Table of Contents


```
function [] = VolumeThresholdhelper(fileName)
```

VolumeThresholdhelper Information

VolumeThresholdhelper is used to help the user find parameters to use for batch processing with LC3B_Tandem_Puncta_QuantificationV2. The function takes in the fileName of a file in the current directory. It then asks the user to input parameters.

Function asks the user to enter the following information which is used for thresholding and identifying the cell boundaries. Three GUIs will be opened for the user to do this.

Three GUIs propompted to the user are:

1.Asks the user to input integers in the range [1,3] which is used to specify the order for the cell markers. This is, the channels for the nuclear marker, eGFP vacuoles and mCherry vacuoles.

2.Asks the user to input integers which are used to threshold the mature eGFP and mCherry vacuoles. Code assumes images are 16-bit so inputs should be in the range of [0, 65535] which specifies the intensity value for unit16 arrays.

3.Asks the user to input approximate volume sizes, in voxels, for the nuclear marker, eGFP vacuoles, mCherry vacuoles and the solidity for the vacuoles which is type float between [0,1].

Once the parameters are input by the user, the function will use the inputs to label each slice in the image data with yellow circles wherever it identifies a vacuole and presents the volume data using sliceViewer with the yellow circles overlaid for the users to check if the parameters worked or should be adjusted. A dialogue box is presented for the user to enter Y/N if they would like to try other parameters to improve the segmentation. Once satisfied the user can enter N and proceed to see what the Ridgelines and single cell containers look like to verify the SNR is high enough to perform segmentation with Otsu's method. Finally, subplots for the single cell data with the nuclear marker, eGFP and mCherry vacuoles are presented for viewing as isosurface plots. Note: user should place waitfor() after figures to stop the function at each plot for viewing in the functions WatershedOberserver and IsosurfaceVacuoles.

Once the user has identified appropriate parameters for their data they can use the parameters in LC3B_Tandem_Puncta_QuantificationV2 to perform batch processing for their data.

Created by: Joaquin Quintana (last modified: 05-18-2021) Email: Joaquin.Quintana@Colorado.edu

Run while loop until user finds parameters for their data to use

```
while 1

%clear work space and cmd on each run
close all;
clc;

%store inputs in struct to return at the end so they know what they
 decided
%worked for their data
Input.Nuclear_channel = [];
Input.eGFP_channel = [];
Input.mCherry_channel = [];
Input.threshold_eGFP = [];
Input.threshold_mCherry = [];
Input.minGFPvacVolume = [];
Input.minRFPvacVolume = [];
Input.minNucleusVolume = [];
Input.Solidity = [];

%get some info from the user

%ask user what order the channels are in
waitfor(msgbox('Please let us know what order the eGFP, mCherry and
 nuclear marker are in'));
input = inputdlg({'Nuclear marker','eGFP','mCherry'},...
'Channels',1,{'1','2','3'},'on');

%convert char to double
Input.Nuclear_channel = str2double(input(1));
Input.eGFP_channel = str2double(input(2));
Input.mCherry_channel = str2double(input(3));

%ask user to input thresholds for eGFP and mCherry channels
waitfor(msgbox('Please enter thresholds for developed puncta. This
 should be for developed eGFP and mCherry vacuoles which are used to
 identify neutral and acidified vacuoles, respectively.'));
input = inputdlg({'GFP Threshold Value 16-bit [0-65536]','RFP
 Threshold Value 16-bit [0-65536]'},...
'Thresholds',1,{'25000','20000'},'on');

%convert char to double
Input.threshold_eGFP =  str2double(input(1));
Input.threshold_mCherry =  str2double(input(2));

%get min sizes in voxels for cell markers and solidity
input = inputdlg({'Minimum volume of GFP vacuole in voxels is >',...
    'Minimum volume of RFP vacuole in voxels is >',...
    'Minimum volume of Nucleus vacuole in voxels is >',...
    'Solidity Thershold Vacuoles [0,1]'},...
```

```
'Thresholds',1,{'1','15','15000','0.6'},'on');

%convert char to double
Input.minGFPvacVolume =  str2double(input(1));
Input.minRFPvacVolume =  str2double(input(2));
Input.minNucleusVolume =  str2double(input(3));
Input.Solidity =  str2double(input(4));

%unpack user inputs for passing to functions
Nuclear_channel = Input.Nuclear_channel;
eGFP_channel = Input.eGFP_channel;
mCherry_channel = Input.mCherry_channel;
eGFP_threshold = Input.threshold_eGFP;
mCherry_threshold = Input.threshold_mCherry;
MinvacVoxeGFP = Input.minGFPvacVolume;
MinvacVoxmCherry = Input.minRFPvacVolume;
MinNucVox = Input.minNucleusVolume;
Solidity = Input.Solidity;
```

Use bfopen to read file

```
img = bfopen(fileName);
name = fileName;
```

```
 ..................................................................
    ..................................
```

Create subvolumes for each channel

```
[r,g,b] =
 imreadVolume(img,Nuclear_channel,eGFP_channel,mCherry_channel);
```

