USER'S MANUAL

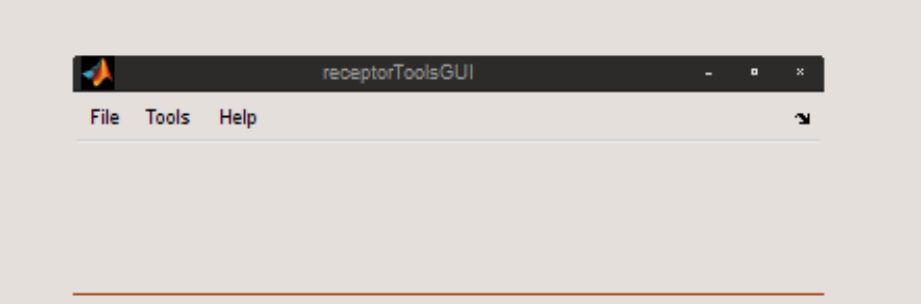


Table of Contents

1 About

The software interface outlined in this document was created to the specific needs of Cornell's Harris-Warrick lab in the department of neurobiology.

Last Updated - May 2011

by Adam McCann ajm232@cornell.edu

2 System Specifications

All MATLAB scipts were created and tested in the MATLAB 2010b version with image processing toolbox included.

3 Installation

All software scripts and files are included in the zipped file

receptorToolsGUI.zip

Including the following scripts:

blobFinder.m
makeHistGUI.fig
makeHistGUI.m
putvar.m
receptorToolsGUI.fig
receptorToolsGUI.m
removeRefMeanGUI.fig
removeRefMeanGUI.m
saveImageGUI.fig
saveImageGUI.m
sphereFinder.m
sphereFinderGUI.fig
sphereFinderGUI.m
uiBlobFinderGUI.fig
uiBlobFinderGUI.m
uigetBitDepth.fig
uigetBitDepth.m
uiThreshGUI.fig
uiThreshGUI.m

the following documents:

McCann – MEng Report.pdf
McCann – Meng Report – User's Manual.pdf

as well as a folder containing a pair 2D sample images, one 3D sample set, and one PSF image.

To install and use the software, place all the scripts above in the same directory, change the current directory in MATLAB to this directory and run receptorToolsGUI.m.

4 Input/Output

As mentioned above, to start the program use the following command:

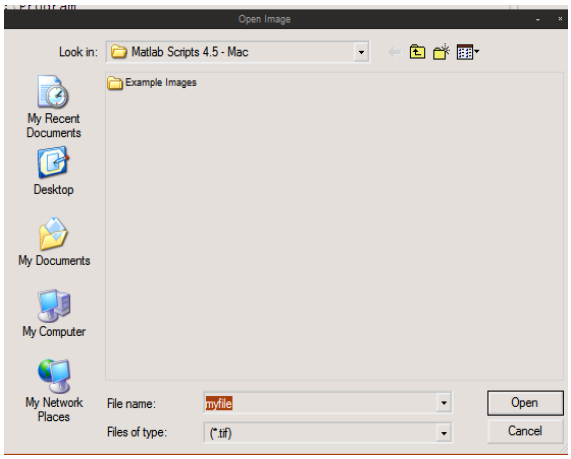
```
>> run receptorToolsGUI
```

For processing 2D images, open the images first then select the desired process from the menu bar on the figure titled receptorToolsGUI. For 3D data sets (image sequences), select the desired process from menu bar on the figure titled receptorToolsGUI and it will prompt you to open a sequence of images.

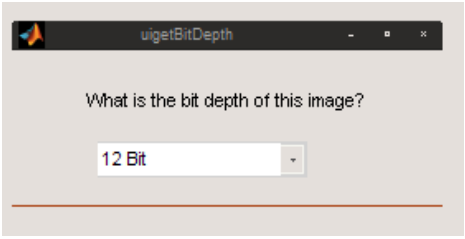
4.1 Opening Images

In the receptorToolsGUI figure, under the file menu select Open Image or use the shortcut CTRL+O when the receptorToolsGUI is active.

You will be prompted to select a file



You may select .tif images of bit depth 8, 12, or 16. After selecting open, you will be prompted to enter the bit depth of the image that's just been selected.



