

# ROIZER

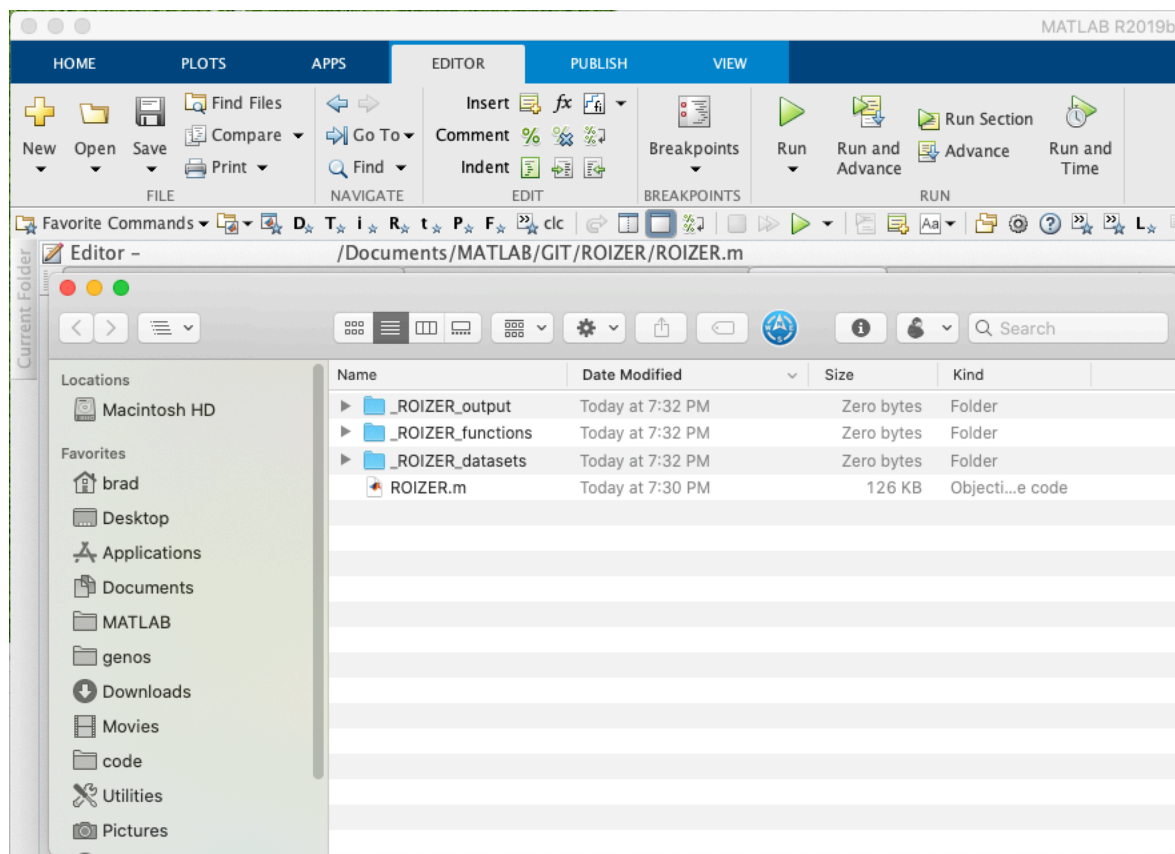
neural segmentation & activity measurement  
app for MATLAB