Label objects (vacuoles) with yellow circles and present to user with a slice viewer to check if parameters are ideal for segmentation

```
%set thresholds from user inputs for developed vacuoles.

%THIS IS ALL REDUNDANT TO THE VACUOLE FINDER AND IS JUST BEING USED TO
 UPDATE AND
%PERFECT THE VACUOLE SEGMENTATION

%convert input thresholds to double precision
max  = 65536;min  = 0;
Green_Thresh  = (eGFP_threshold - min)/(max - min);
Red_Thresh  = (mCherry_threshold - min)/(max - min);
```

```
%segment vacuoles
[pre_green_Vacs,green_Vacs_CC] = VacuoleCleansing(g,Green_Thresh,...
    MinvacVoxeGFP,Solidity);
[red_Vacs,red_Vacs_CC] =
 VacuoleCleansing(r,Red_Thresh,MinvacVoxmCherry...
    ,Solidity);

%get intial stats for vacuoles
red_vacuoles_stats = regionprops3(red_Vacs_CC,red_Vacs,'all');
pre_green_stats = regionprops3(green_Vacs_CC,pre_green_Vacs,'all');

%only keeps green vacoles if a red vacuole is associated with it. Due
 to the
%nature of the probe this should be the case. It helps as a precheck
 for
%ensuring the vacuole is indeed a LC3B vacuole.
[green_vacuoles_stats] = OverlappingChannels(pre_green_stats,...
    red_vacuoles_stats);

%now create the final binary for green vacuoles which overlap mcherry.
[~,idx] = intersect(pre_green_stats.EquivDiameter,...
    green_vacuoles_stats.EquivDiameter);

%return the updated indexes for eGFP binary image
bi_img = ismember(labelmatrix(green_Vacs_CC),idx);

%make sure to get the new connected components for updated eGFP stats
green_Vacs_CC = bwconncomp(bi_img);

%label the found vacuoles
label_r = labelmatrix(red_Vacs_CC);
label_g = labelmatrix(green_Vacs_CC);

%display current stats for vacuoles. Will probably reduce the outputs
 in
%the future but as of now everyhting from regionprops3 is returned
disp('Stats for eGFP vacuoles using current parameters')
disp(green_vacuoles_stats)
disp('Stats for mCherry vacuoles using current parameters')
disp(red_vacuoles_stats)
```