The software will then open a figure with the properties:

Tag	'image#'
Name	['bitDepth' filename]
User Data	raw image data (in raw bit depth)

The image that is displayed inside the figure is the selected raw image converted to 8-bit and resized to be half of its original size. We convert to 8-bit for all displayed images because the human visual system can really only distinguish on the order of 256 different grey levels. The resizing just help the I/0 to run quickly and allows the user to keep many images open at once with minimal memory load.

WARNING:

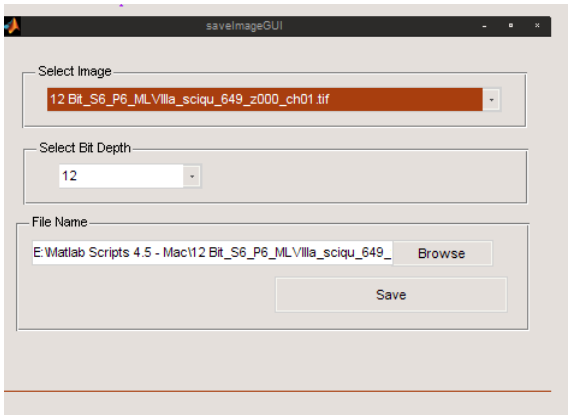
Do not use the File > Save function in the image figures to save the images. This will save a "screen shot" of whatever is in the figure window, so you will be saving an 8-bit half sized version of your original image.

TROUBLESHOOTING TIPS:

If you load in images and they look darker than they are supposed to be, it's probably because you got the wrong bit depth. Typically, images coming directly from the microscope are in 12 bit and deconvolved images are in 16 bit. It is helpful to label saved images with their bit depth.

4.2 Saving Images

In order to save an image that is open in one of the figures select File > Save Image on the receptorToolGUI figure. Presumably you would want to do this after you've performed a Max Z Project or a Reference Mean Subtraction. The interface will bring up a list of currently open images for you to choose from along with a bit depth to be saved in and the file name.



WARNING:

Saving in 12-Bit format will really be in 16-Bit format, but will leave the last four bit blank.

4.3 Close All Images

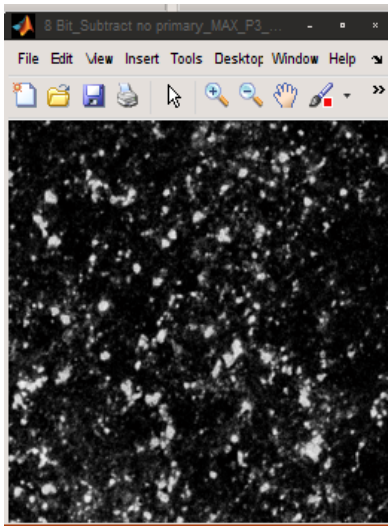
In the receptorToolsGUI figure, selecting File > Close All Images will close only the images with the 'Tag' starting with the 'image', or equivalently all the figures that aren't plots or interfaces.

5 Tools

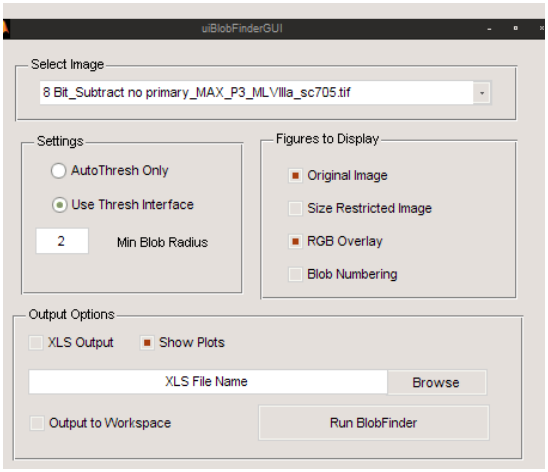
5.1 2D BlobFinder

The blobFinder tool is used to identify and characterize the blobs in a given image. The output is a list of blobs and their corresponding area, total intensity, and average intensity values.

I'll use the 8-bit example image, Subtract no primary_MAX_P3_MLVIIIa_sc705.tif, for the following walk-through.