## Setup

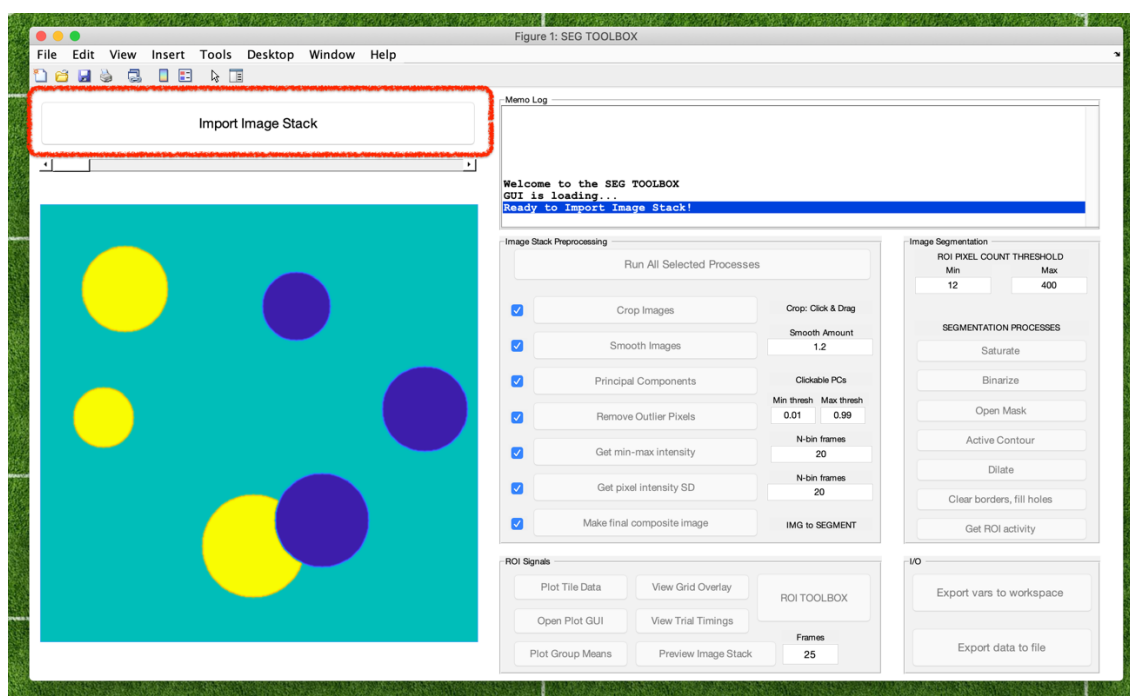
You can download the latest version of ROIZER on GitHub:

- [github.com/subroutines/ROIZER](https://github.com/subroutines/ROIZER)

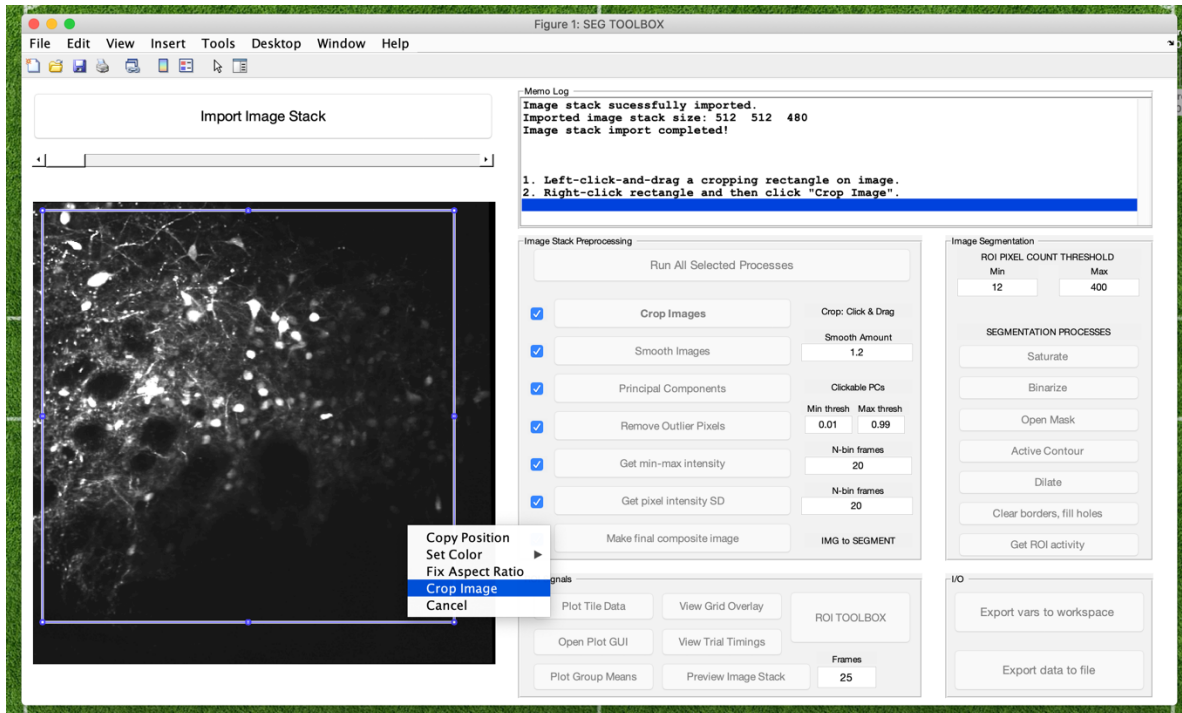
Unzip the downloaded file and put the folder contents in your MATLAB path. Then in MATLAB navigate to the ROIZER folder, and you should see the ROIZER.m app and three subdirectories. The app requires you to retain this directory structure...



