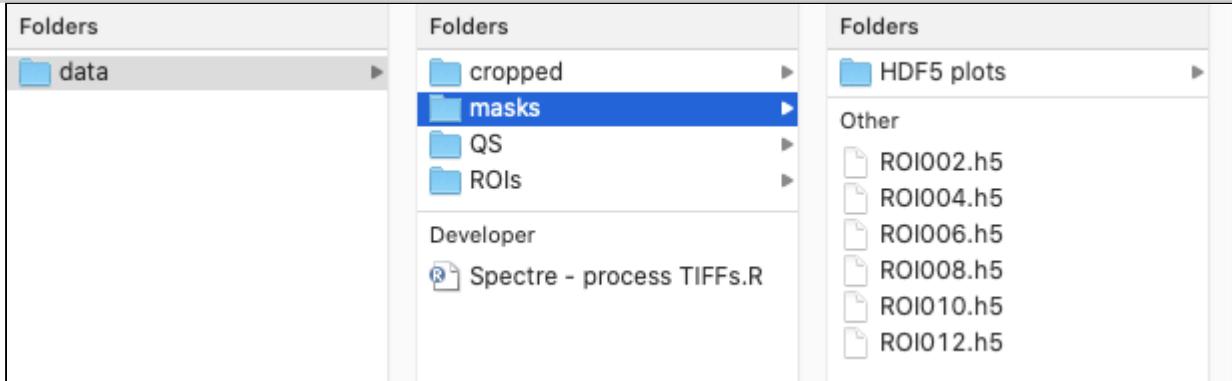
## Whole ROIs for Ilastik

setwd(MaskDirectory)
write.hdf5(dat = spatial.dat,
           channels = for.ilastik,

```

```
merge.channels = merge.channels,
plots = FALSE)

fwrite(data.table('Channels' = for.ilastik), 'ilastik.channels.csv')
```



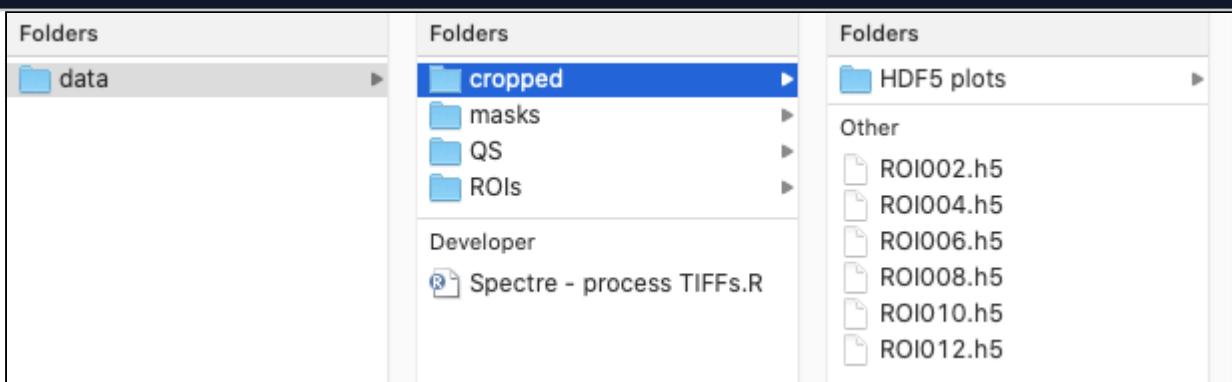
### Secondly, we will export some randomly cropped HDF5 files of the ROIs.

In this case, we are going to create random cropped areas of 350x300 pixels. You can make these larger or smaller if you wish.

```
## Cropped ROIs to train Ilastik

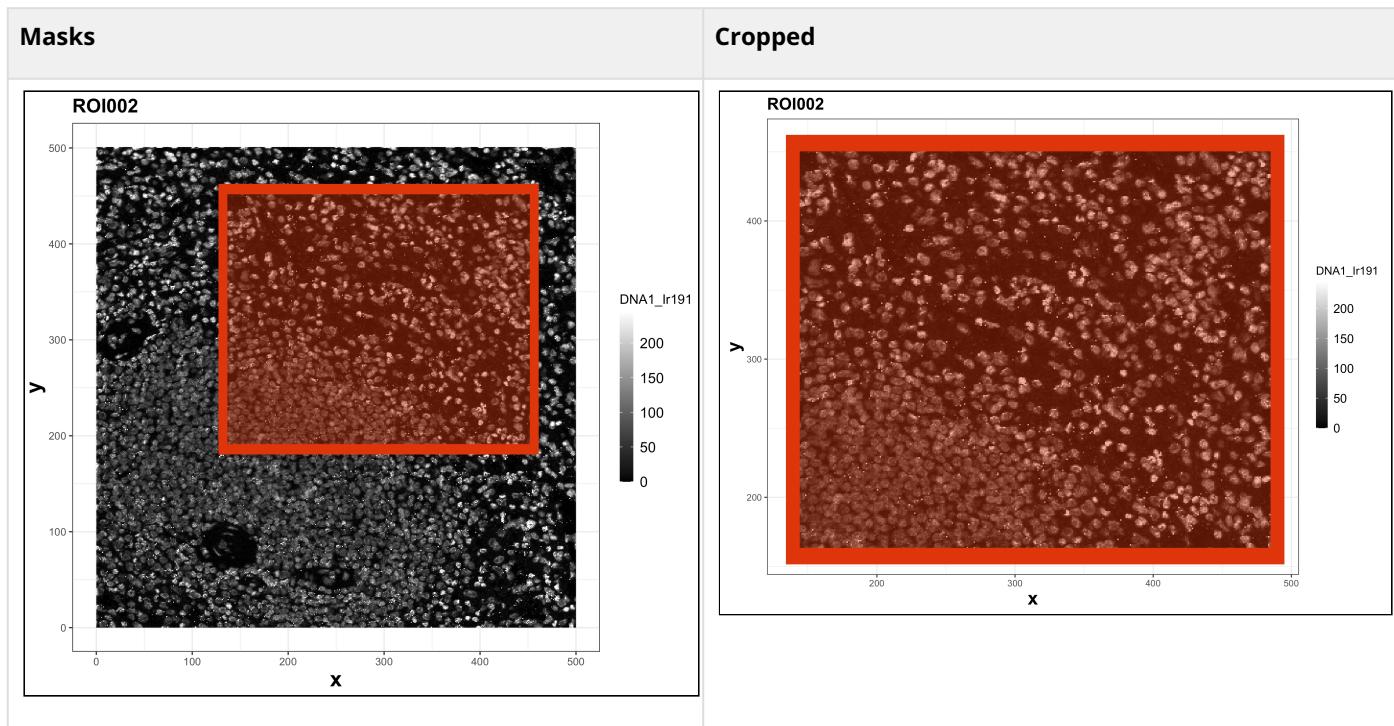
setwd(CroppedDirectory)
write.hdf5(dat = spatial.dat,
           channels = for.ilastik,
           merge.channels = merge.channels,
           random.crop.x = 350,
           random.crop.y = 300,
           plots = TRUE)

fwrite(data.table('Channels' = for.ilastik), 'ilastik.channels.csv')
```



You can also check the 'HDF5 plots' folder in both the 'masks' and 'cropped' folders.

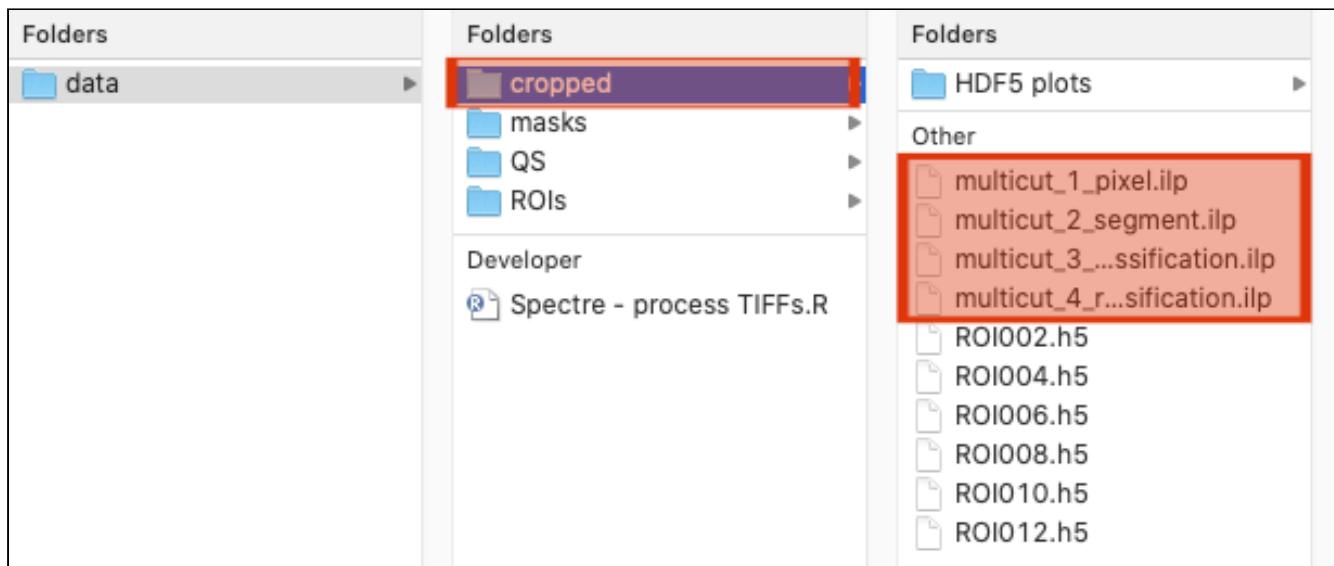
Masks	Cropped
Representative plots of the full ROI area	Representative plots of the cropped ROI area used for training



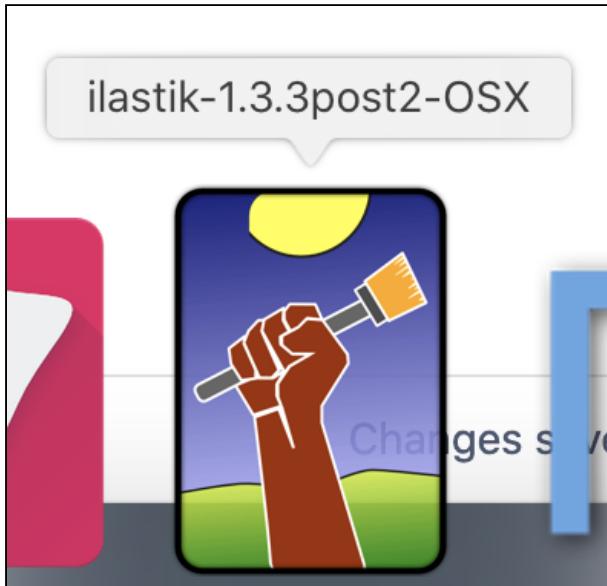
## 2. Ilastik - pixel classification of cell borders

**Copy the template Ilastik files into the 'cropped' folder (NOT the masks folder)**

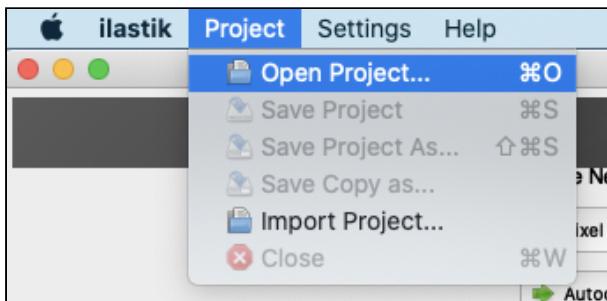
Moving these files is not strictly necessary, but is convenient.



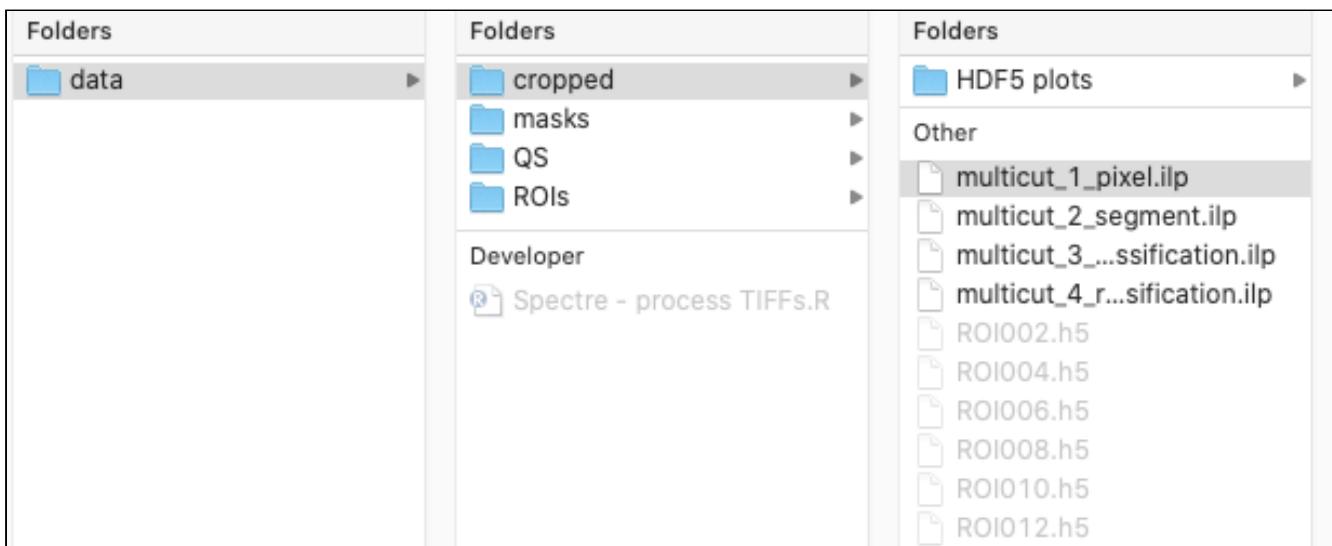
**Open Ilastik**



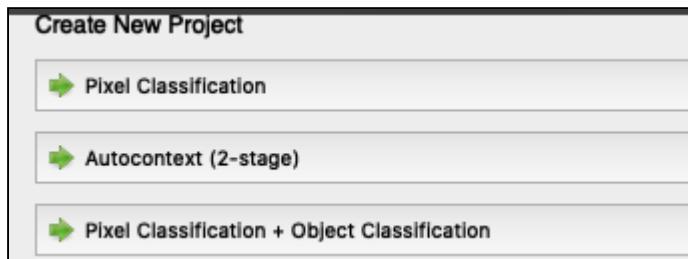
Go to Project / Open Project...



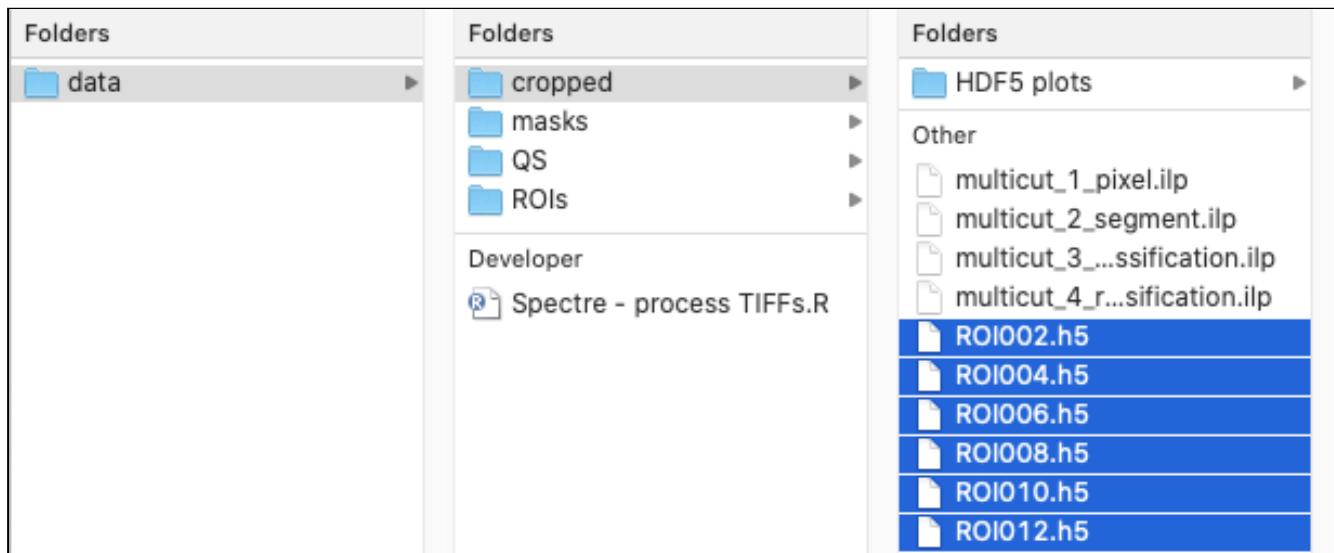
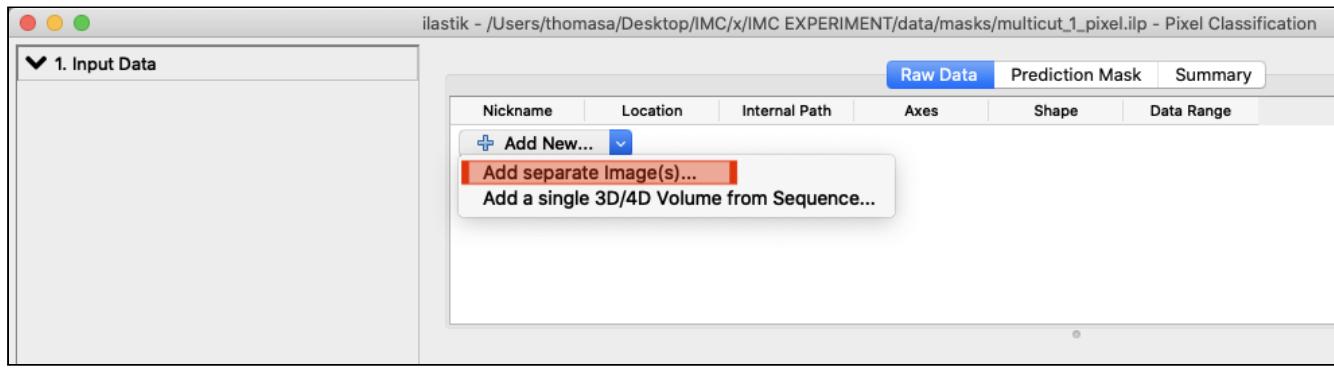
And select the 'multicut\_1\_pixel.ilp' file.



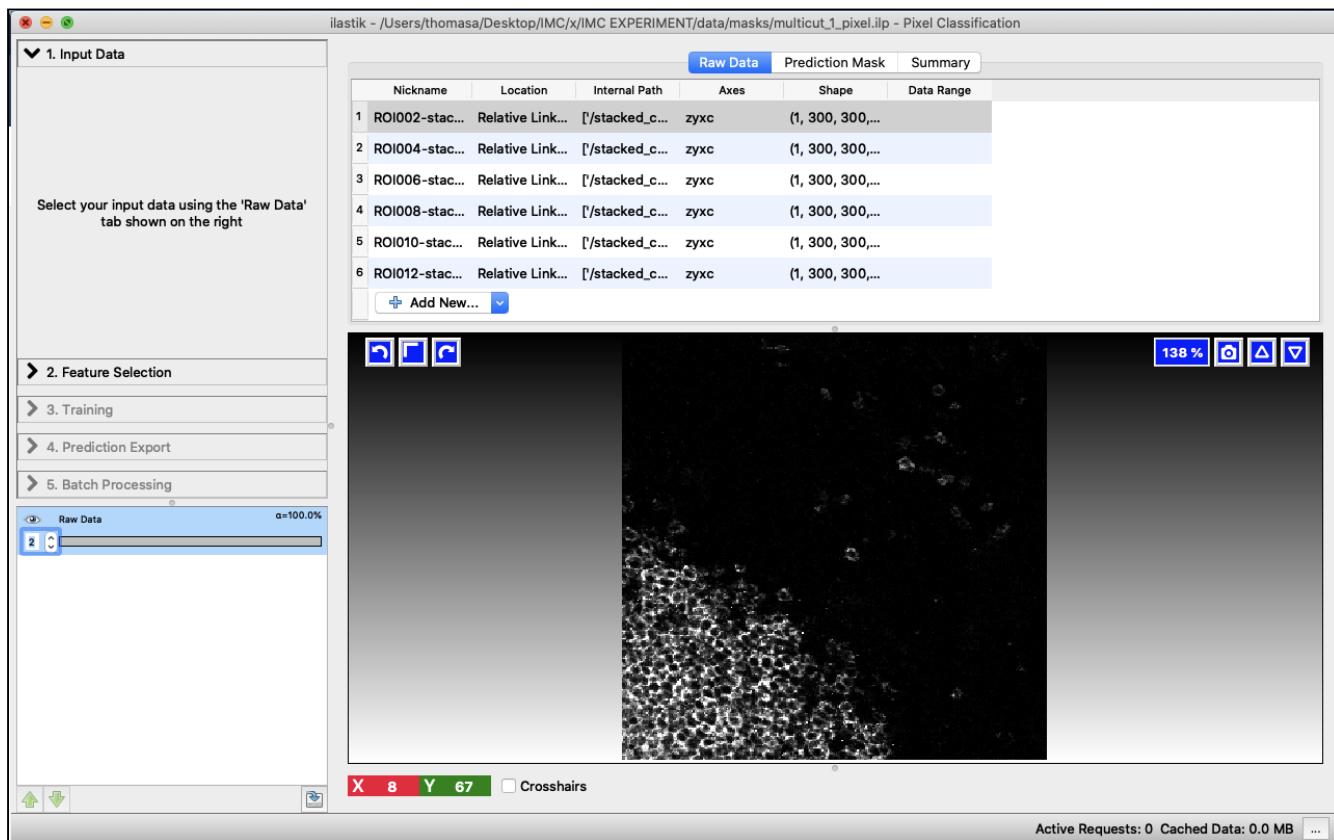
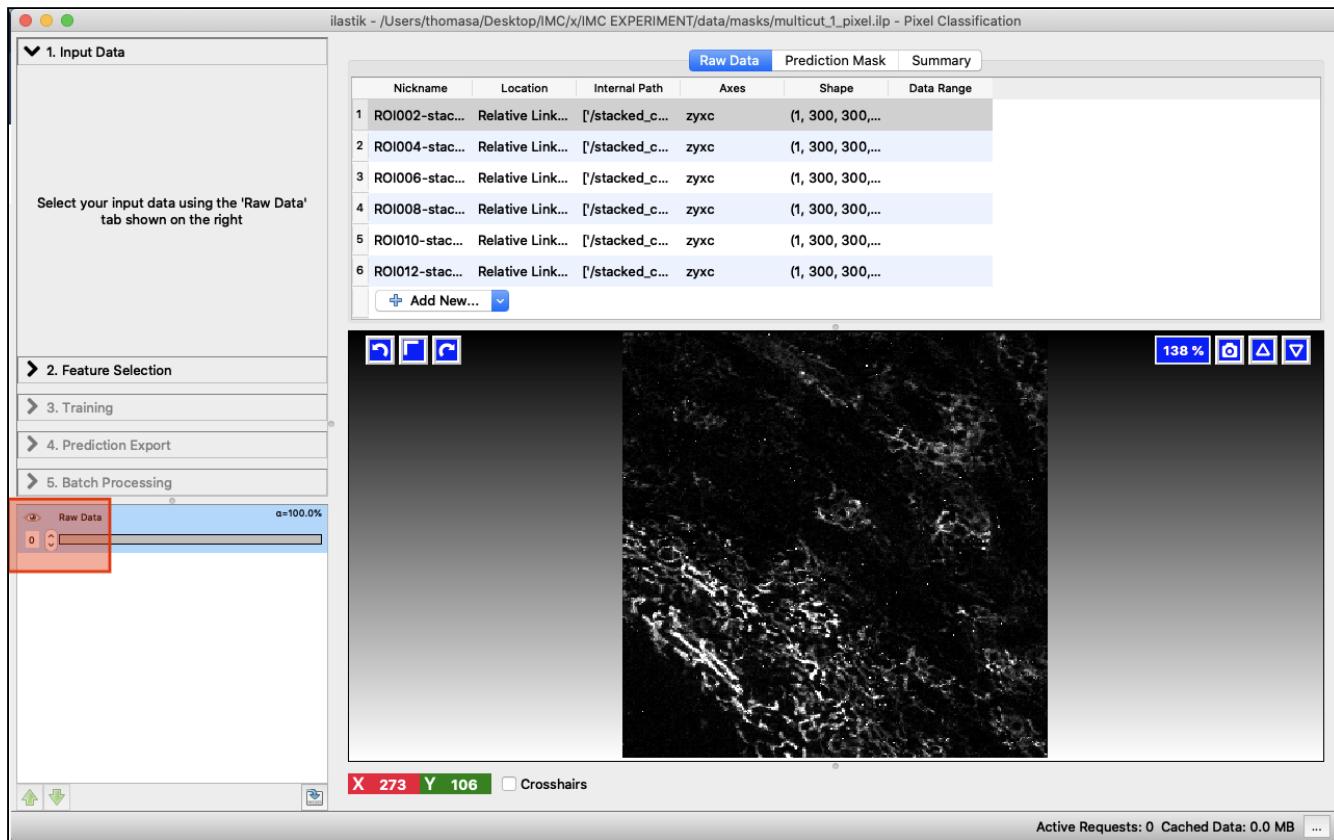
Alternatively, create and save a new Ilastik file of the 'pixel classification' type.



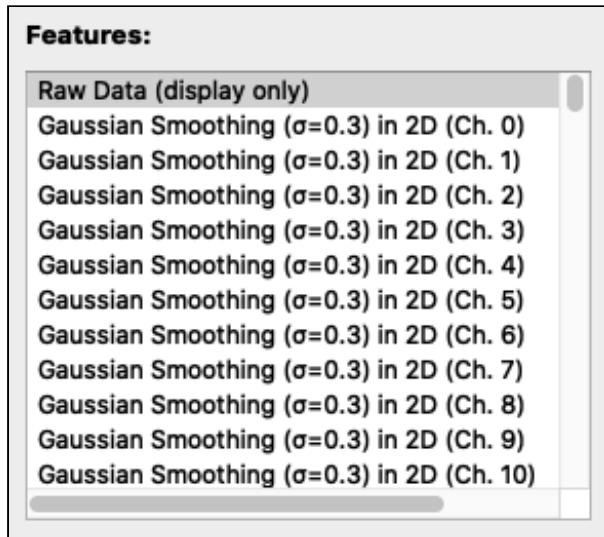
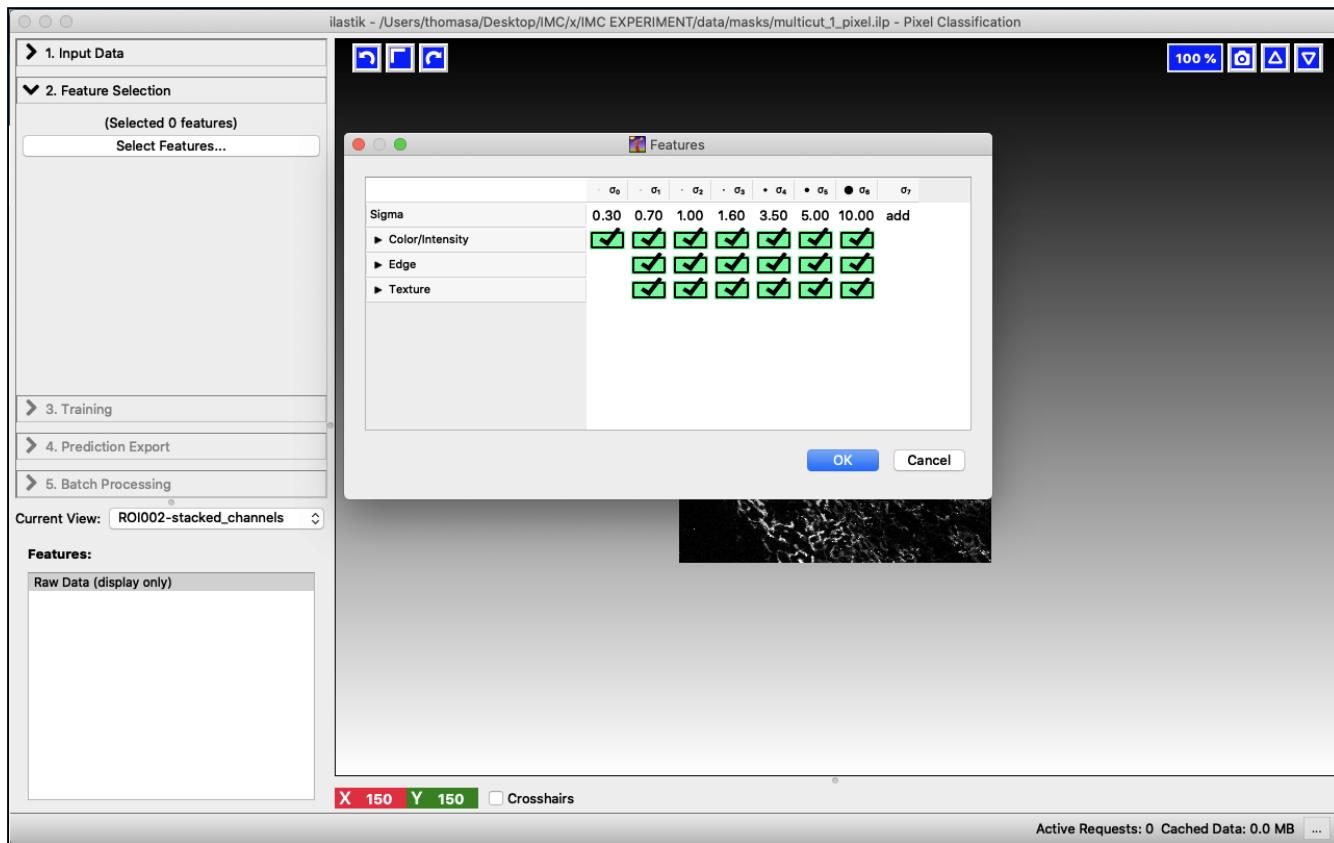
Add the HDF5 (.h5) files from the 'cropped' folder.



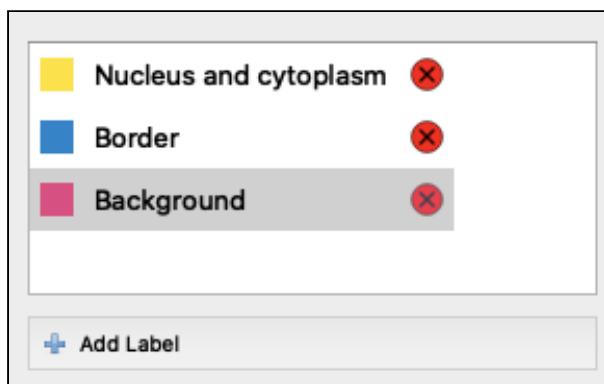
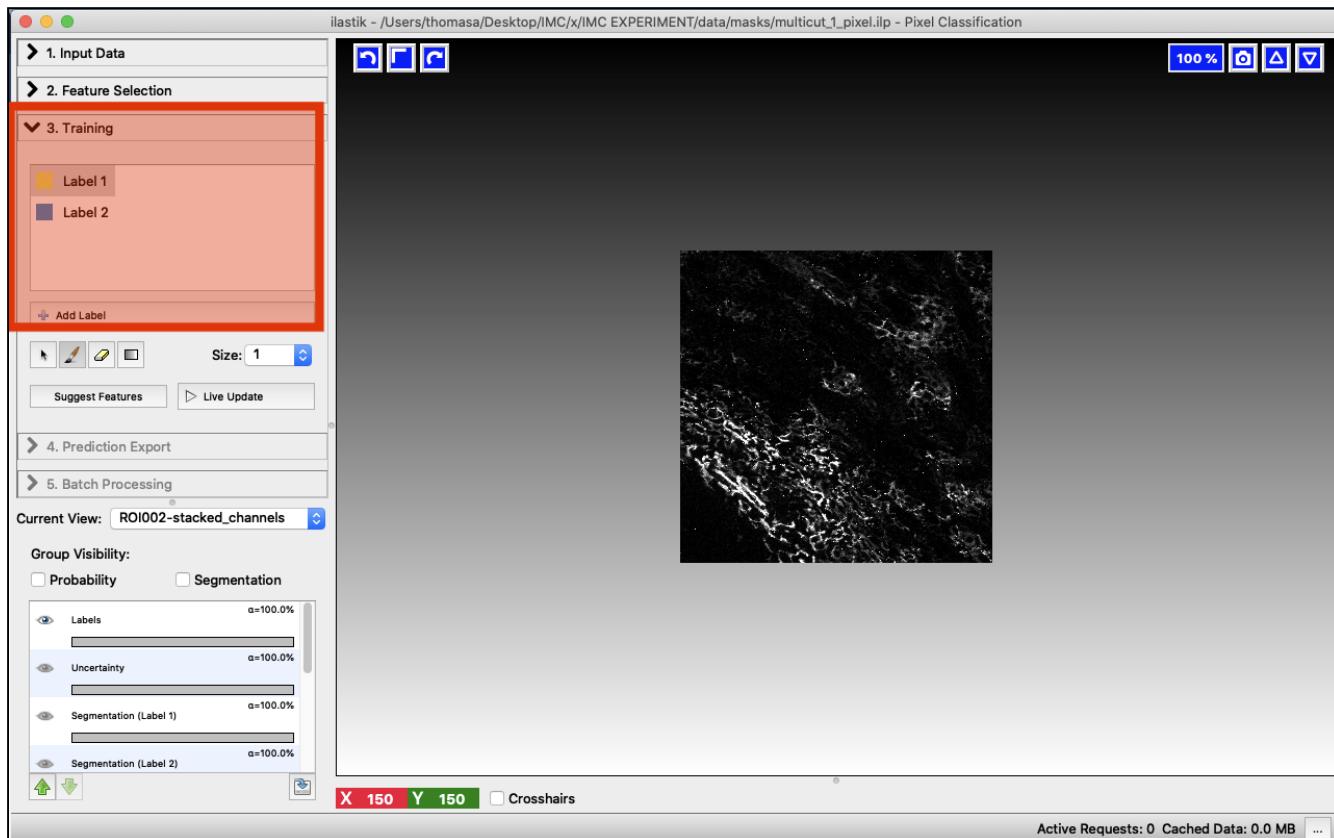
These should load with no changes to the preference required. You can flick through the different layers (channels) using the up and down arrows highlighted in red. The first channel TIFF will be '0', and will increase from there. You can check the files in the HDF5 cropped plots directory.



Go to '2. Feature selection' and select all the feature options.

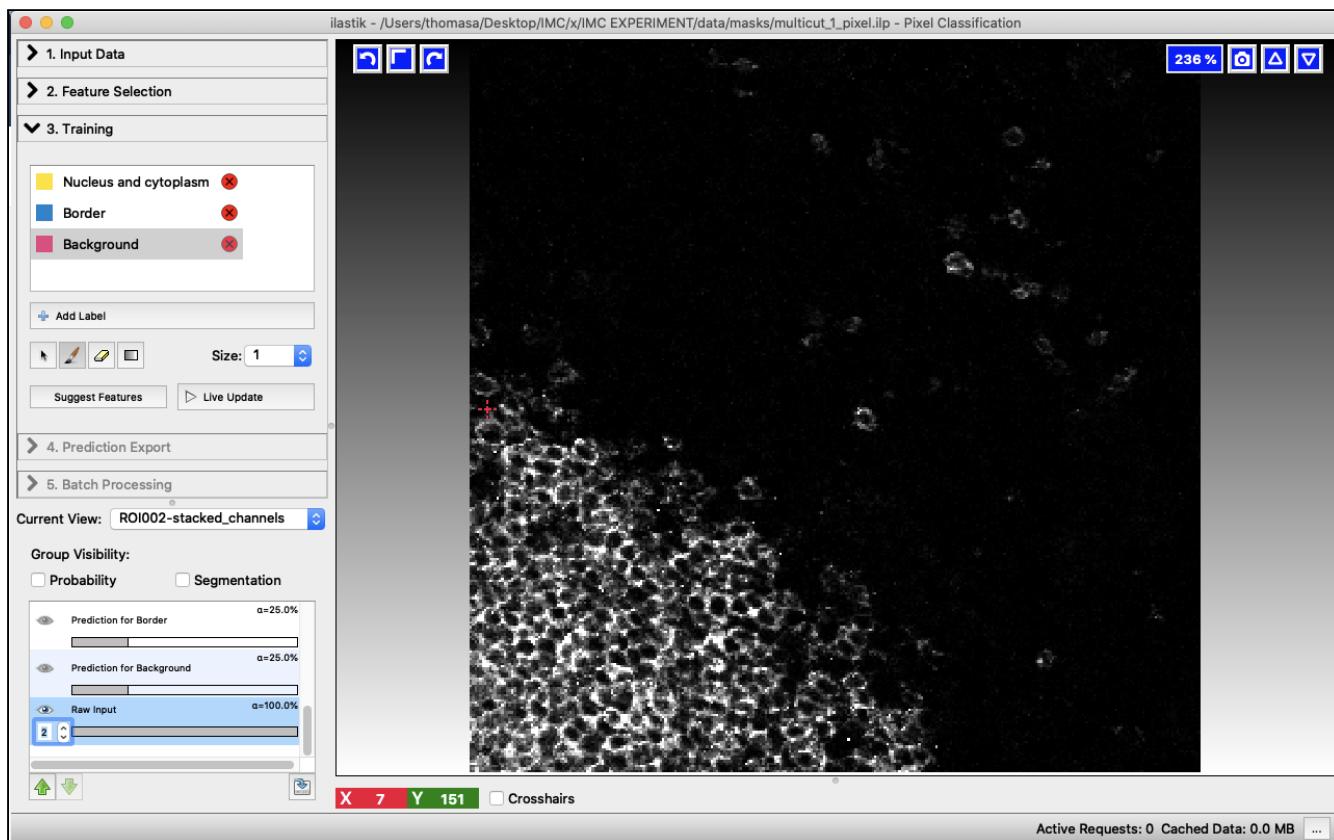


Under training, create three label types



### Trace cell boundaries

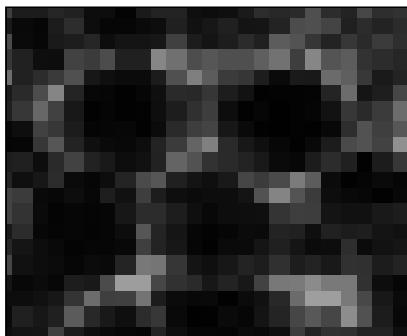
Select the 'cell boundaries' label, and trace the cellular boundaries. Then use the 'nuclear/cytoplasm' label and trace the cell centre, as well as 'background' for the background areas.