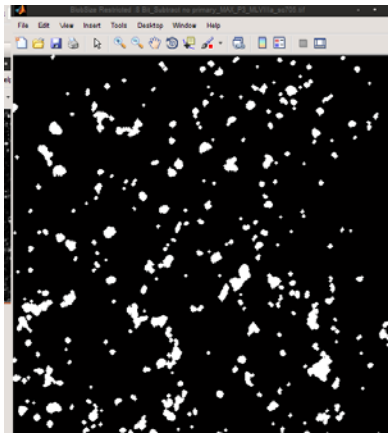
Selecting Tools > Filters > 2D BlobFinder in the receptorToolsGUI figure will bring up the following interface.



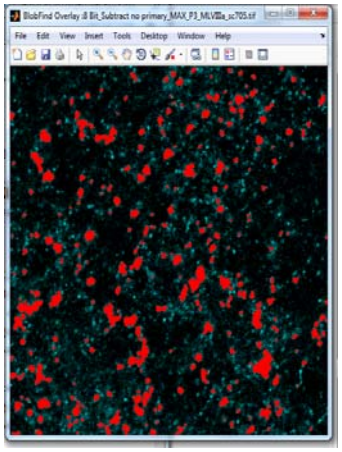
The Select Image drop down box will list all the currently open images, I've selected the example image shown above. In the Settings group, the AutoThresh Only will just disable the user interactive thresholding of the image and just use the value of the image mean intensity + one standard deviation of the intensity. The Min Blob Radius setting controls the radius of the minimum allowable blob.

The possible figures to display are:

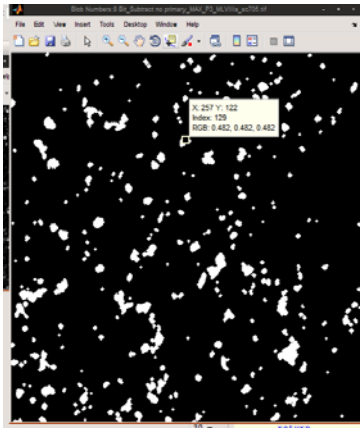
- *Original Image* – When selected, this will keep the original image from closing.
- *Size Restricted Image* – When selected, the software will show the black and white, thresholded and blob size restricted image.



- *RGB Overlay* – When selected, the software will show the identified blobs in red overlaying the original image in blue and green.



- *Blob Numbering* – When selected, the software will show the identified blobs and if the user selects the Data Cursor in the figure window they can click on a blob the label number will be shown.



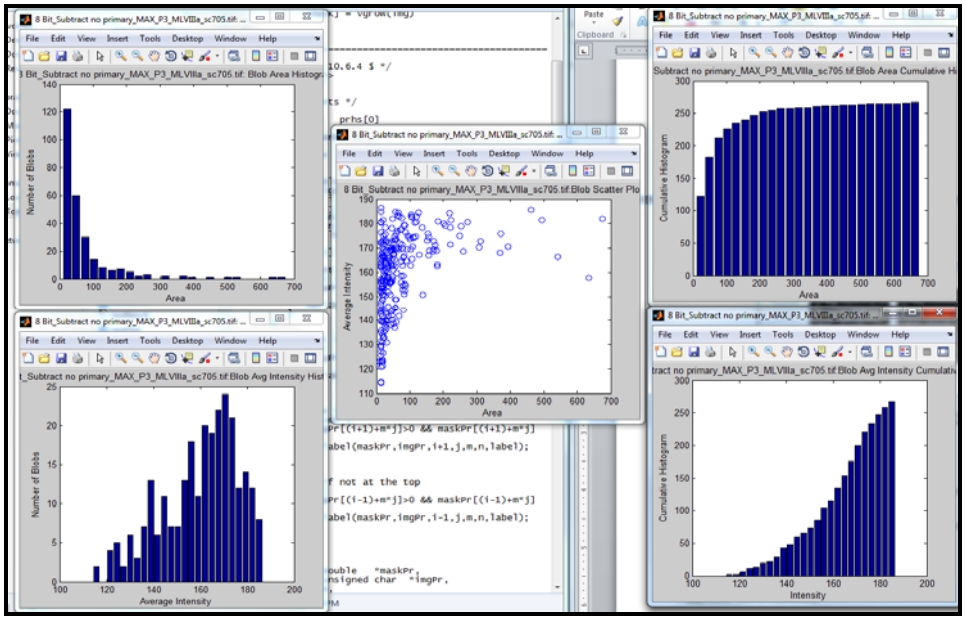
The possible output options are:

- *XLS Output* – When selected, this will save a copy of the blob characteristic data to an XLS file of the name specified by the user in the text dialog box below the XLS Output button. You cannot use this

function to append the results onto an existing XLS spreadsheet because the data values will always be written in the first three columns of the first sheet.