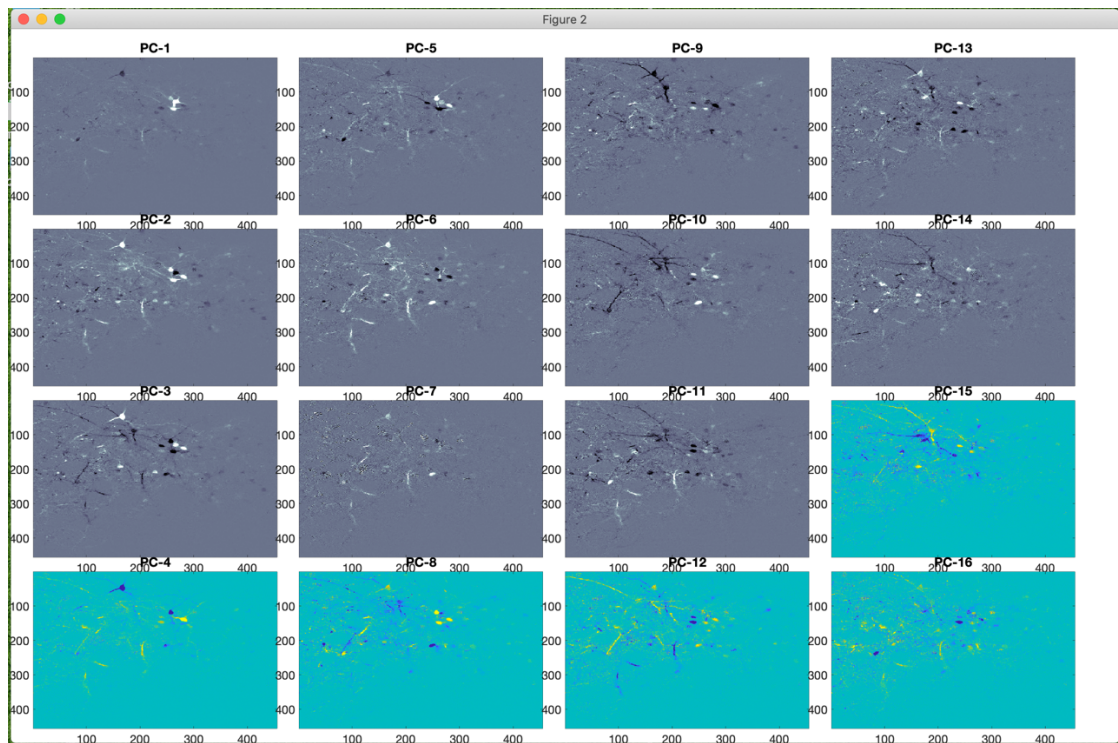
Run the `ROIZER.m` file to open a GUI interface; click the “Import Image Stack” button and select a tiff stack (you can keep tiff images in the `_ROIZER_datasets` folder for convenience)...



After the image stack has finished importing you can click on “Run All Selected Processes” to start performing the auto-image segmentation. Immediately after pressing this button you will be prompted to crop the image. To do this, mouse over to the top-left corner of the image, then *right-click-and-drag* towards the bottom-right corner of the image, to your satisfaction. Then left-click somewhere in the box you drew and select “Crop Image” (as shown below)...



The next user-interaction will ask you to click on PCA-generated images, that appear to work well for segmentation. Choose a minimum of 3, and up to all 16 component images by clicking on them (they will turn grey upon clicking) then close this window.



After those set of processes finish running, the final 7-steps are interactively performed by clicking on each of the 7 buttons in the “Image Segmentation” panel from top to bottom. You may click each button more than once, but avoid going back a button or skipping a button. Most of the time it’s best to just click each button once, in top-to-bottom order...