### Trace cell boundaries

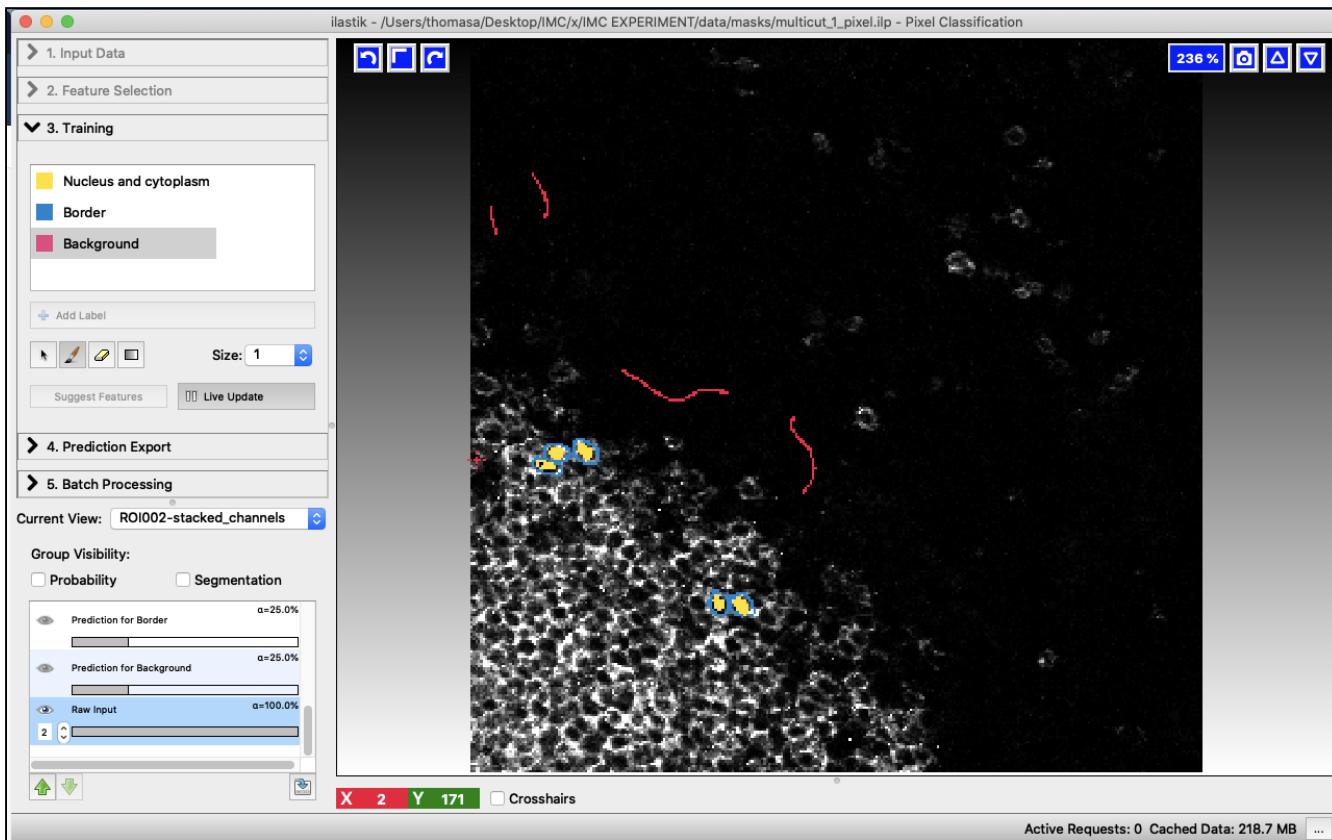
Select the 'cell boundaries' label, and trace the cellular boundaries. Then use the 'other' label and trace the cell centre, as well as background areas.

*Before:*

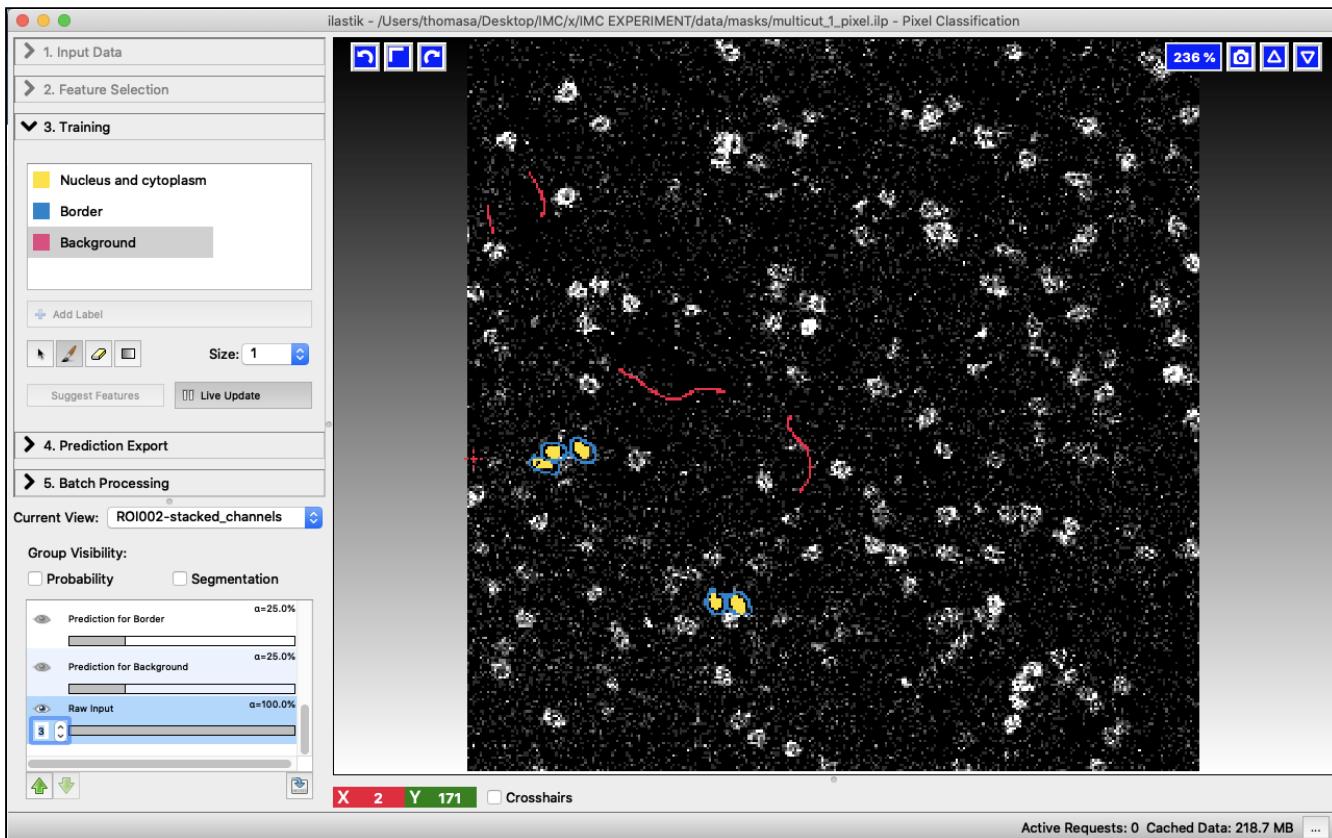


*After:*

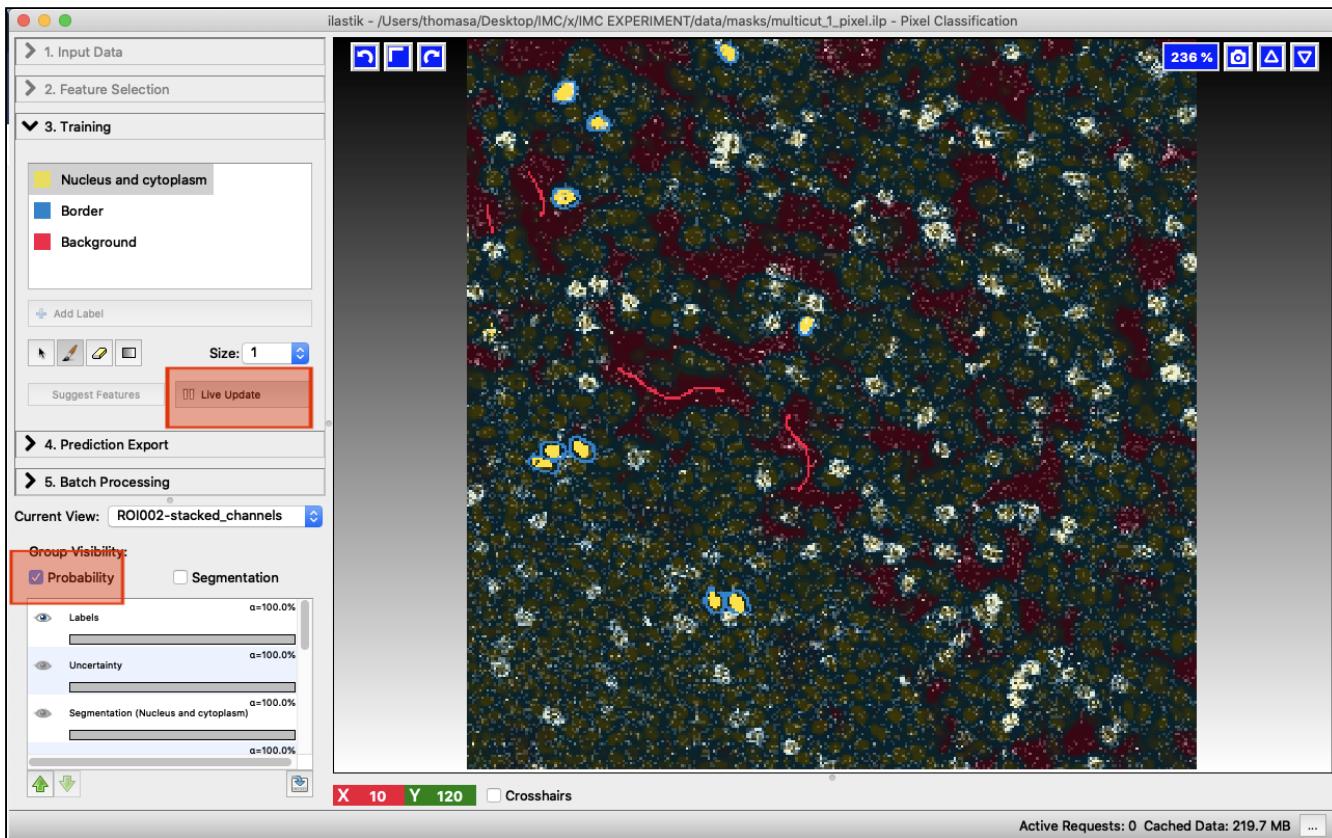




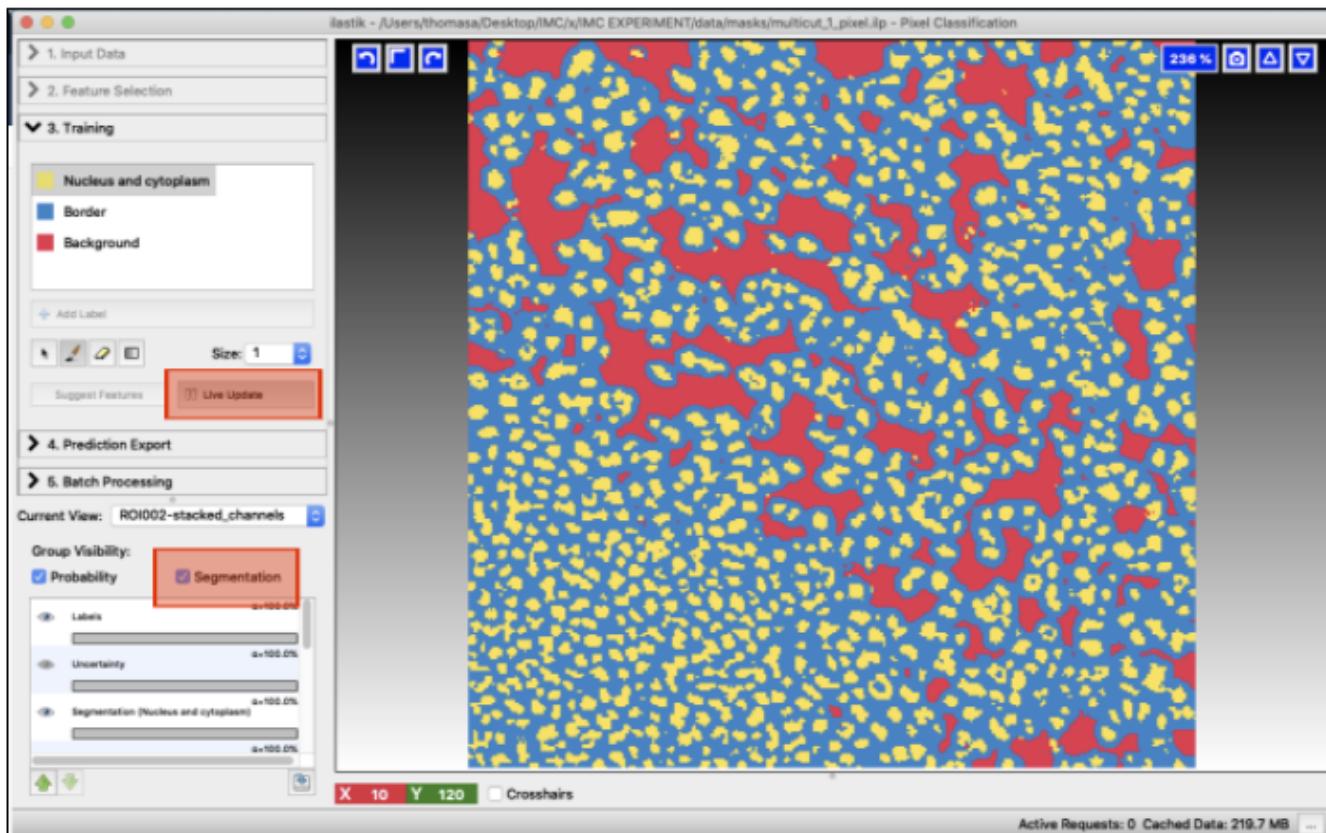
Cycle through different channels to ensure that the painted pixels are appropriate for all layers, as Ilastik will use information from all layers for pixel prediction.



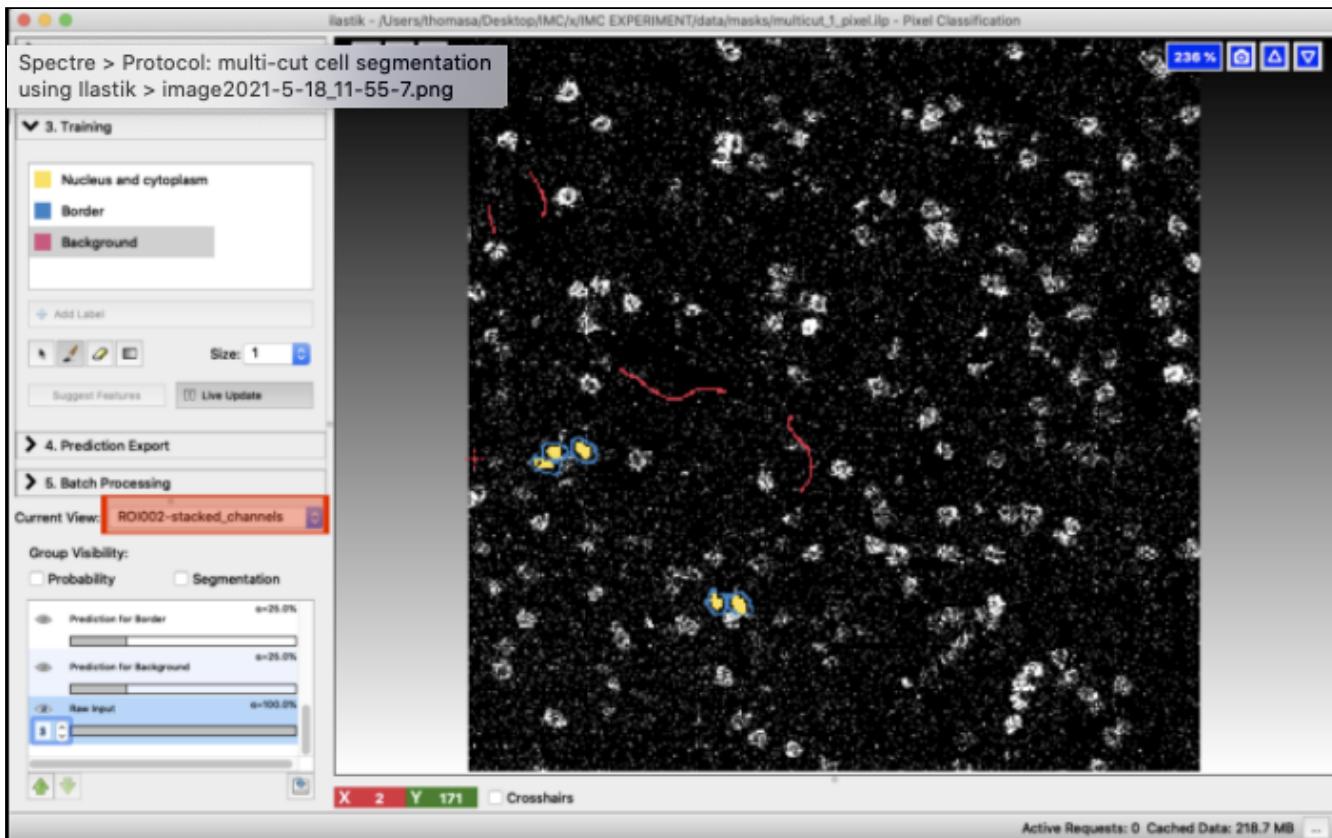
You can switch on 'Live Update' to visualise and overlay of the 'probability' for each pixel according to Ilastik.



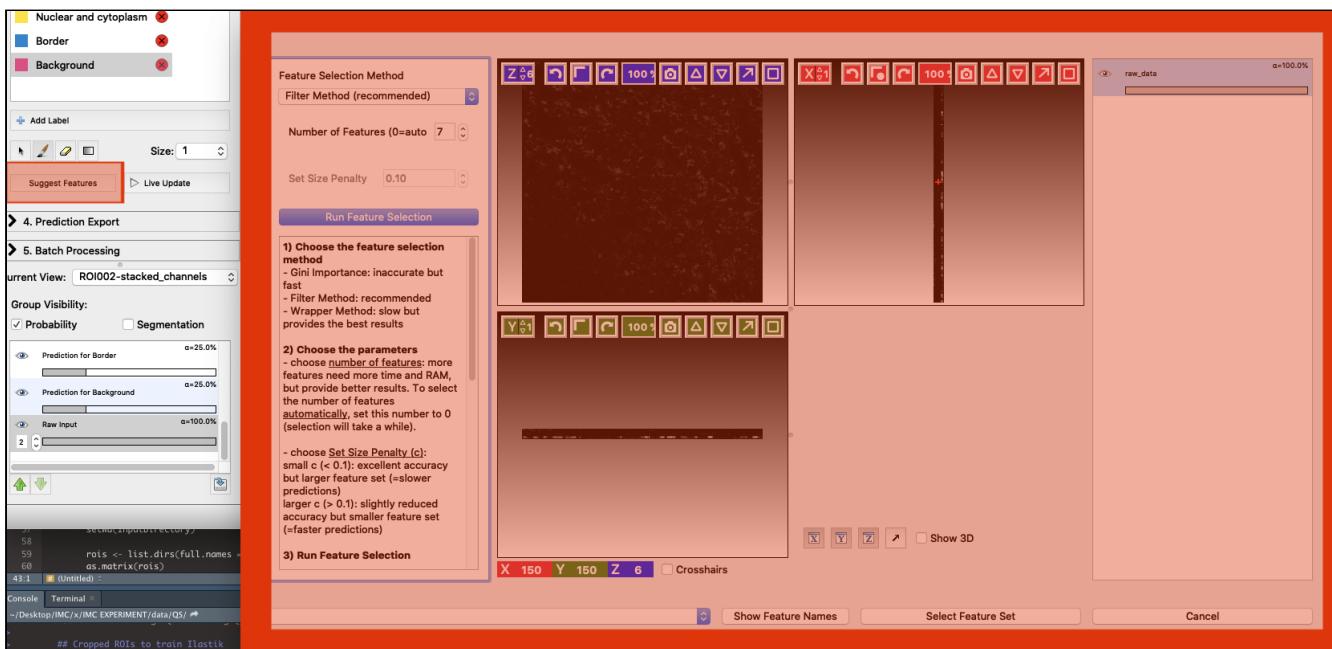
You can also see the segmentation decision.



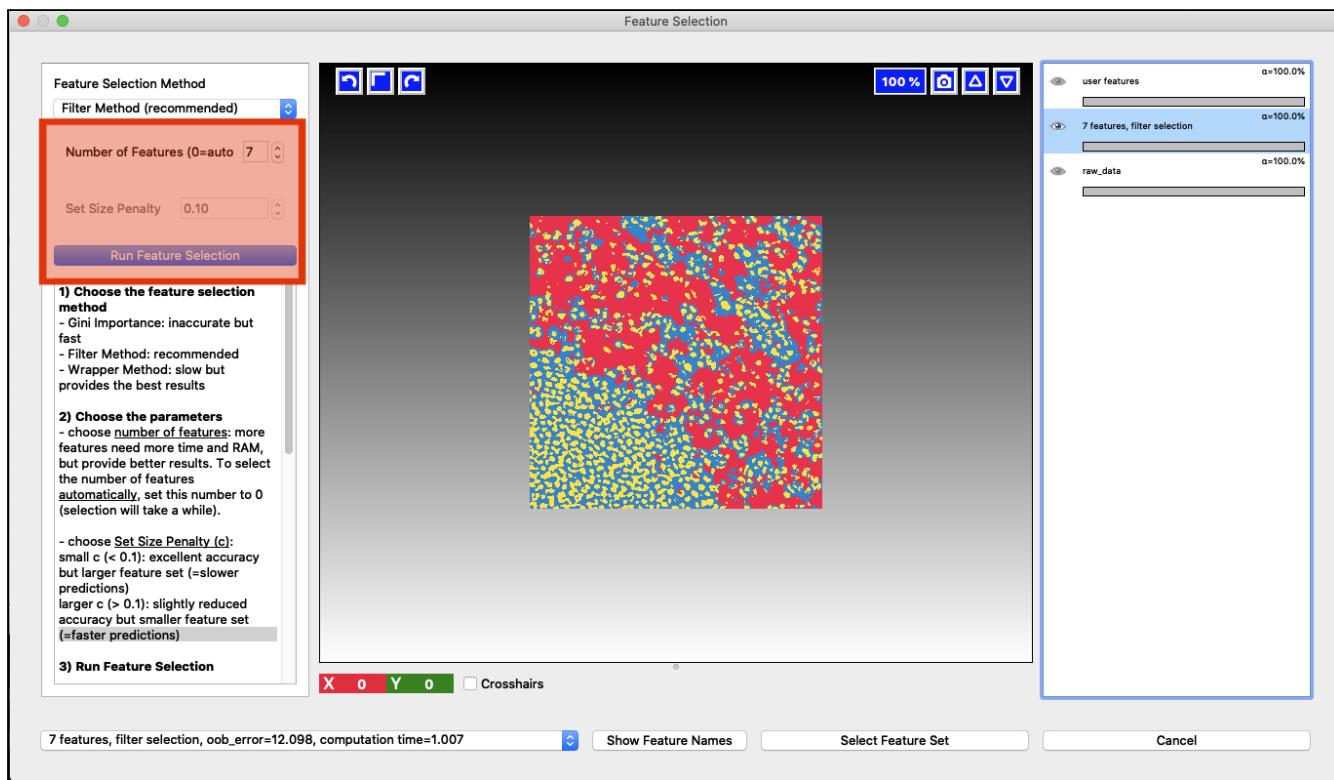
You should also cycle through each ROI and provide training for all three layers on each ROI.



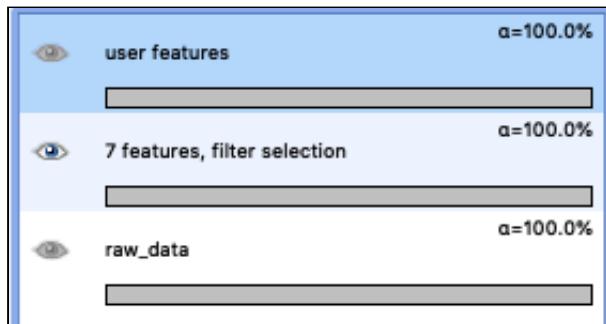
Once you have provided a decent level of pixel training, you can select 'suggest features'.

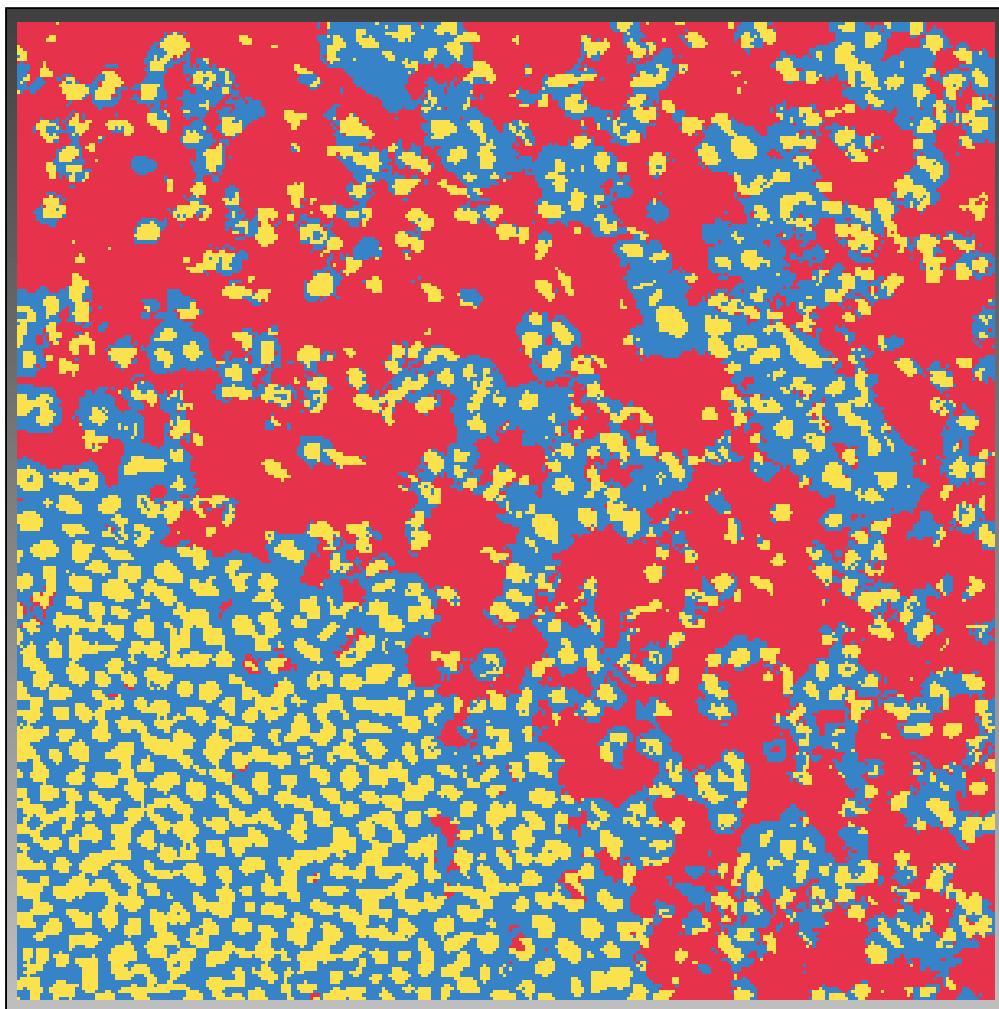


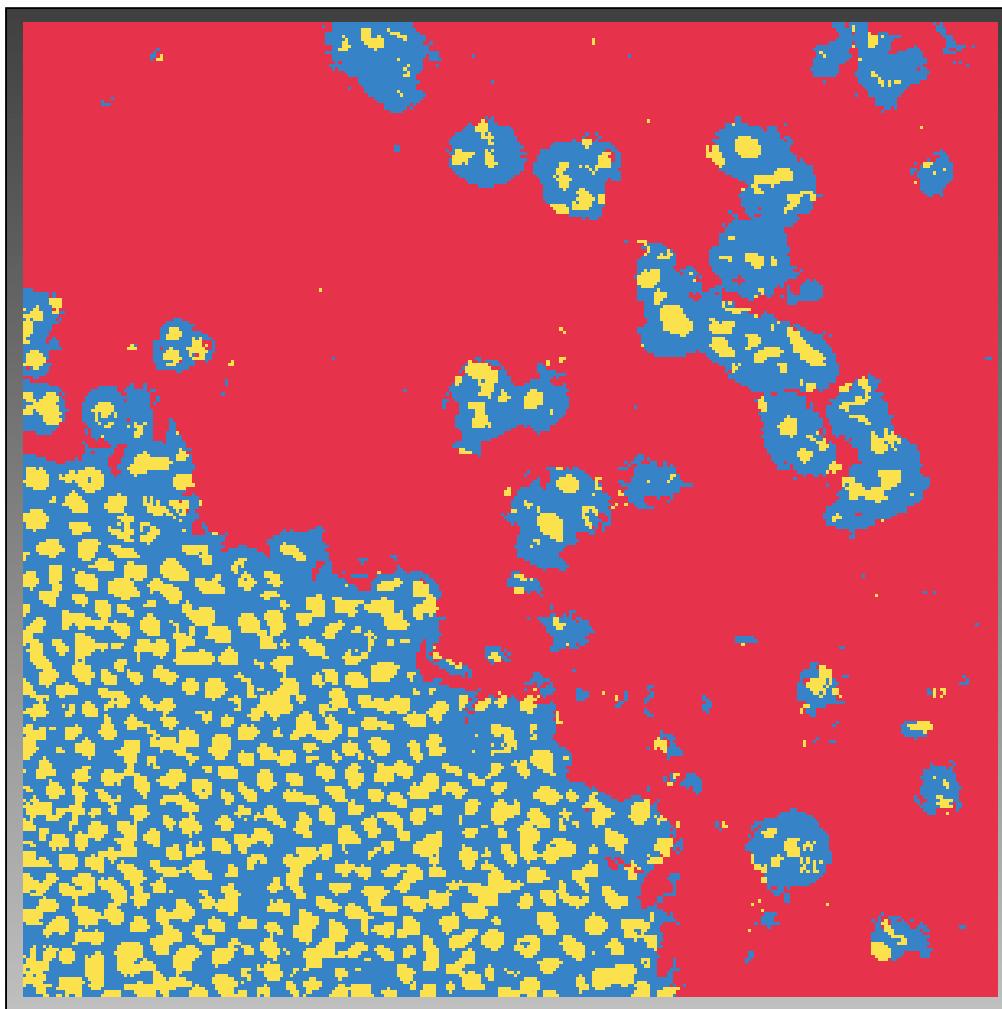
Select 'run feature selection' with a target of 7.



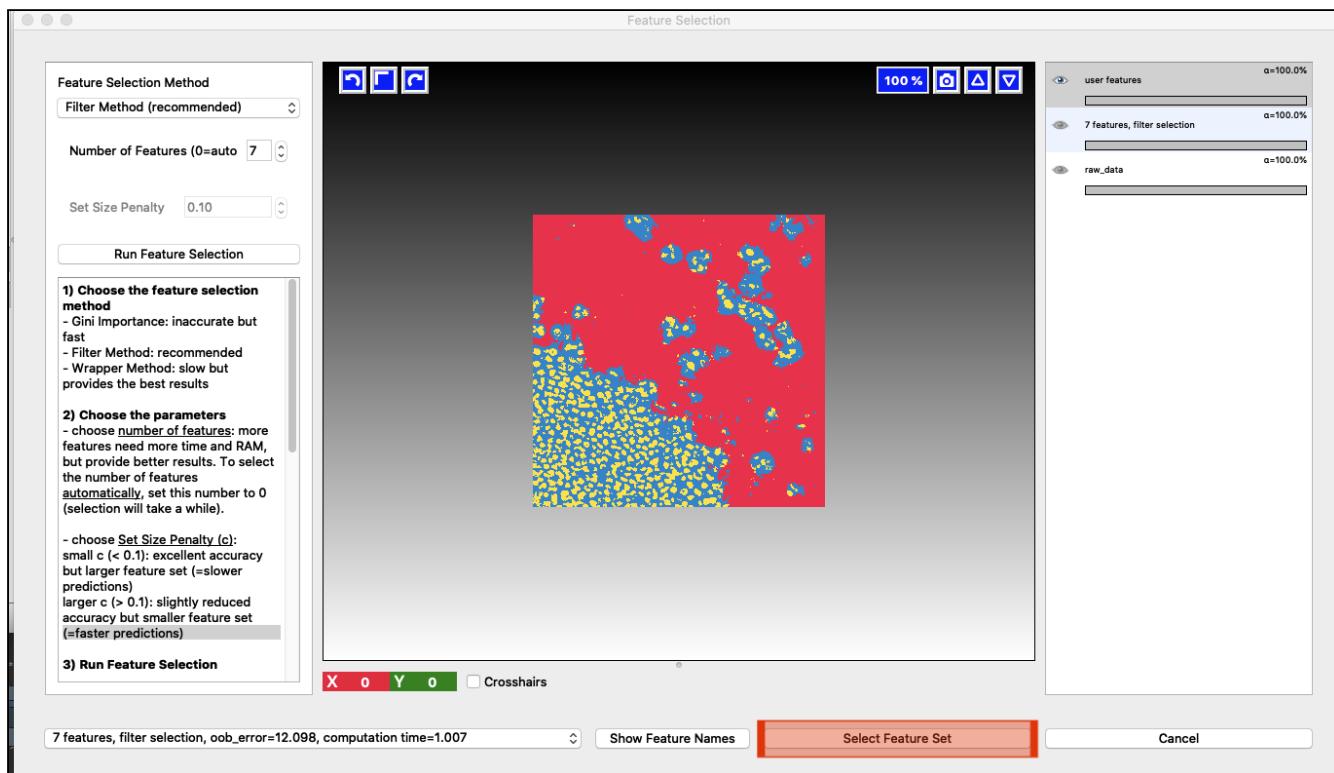
Flick between the 'user features' (i.e. prediction with all features) and the '7 features, filter selection' by clicking on the eye, and see the difference.