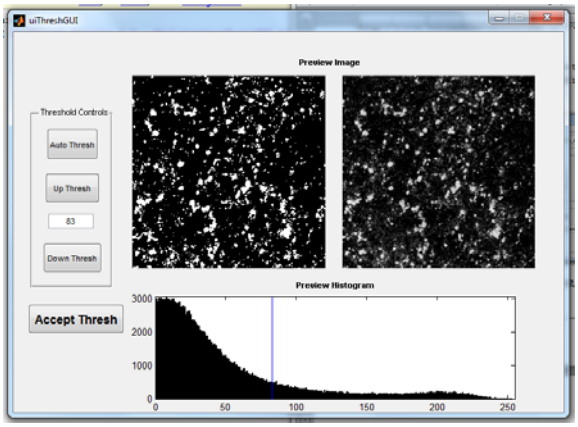
- *Show Plots* – When selected, the software will display the following 5 plots regarding the blob characteristic data:

    - Histograms for receptor area and average intensity
    - Cumulative histograms for area and average intensity
    - Scatter plot of receptor area vs avg intensity



- *Output to Workspace* – When selected, the software will place a copy of the 3xnumblobs characteristic data matrix in the workspace under the variable name blob_info. If there is a variable in the workspace already call blob_info, this will not append the new data to the old data, but will instead write over the blob_info variable.

Selecting <Run BlobFinder> will open up the user interactive threshold panel if auto thresh was not selected.



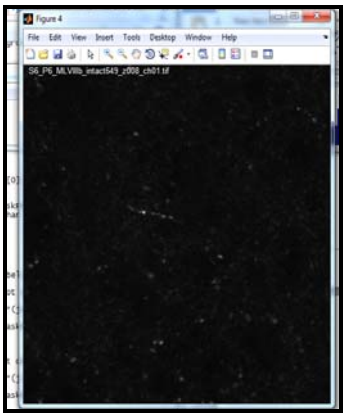
The user can adjust the threshold with the up or down buttons or by just typing it in the box and pressing enter. When the selected threshold changes, the left image in the Preview image set will show the resulting, thresholded image while the right image shows the unchanged original. The user may also want to use the image histogram provided at the bottom of the panel with a blue line to indicate the current threshold. Selecting the Accept Thresh button will cause the program to finish and output all the selected options.

5.2 3D BlobFinder

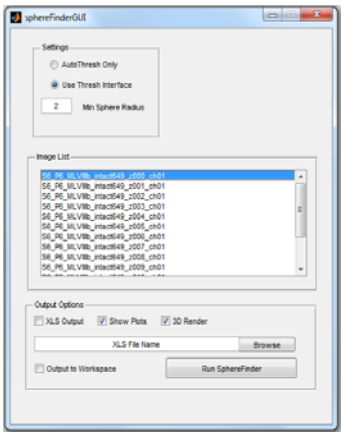
The 3D blobFinder tool is used to identify and characterize the blobs in a given image sequence. The output is a list of 3D blobs and their corresponding volume, total intensity, and average intensity values.

I'll use the 12-bit example sequence in the folder 3D Test Set.

Selecting Tools > Filters > 3D BlobFinder in the receptorToolsGUI figure will prompt the user to select a group of files. It is assumed that the files to be analyzed are all the same bit depth and are sorted in the file so that when they are loaded the stack is in the correct order. This is simple to do if put the z parameter in the file name. Upon selecting the images, they will be displayed as they are loaded in along with their filename in text.



After the files have been loaded the following interface will appear.