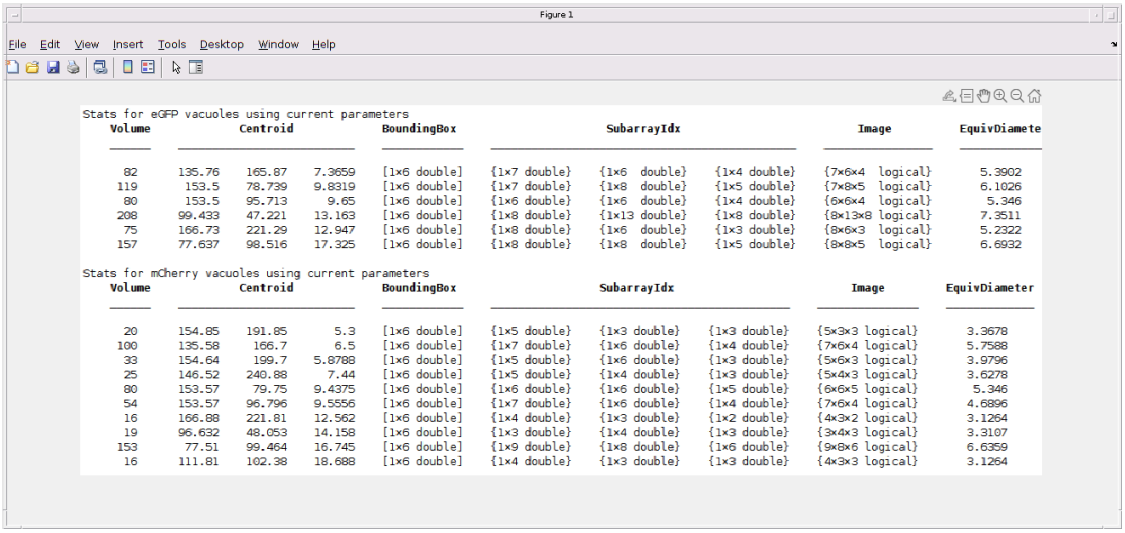
```
Stats for eGFP vacuoles using current parameters
    Volume             Centroid             BoundingBox
               SubarrayIdx                          Image
  EquivDiameter    Extent      VoxelIdxList       VoxelList
     PrincipalAxisLength                Orientation
 EigenVectors    EigenValues      ConvexHull        ConvexImage
     ConvexVolume    Solidity    SurfaceArea     VoxelValues
       WeightedCentroid         MeanIntensity    MinIntensity
 MaxIntensity
    ______    ____________________________    ____________
 ___________________________________________    ________________
    ____________    _______    _____________    ________________
   ________________________    ___________________________________
```

```
_____ _____ _____ _____
_____ _____ _____ _____
_____ _____ _____
_____

    82      135.76    165.87    7.3659    [1×6 double]    {1×7
double}    {1×6  double}    {1×4 double}    {7×6×4  logical}
5.3902         0.4881    { 82×1 double}    { 82×3 double}    6.2863
   5.4065    3.6914     8.6623    -3.8624    -11.961    {3×3 double}
   {3×1 double}    {44×3 double}    {7×6×4  logical}         97
    0.84536       95.657       { 82×1 double}    135.78    165.84
7.3363       0.41188         0.20423         0.76669
    119       153.5    78.739    9.8319    [1×6 double]    {1×7
double}    {1×8  double}    {1×5 double}    {7×8×5  logical}
6.1026          0.425    {119×1 double}    {119×3 double}     6.908
    6.303    4.4407     80.906     15.928    -2.2192    {3×3 double}
   {3×1 double}    {46×3 double}    {7×8×5  logical}        151
    0.78808       130.13       {119×1 double}    153.57    78.836
9.8243       0.39048         0.18772         0.75219
     80       153.5    95.713      9.65    [1×6 double]    {1×6
double}    {1×6  double}    {1×4 double}    {6×6×4  logical}
5.346         0.55556    { 80×1 double}    { 80×3 double}    5.6377
  5.5397    3.9046    -7.4139        -16     175.95    {3×3 double}
   {3×1 double}    {41×3 double}    {6×6×4  logical}         91
    0.87912       92.061       { 80×1 double}    153.52    95.764
9.7042       0.38502         0.18906         0.68329
    208      99.433    47.221    13.163    [1×6 double]    {1×8
double}    {1×13 double}    {1×8 double}    {8×13×8 logical}
7.3511           0.25    {208×1 double}    {208×3 double}    13.001
    6.614    5.1231     85.642     15.116    -31.639    {3×3 double}
   {3×1 double}    {48×3 double}    {8×13×8 logical}        331
     0.6284       218.72       {208×1 double}    99.135    47.194
13.299       0.35992         0.18873         0.67758
     75      166.73    221.29    12.947    [1×6 double]    {1×8
double}    {1×6  double}    {1×3 double}    {8×6×3  logical}
5.2322         0.52083    { 75×1 double}    { 75×3 double}    7.2159
   4.8882    3.2221    -17.635     -2.074    -7.6291    {3×3 double}
   {3×1 double}    {39×3 double}    {8×6×3  logical}         84
    0.89286       91.13       { 75×1 double}     166.7    221.32
12.974        0.3774         0.19127         0.65256
    157      77.637    98.516    17.325    [1×6 double]    {1×8
double}    {1×8  double}    {1×5 double}    {8×8×5  logical}
6.6932         0.49062    {157×1 double}    {157×3 double}    7.8519
   6.6215    4.4466    -61.264    -4.5475     -3.587    {3×3 double}
   {3×1 double}    {53×3 double}    {8×8×5  logical}        181
     0.8674       148.33       {157×1 double}    77.693    98.504
17.342       0.48249         0.18855         0.91596

Stats for mCherry vacuoles using current parameters
   Volume             Centroid             BoundingBox
            SubarrayIdx                             Image
EquivDiameter    Extent      VoxelIdxList       VoxelList
   PrincipalAxisLength                  Orientation
 EigenVectors    EigenValues      ConvexHull        ConvexImage
```

```
     ConvexVolume    Solidity    SurfaceArea     VoxelValues
       WeightedCentroid          MeanIntensity    MinIntensity
MaxIntensity
   ______    ____________________________    ____________
_____________________________________________    ______________
  _____________    _______    _______________    _____________
 ___________________________    ___________________________________
 _____________    _____________    _____________    ________________
   _____________    _________    ____________    _______________
 ___________________________    _____________    _____________
_____________

     20      154.85    191.85       5.3    [1×6 double]    {1×5
double}    {1×3 double}    {1×3 double}    {5×3×3 logical}
3.3678        0.44444    { 20×1 double}    { 20×3 double}     4.436
  3.1484    2.4186     6.6043     -6.0081     -15.642    {3×3 double}
   {3×1 double}    {21×3 double}    {5×3×3 logical}         21
   0.95238       35.488       { 20×1 double}    154.87    191.83
5.2978       0.19315         0.13134         0.28973
    100      135.58     166.7       6.5    [1×6 double]    {1×7
double}    {1×6 double}    {1×4 double}    {7×6×4 logical}
5.7588        0.59524    {100×1 double}    {100×3 double}     6.825
  5.6103    3.9092    -1.2438      15.362      178.19    {3×3 double}
   {3×1 double}    {40×3 double}    {7×6×4 logical}        108
   0.92593       108.12       {100×1 double}    135.65    166.72
6.5692       0.26936         0.1177         0.48489
     33      154.64     199.7    5.8788    [1×6 double]    {1×5
double}    {1×6 double}    {1×3 double}    {5×6×3 logical}
3.9796        0.36667    { 33×1 double}    { 33×3 double}    6.0842
  4.5438    2.7361    -83.854      9.9176      -11.34    {3×3 double}
   {3×1 double}    {28×3 double}    {5×6×3 logical}         49
   0.67347       61.071       { 33×1 double}    154.73    199.73
5.8806       0.15794         0.11916         0.21847
     25      146.52    240.88      7.44    [1×6 double]    {1×5
double}    {1×4 double}    {1×3 double}    {5×4×3 logical}
3.6278        0.41667    { 25×1 double}    { 25×3 double}    5.2829
  3.8198    2.3845    -2.3753     -9.2406      169.34    {3×3 double}
   {3×1 double}    {24×3 double}    {5×4×3 logical}         32
   0.78125       45.825       { 25×1 double}    146.54    240.87
7.4353       0.17526         0.12249         0.24559
     80      153.57     79.75    9.4375    [1×6 double]    {1×6
double}    {1×6 double}    {1×5 double}    {6×6×5 logical}
5.346        0.44444    { 80×1 double}    { 80×3 double}    5.9165
  5.295    3.9756     85.204      14.879      -7.598    {3×3 double}
   {3×1 double}    {37×3 double}    {6×6×5 logical}         96
   0.83333       93.427       { 80×1 double}    153.61    79.822
9.4625       0.24282         0.11853         0.42815
     54      153.57    96.796    9.5556    [1×6 double]    {1×7
double}    {1×6 double}    {1×4 double}    {7×6×4 logical}
4.6896        0.32143    { 54×1 double}    { 54×3 double}      6.02
   4.412    3.3901     -23.32     -14.219    -0.83694    {3×3 double}
   {3×1 double}    {36×3 double}    {7×6×4 logical}         66
   0.81818        73.87       { 54×1 double}    153.56    96.863
9.6123       0.24218         0.11795         0.40777
```