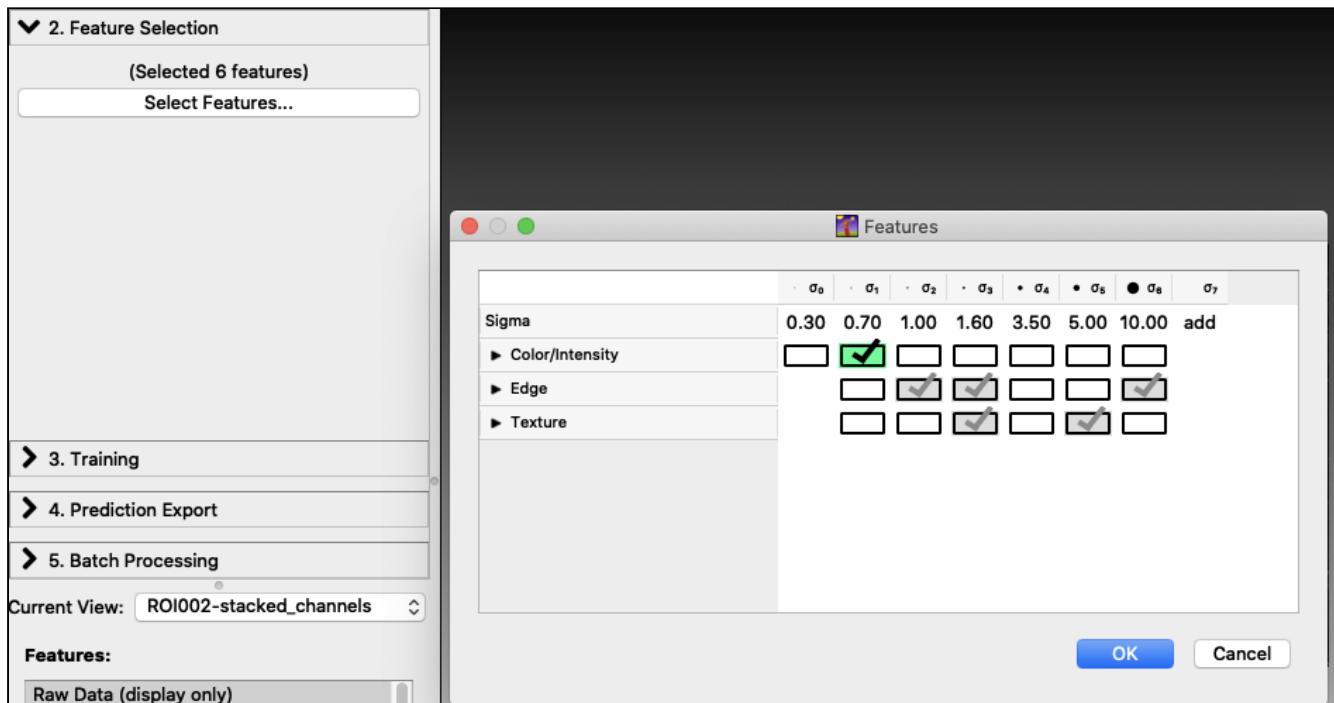




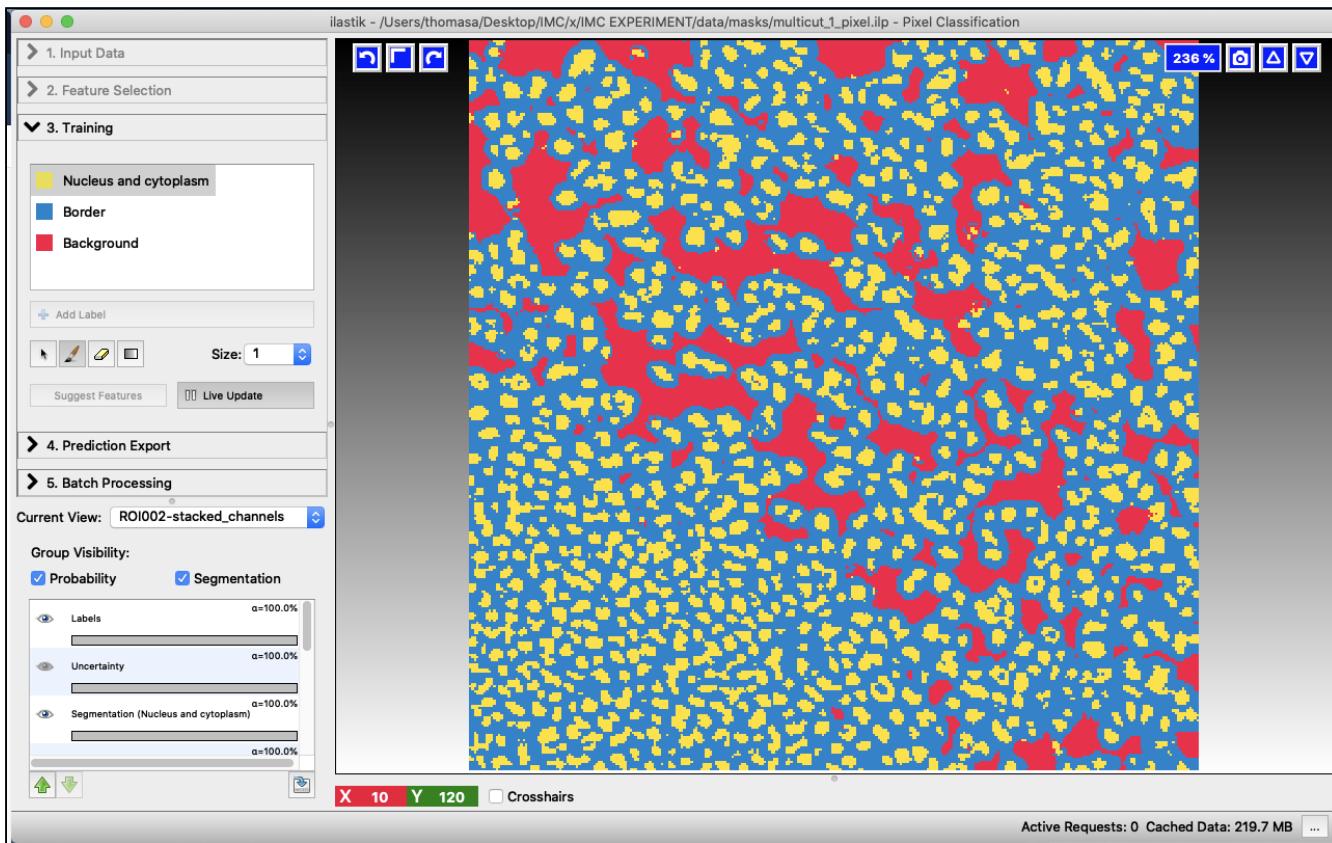
Decide on which one produces a better result, and select 'Select Feature Set'.



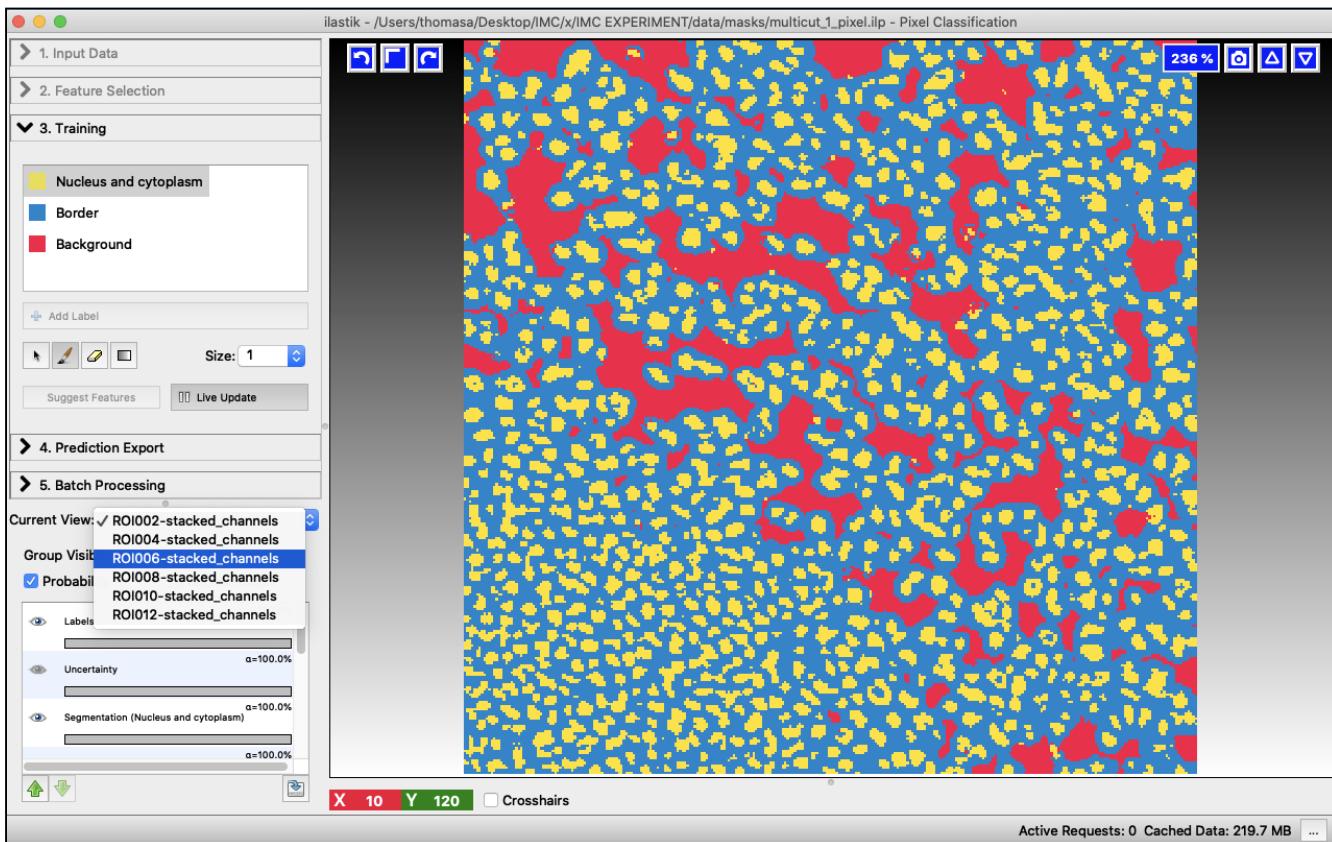
You can check which features are in use by flicking back to the 'feature selection' section.



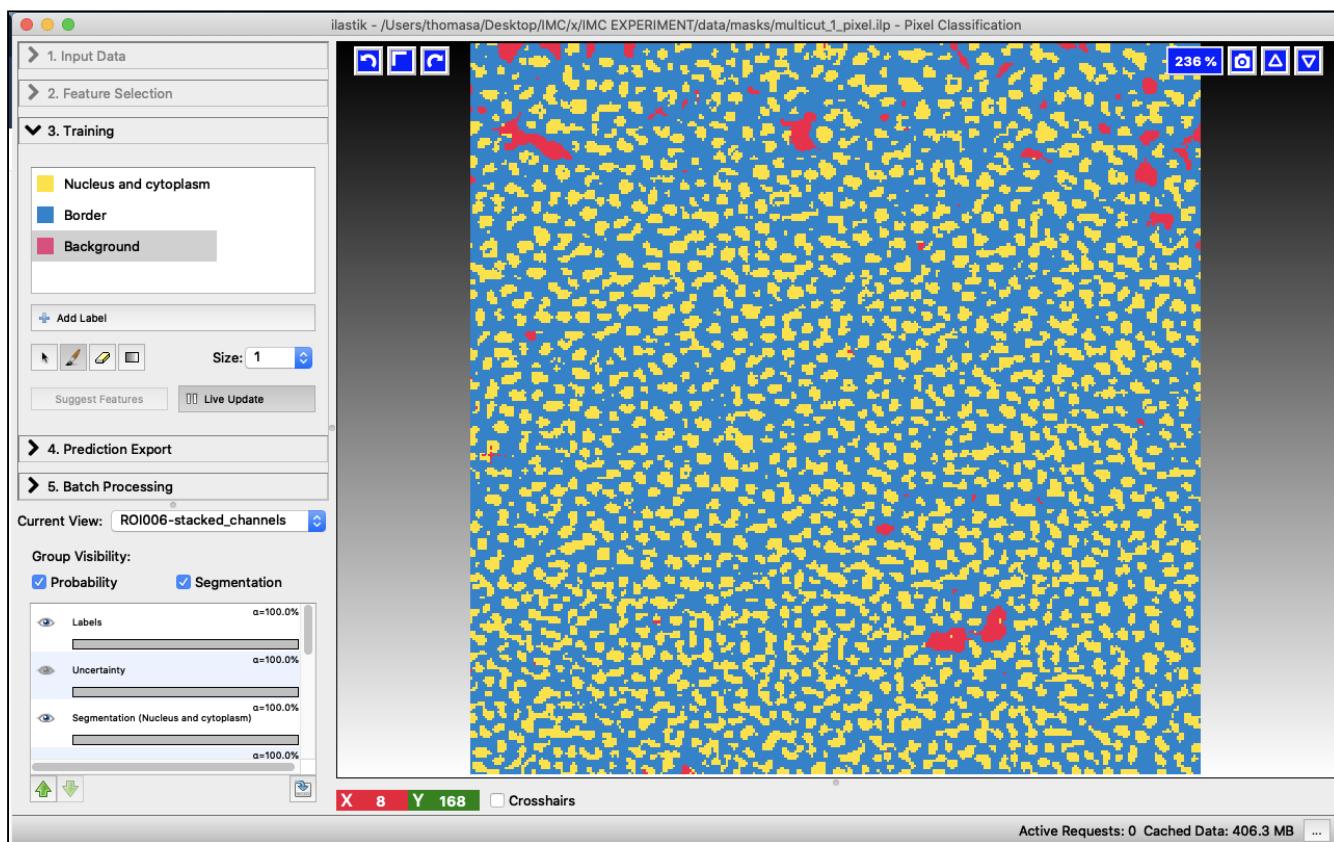
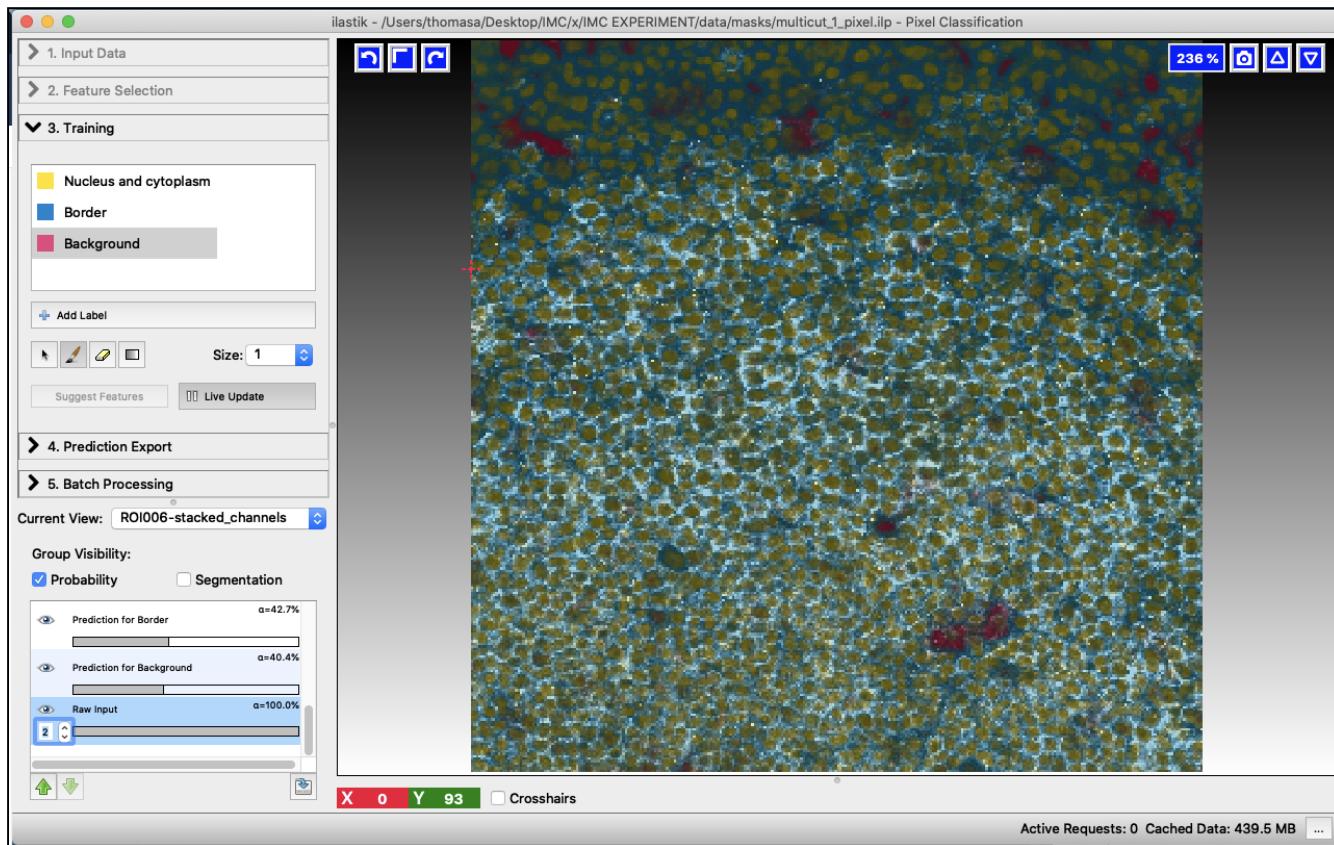
Check segmentation results.



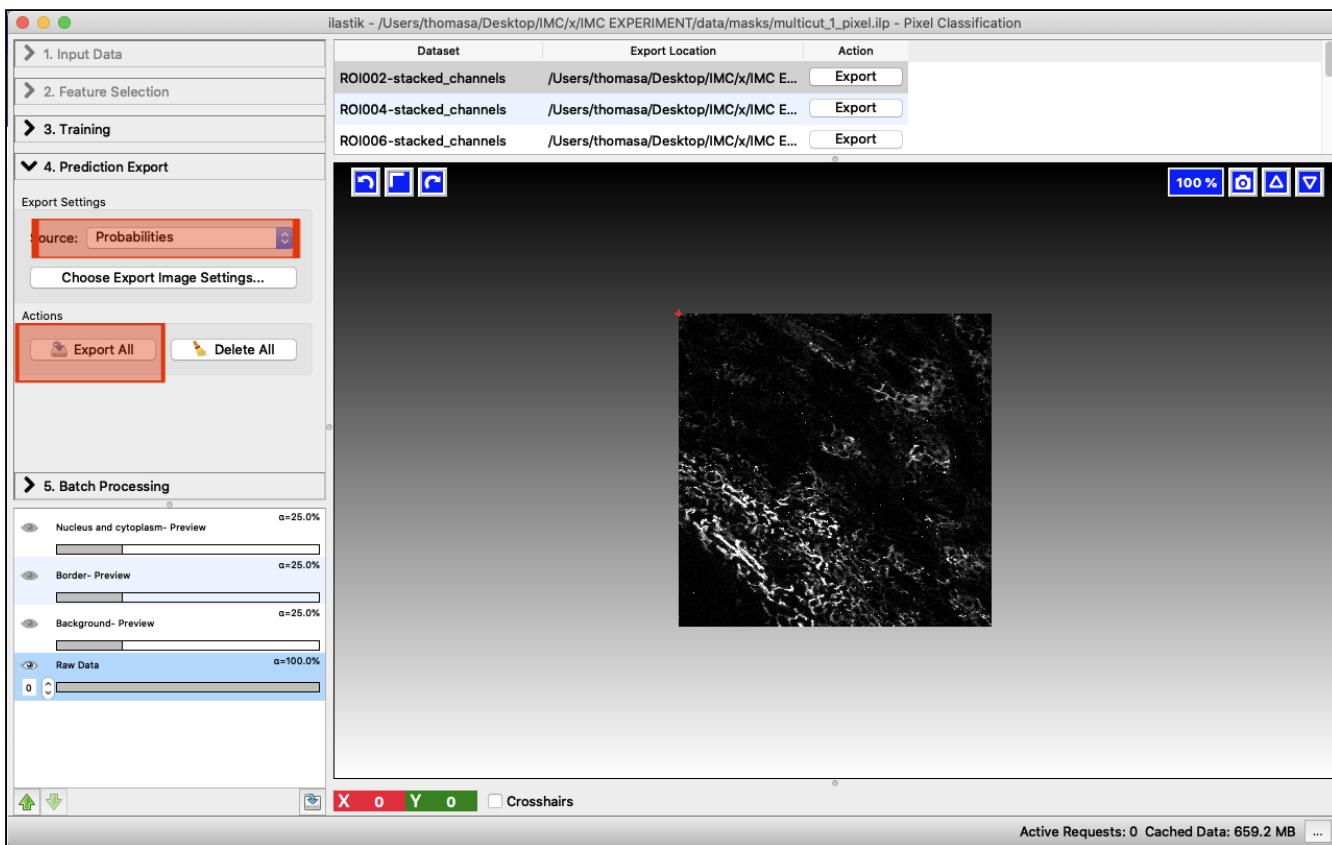
Check each ROI.



Also check the probability overlay.



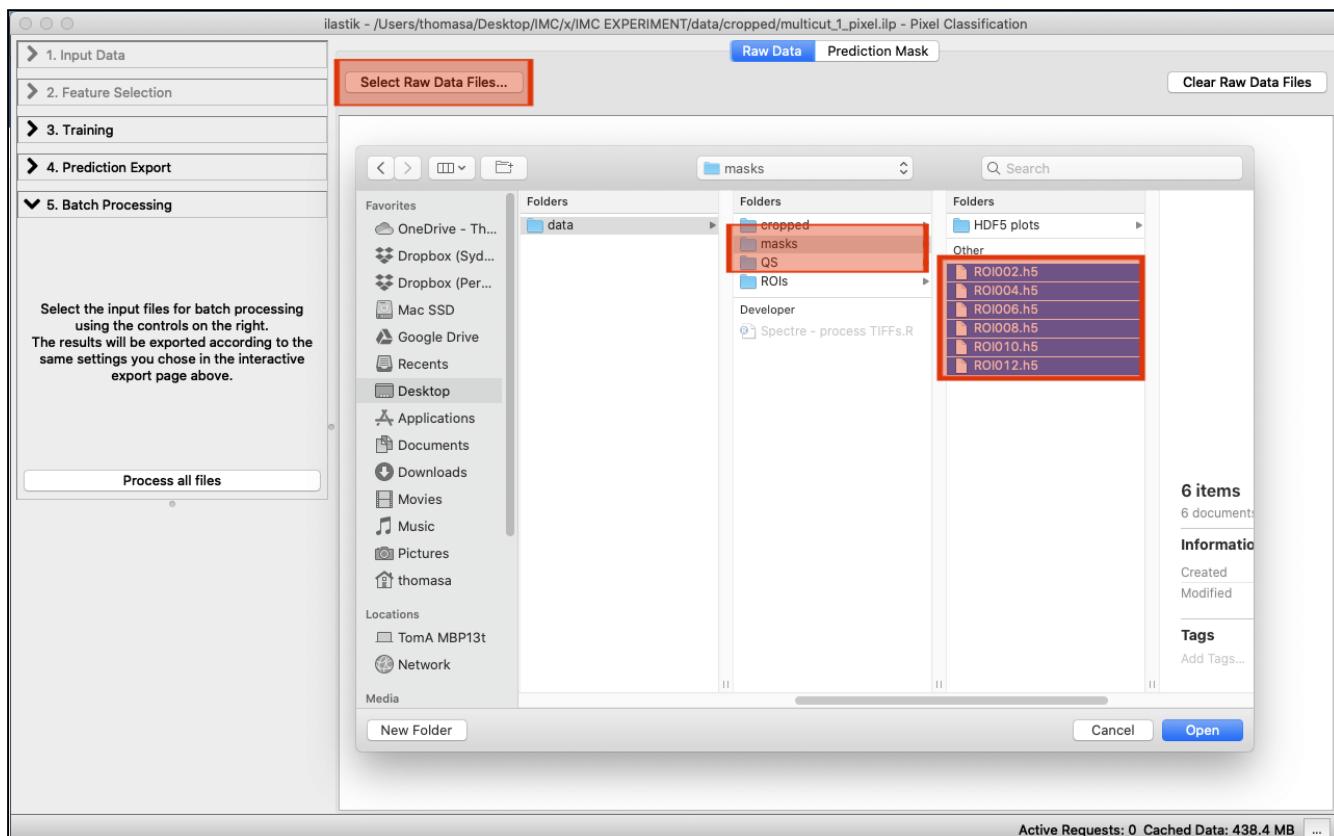
When you are happy that the pixel prediction is working well, go the '4. Prediction Export' section and hit 'Export All' with source 'Probabilities'.



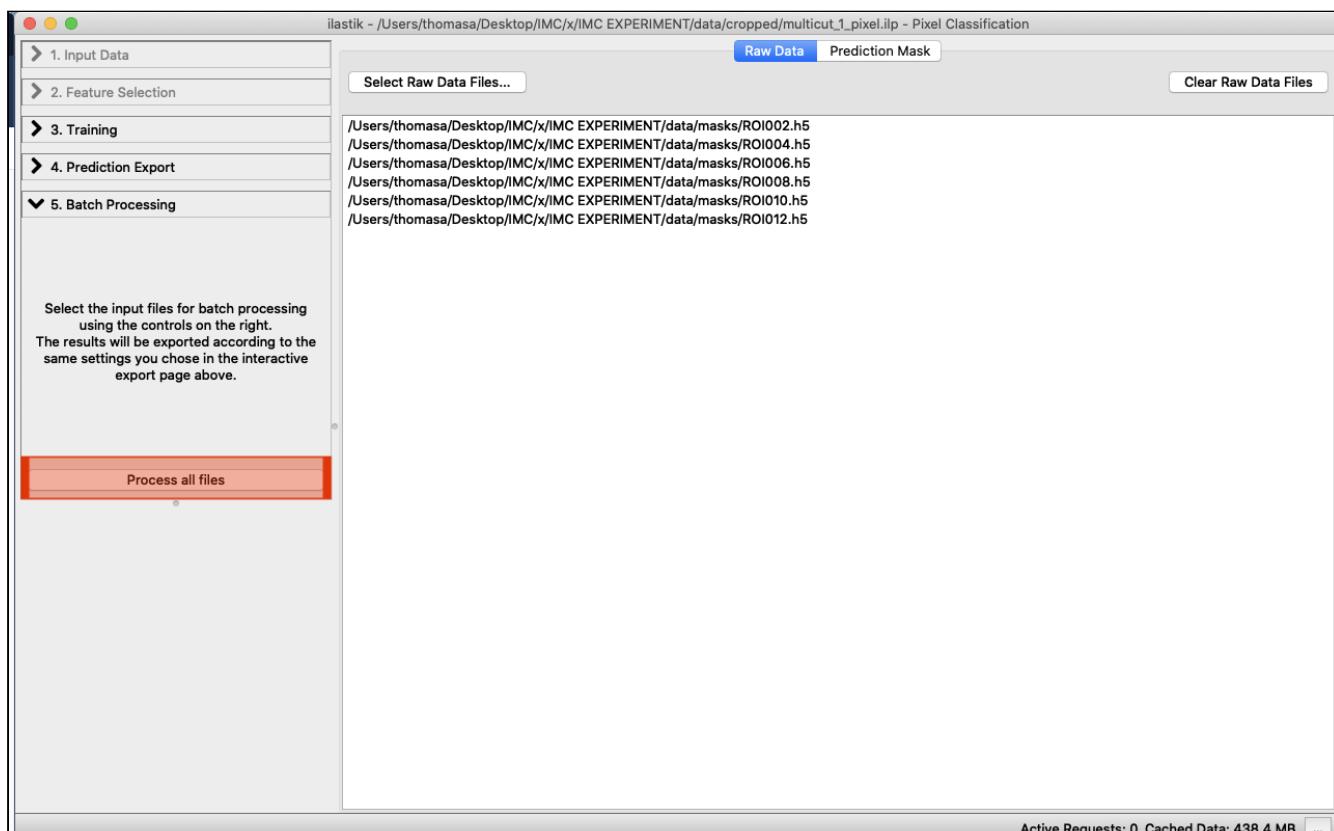
This will create 'probabilities' .h5 files in the **cropped** directory.

File
multicut_1_pixel.h5
multicut_2_segment.h5
multicut_3_segmentation.h5
multicut_4_segmentation.h5
ROI002-stacked_probabilities.h5
ROI002.h5
ROI004-stacked_probabilities.h5
ROI004.h5
ROI006-stacked_probabilities.h5
ROI006.h5
ROI008-stacked_probabilities.h5
ROI008.h5
ROI10-stacked_probabilities.h5
ROI10.h5
ROI12-stacked_probabilities.h5
ROI12.h5

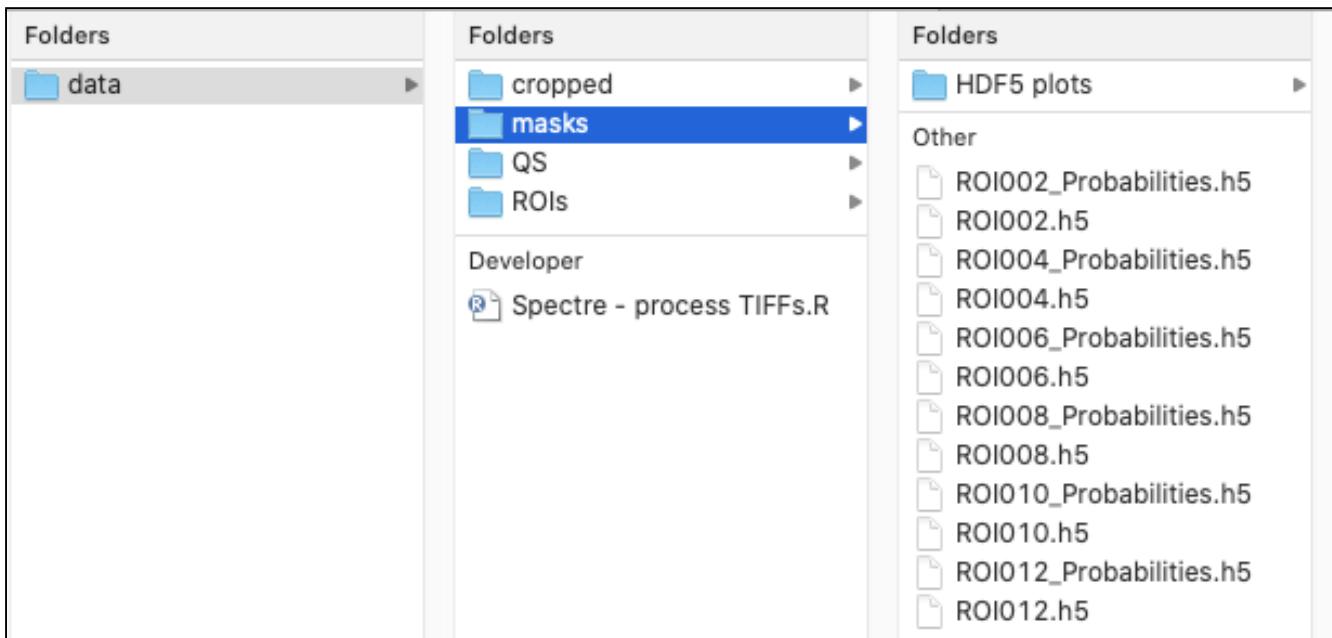
Go to the '5. Batch Processing' and click 'select raw data files'. Select the .h5 files in the '**masks**' directory.



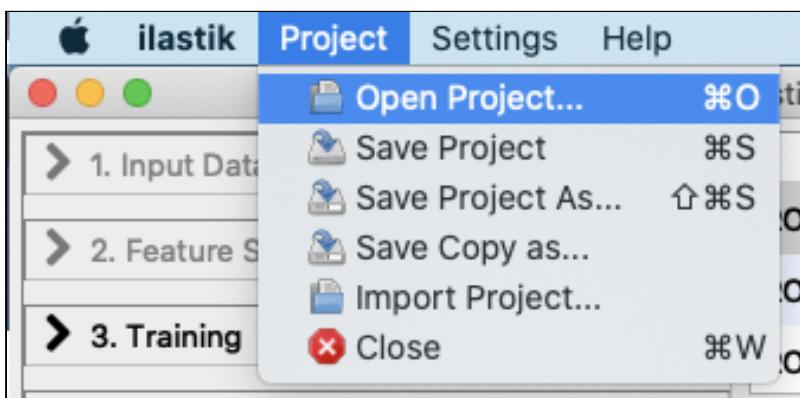
Click 'Process all files'.



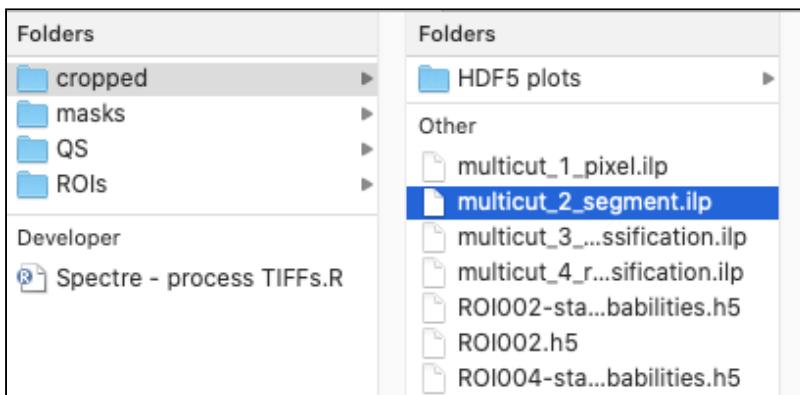
This will create 'probabilities' .h5 files in the '**masks**' directory.