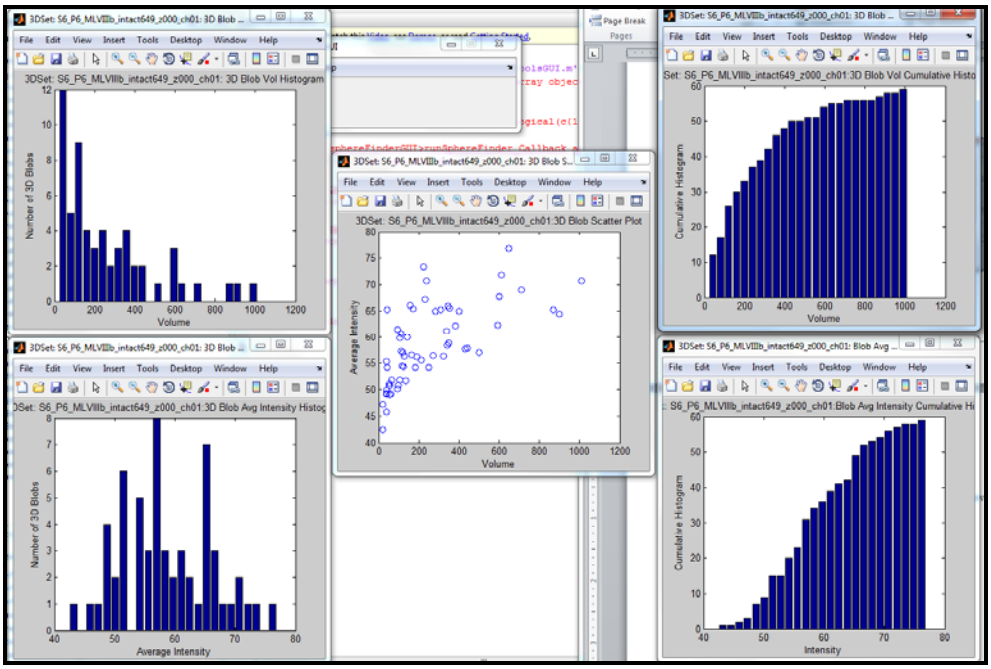
When clicked, the Image List will list all the image loaded into the sequence in the order that they will be stacked. In the Settings group, the AutoThresh Only will just disable the user interactive thresholding of the image and just use the value of the image mean intensity + one standard deviation of the intensity. The Min Blob Radius setting controls the radius of the minimum allowable 3D blob.

The possible output options are:

- *XLS Output* – When selected, this will save a copy of the 3D blob characteristic data to an XLS file of the name specified by the user in the text dialog box below the XLS Output button. You cannot use this function to append the results onto an existing XLS spreadsheet because the data values will always be written in the first three columns of the first sheet.

- *Show Plots* – When selected, the software will display the following 5 plots regarding the blob characteristic data:

    - Histograms for receptor volume and average intensity
    - Cumulative histograms for volume and average intensity
    - Scatter plot of receptor volume vs avg intensity



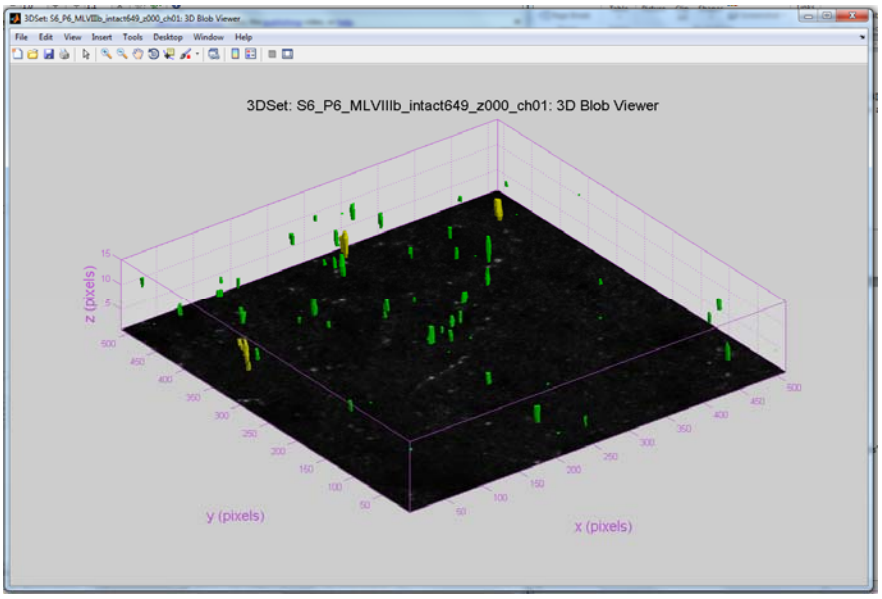
- *Output to Workspace* – When selected, the software will place a copy of the 3xnumblobs characteristic data matrix in the workspace under the variable name sphere_info. If there is a variable in the workspace already call sphere_info, this will not append the new data to the old data, but will instead write over the sphere_info variable.