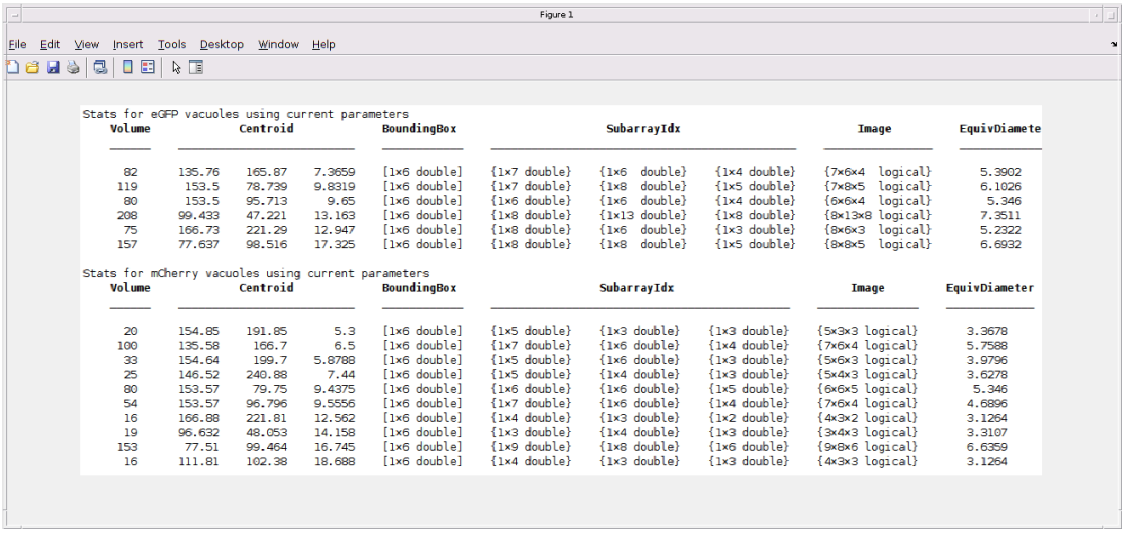
```
     16      166.88    221.81    12.562    [1×6 double]    {1×4
 double}    {1×3 double}    {1×2 double}    {4×3×2 logical}
 3.1264        0.66667    { 16×1 double}    { 16×3 double}    4.2419
   3.1487    2.0535    -8.3792      -1.668      25.333    {3×3 double}
    {3×1 double}    {23×3 double}    {4×3×2 logical}         17
    0.94118       30.521       { 16×1 double}    166.85    221.86
 12.558       0.18293         0.13857         0.24761
     19      96.632    48.053    14.158    [1×6 double]    {1×3
 double}    {1×4 double}    {1×3 double}    {3×4×3 logical}
 3.3107        0.52778    { 19×1 double}    { 19×3 double}    4.6708
   2.6999    2.6087    -59.719     -19.712     -1.4611    {3×3 double}
    {3×1 double}    {26×3 double}    {3×4×3 logical}         21
    0.90476       34.425       { 19×1 double}    96.603    48.081
 14.211       0.20339         0.12047          0.3221
    153       77.51    99.464    16.745    [1×6 double]    {1×9
 double}    {1×8 double}    {1×6 double}    {9×8×6 logical}
 6.6359        0.35417    {153×1 double}    {153×3 double}    7.8497
   7.155    4.2759    -56.692      -9.258      4.9168    {3×3 double}
    {3×1 double}    {44×3 double}    {9×8×6 logical}         188
    0.81383       153.03       {153×1 double}    77.569    99.484
 16.826       0.3136         0.11792         0.60988
     16      111.81    102.38    18.688    [1×6 double]    {1×4
 double}    {1×3 double}    {1×3 double}    {4×3×3 logical}
 3.1264        0.44444    { 16×1 double}    { 16×3 double}    4.282
   2.8013    2.3329    -31.766      10.015      39.377    {3×3 double}
    {3×1 double}    {20×3 double}    {4×3×3 logical}         17
    0.94118       30.352       { 16×1 double}    111.81    102.35
 18.679       0.19713         0.1436          0.29888
```

```
%The published out for disp looks horrible without using a Latex
 wrapper
%so I've included an image of how it looks in the command window.
%I need to add a Latex wrapper for this
imshow('StatsCmdOutput.png')
snapnow
```



Label vacuoles with circles and display using sliceViewer

```
CircleVacuoles(label_r,label_g,r,g)

%end while loop if user has found what they seek
options.WindowStyle = 'normal';
resp = inputdlg({'Do you wish to try other inputs? Y/N: '},...
'Retry?',1,{'N'},options);

    if strcmpi(resp,'n')
        % user has typed in N or n so break out of the while loop
        break;
    end
```