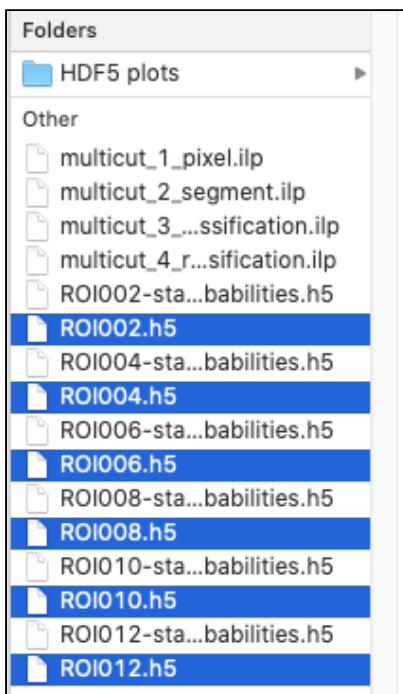
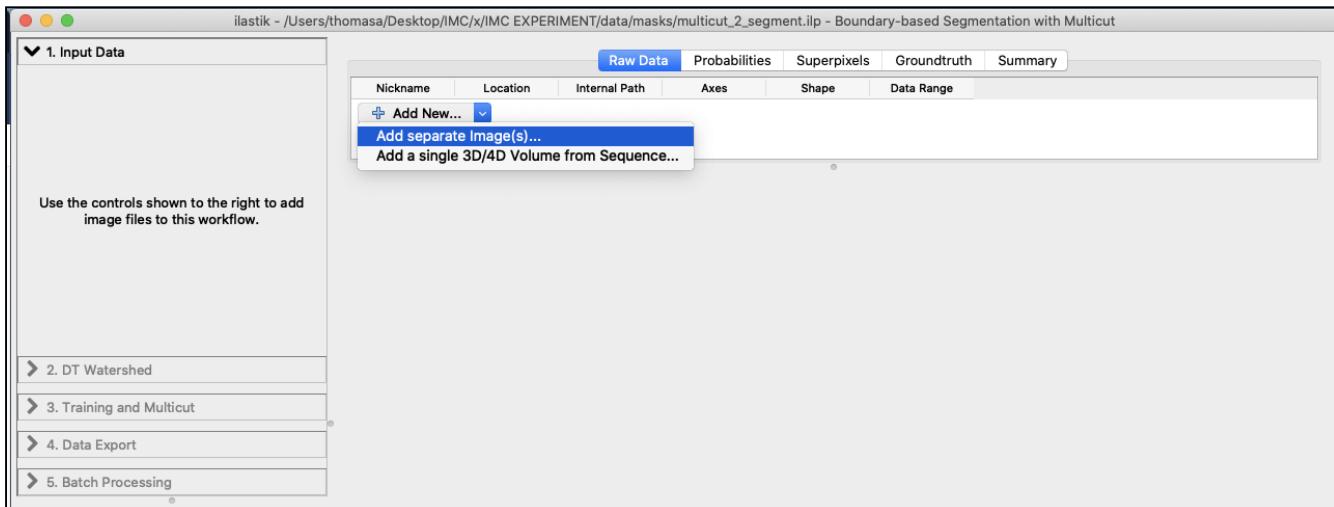
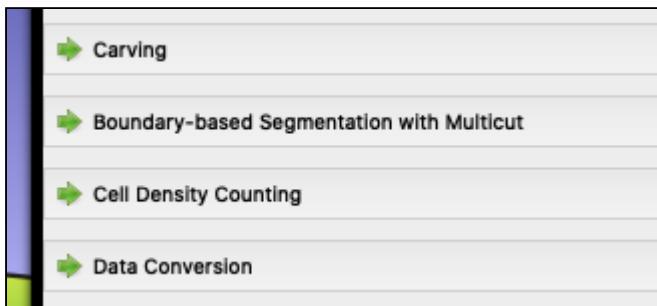
### 3. Ilastik - multicut processing

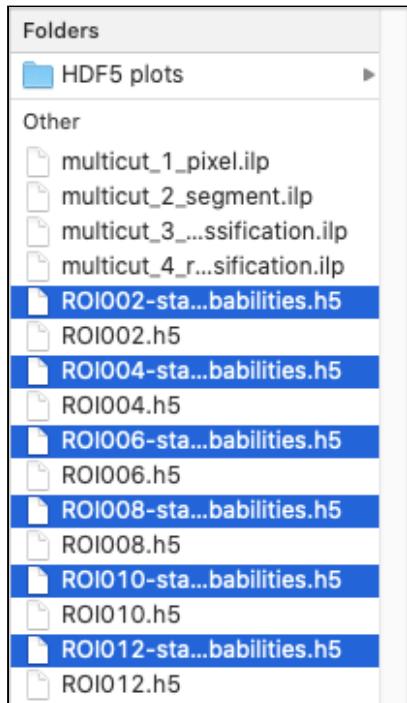
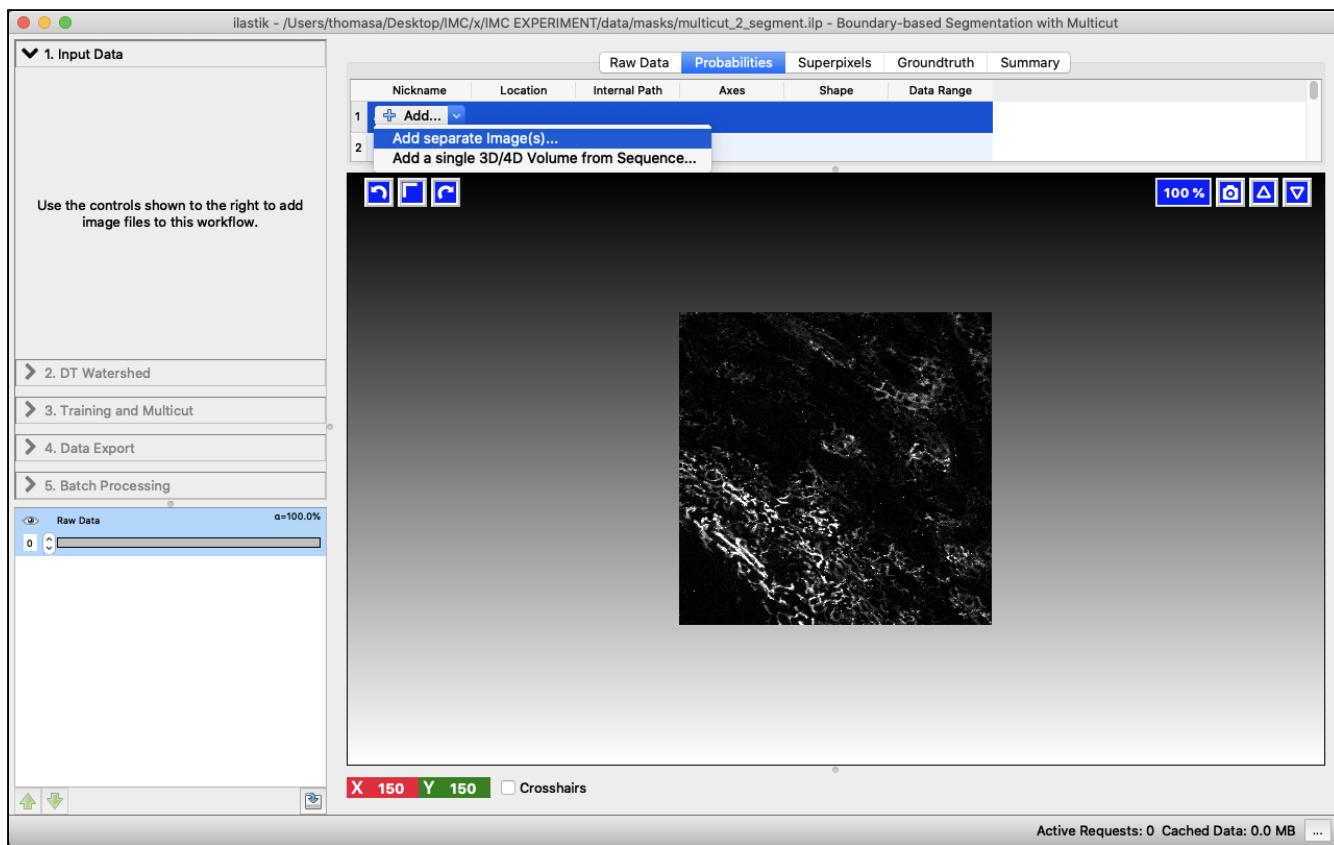


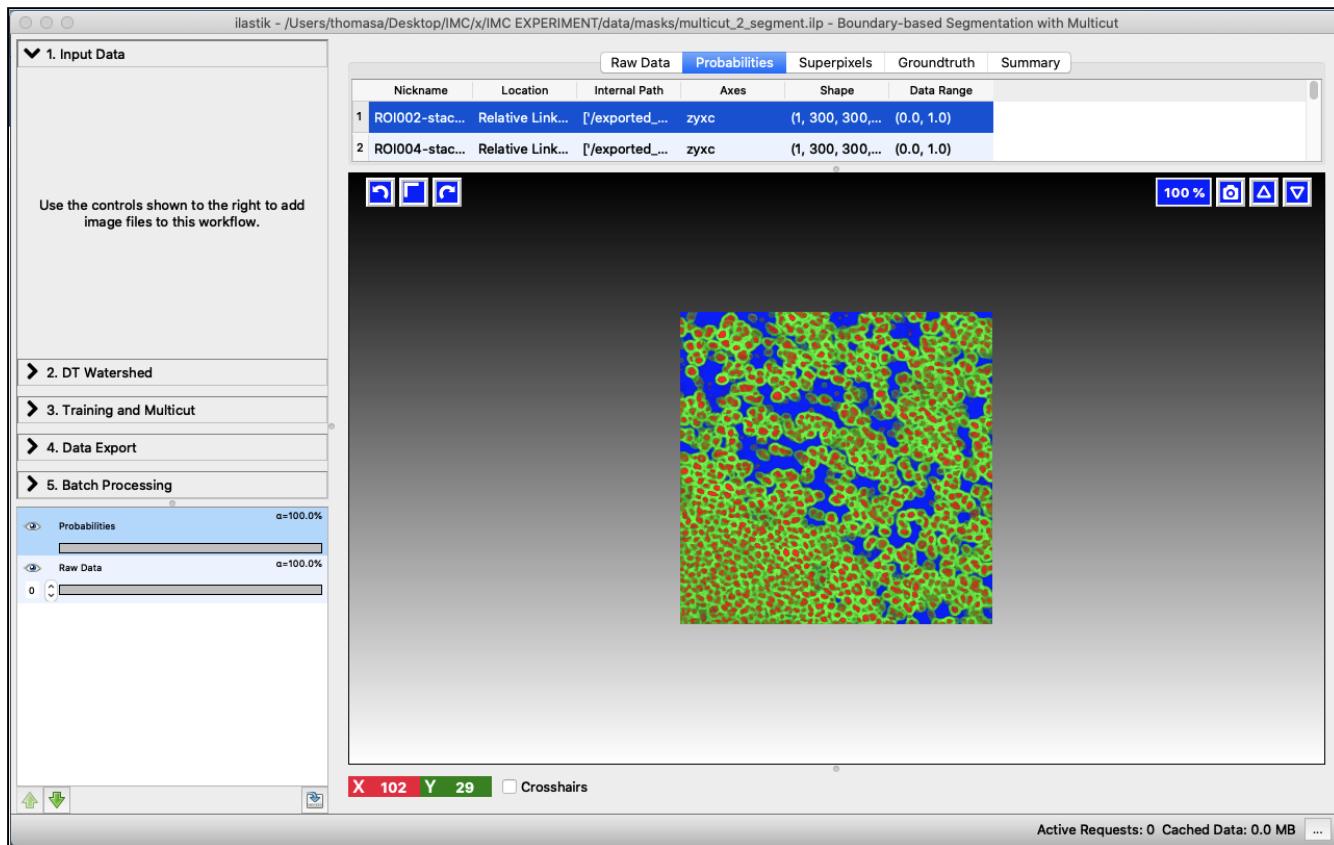
And select the 'multicut\_2\_segment.ilp' file.



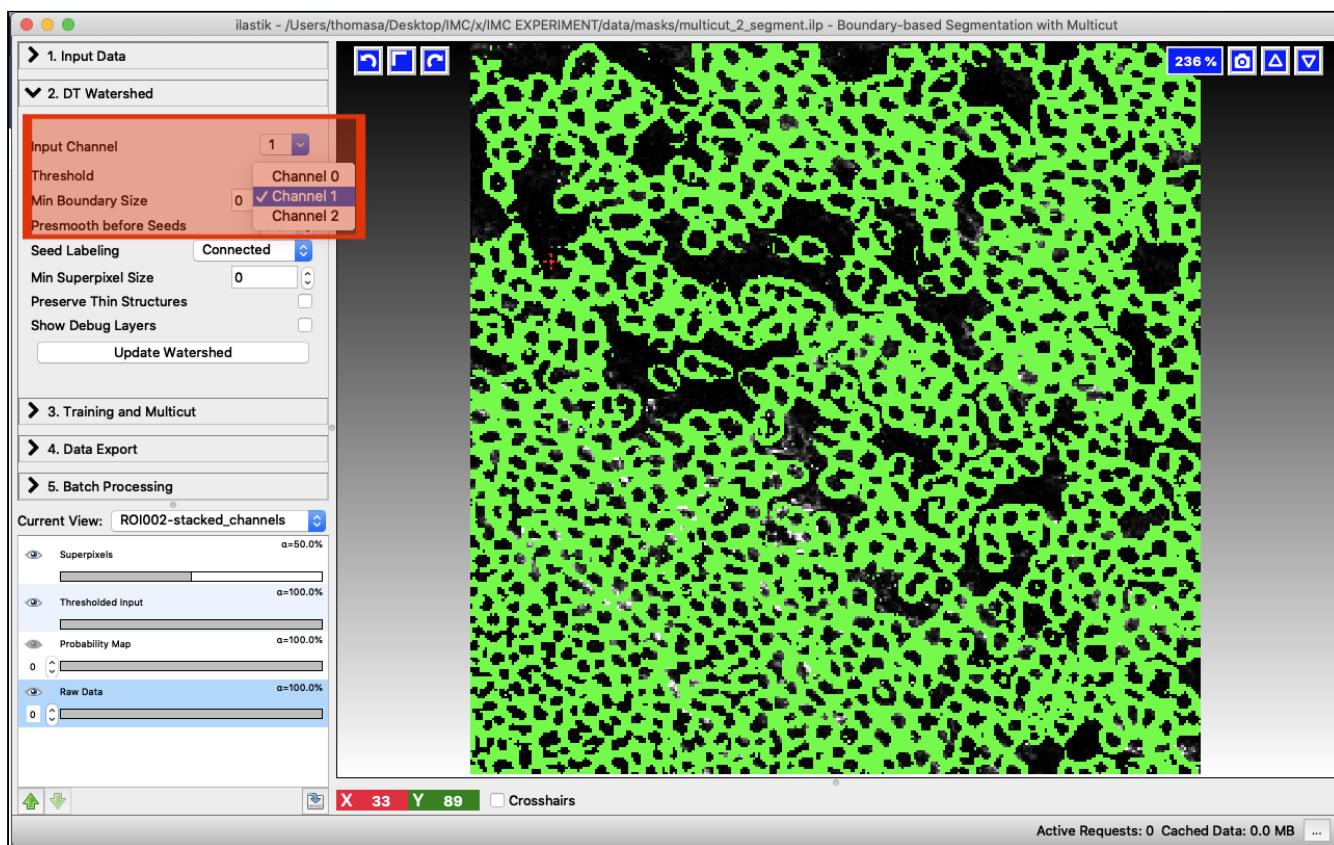
Alternatively, create and save a new Ilastik file of the 'Boundary-based Segmentation with Multicut' type).

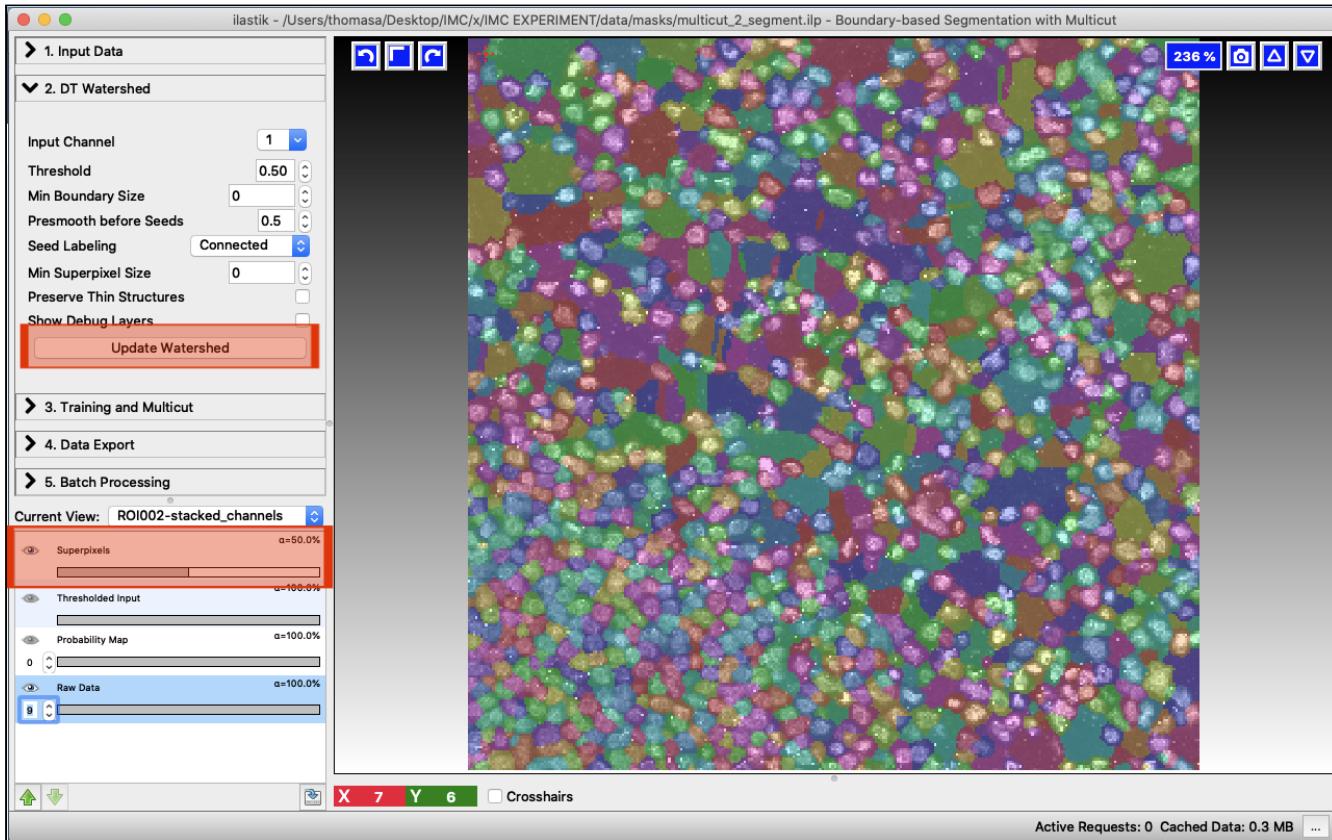
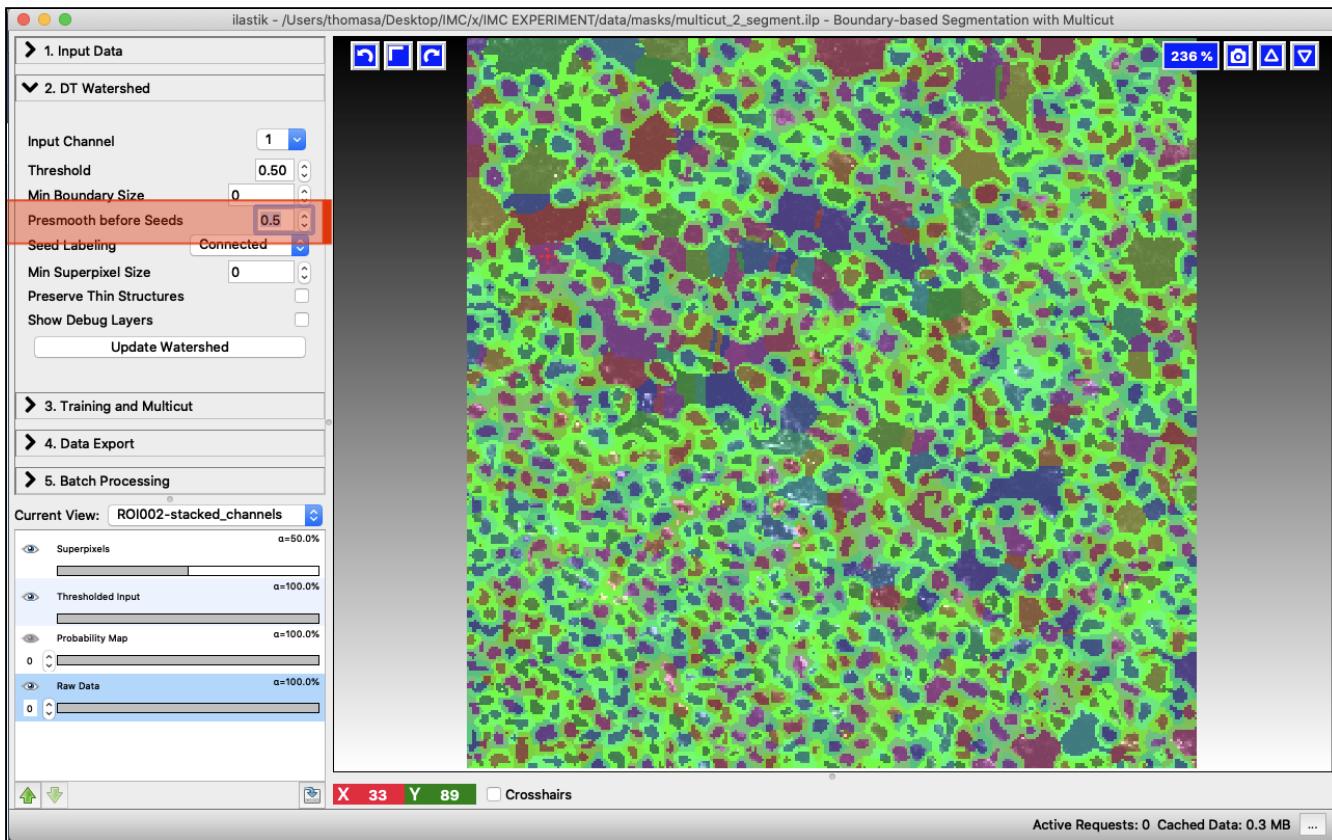


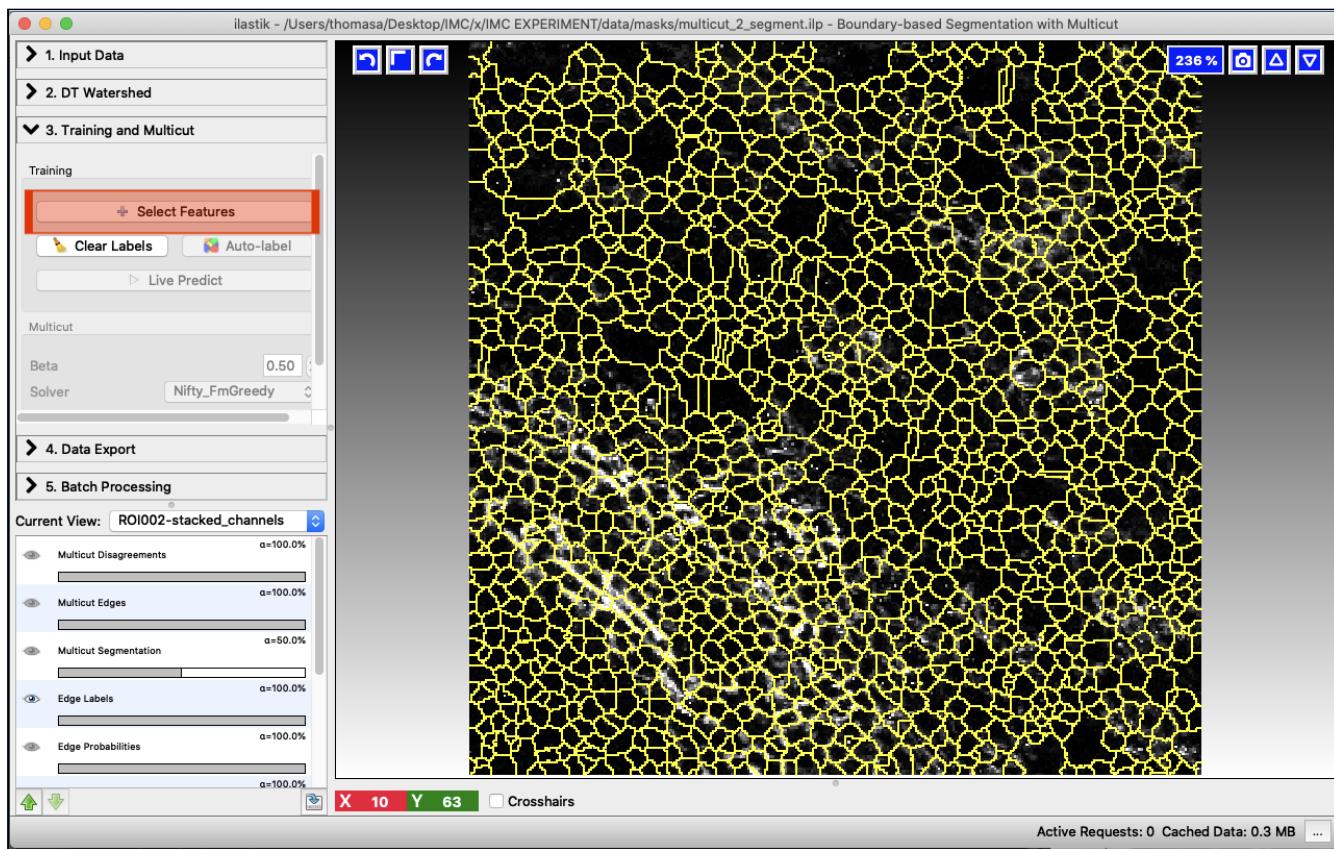
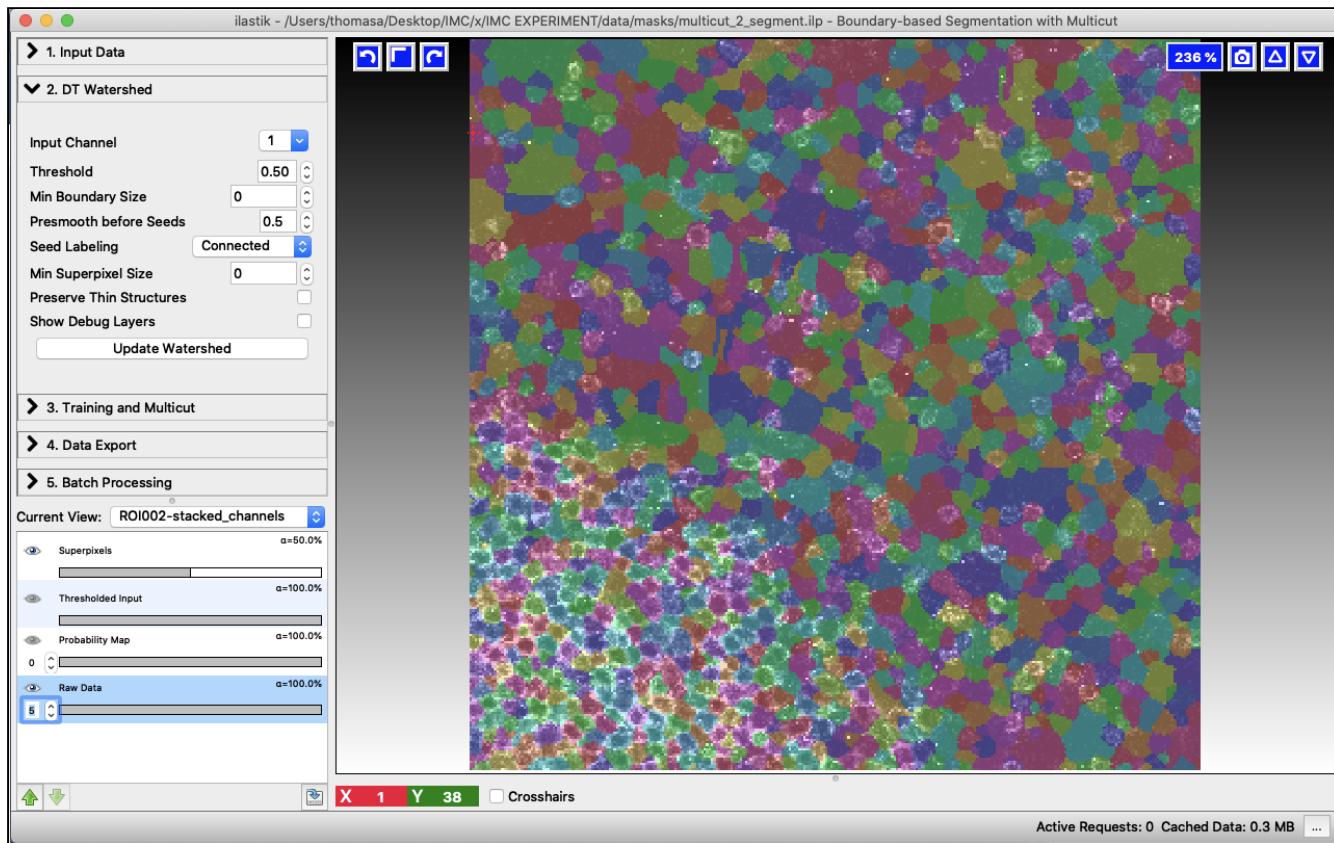


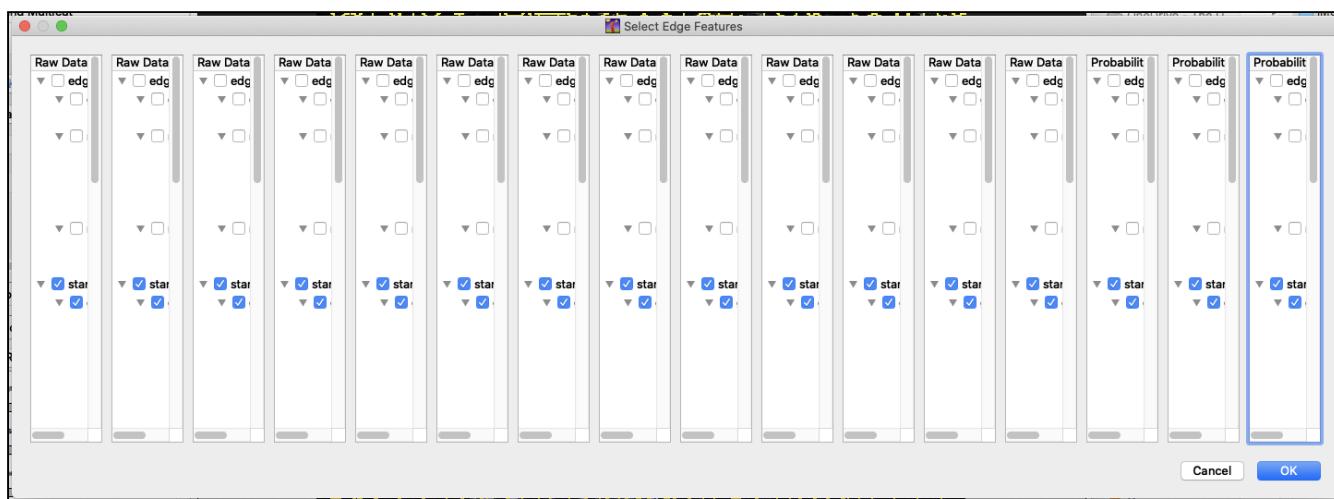


Change the input channel to reflect the 'border' layer that we made in the previous section (in this case, channel 1). If the border was the second layer we made, it should be the second channel (out of channel 0, channel 1, and channel 2).





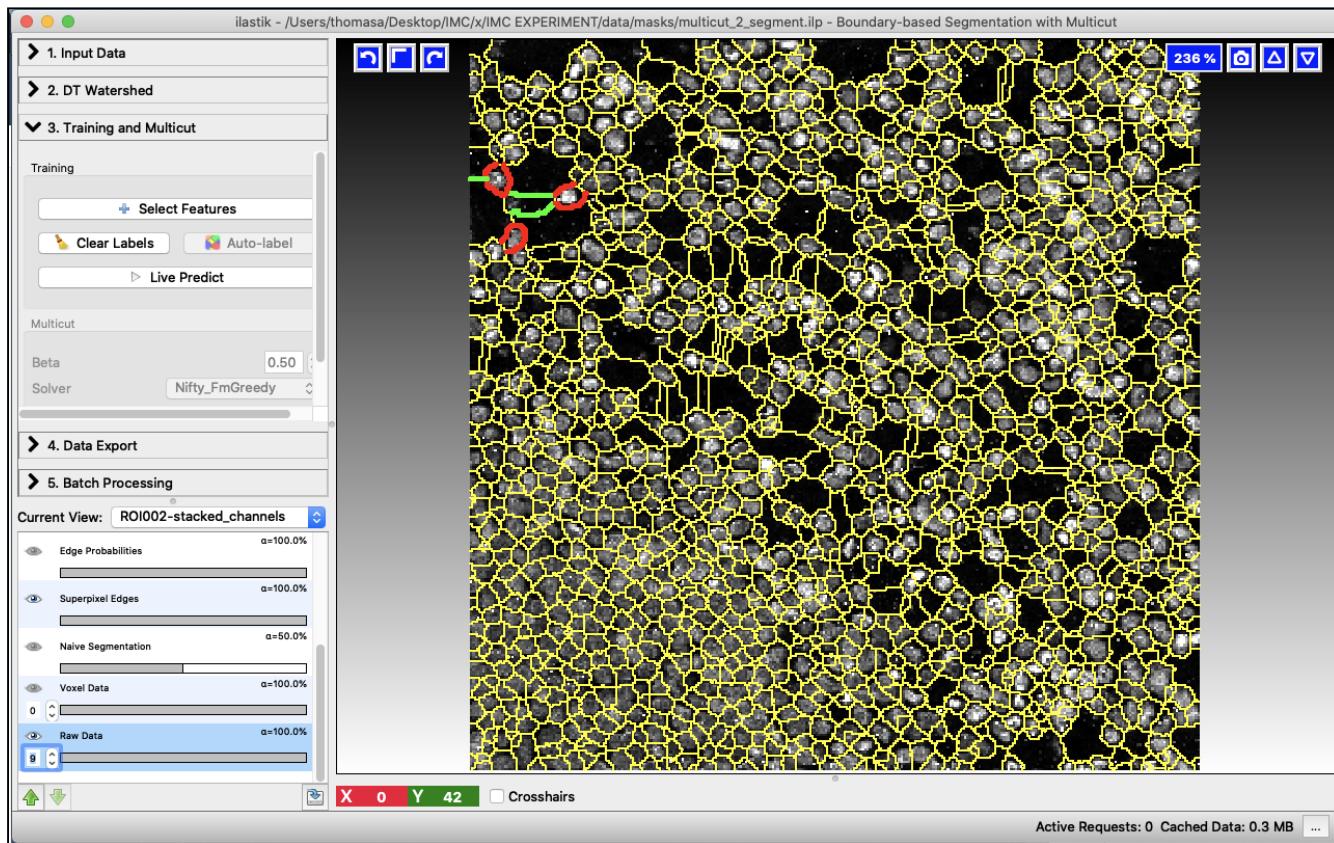


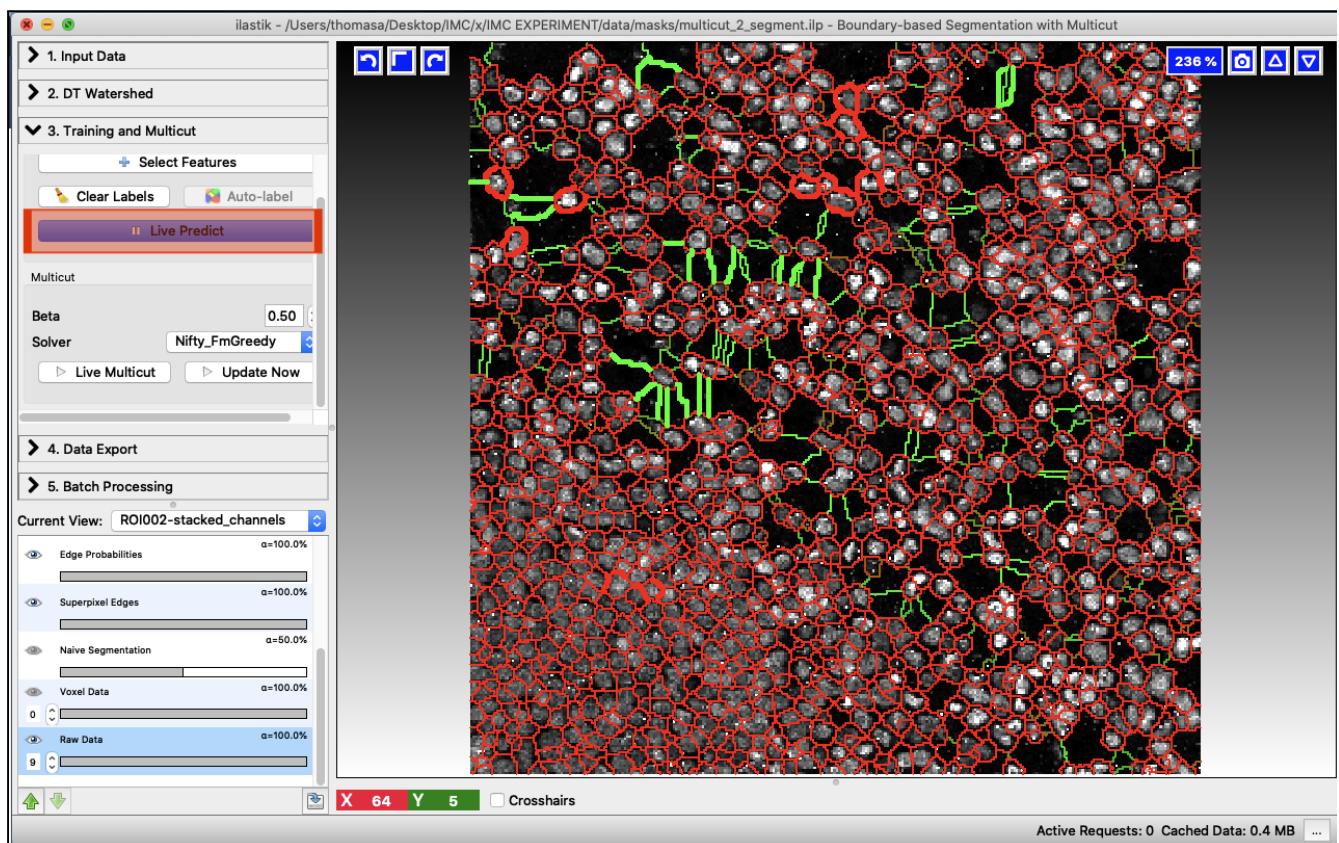
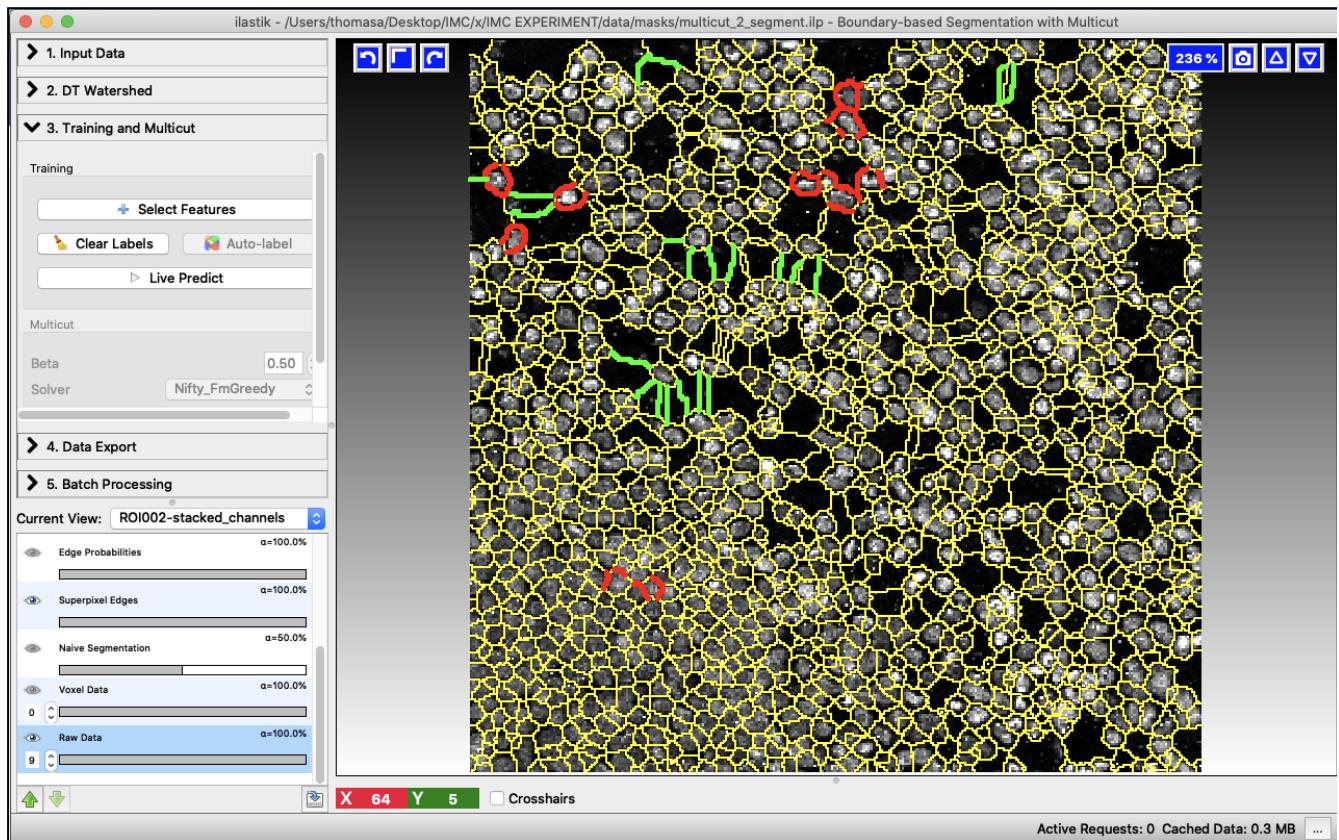


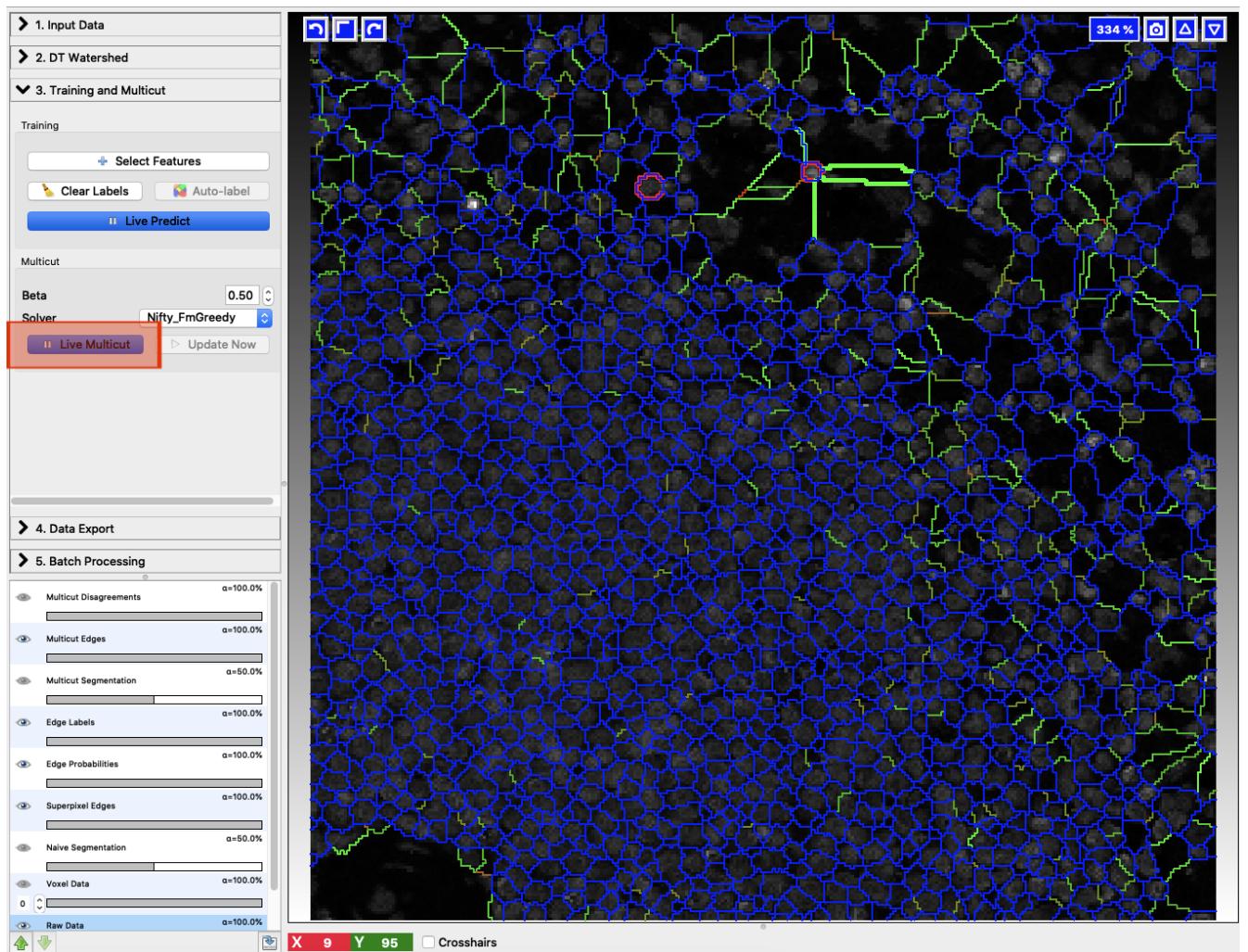
It is more important here to **exclude** lines that are splitting cells. It's also helpful to **exclude** lines that are dividing up empty space – the larger the empty space object is, the easier it is to filter out down the track.