- *3D Render* – When selected, a 3D rendering of the identified receptors is generated using MATLAB's isosurface function. Before rendering, the identified receptors are given a size category number. Small receptors (volume < 800 voxels) are labeled first. Once they're labeled the label data is passed to the isosurface function along with the label value. This function looks for points in the data with the specified label value and tries to connect then with patches. These patches are calculated as vertices and

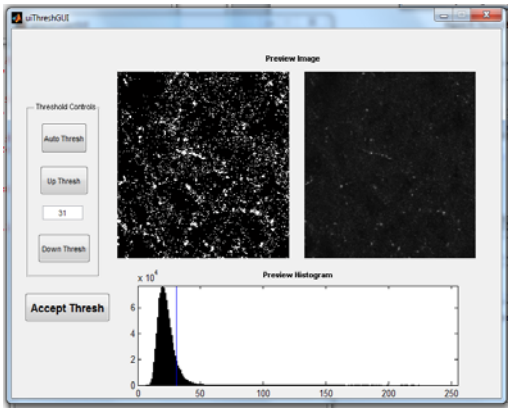
edges and are then passed to the patch renderer function with instructions to make them green. The same process happens for medium receptors (800 > volume > 3000) that are colored yellow and large receptors (volume > 3000 voxels) which are colored red. These rendered patch surfaces are displayed on a graph with dimensional aspect ratio adjusted to account for the physical distance between confocal images and the pixel to pixel distance. For most cases in data I've seen the following values determine the aspect ratio:

```
pixSize_xy = 91.55e-9;
 pixSize_z = 1.18e-6;
 zFactor = pixSize_z/(pixSize_xy*2);
```

The factor of two in the denominator of the zFactor is due to the fact that the 3D view is rendered at half resolution (512x512 instead of 1024x1024). Some lighting and background color is also added to the plot to help the 3D effect. The max projected image is also shown on the floor of the plot to help the researcher relate the 3D rendering with the 2D images.



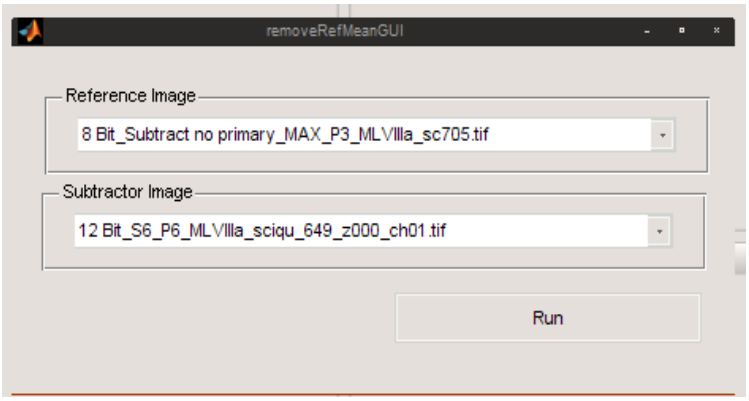
Selecting <Run SphereFinder> will open up the user interactive threshold panel if auto thresh was not selected.



The user can adjust the threshold with the up or down buttons or by just typing it in the box and pressing enter. When the selected threshold changes, the left image in the Preview image set will show the resulting, thresholded Max Z projection of the set while the right image shows the unchanged original. The user may also want to use the image histogram provided at the bottom of the panel with a blue line to indicate the current threshold. Selecting the Accept Thresh button will cause the program to finish and output all the selected options.