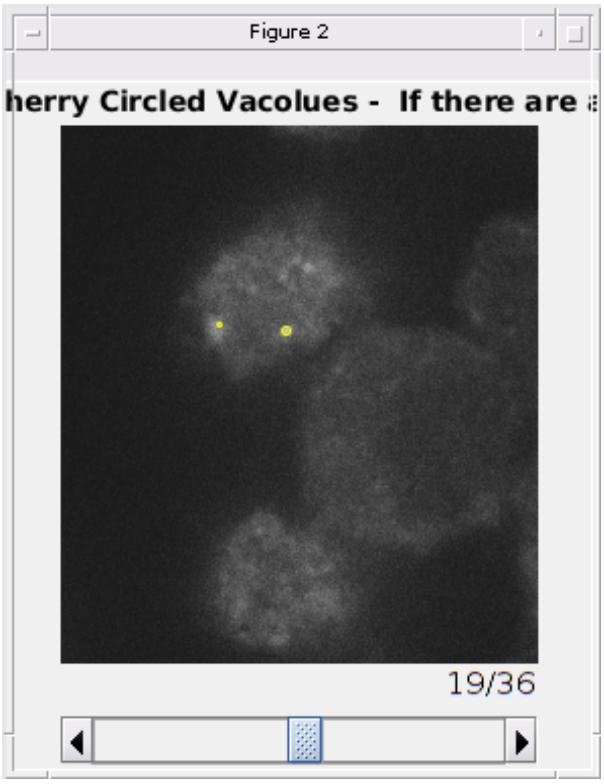


Figure 2
herry Circled Vacolues - If there are
19/36

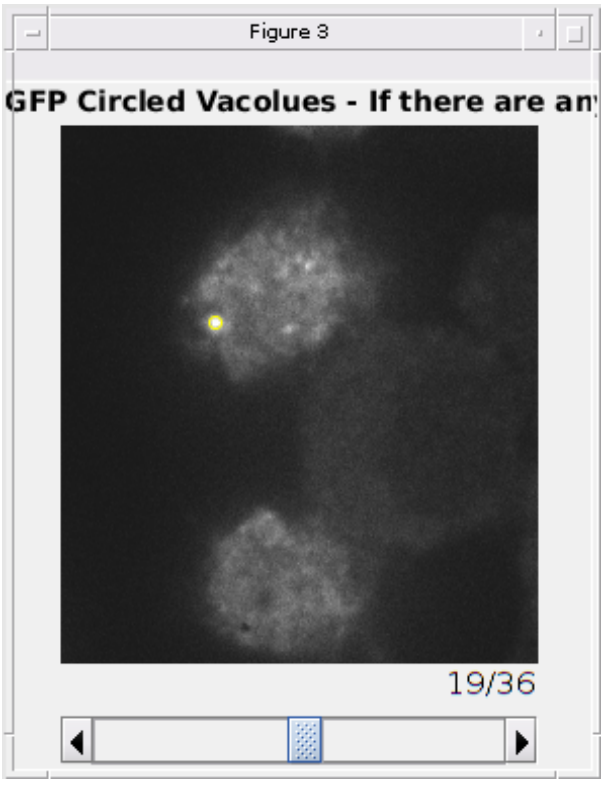
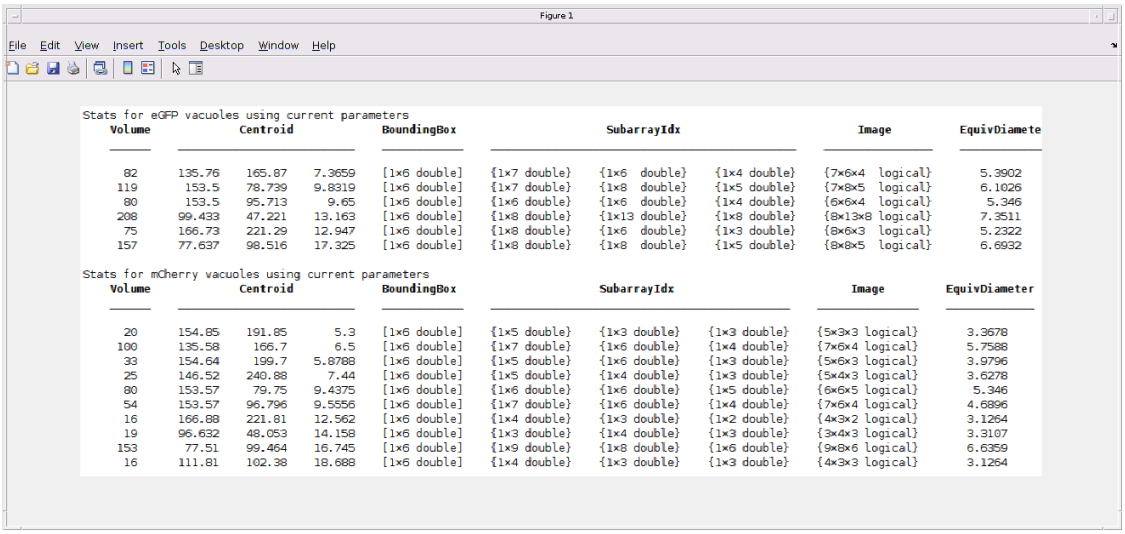


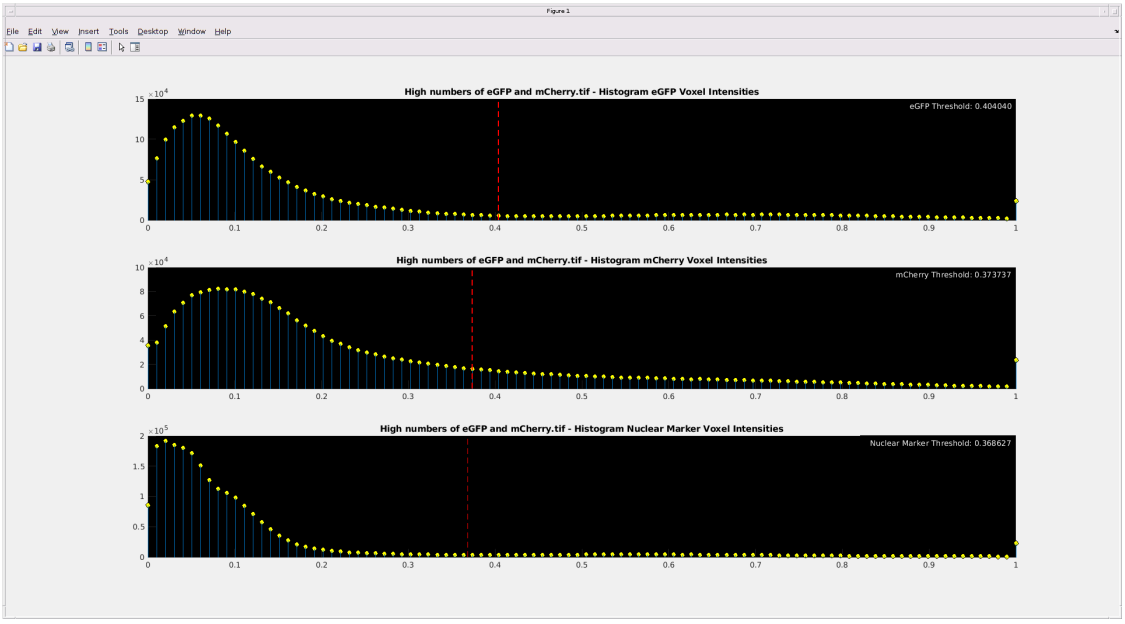
Figure 3
GFP Circled Vacolues - If there are an
19/36

end




```
%if user likes the parameters used, proceed and show them what is
 created
%using the rest of the script.

%segment cells and get ROIs
[bw_stats] = Watershed2(r,g,b,name,MinNucVox,true);
```