Image Segmentation

ROI PIXEL COUNT THRESHOLD

Min

12

Max

400

SEGMENTATION PROCESSES

Saturate

Binarize

Open Mask

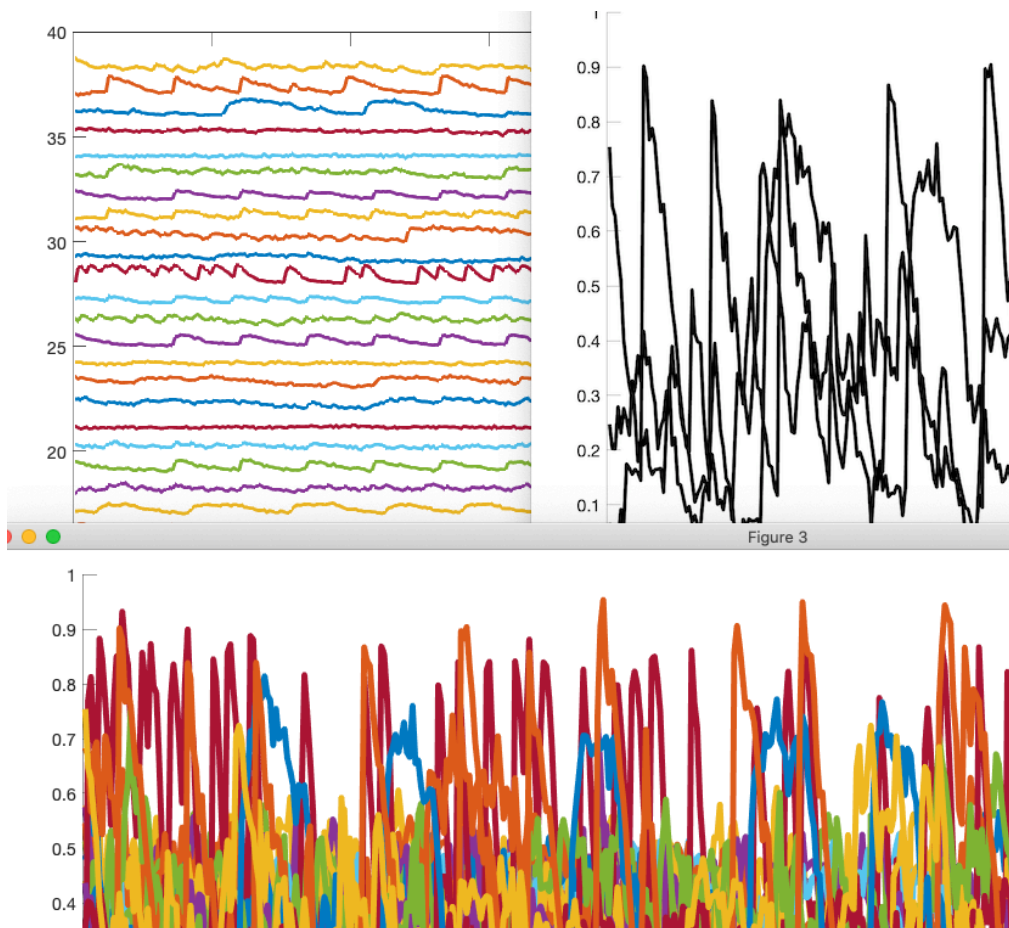
Active Contour

Dilate

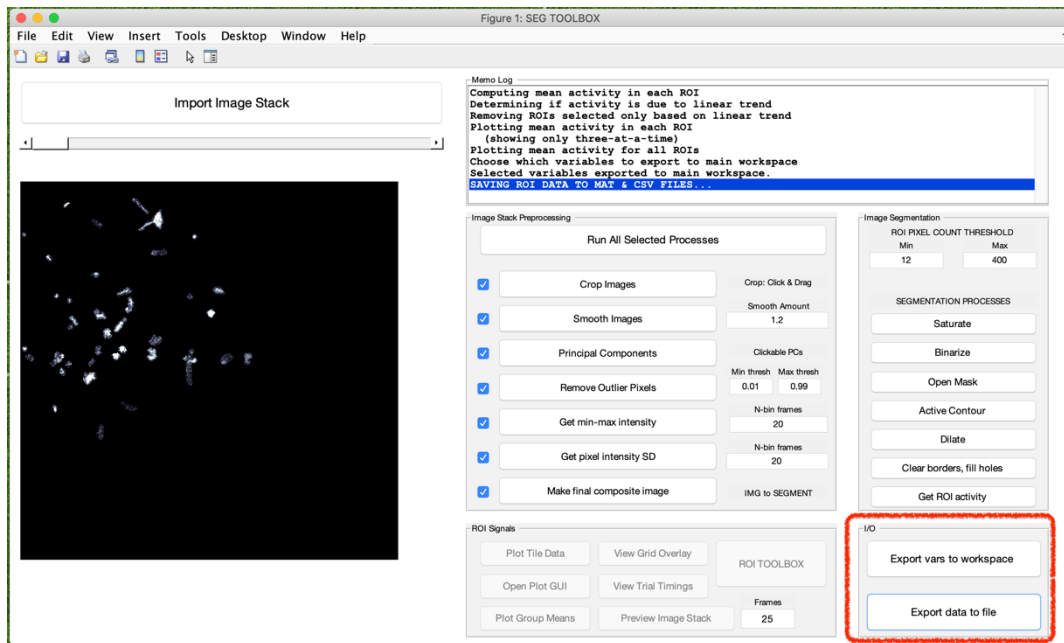
Clear borders, fill holes

Get ROI activity

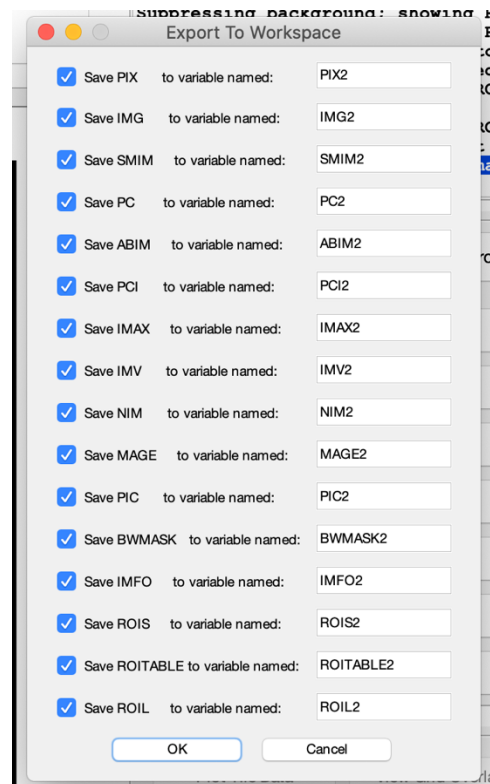
Upon clicking on “Get ROI activity” (the last button above), a series of steps are performed to quantify the activity of the ROIs that were just segmented. This takes about a minute and will end by showing the average ROI pixel value over time...



Once you see these images have been output, you can export the final dataset to the MATLAB workspace and/or to a .csv file. The popup on the right shows you the final set of variables that are export to the MATLAB workspace at the end of running the ROIZER.m script...



In the order they are generated in the script...



















PIX

metadata about the imported tiff file

IMG

the tiff stack (an 3-dim rectangular double-precision matrix)

Name ▲	Size	Bytes	Class
 ABIM	512x512x25	26214400	single
 BWMASK	512x512	262144	logical
 IMAX	512x512	2097152	double
 IMFO	1x1	2126409	struct
 IMG	512x512x480	1006632960	double
 IMV	512x512	2097152	double
 MAGE	512x512x6	12582912	double
 NIM	1x1	269747232	struct
 PC	1x2	1033692208	struct
 PCI	512x512x7	7340032	single
 PIC	512x512	2097152	double
 PIX	1x1	1268002	struct
 ROIL	480x37	142080	double
 ROIS	480x37	142080	double
 ROITABLE	480x1	143120	table
 SMIM	512x512x480	1006632960	double

PC

principal component scores and coefficients

ABIM

absolute value after mean deviation of the first 25 principal component images

PCI

hand-chosen set of principal component images



## IMAX

max\_pixel – min\_pixel value, every 20 frames, then averaged over all differences.