**Right click = keep**

**Left click = exclude**

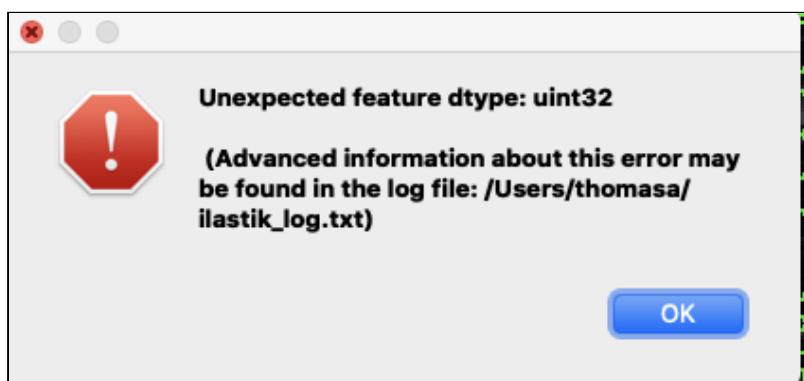


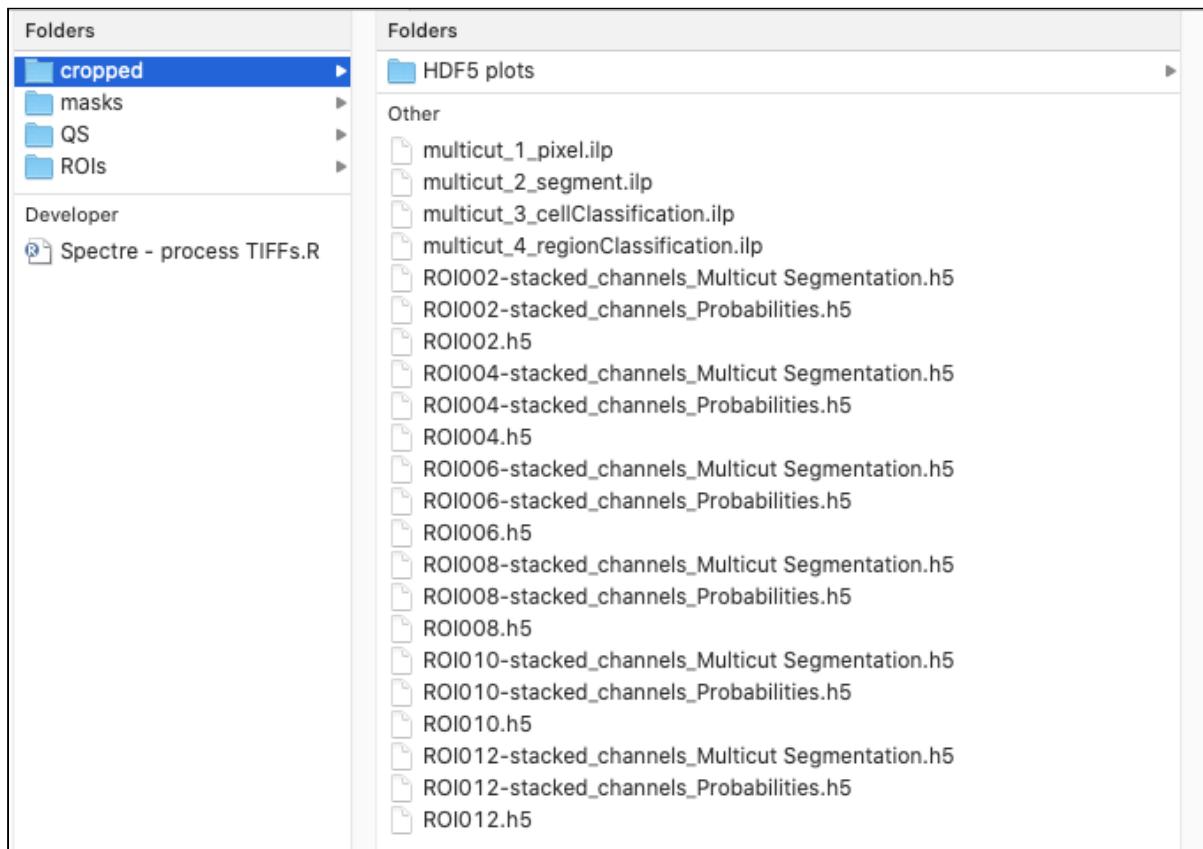
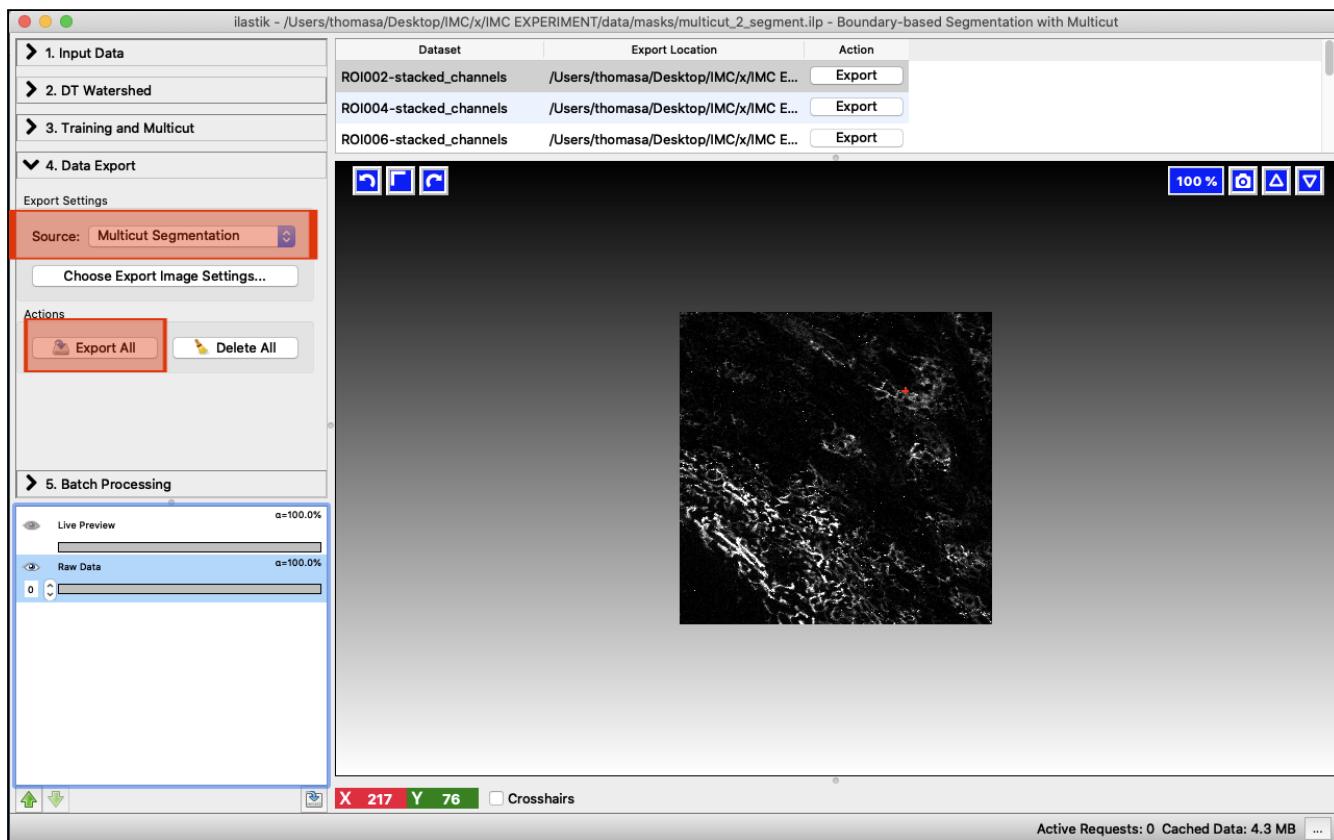


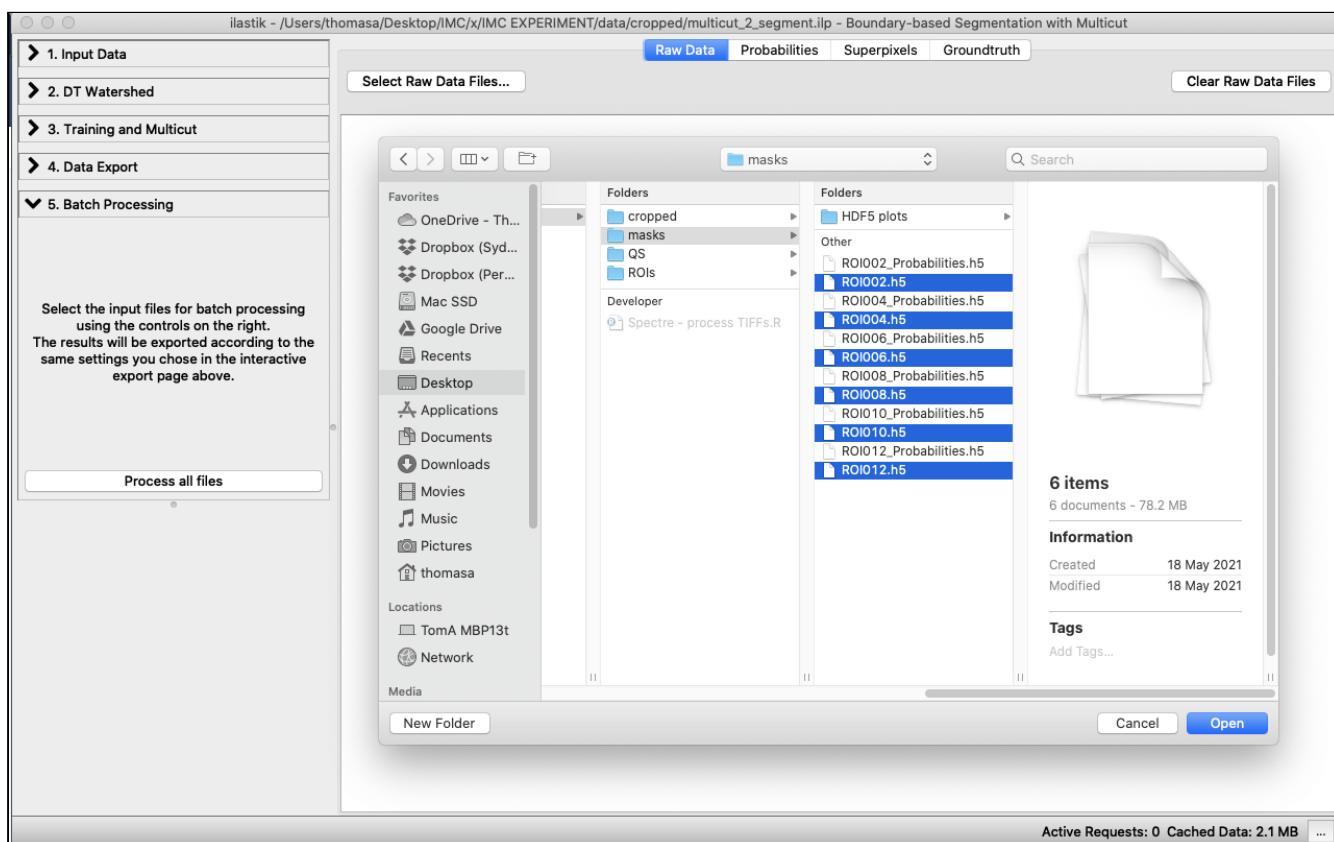
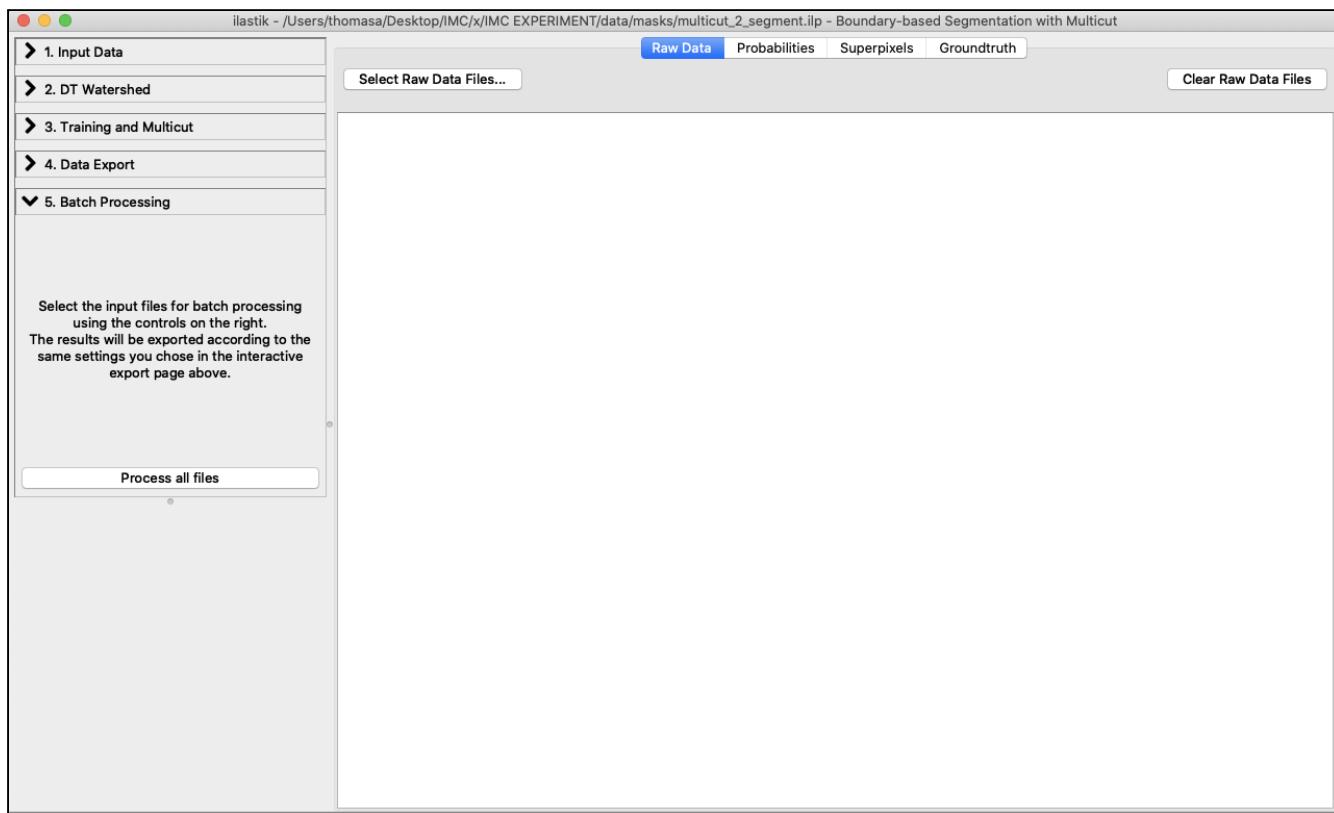


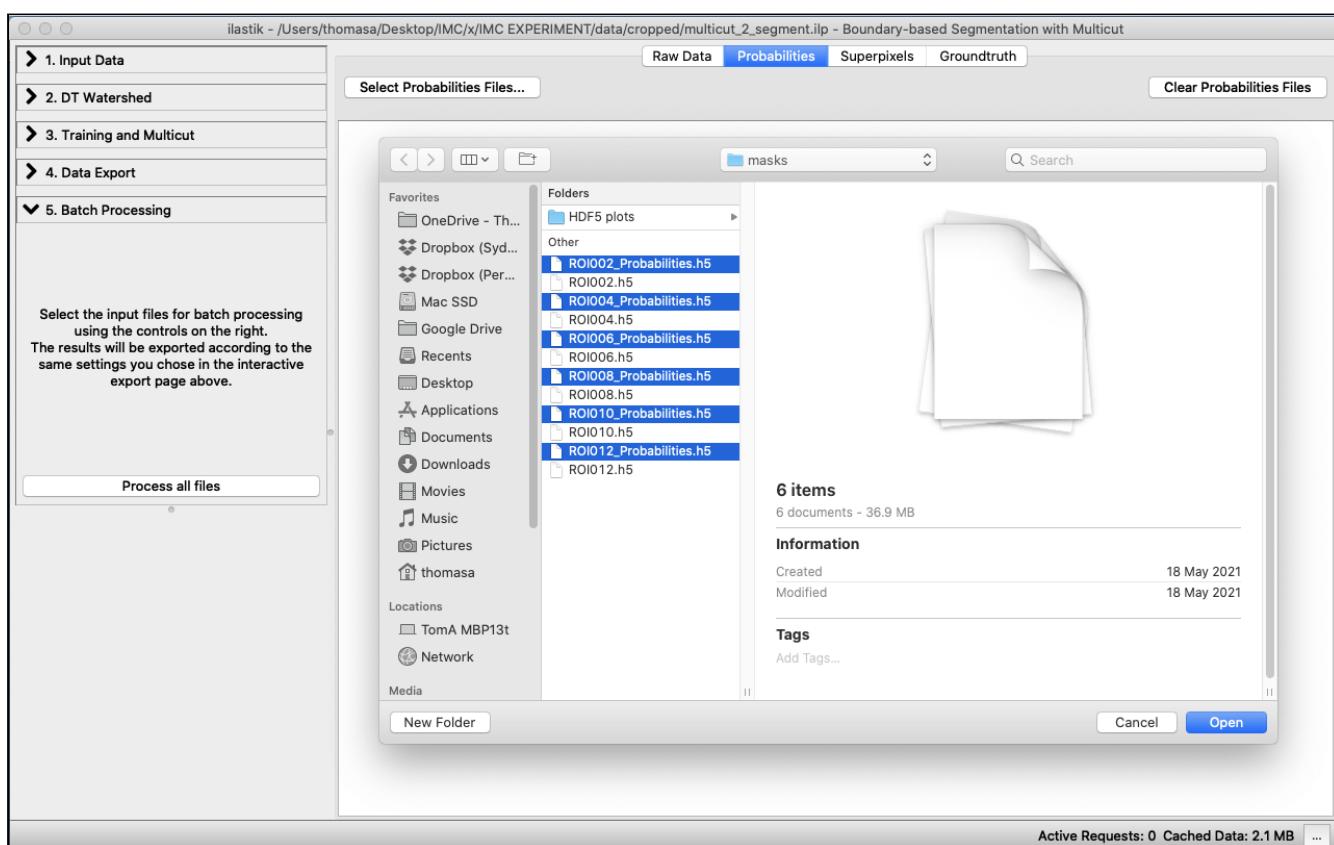
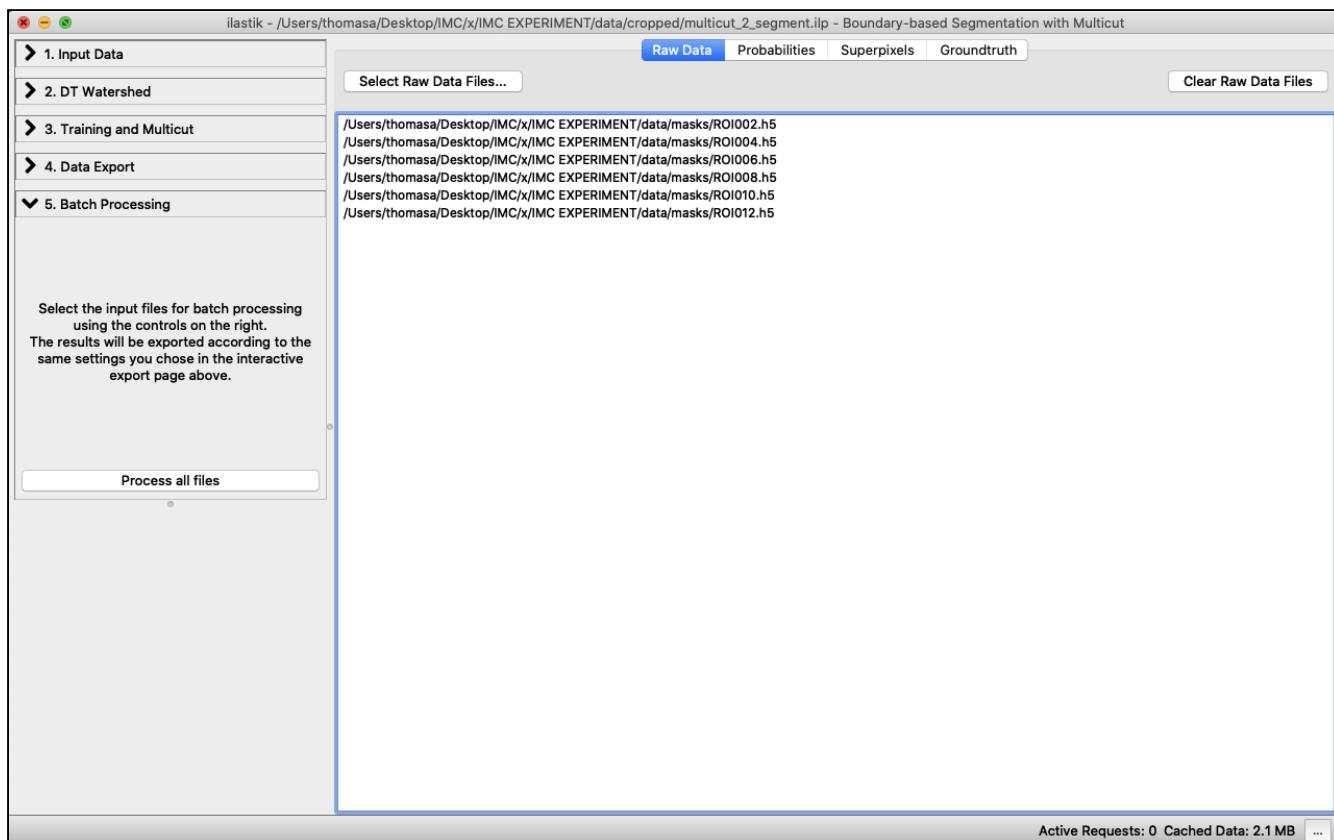
Press *i* on keyboard to flip between the layers and raw data only.

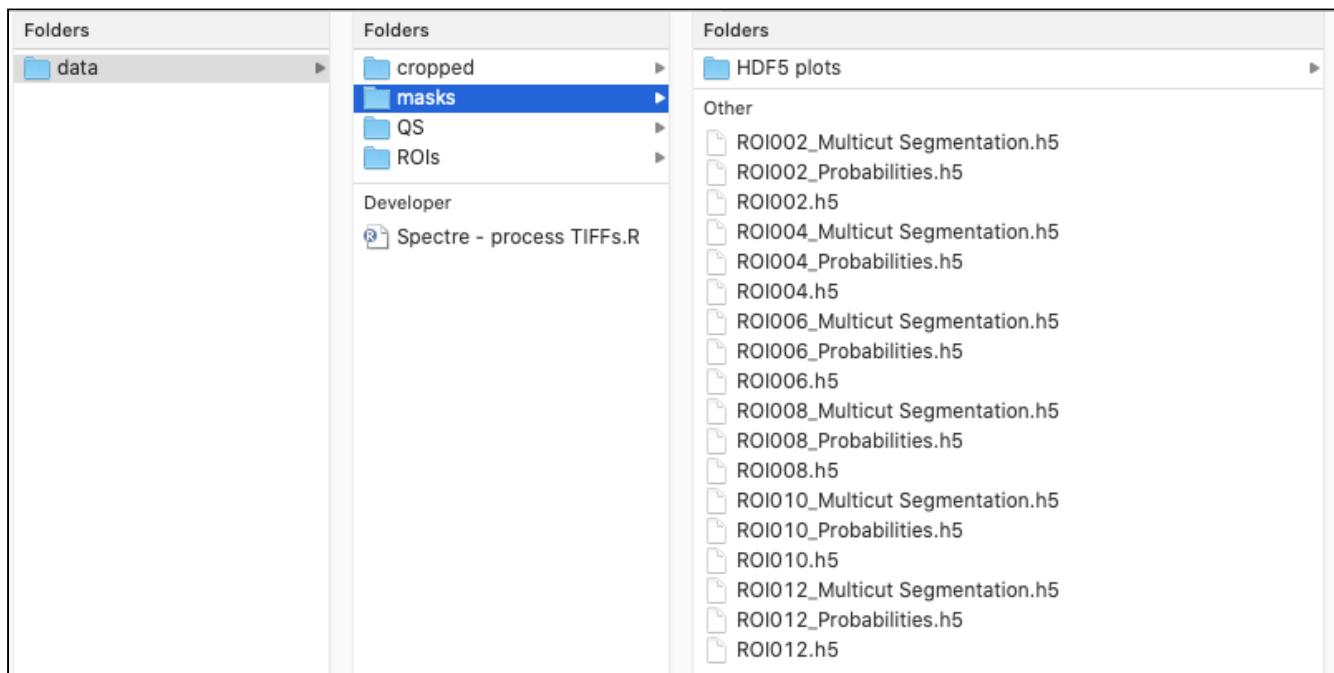
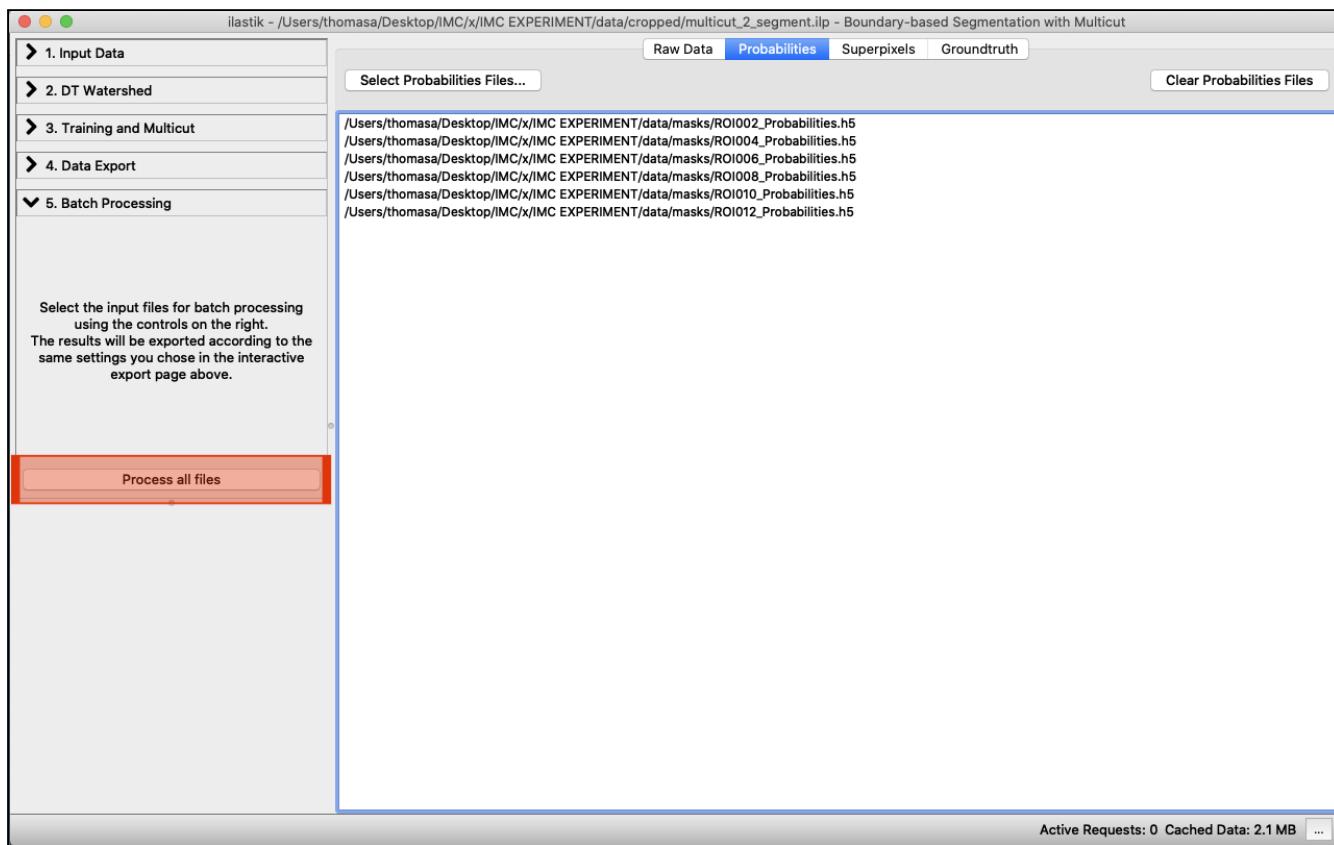
If this error comes up, best to create a new ilastik multicut file and start this step again.