Data for plots generated from - High numbers of eGFP and mCherry.tif



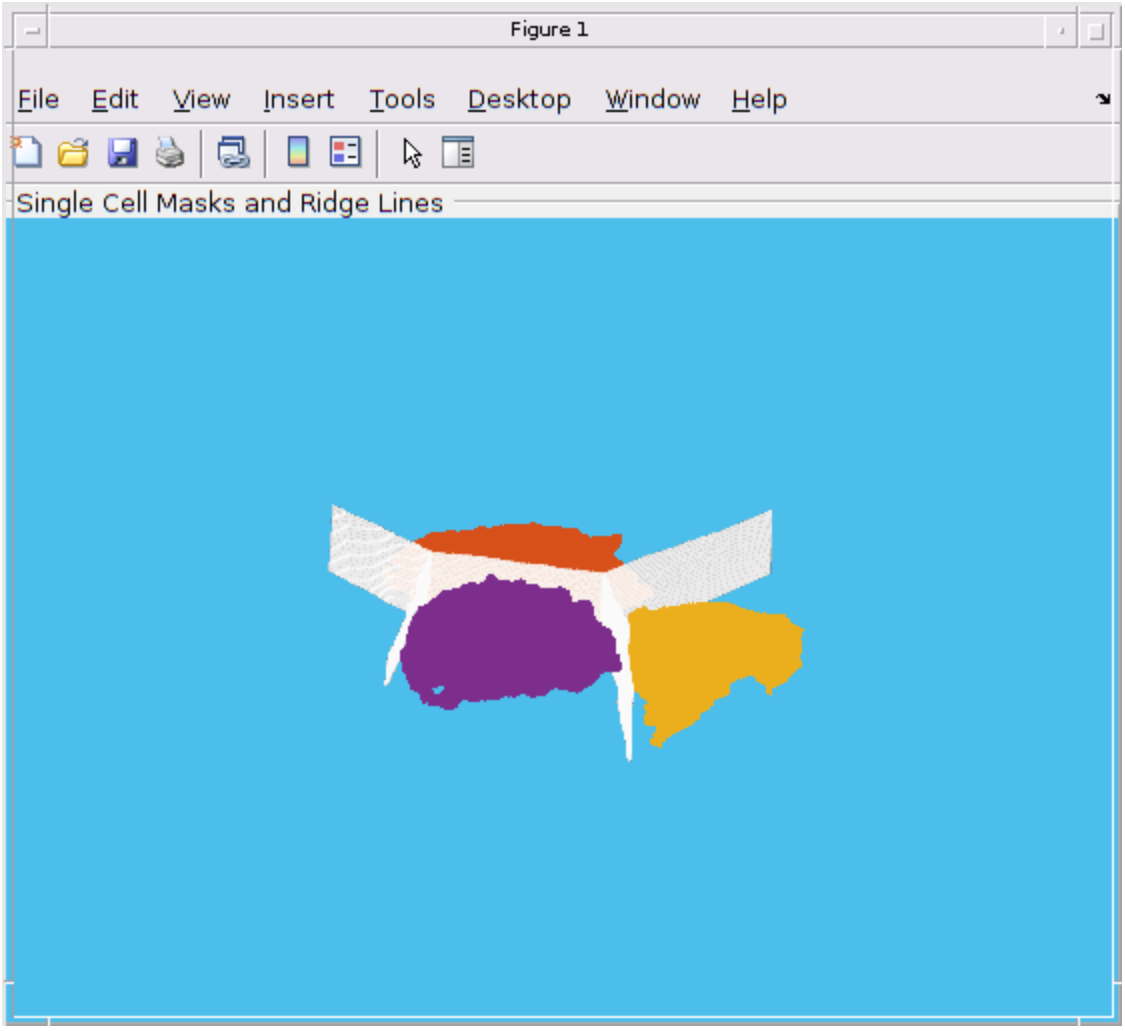
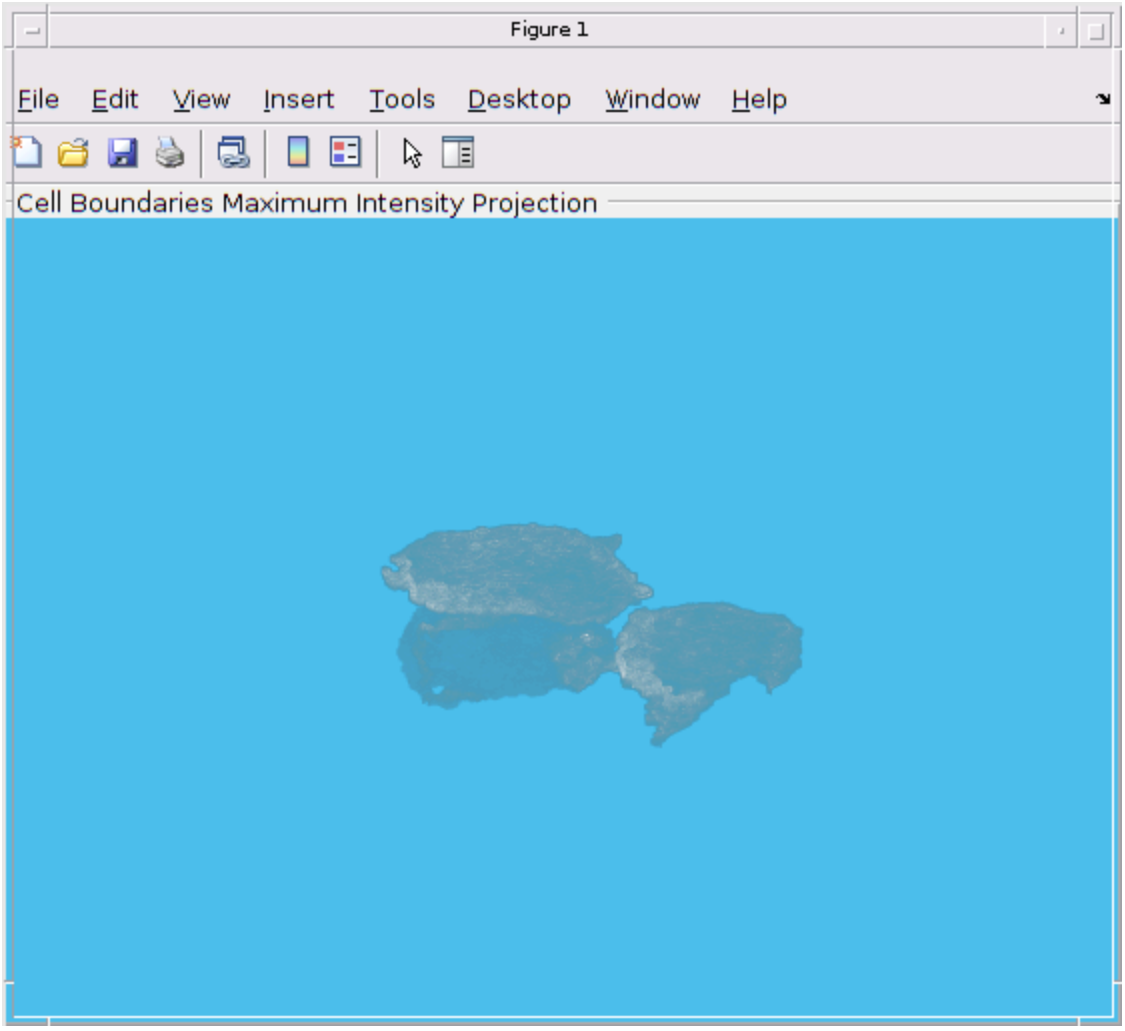