## IMV

standard deviation, every 20 frames, then average of the sd for the stack use imstats() to get statistics for any of the above variables...

```
#####  
%%      CREATE COMPOSITE IMAGE USING COMBINATION OF ABOVE  
#####  
  
disp('IMG'); imstats(IMG); % RAW IMAGE STACK  
disp('SMIM'); imstats(SMIM); % GAUSSIAN SMOOTHED VERSION OF IMG  
disp('IMAX'); imstats(IMAX); % MEAN MAX PIXEL INTENSITY OF IMG  
disp('ABIM'); imstats(ABIM); % ABSOLUTE MEAN DEVIATION OF ALL PCs  
disp('PCI'); imstats(PCI); % CHOSEN PRINCIPAL COMPONENTS  
disp('IMV'); imstats(IMV); % STDEV OF EACH IMG PIXEL ALONG 3RD DIM
```

## MAGE

A stack of each image transformation above to be used to perform image segmentations

## PIC

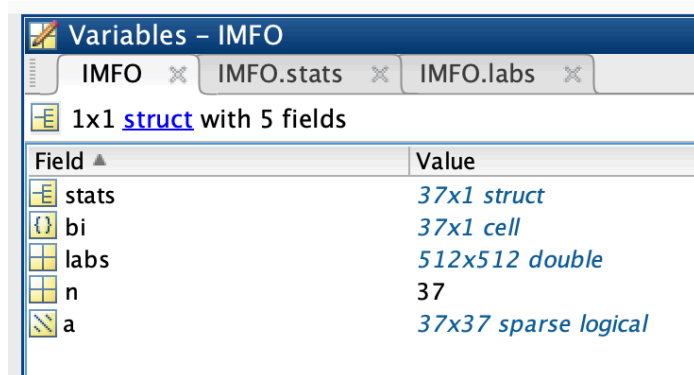
an average of the images in MAGE

## BWMASK

image mask the same size as the first image of the tiff stack, where each ROI have pixels represented by 1s while the rest of the pixels are assigned 0s.

## IMFO

contains a number of useful ROI stats and masks...



Variables – IMFO	
IMFO x IMFO.stats x IMFO.labs x	
1x1 struct with 5 fields	
Field ▲	Value
stats	37x1 struct
bi	37x1 cell
labs	512x512 double
n	37
a	37x37 sparse logical

Variables – IMFO.stats			
IMFO.stats			
Fields	Area	Centroid	BoundingBox
1	13	[27,83]	[24.5000,80.5000,5,5]
2	33	[34.9394,...	[30.5000,30.5000,9,7]
3	13	[34,23]	[31.5000,20.5000,5,5]
4	18	[35,102.5...	[32.5000,99.5000,5,6]
5	13	[41,81]	[38.5000,78.5000,5,5]
6	13	[48,104]	[45.5000,101.5000,5,5]
7	24	[69,92.50...	[65.5000,89.5000,7,6]
8	13	[69,83]	[66.5000,80.5000,5,5]
9	13	[70,16]	[67.5000,13.5000,5,5]
10	24	[77.5000,...	[74.5000,252.5000,6,6]
11	60	[83.4333,...	[79.5000,29.5000,8,14]
12	13	[83,22]	[80.5000,19.5000,5,5]

Variables – IMFO.labs															
IMFO.labs															
	144	145	146	147	148	149	150	151	152	153	154	155	156	157	
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
44	0	0	0	0	0	0	0	20	20	20	0	0	0	0	
45	0	0	0	0	0	20	20	20	20	20	0	0	0	0	
46	0	0	0	0	20	20	20	20	20	20	20	0	0	0	
47	0	0	0	20	20	20	20	20	20	20	20	0	0	0	
48	0	0	0	20	20	20	20	20	20	20	20	20	0	0	
49	0	0	20	20	20	20	20	20	20	20	20	20	20	20	
50	20	0	0	20	20	20	20	20	20	20	20	20	20	20	
51	20	20	0	0	20	20	20	20	20	20	20	20	20	20	
52	20	20	20	0	0	20	20	20	20	20	20	20	20	20	
53	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
54	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
55	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
56	0	20	20	20	20	20	20	20	20	20	20	20	0	0	
57	0	0	20	20	20	20	20	20	20	20	20	20	0	0	
58	0	0	0	20	20	20	20	20	20	0	0	0	0	0	
59	0	0	20	20	20	20	20	0	0	0	0	0	0	0	
60	0	0	0	20	20	20	0	0	0	0	0	0	0	0	
61	0	0	0	0	20	0	0	0	0	0	0	0	0	0	
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

ROIS ROIL ROITABLE  
 The mean pixel value for each segmented ROI for each image in the stack