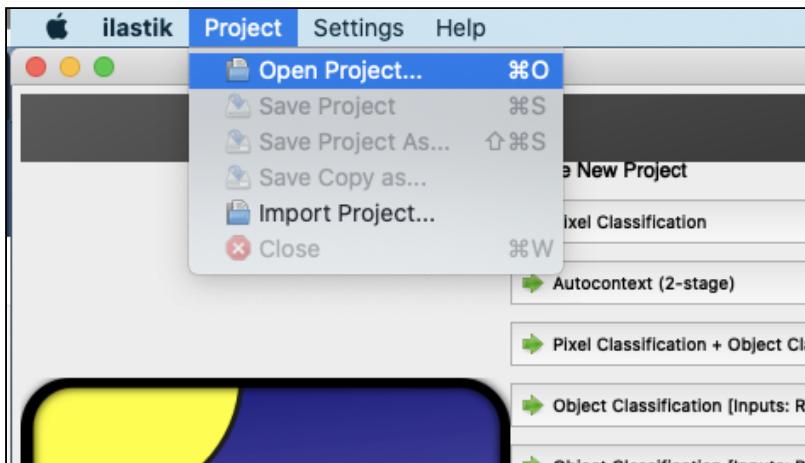




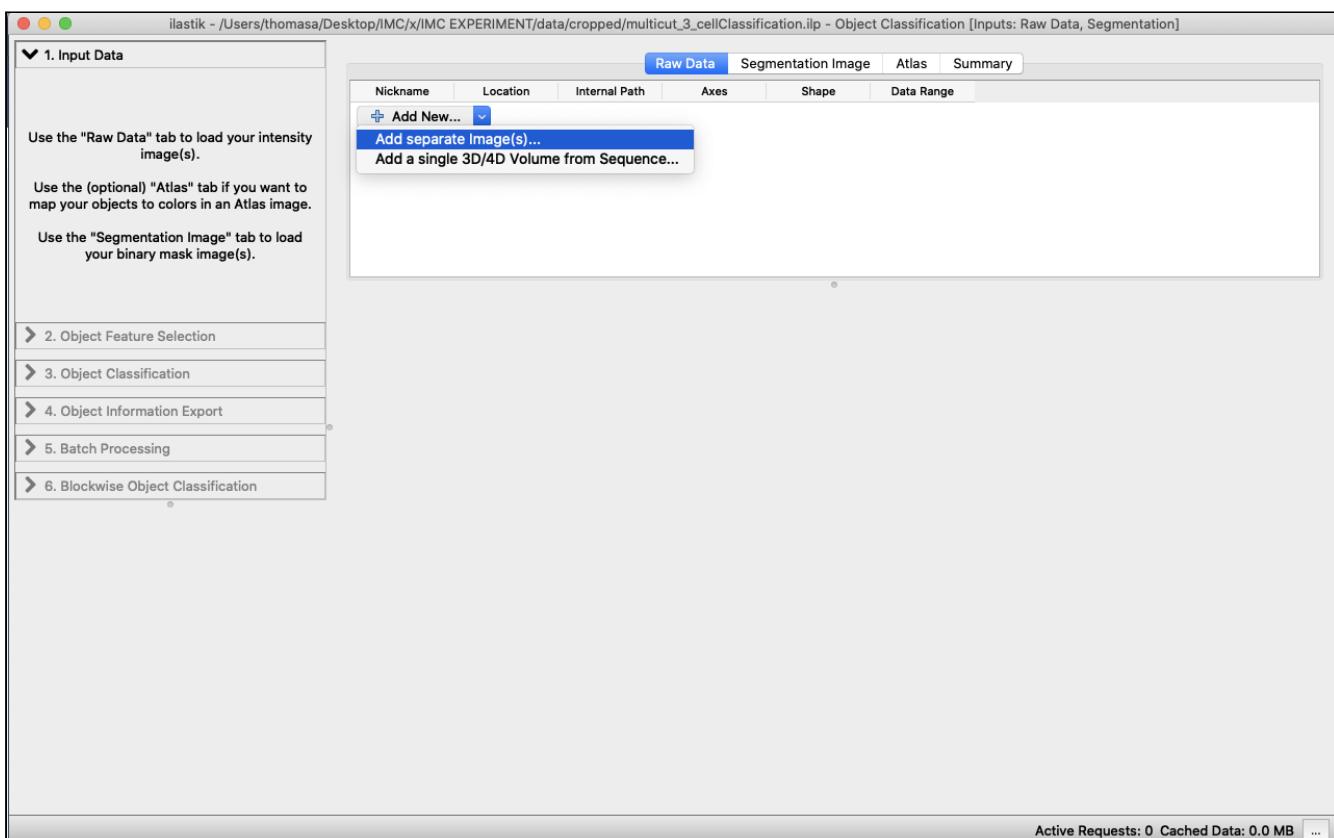
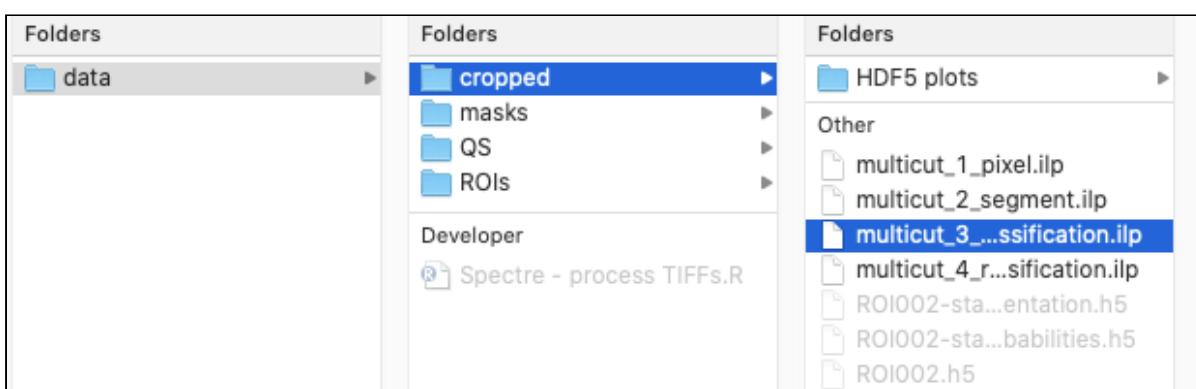


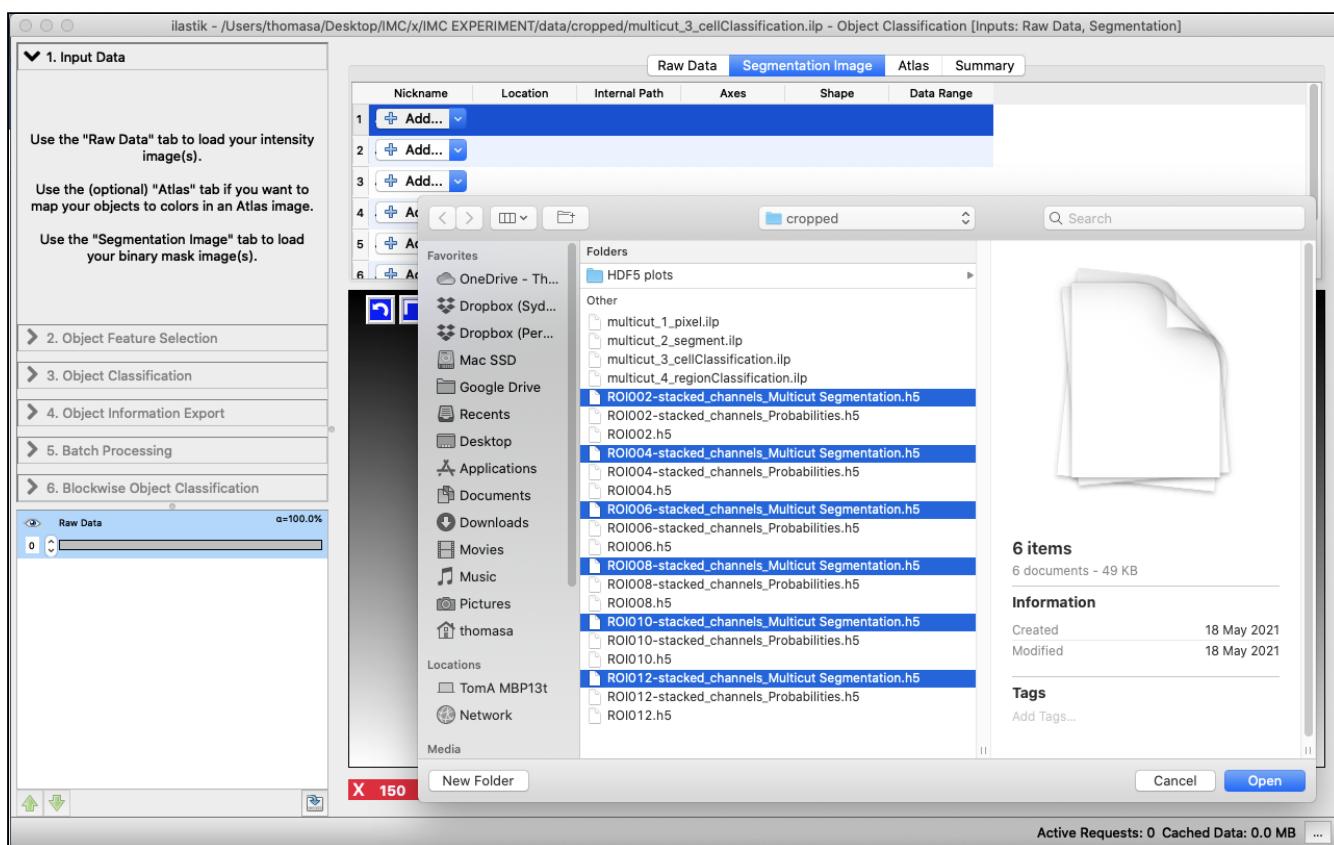
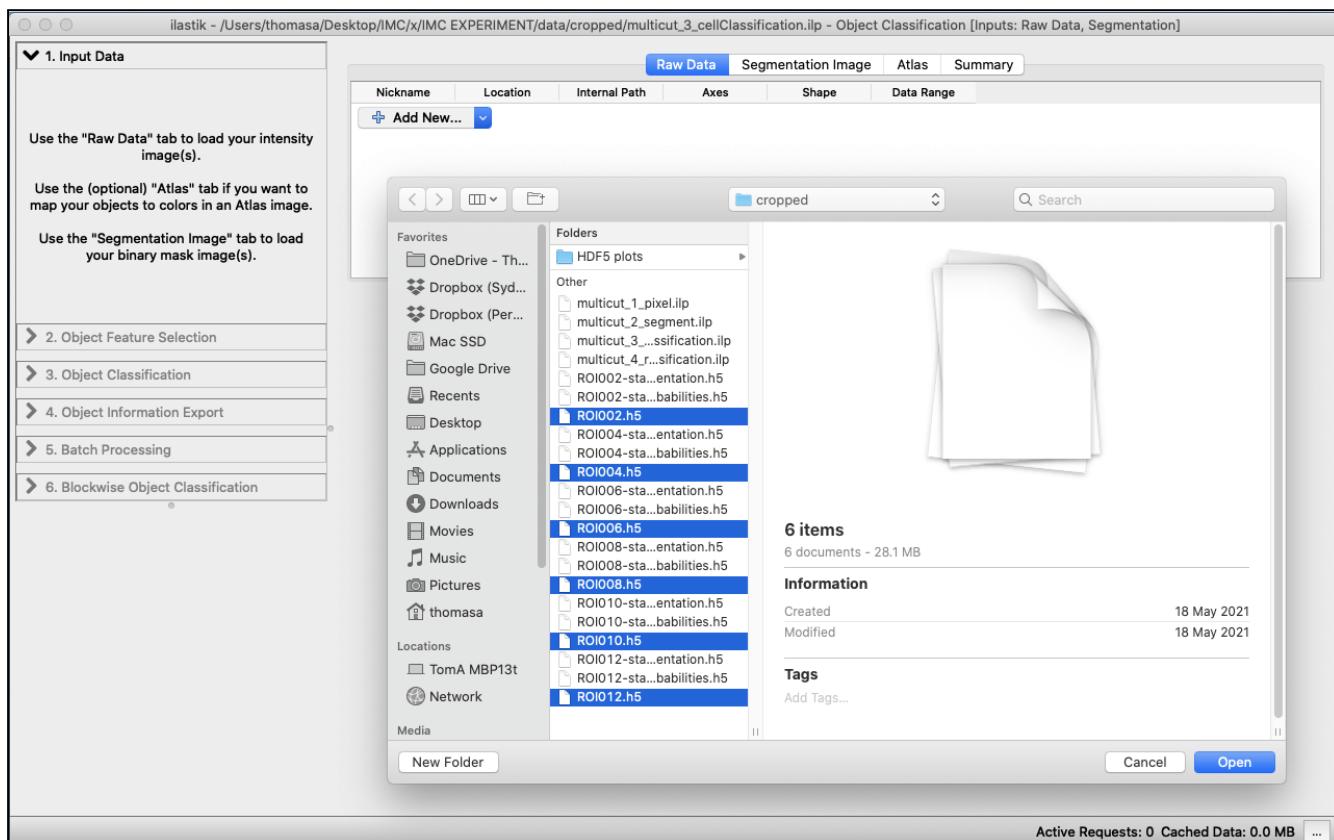


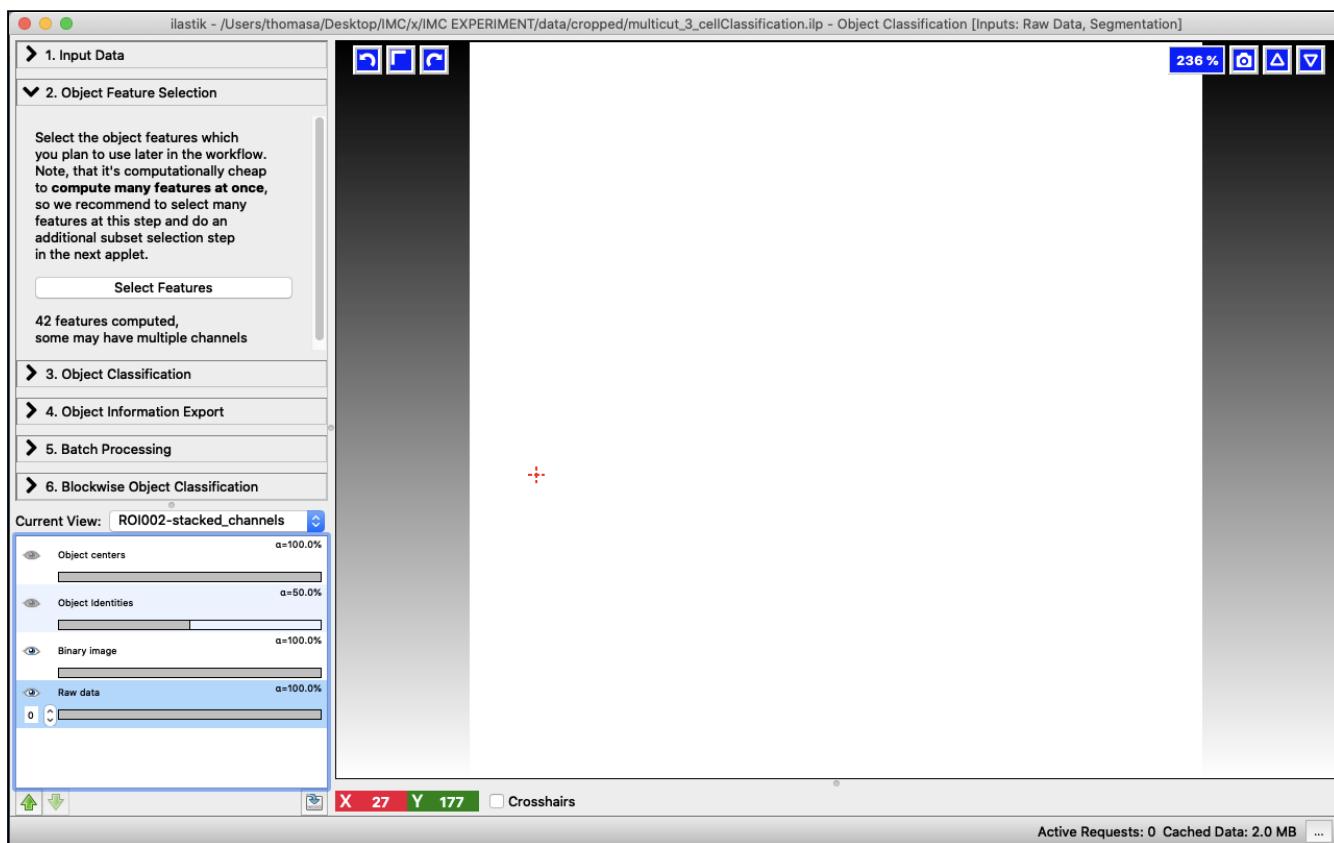
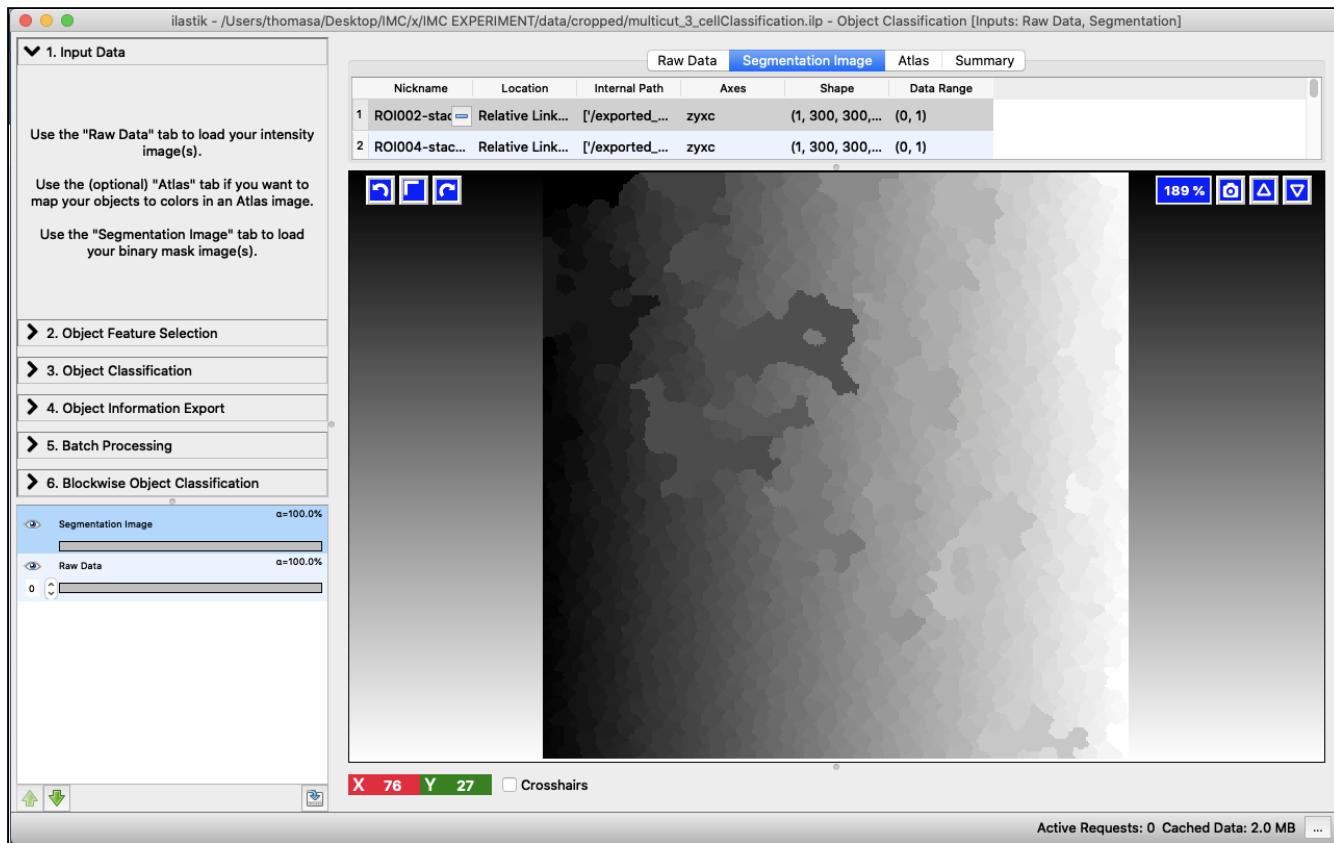
## 4. Ilastik - cell object classification

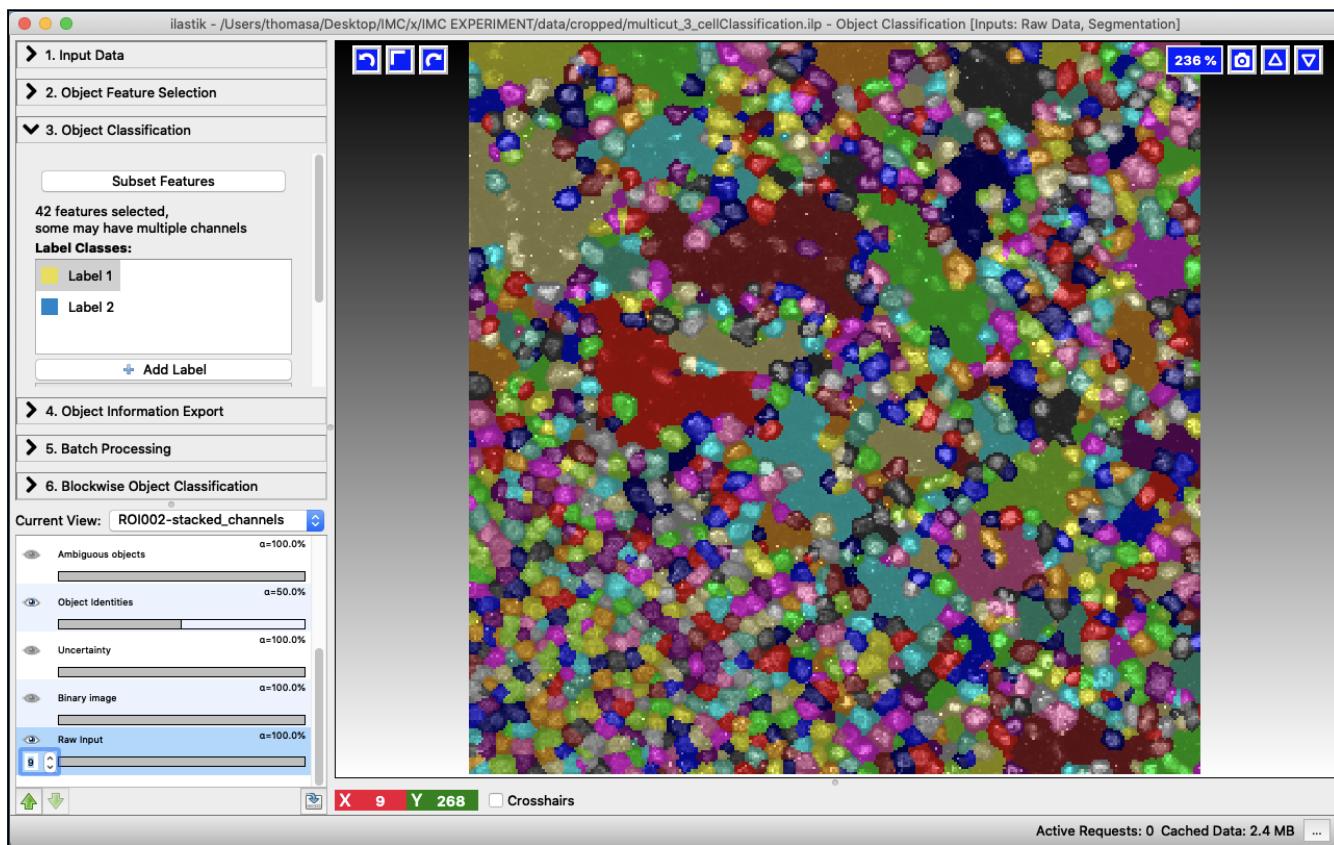
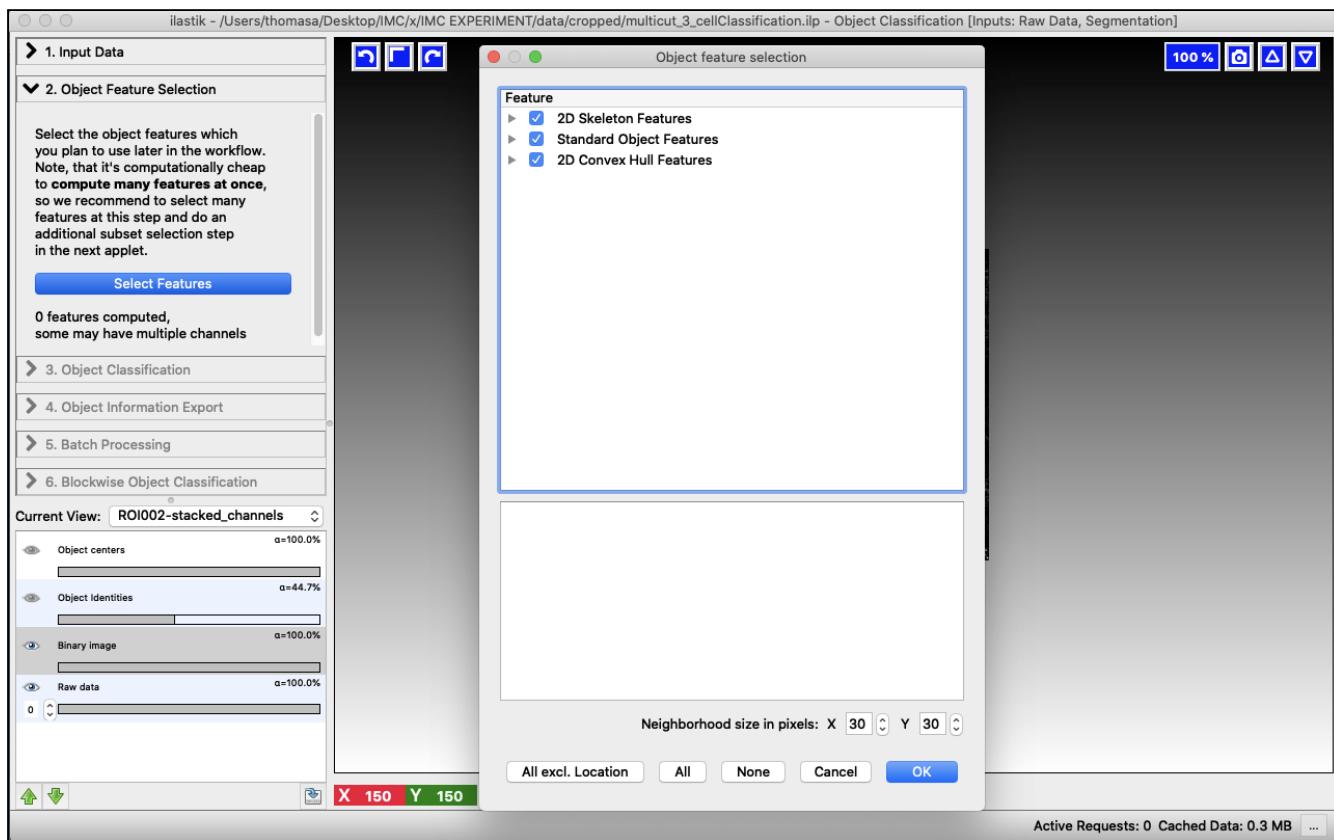


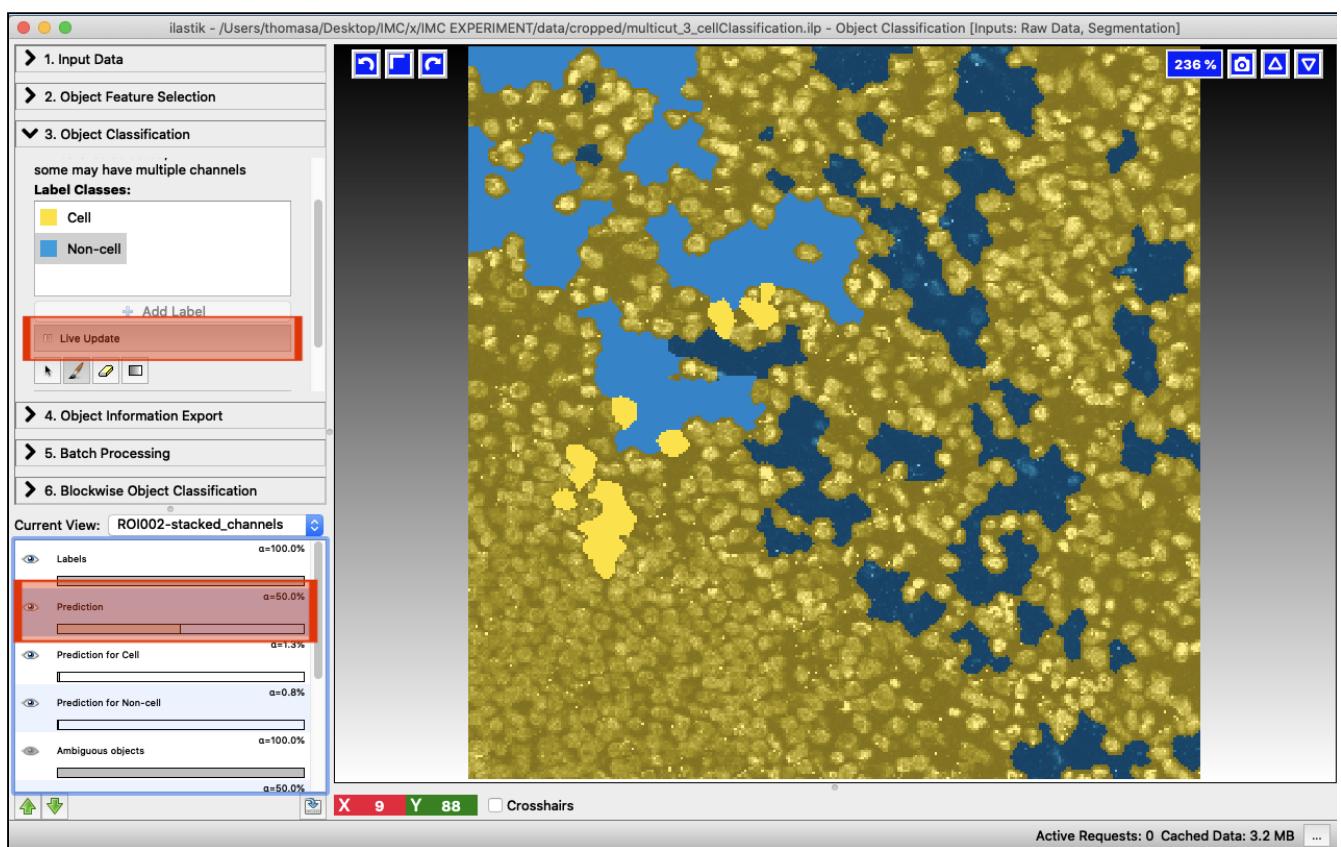
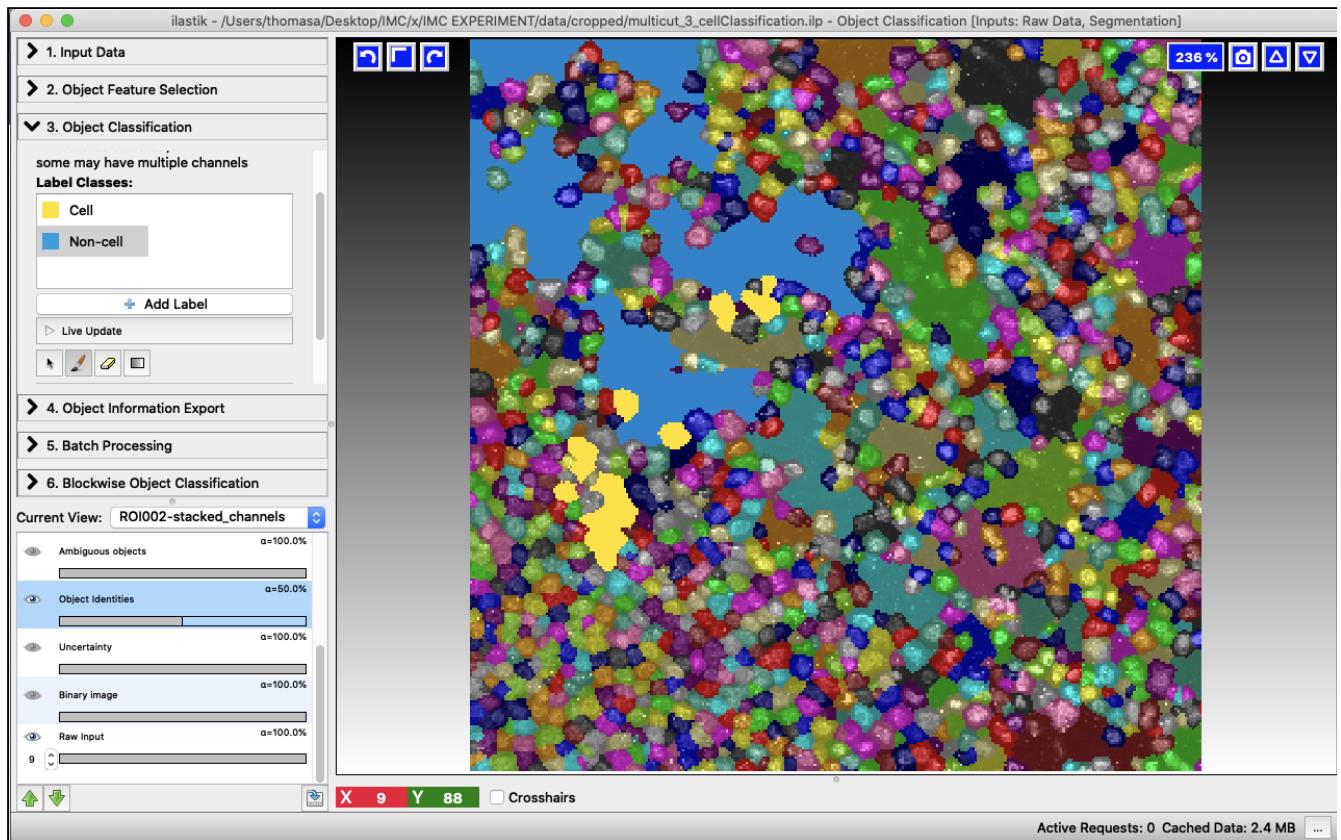
And select the 'multicut\_3\_cellClassification.ilp' file (or alternatively, create and save a new Ilastik file of the type: 'object classification' (inputs: Raw Data, Segmentation)).