Figure 1
File Edit View Insert Tools Desktop Window Help
Single Cell Masks and Ridge Lines




```
%find vacuoles in single cells and return the results in a structure
[SingleCellStructure] =
 FindVaculoes(r,g,b,bw_stats,name,eGFP_threshold,...

 mCherry_threshold,MinvacVoxeGFP,MinvacVoxmCherry,MinNucVox,Solidity);

%close all figures and clear command window as we are done
close all; clc;
```

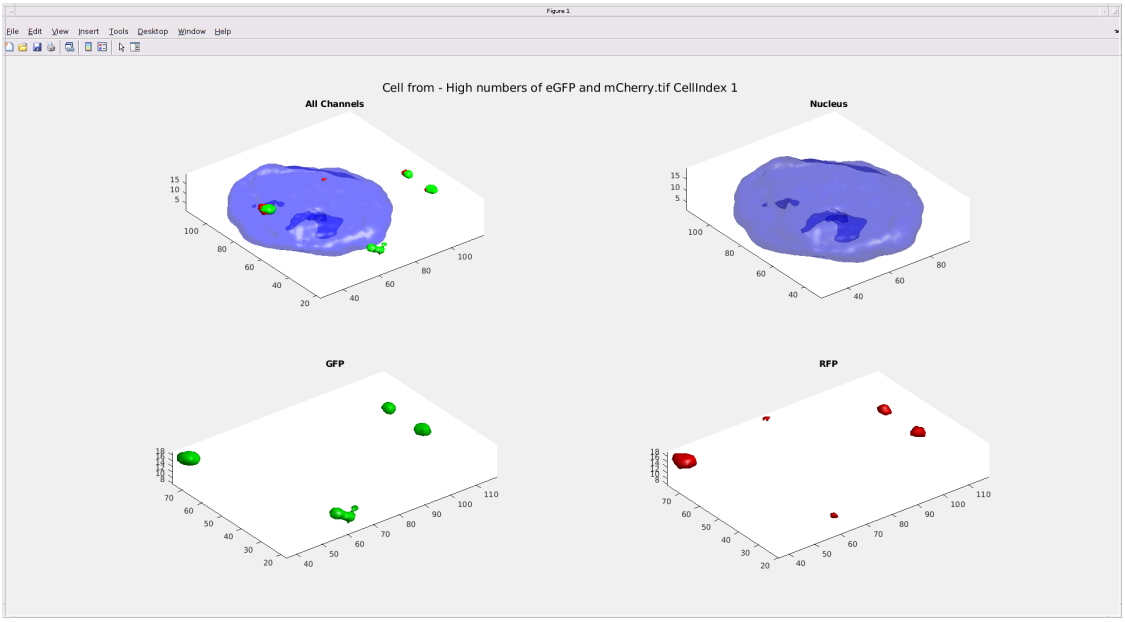


Figure 1
File Edit View Insert Tools Desktop Window Help
Cell from - High numbers of eGFP and mCherry.tif CellIndex 1
All Channels
Nucleus
GFP
RFP