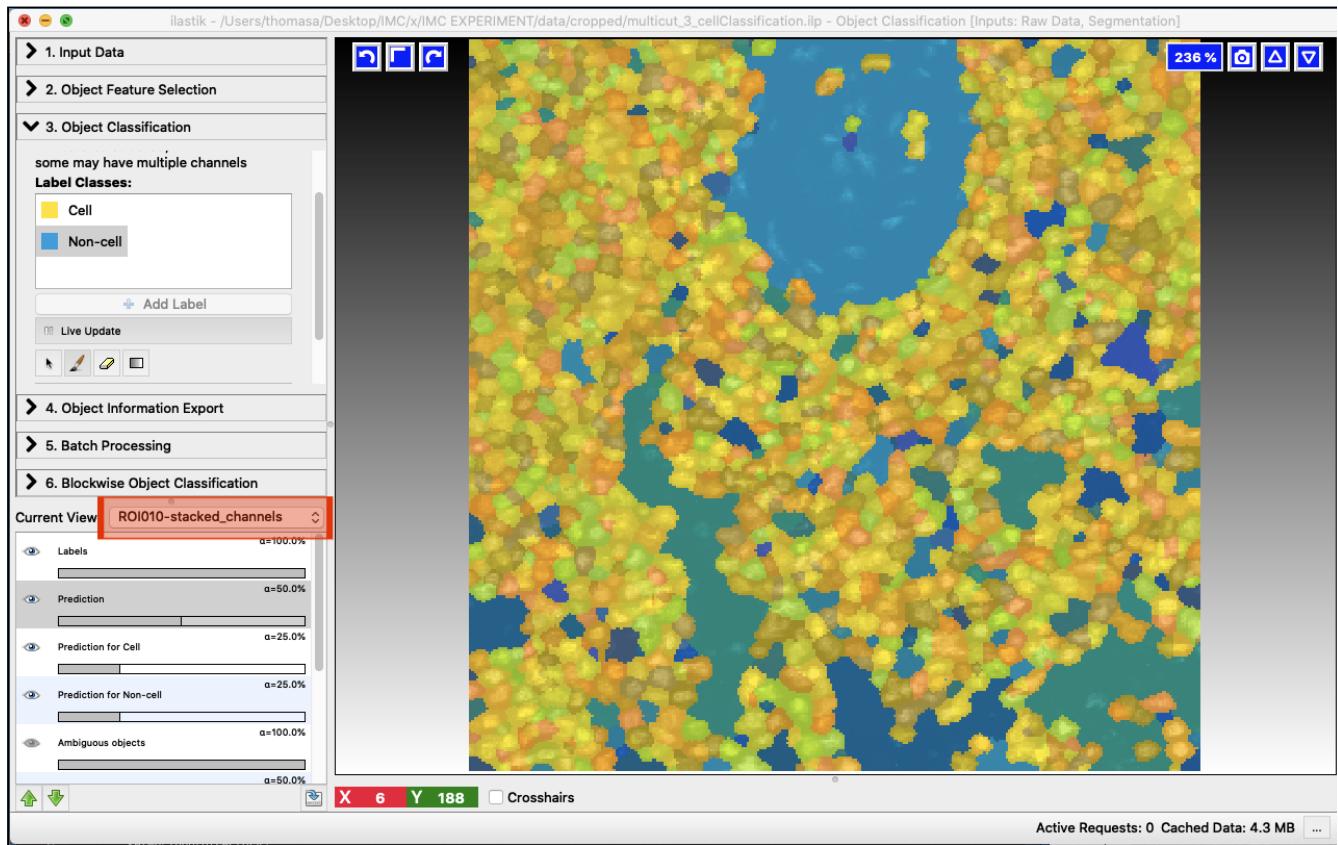
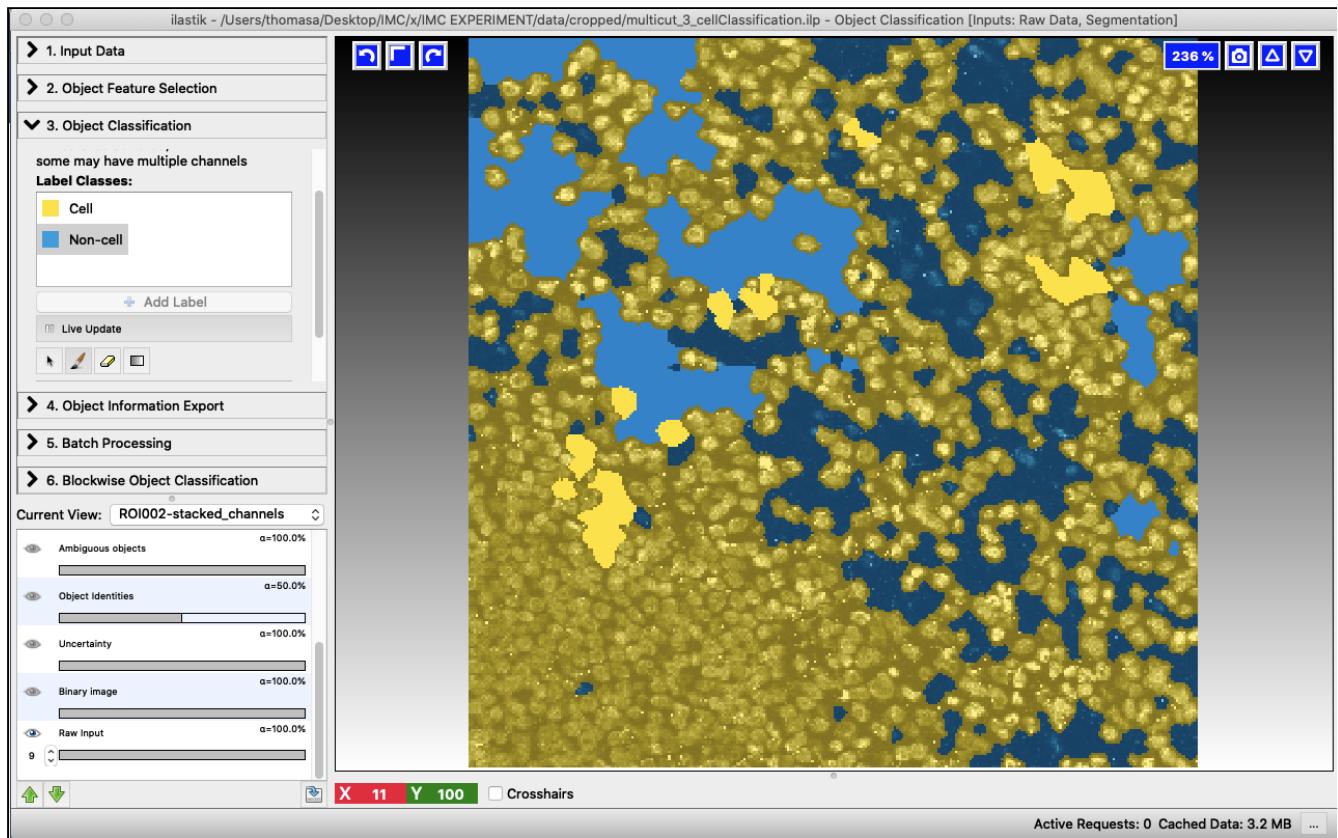










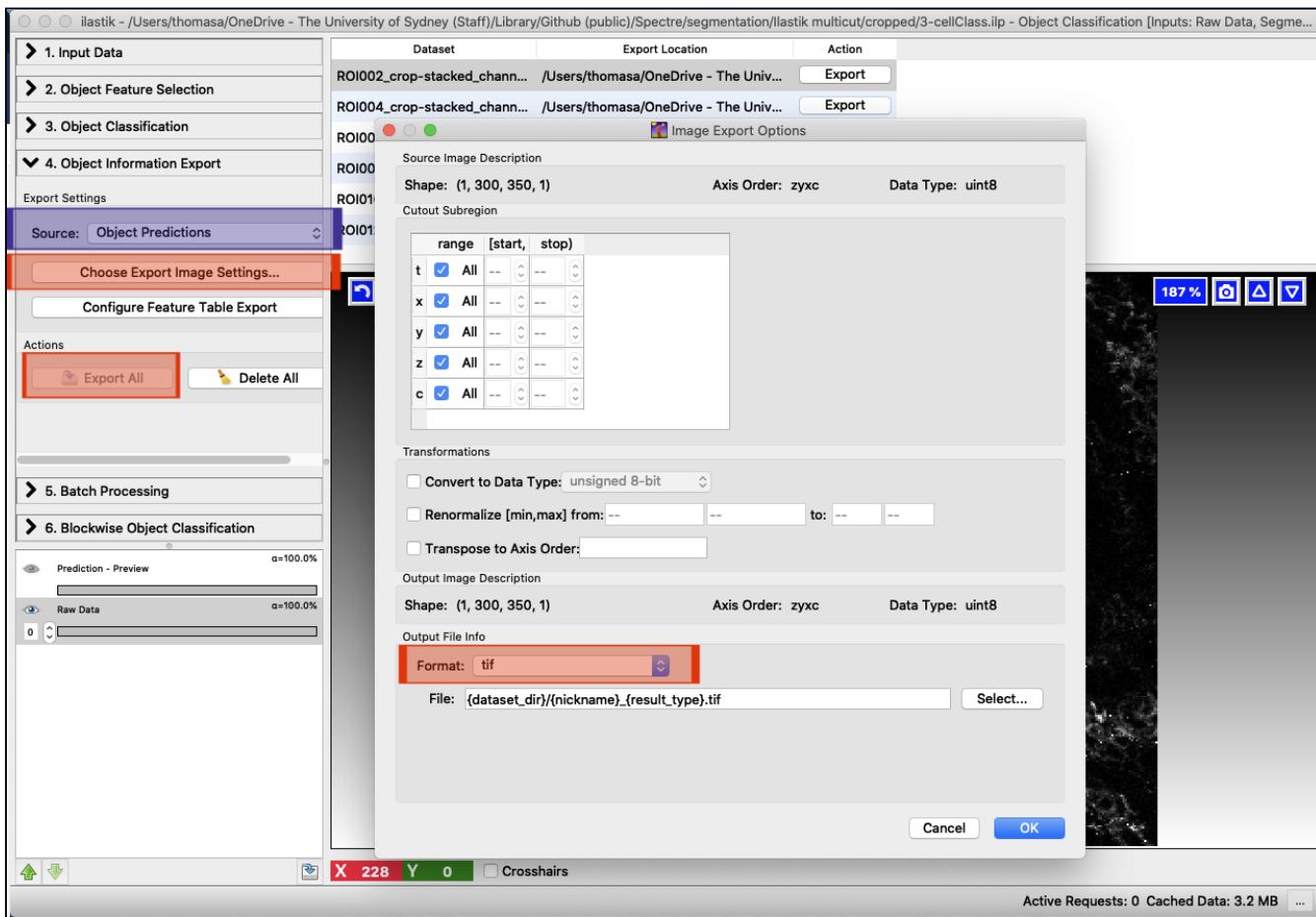
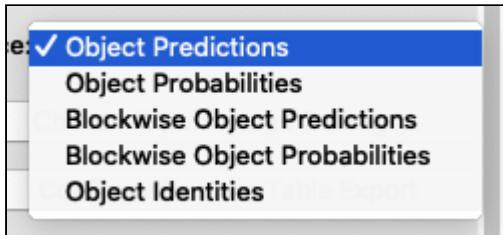


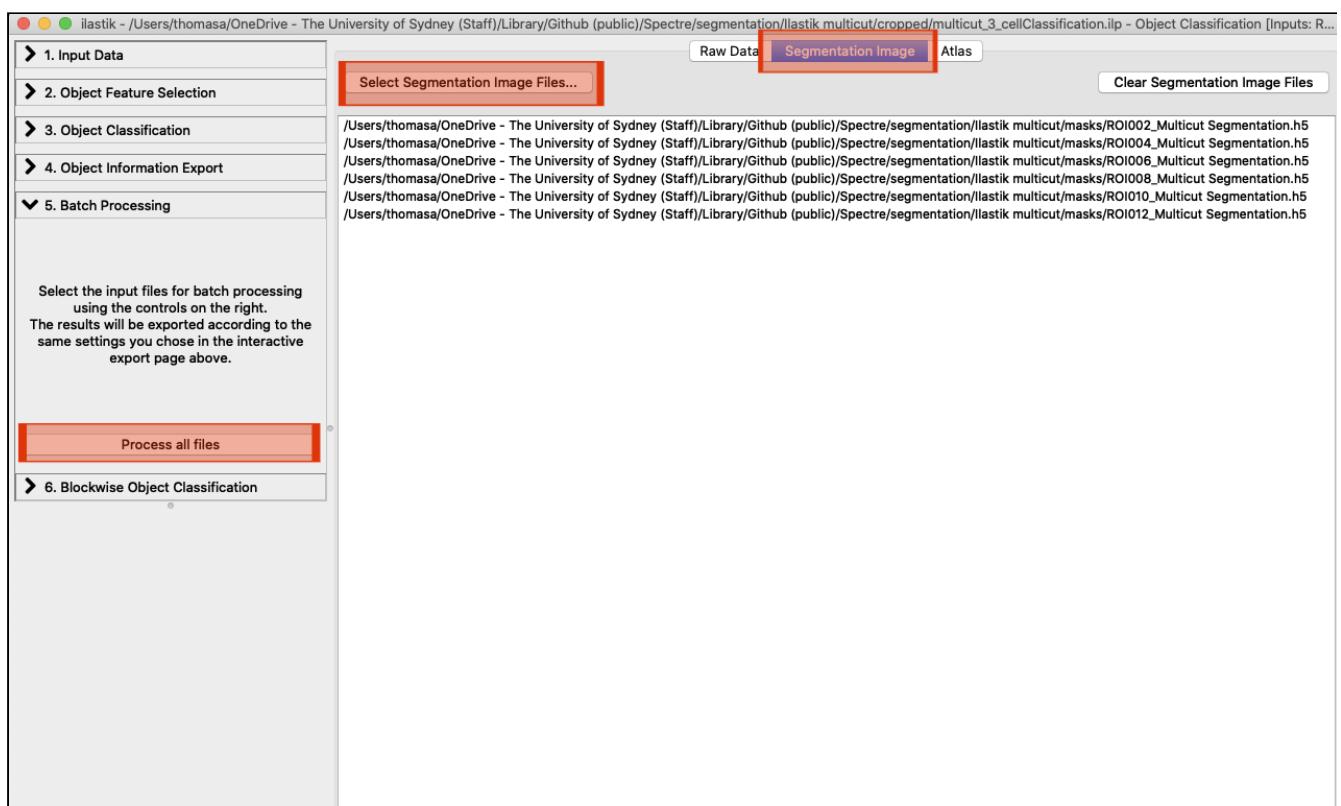
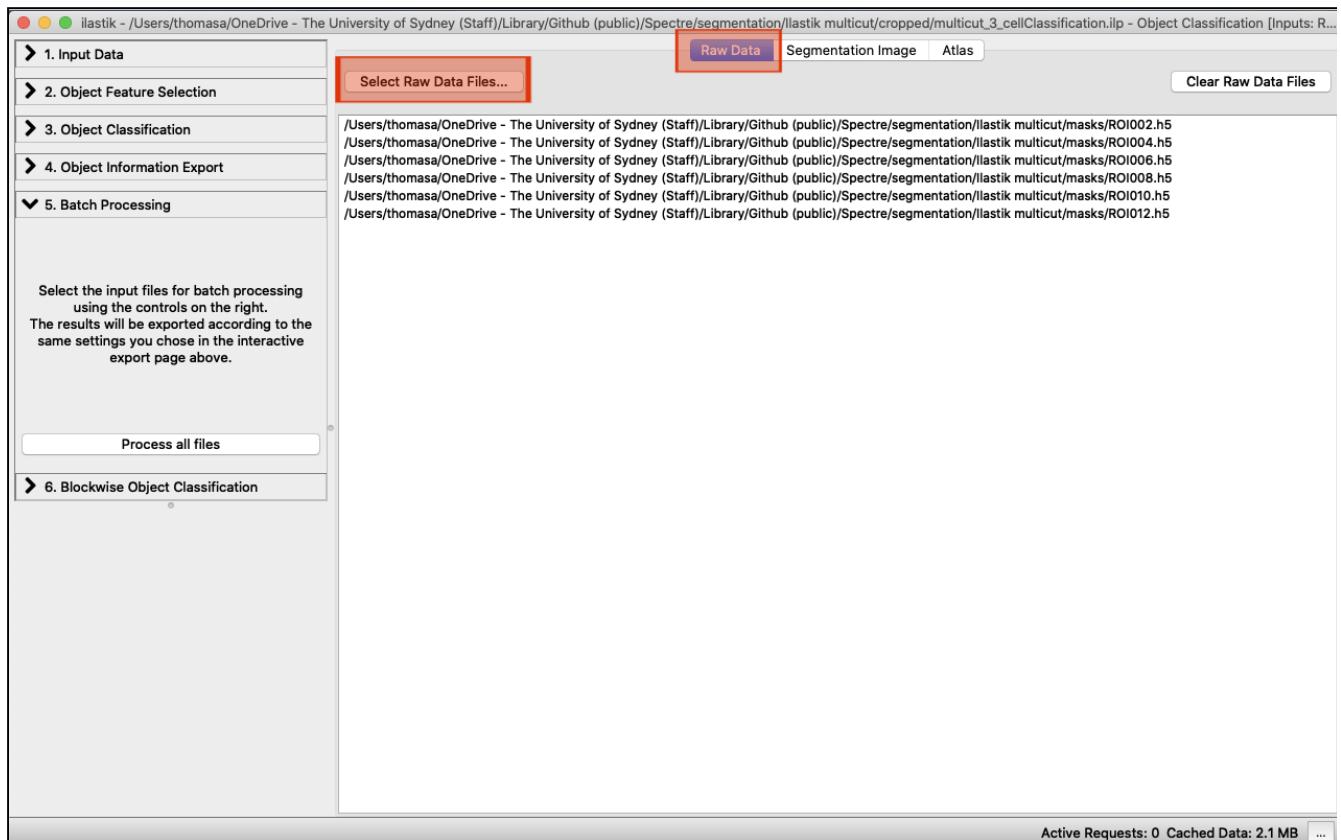
## ⚠ Two TIFF exports

Follow the export instructions below carefully, as there are two rounds of TIFF exports

### Export 1. Object predictions (as TIFFs)

These are cell classifications (e.g. cell, non-cell)

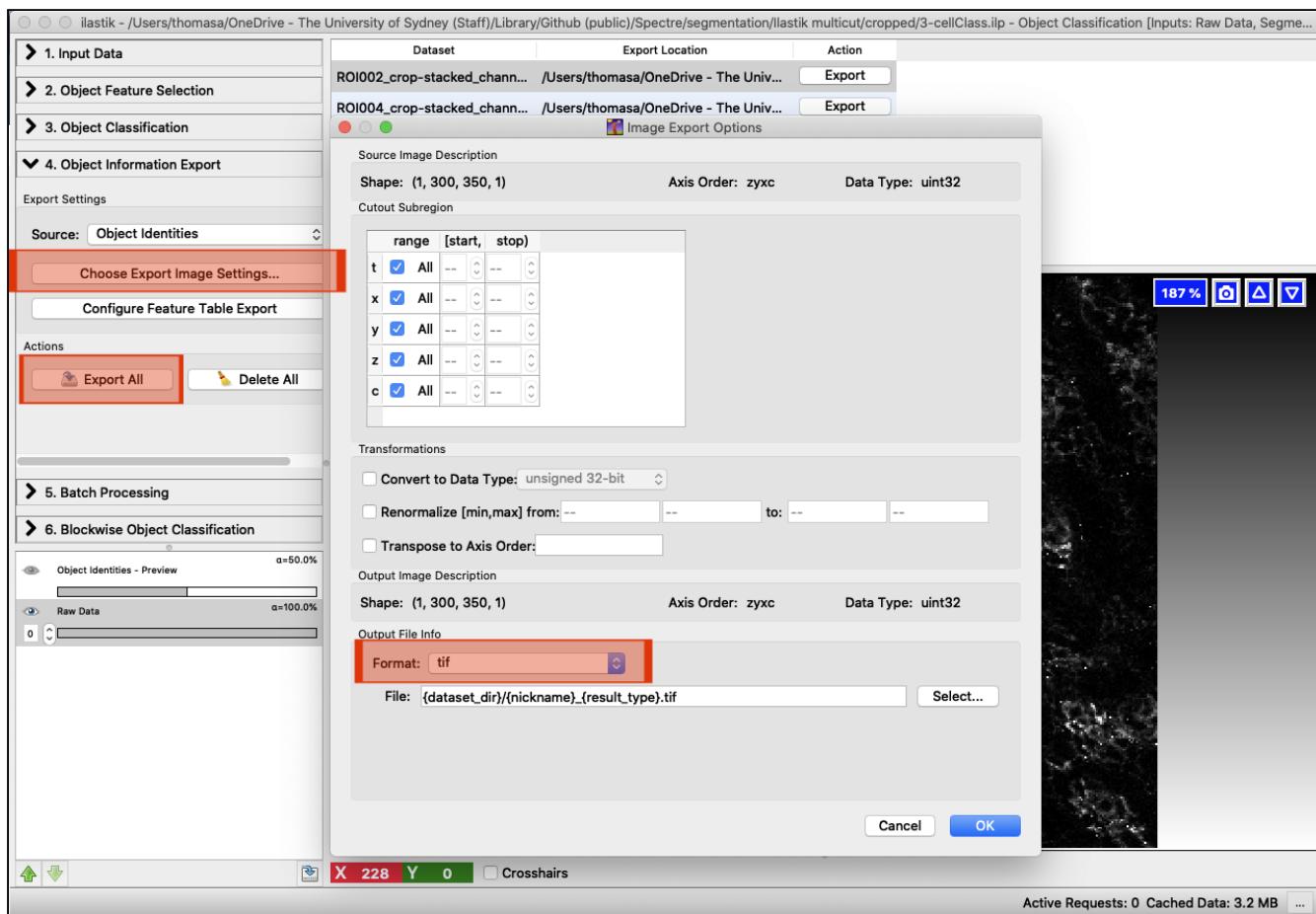




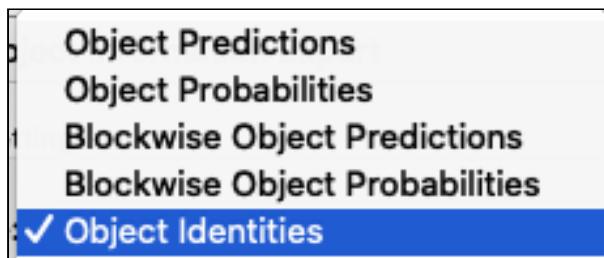
Images
ROI02_Object Predictions.tif
ROI04_Object Predictions.tif
ROI06_Object Predictions.tif
ROI08_Object Predictions.tif
ROI10_Object Predictions.tif
ROI12_Object Predictions.tif
Other
ROI02_Multicut Segmentation.h5
ROI02_Probabilities.h5
ROI02.h5
ROI04_Multicut Segmentation.h5
ROI04_Probabilities.h5
ROI04.h5
ROI06_Multicut Segmentation.h5
ROI06_Probabilities.h5
ROI06.h5
ROI08_Multicut Segmentation.h5
ROI08_Probabilities.h5
ROI08.h5
ROI10_Multicut Segmentation.h5
ROI10_Probabilities.h5
ROI10.h5
ROI12_Multicut Segmentation.h5
ROI12_Probabilities.h5
ROI12.h5

## Export 2. Object identities (as TIFFs)

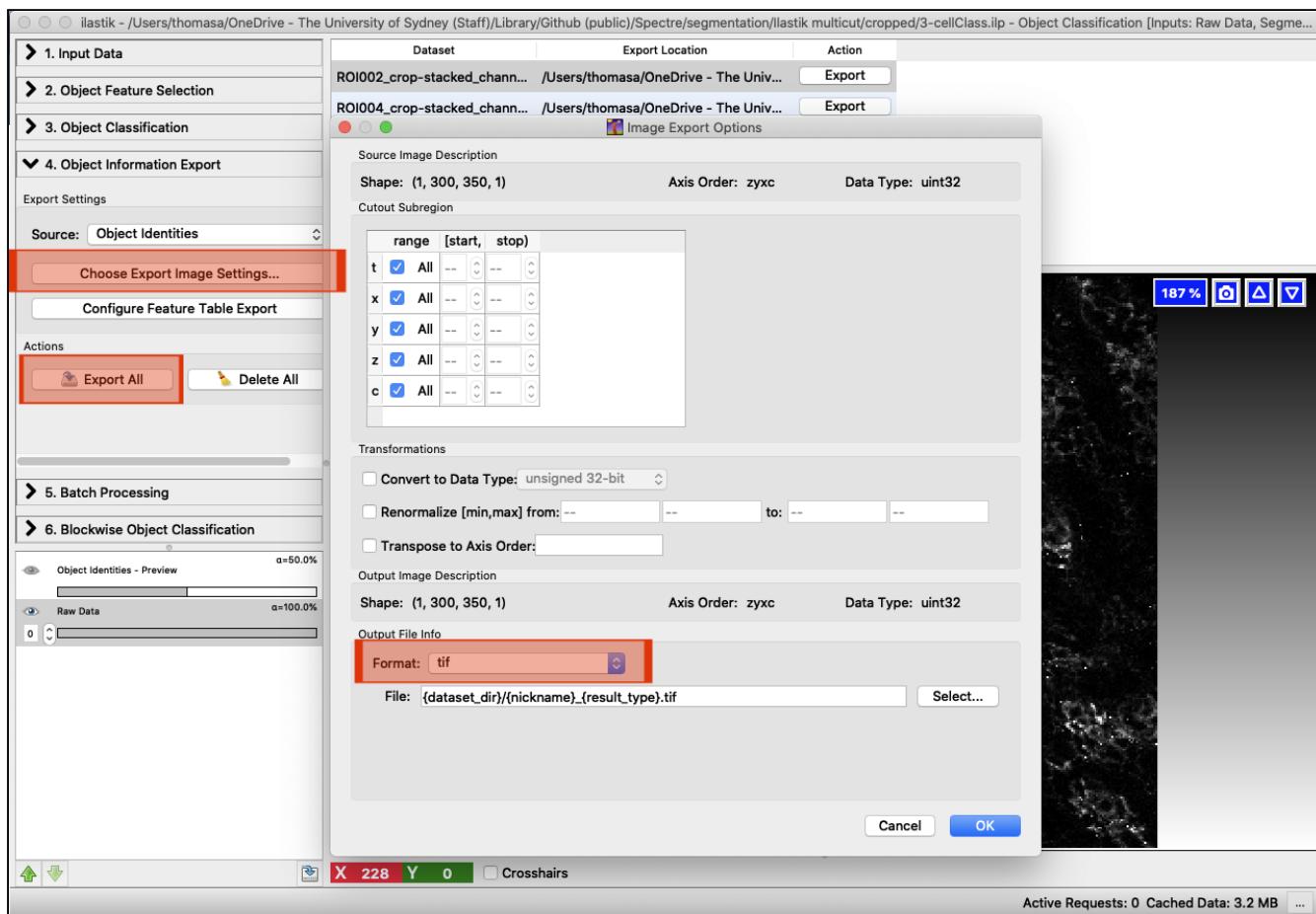
Go back to '4. Object Information Export'. Select Object Identities – these are individual cell masks.



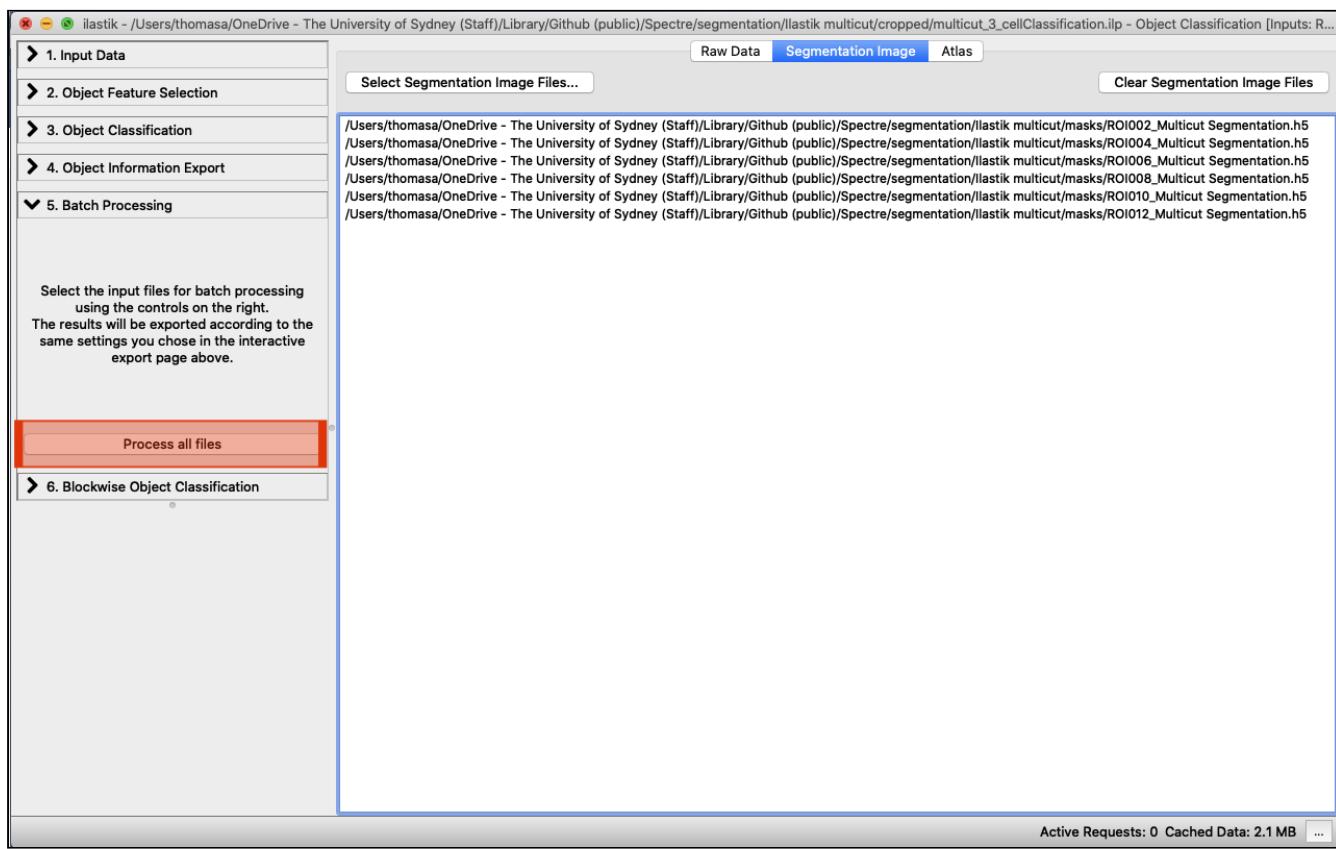
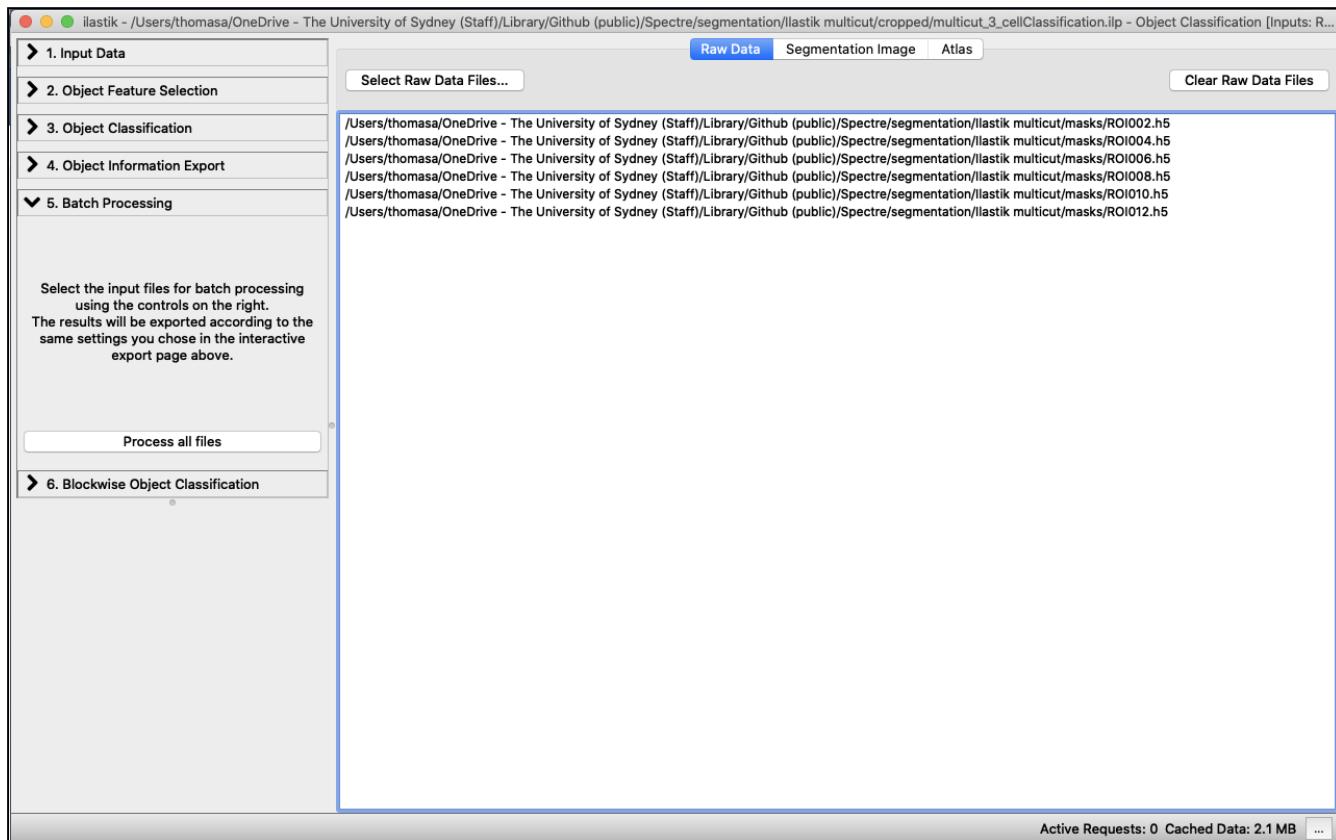
Under 'Source', select 'Object Identities'.

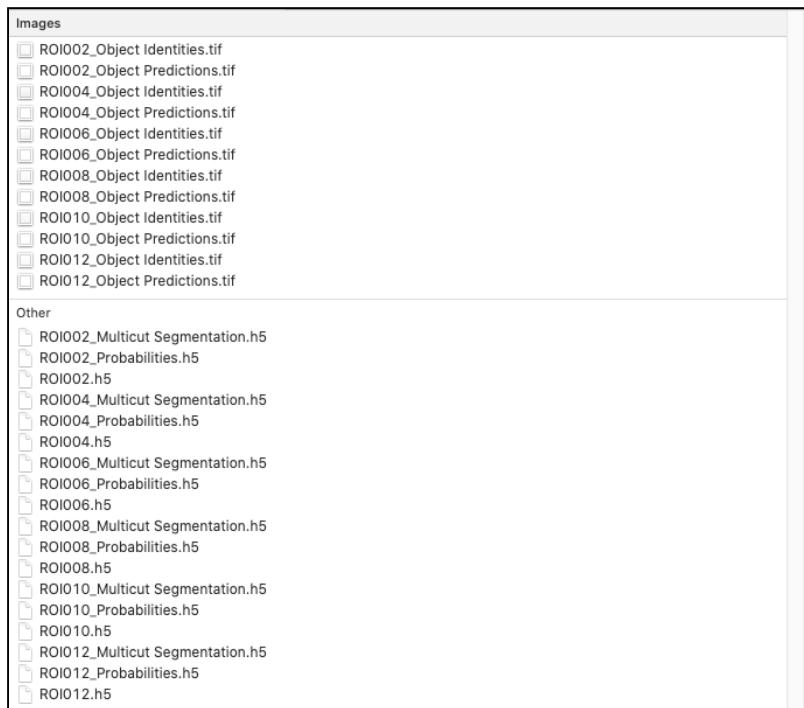


Change output format to tif.

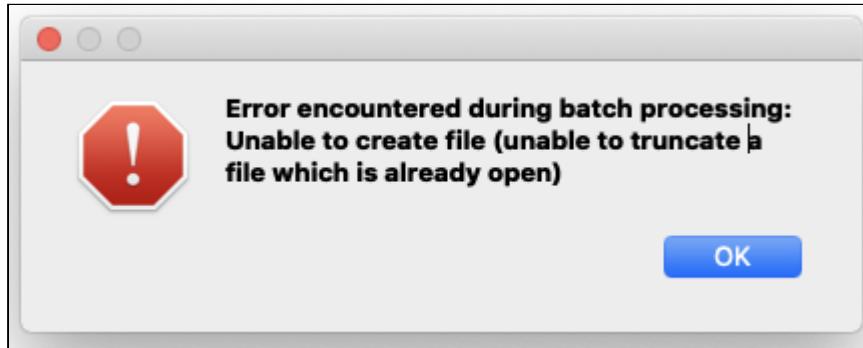


Double check the raw data and segmentation images are still listed.



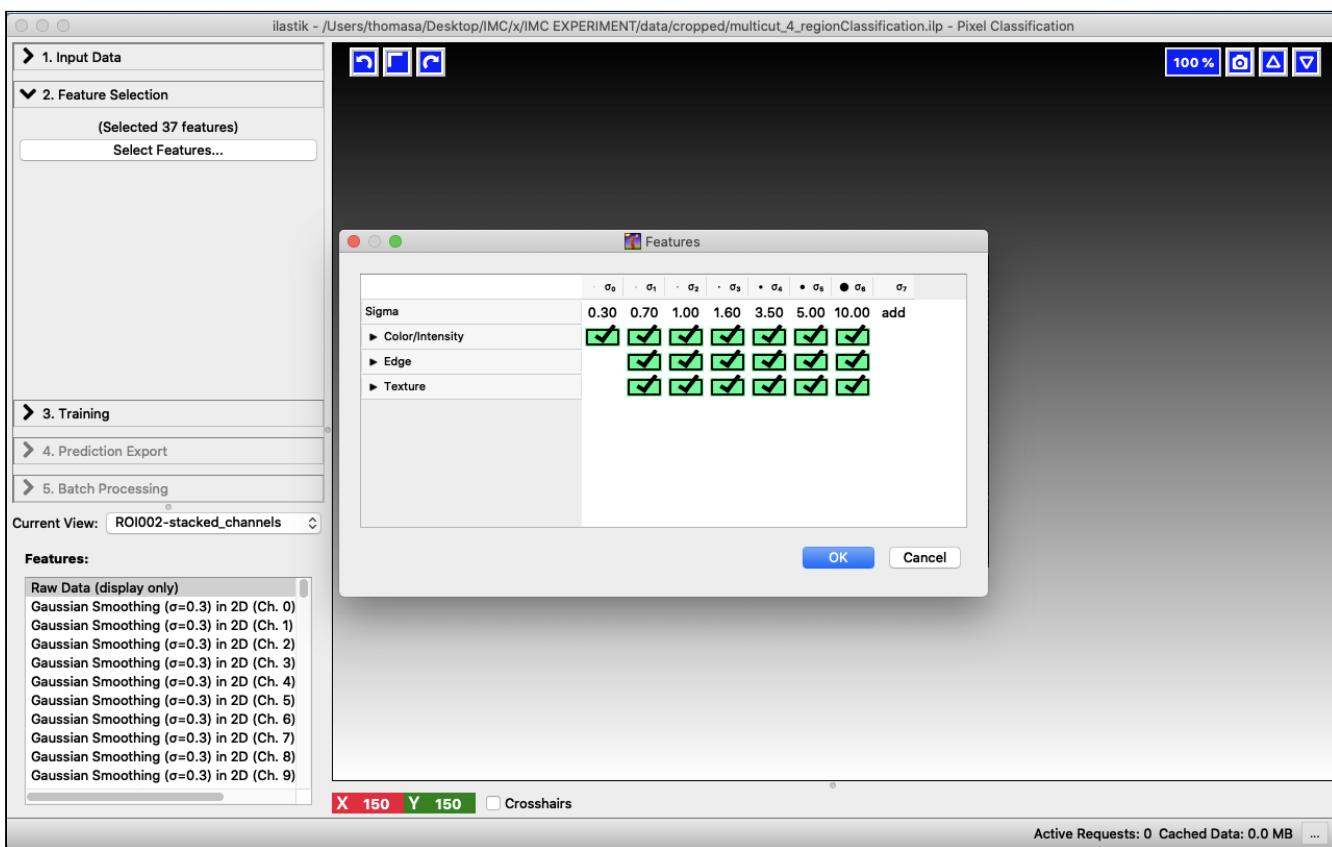
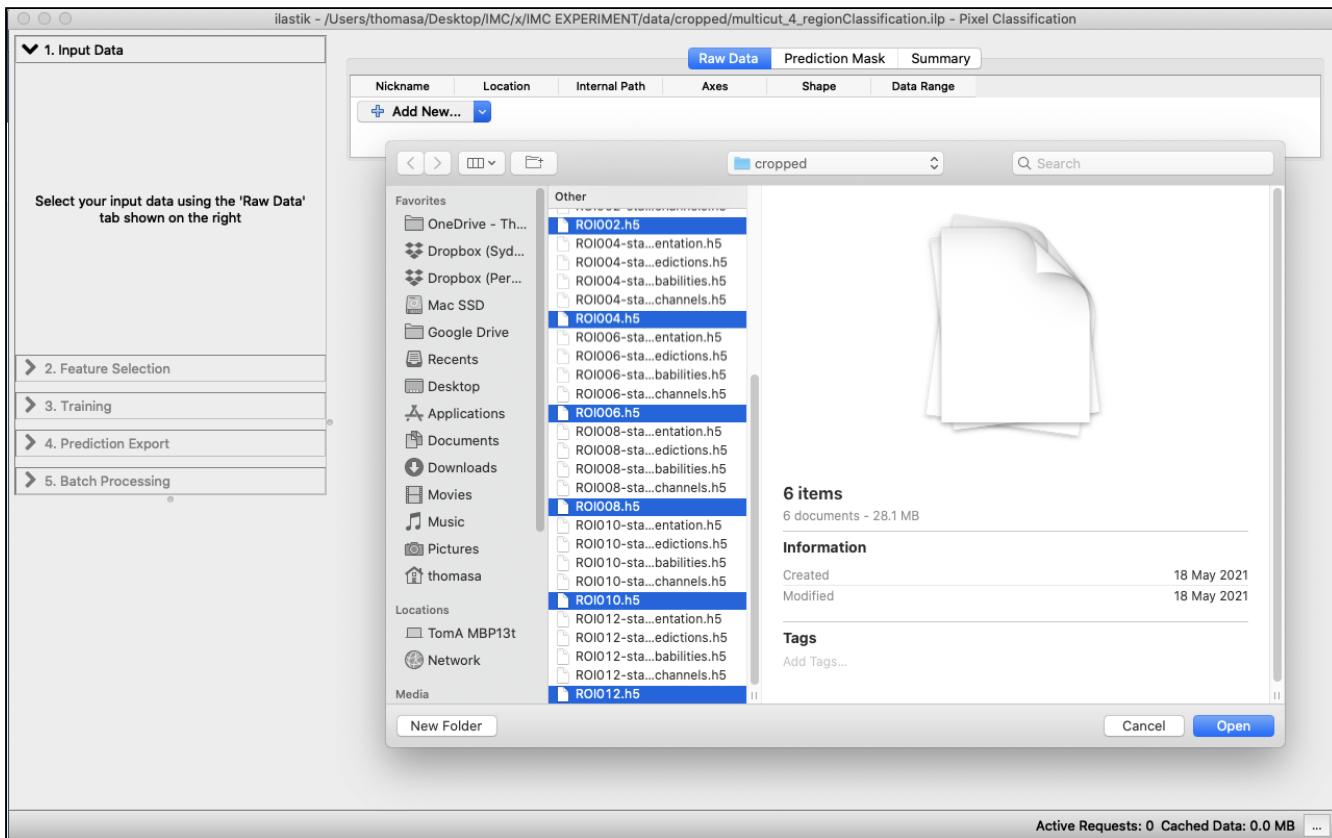


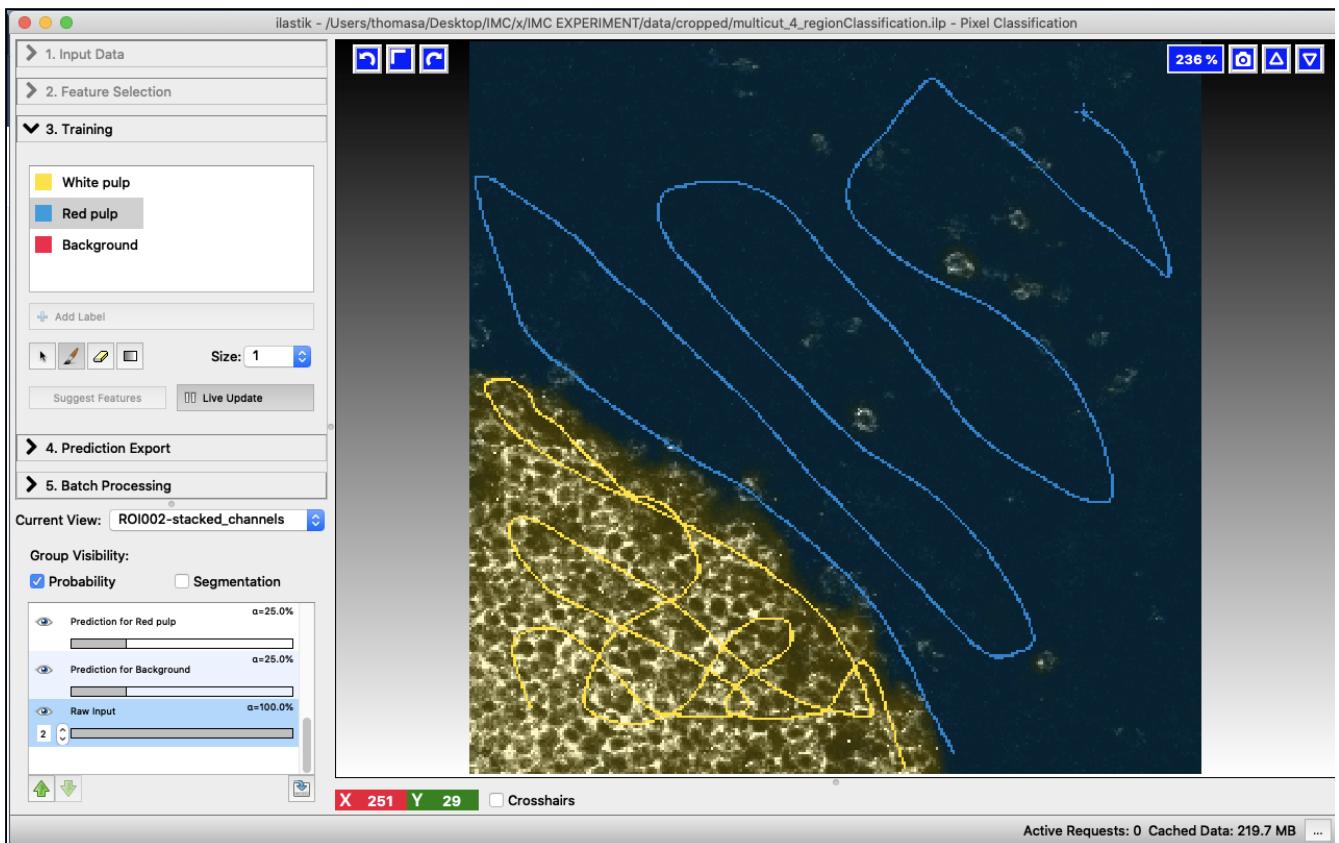
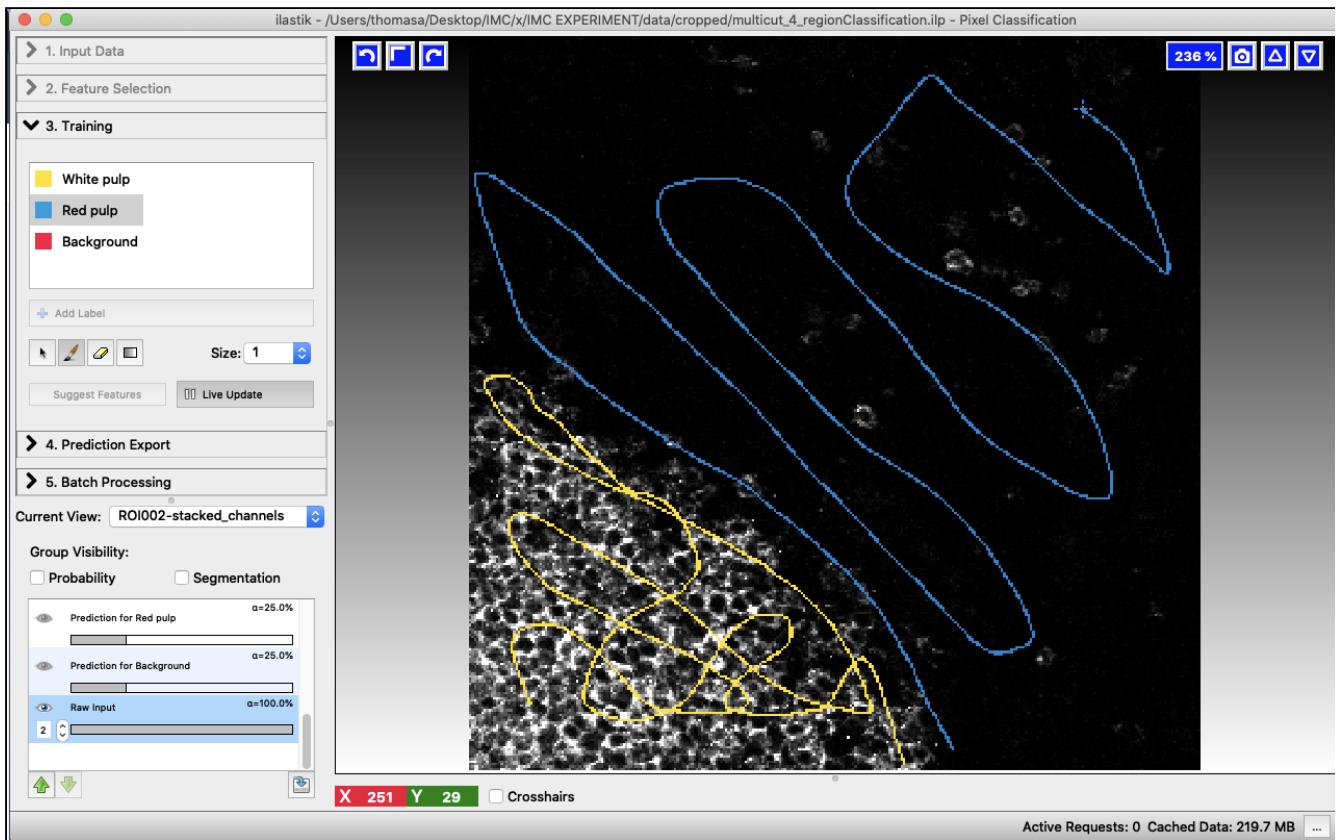
If this error comes up, delete the Ilastik file and start this section again.

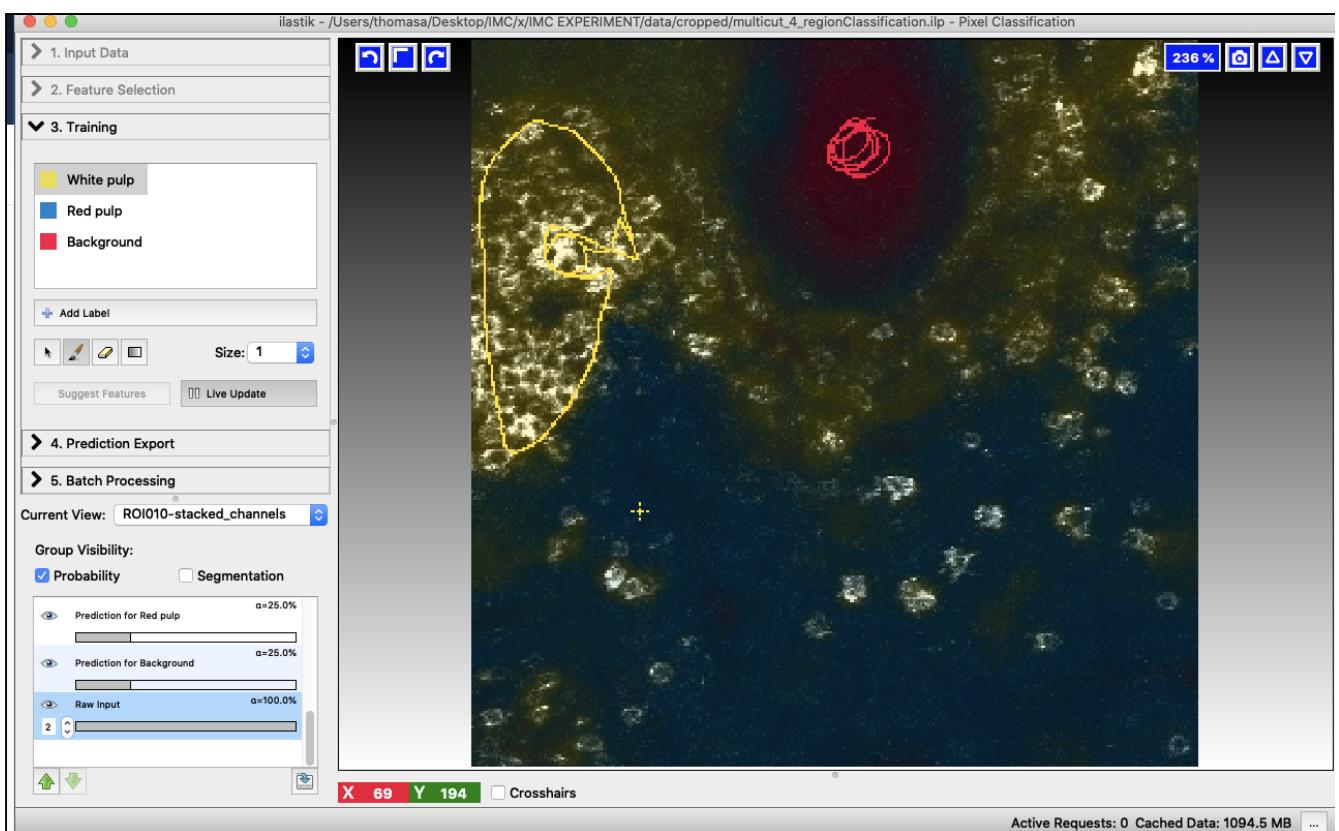
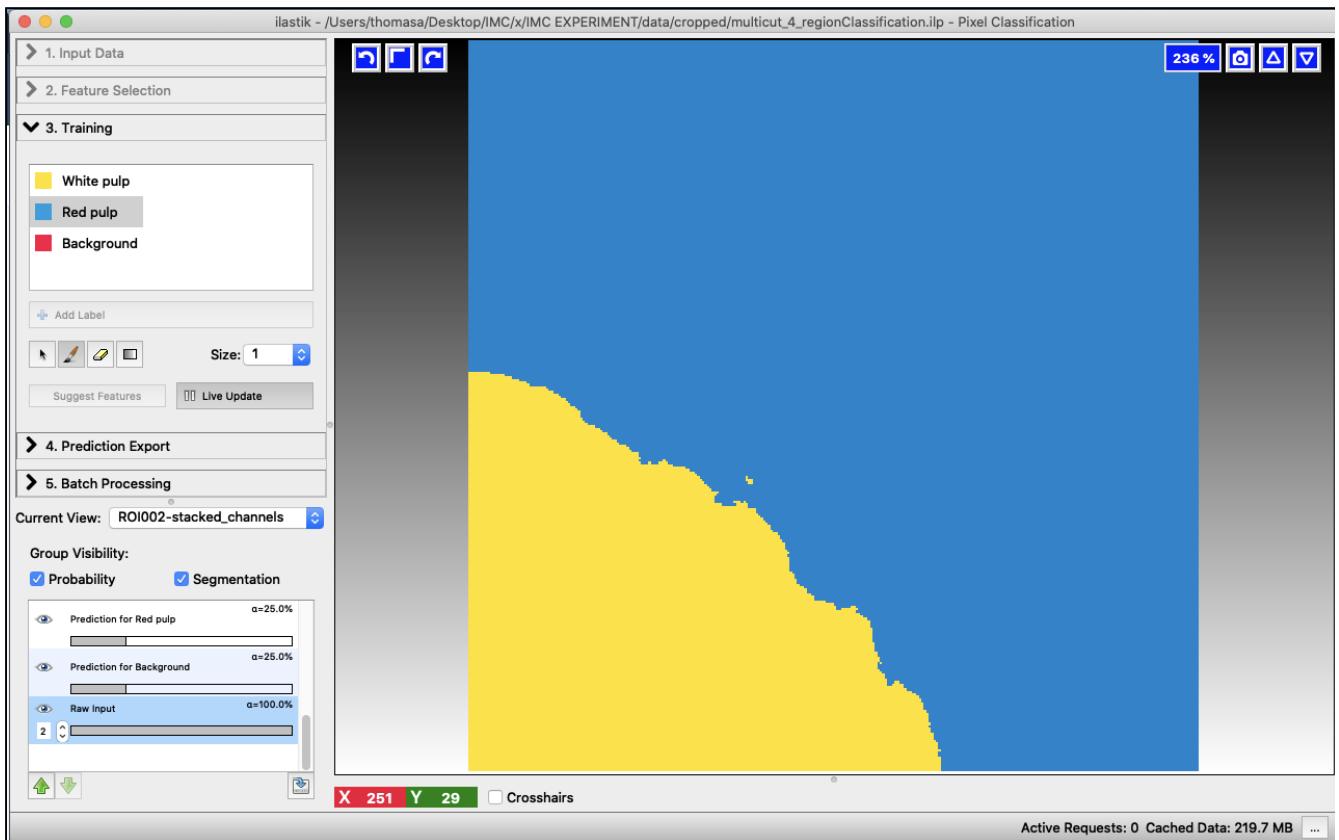


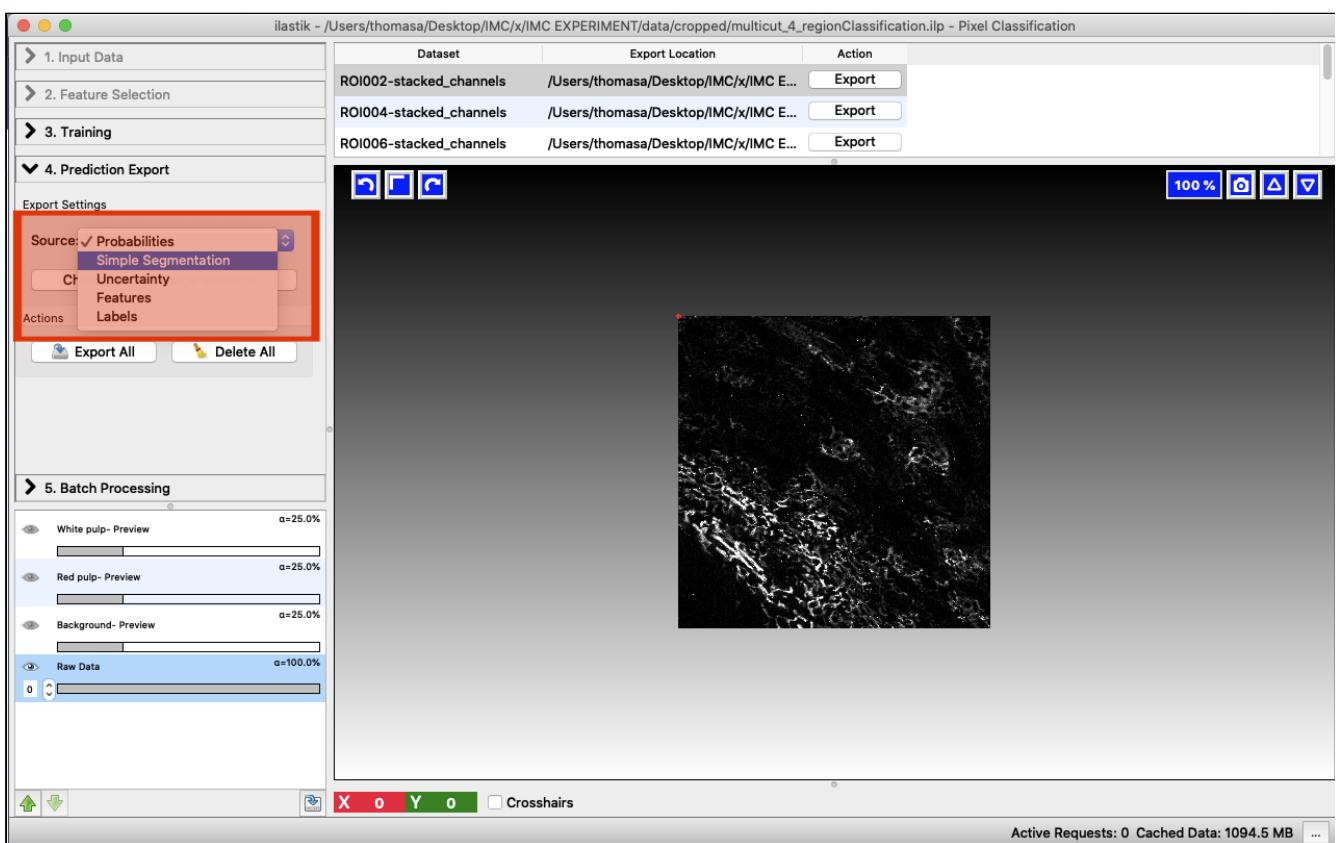
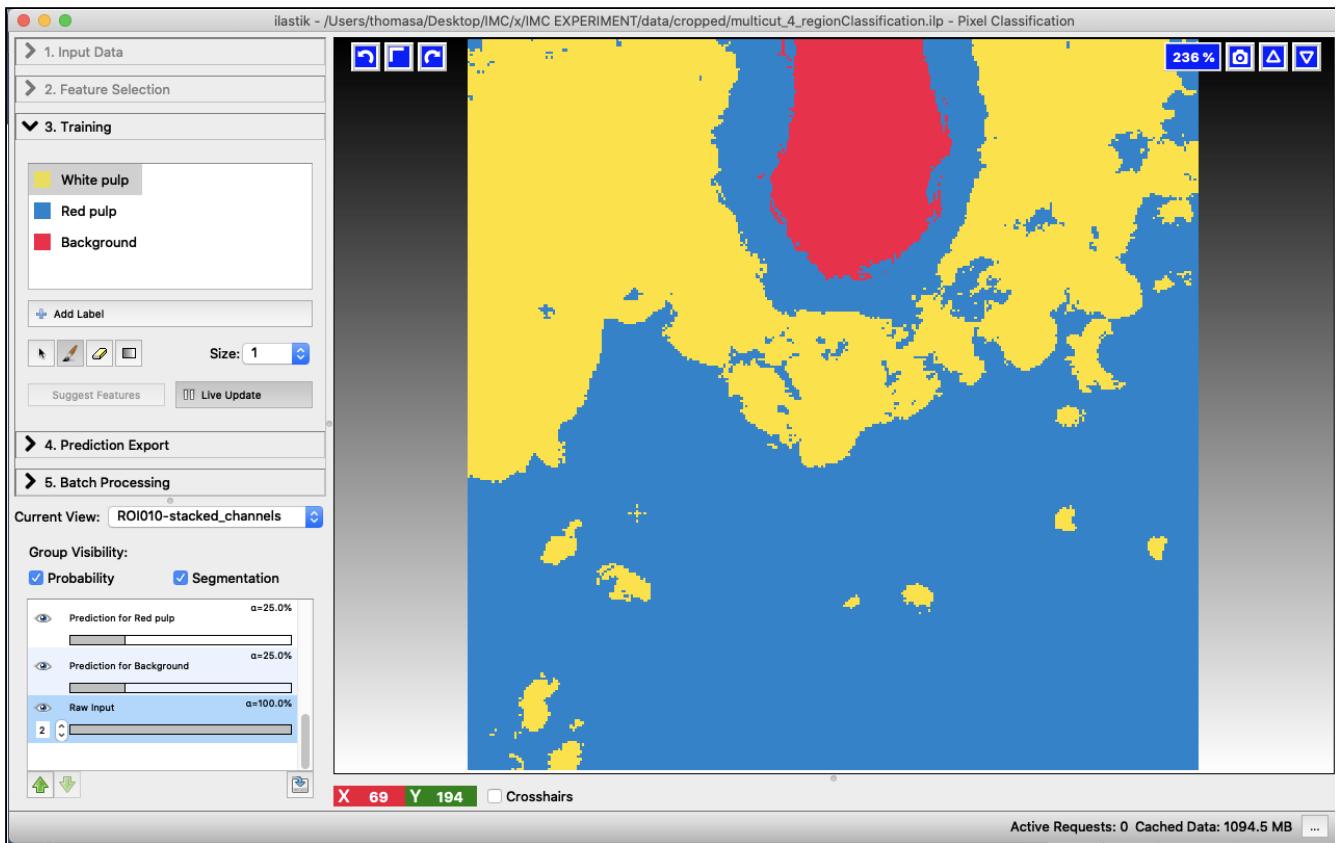
## 5. Ilastik - (optional, but recommended) region classification

And select the 'multicut\_4\_regionClassification.ilp' file (or alternatively, create and save a new Ilastik file of the 'pixel classification' type).

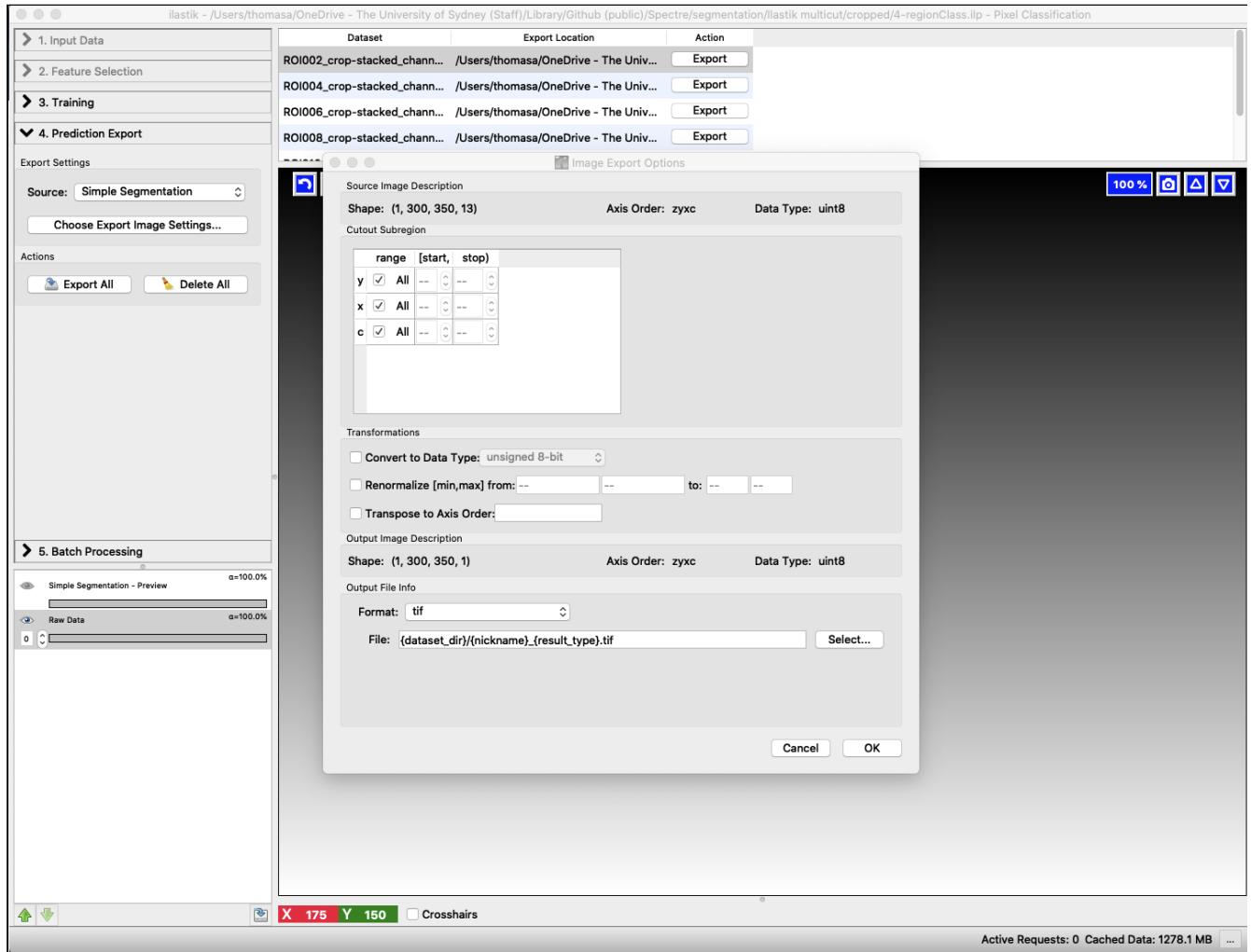
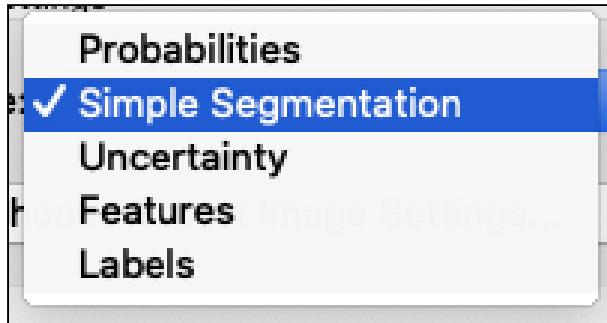


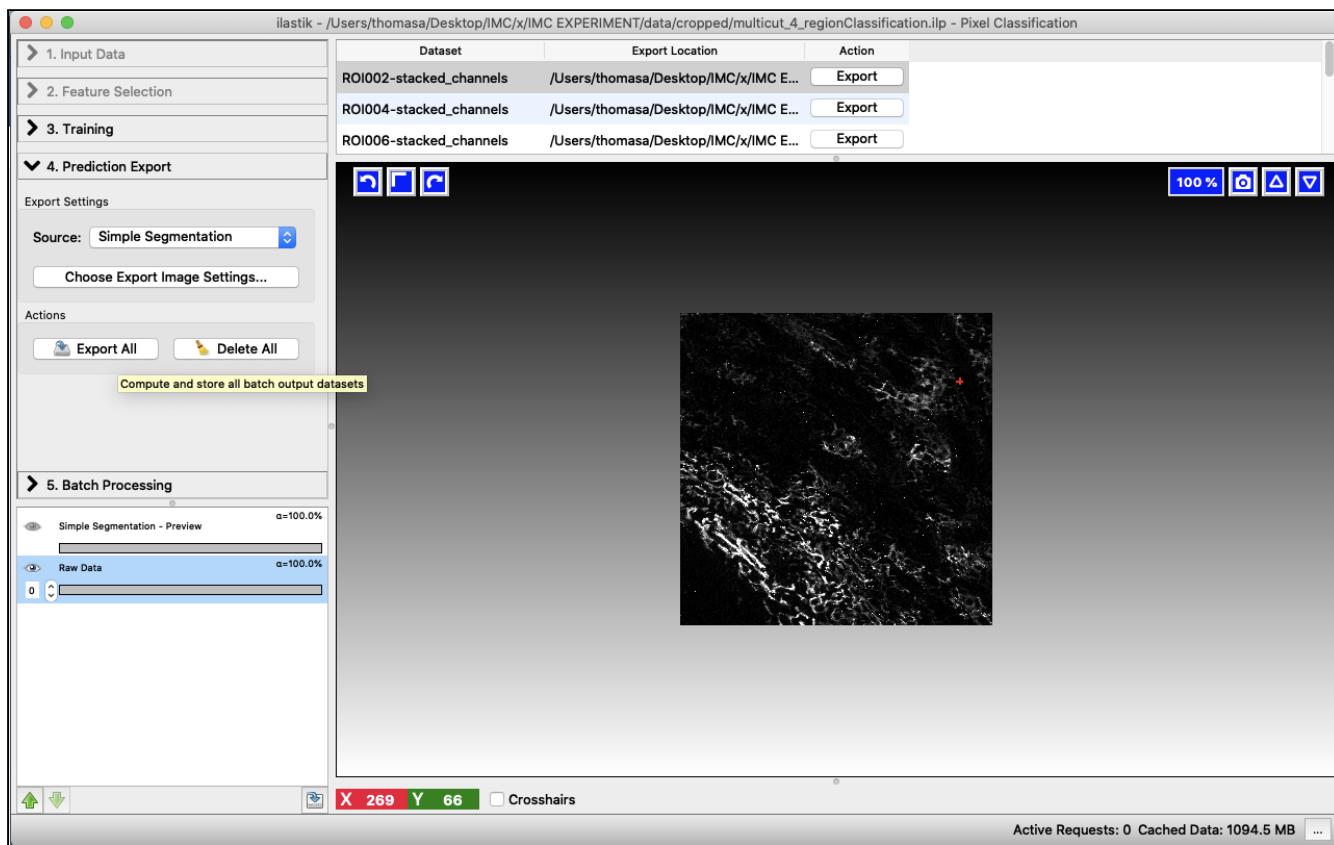




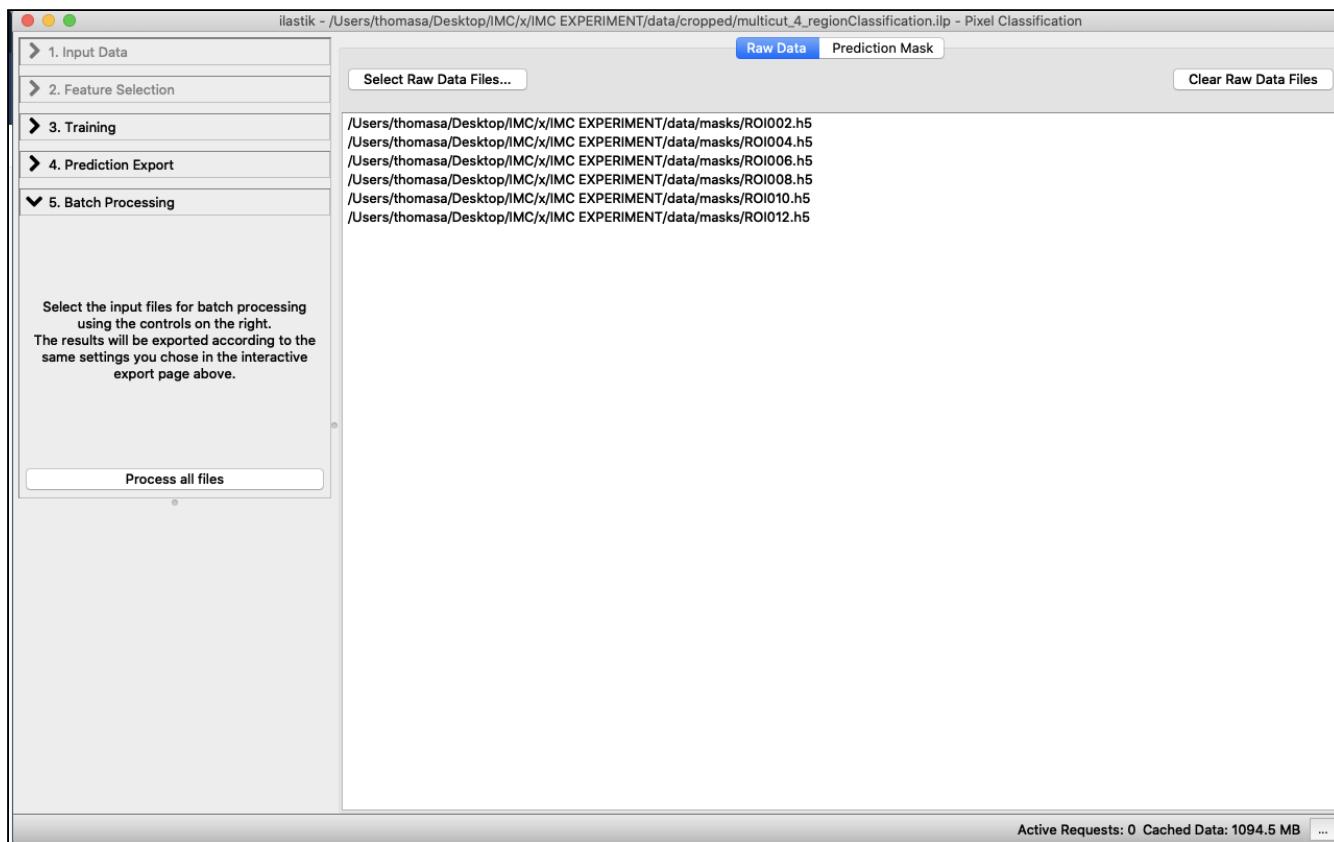


Select simple segmentation. This is different to the 'cell segmentation' we did before, as masks that belong to a specific type are not distinguished from each other.





Images	
<input type="checkbox"/>	ROI002_crop-stacked_channels_Object Identities.tif
<input type="checkbox"/>	ROI002_crop-stacked_channels_Object Predictions.tif
<input checked="" type="checkbox"/>	ROI002_crop-stacked_channels_Simple Segmentation.tif
<input type="checkbox"/>	ROI004_crop-stacked_channels_Object Identities.tif
<input type="checkbox"/>	ROI004_crop-stacked_channels_Object Predictions.tif
<input checked="" type="checkbox"/>	ROI004_crop-stacked_channels_Simple Segmentation.tif
<input type="checkbox"/>	ROI006_crop-stacked_channels_Object Identities.tif
<input type="checkbox"/>	ROI006_crop-stacked_channels_Object Predictions.tif
<input checked="" type="checkbox"/>	ROI006_crop-stacked_channels_Simple Segmentation.tif
<input type="checkbox"/>	ROI008_crop-stacked_channels_Object Identities.tif
<input type="checkbox"/>	ROI008_crop-stacked_channels_Object Predictions.tif
<input checked="" type="checkbox"/>	ROI008_crop-stacked_channels_Simple Segmentation.tif
<input type="checkbox"/>	ROI010_crop-stacked_channels_Object Identities.tif
<input type="checkbox"/>	ROI010_crop-stacked_channels_Object Predictions.tif
<input checked="" type="checkbox"/>	ROI010_crop-stacked_channels_Simple Segmentation.tif
<input type="checkbox"/>	ROI012_crop-stacked_channels_Object Identities.tif
<input type="checkbox"/>	ROI012_crop-stacked_channels_Object Predictions.tif
<input checked="" type="checkbox"/>	ROI012_crop-stacked_channels_Simple Segmentation.tif
Other	
<input type="checkbox"/>	multicut_1_pixel.ilp
<input type="checkbox"/>	multicut_2_segment.ilp
<input type="checkbox"/>	multicut_3_cellClassification.ilp
<input type="checkbox"/>	multicut_4_regionClassification.ilp
<input type="checkbox"/>	ROI002_crop-stacked_channels_Multicut Segmentation.h5
<input type="checkbox"/>	ROI002_crop-stacked_channels_Probabilities.h5
<input type="checkbox"/>	ROI002_crop.h5
<input type="checkbox"/>	ROI004_crop-stacked_channels_Multicut Segmentation.h5
<input type="checkbox"/>	ROI004_crop-stacked_channels_Probabilities.h5
<input type="checkbox"/>	ROI004_crop.h5
<input type="checkbox"/>	ROI006_crop-stacked_channels_Multicut Segmentation.h5
<input type="checkbox"/>	ROI006_crop-stacked_channels_Probabilities.h5
<input type="checkbox"/>	ROI006_crop.h5
<input type="checkbox"/>	ROI008_crop-stacked_channels_Multicut Segmentation.h5
<input type="checkbox"/>	ROI008_crop-stacked_channels_Probabilities.h5
<input type="checkbox"/>	ROI008_crop.h5
<input type="checkbox"/>	ROI010_crop-stacked_channels_Multicut Segmentation.h5
<input type="checkbox"/>	ROI010_crop-stacked_channels_Probabilities.h5
<input type="checkbox"/>	ROI010_crop.h5
<input type="checkbox"/>	ROI012_crop-stacked_channels_Multicut Segmentation.h5
<input type="checkbox"/>	ROI012_crop-stacked_channels_Probabilities.h5
<input type="checkbox"/>	ROI012_crop.h5



Images	
<input type="checkbox"/>	ROI002_Object Identities.tif
<input type="checkbox"/>	ROI002_Object Predictions.tif
<input checked="" type="checkbox"/>	<b>ROI002_Simple Segmentation.tif</b>
<input type="checkbox"/>	ROI004_Object Identities.tif
<input type="checkbox"/>	ROI004_Object Predictions.tif
<input checked="" type="checkbox"/>	<b>ROI004_Simple Segmentation.tif</b>
<input type="checkbox"/>	ROI006_Object Identities.tif
<input type="checkbox"/>	ROI006_Object Predictions.tif
<input checked="" type="checkbox"/>	<b>ROI006_Simple Segmentation.tif</b>
<input type="checkbox"/>	ROI008_Object Identities.tif
<input type="checkbox"/>	ROI008_Object Predictions.tif
<input checked="" type="checkbox"/>	<b>ROI008_Simple Segmentation.tif</b>
<input type="checkbox"/>	ROI010_Object Identities.tif
<input type="checkbox"/>	ROI010_Object Predictions.tif
<input checked="" type="checkbox"/>	<b>ROI010_Simple Segmentation.tif</b>
<input type="checkbox"/>	ROI012_Object Identities.tif
<input type="checkbox"/>	ROI012_Object Predictions.tif
<input checked="" type="checkbox"/>	<b>ROI012_Simple Segmentation.tif</b>
Other	
<input type="checkbox"/>	ROI002_Multicut Segmentation.h5
<input type="checkbox"/>	ROI002_Probabilities.h5
<input type="checkbox"/>	ROI002.h5
<input type="checkbox"/>	ROI004_Multicut Segmentation.h5
<input type="checkbox"/>	ROI004_Probabilities.h5
<input type="checkbox"/>	ROI004.h5
<input type="checkbox"/>	ROI006_Multicut Segmentation.h5
<input type="checkbox"/>	ROI006_Probabilities.h5
<input type="checkbox"/>	ROI006.h5
<input type="checkbox"/>	ROI008_Multicut Segmentation.h5
<input type="checkbox"/>	ROI008_Probabilities.h5
<input type="checkbox"/>	ROI008.h5
<input type="checkbox"/>	ROI10_Multicut Segmentation.h5
<input type="checkbox"/>	ROI10_Probabilities.h5
<input type="checkbox"/>	ROI10.h5
<input type="checkbox"/>	ROI12_Multicut Segmentation.h5
<input type="checkbox"/>	ROI12_Probabilities.h5
<input type="checkbox"/>	ROI12.h5