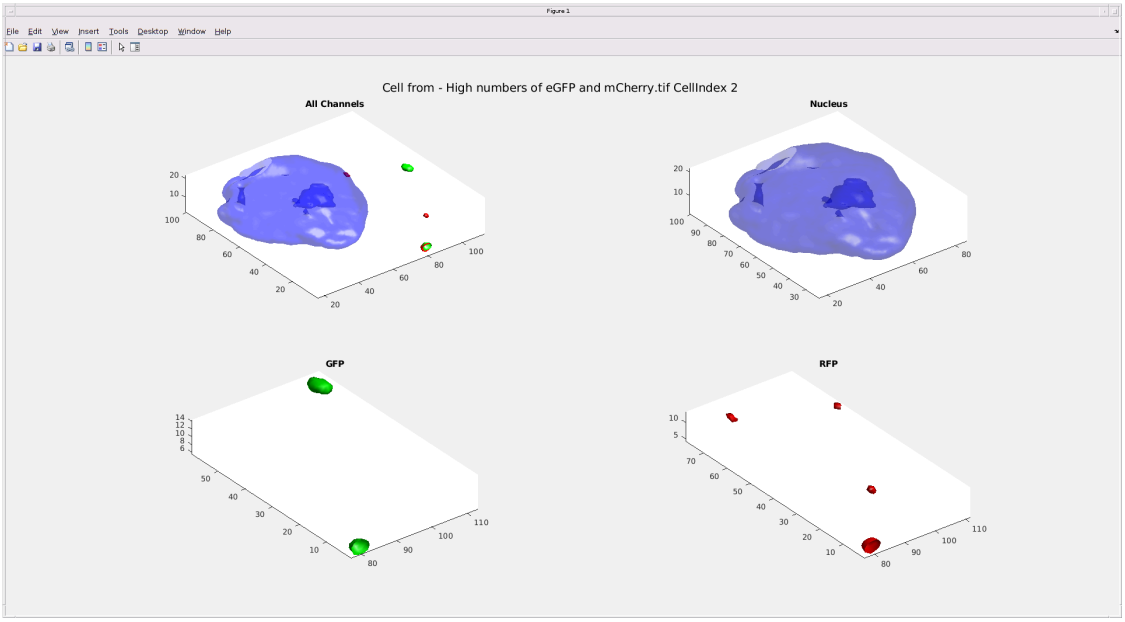
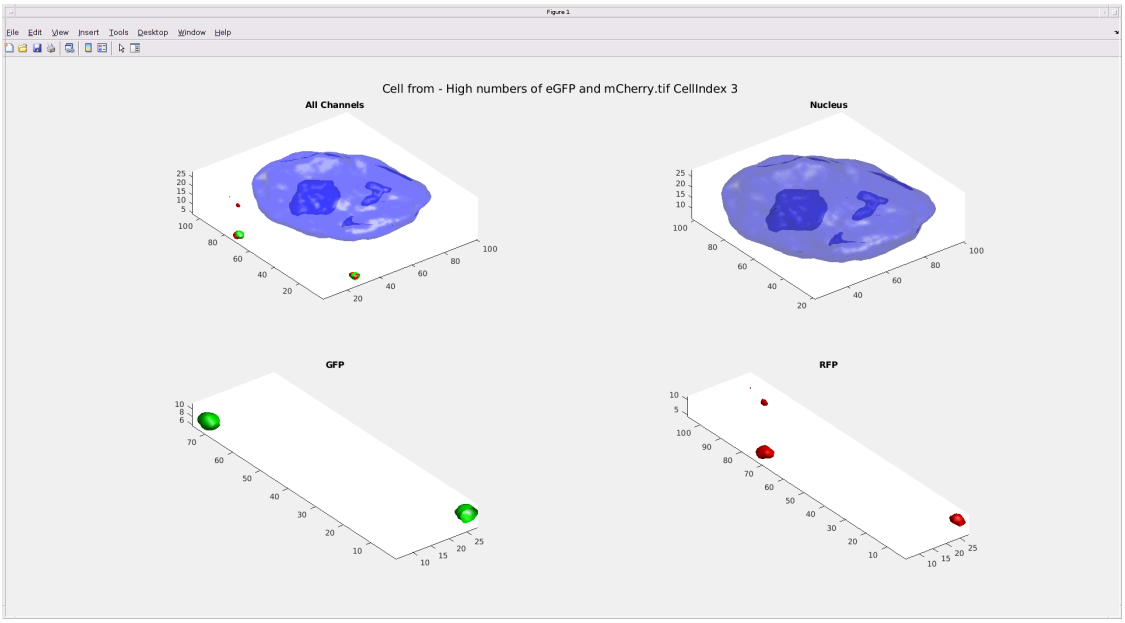


Figure 1
File Edit View Insert Tools Desktop Window Help
Cell from - High numbers of eGFP and mCherry.tif CellIndex 2
All Channels
Nucleus
GFP
RFP



Accumulate results

```
dat.FileName = name;
dat.SingleCell = SingleCellStructure;

%return single cell structure containg counts as a table
[Results_Table] = TabularCountVacuoles(dat);

%report the parameters the user decided worked for their data
disp('Inputs used: ')
disp(struct2table(Input))
```

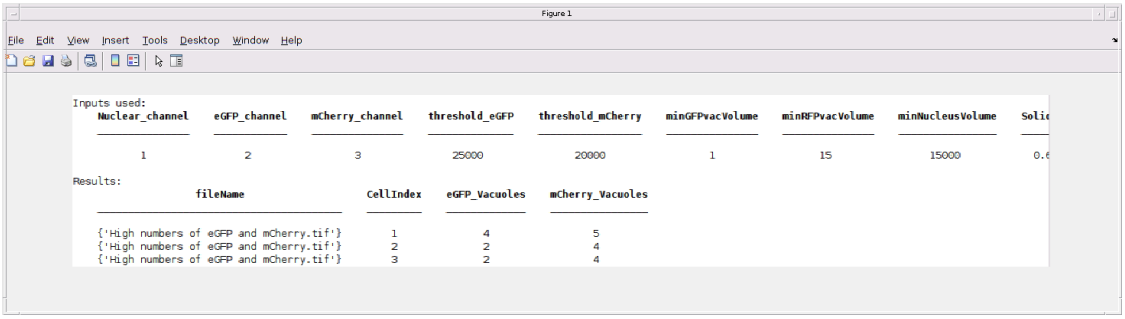
```
Inputs used:
    Nuclear_channel    eGFP_channel    mCherry_channel
 threshold_eGFP    threshold_mCherry    minGFPvacVolume
 minRFPvacVolume    minNucleusVolume    Solidity
    _____________    _____________    _____________
 _____________    _____________________    _____________
 _____________    _____________________    _________

          1                2                3                25000
            20000                  1                  15
 15000            0.6
```

```
%diplay the fruits of your labor for this image using the inputs
 accepted
disp('Results: ')
disp(Results_Table)
```

```
Results:
                  fileName                    CellIndex
 eGFP_Vacuoles    mCherry_Vacuoles
```

```
    {'High numbers of eGFP and mCherry.tif'}        1              4
                5
    {'High numbers of eGFP and mCherry.tif'}        2              2
                4
    {'High numbers of eGFP and mCherry.tif'}        3              2
                4
```

```
%The published display looks horrible without using a Latex wrapper so
 I've
%included an image of how it looks in the command window.
%I need to add a Latex wrapper for this
imshow('Resultscmd.png')
snapnow
```




```
end
```

Published with MATLAB® R2020b