5.3 Remove Ref Mean

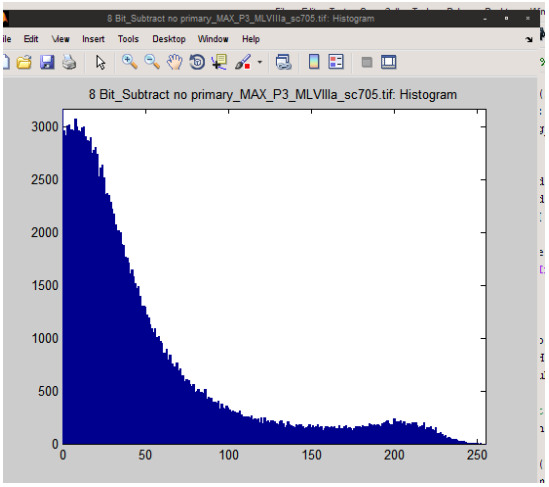
This tool is used to subtract the mean of one image from another. Selecting Tools > Math Operators > Remove Ref Mean in the receptorToolsGUI figure will open the following dialog box.



The mean of the reference image will be subtracted from the subtractor image. The resulting image is always in 8 Bit and will labeled so in the name of the resulting figure.

5.4 Histogram

This tool is used to show the histogram of a selected image. Selecting Tools > Histogram in the receptorToolsGUI figure will open an image select list. Selecting <Generate Histogram> will produce the histogram for the image. If the image is mostly black the histogram will let the zeroth bin of the histogram run off the chart and display a warning box.



5.5 Max Z Project

This tool is will compute the Maximum intensity for the each z projected pixel in a image sequence. For all notation in the software and in this manual, the image axes are x and y while the axis perpendicular to the plane of the image is the z plane.

Selecting Tools > Histogram in the receptorToolsGUI figure will open an image select list. Selecting multiple files of the same bit depth will compute the max z project and create a figure of the same bit depth with the annotation Z PROJECT in the figure name.

6 Sample Work Flow

In this example, the goal is to use a 12 bit raw image and its 12 bit corresponding reference image to get a blob_info data set.

- Open both images with receptorToolsGUI
- Use the Remove Ref Mean to subtract the raw reference mean from the raw image.
- Save image (not the reference image) in 8-bit
- Open the 8-bit image in ImageJ and apply deconvolution, then save in 8-bit from ImageJ
  - ImageJ has a plugin called Iterative Deconvolution. Run this 10 iterations.
- Open the 8-bit deconvoled image with receptorToolsGUI
- Apply 2D BlobFinder and output data to workspace

7 Adding Future Functionality

Adding new functionality to receptorToolsGUI can be very easy if you use the following steps and copy certain sections of existing code in order to get the I/O right.

If for example, I wanted to add a function that generated a random integer for every pixel in a specified image. I'd call this function randomize and save it in a file called randomize.m. It is important the raw image and it's bit depth be input parameters into the function. The script for randomize.m might be something like this, where I've included the bitdepth conversion just to show:

```
function [img_8_rando] = randomize(raw_img,bitDepth)

   imgHi = 2^bitDepth-1;
   imgLo = 0;

   img_8_rando = uint8(255.*double((img-imgLo))./double((imgHi-imgLo)));
```

```
for i = 1:size(img_8,1)
      for j = 1:size(img_8,1)
            img_8_rando = randi(255);
      end
end
```

To connect this function to receptorToolGUI, start by typing the command

```
>> guide
```

Open receptorToolsGUI.fig and Tools > Menu Editor. I'll add randomize to Tools and give it the Label: Randomize along with the Tag: menu_tools_randomize. Save receptorToolsGUI.fig and open receptorToolsGUI.m.

In the receptorToolsGUI.m file there should be a new function listing called menu_tools_randomize_Callback.

Now I want to use randomize on the images that are already open in receptorToolsGUI, so I'm going to implement a new GUI called randomizeGUI.m and randomizeGUI.fig

Again type `>> guide` and this time select Blank GUI. Add a pop up menu for image list and make tag for it 'imageList' using the property editor for the pop up menu. Next add a run button and give it the tag 'runRandomize'. Save the figure as randomizeGUI.fig and it will generate the file randomizeGUI.m.

In randomizeGUI.m, there should be a function called imageList_CreateFcn. Paste the following code under it:

```
ihandles = findobj('-regexp','Tag','^image');
ilist = cell(0);

for i=1:size(ihandles,1)
    ilist = [ilist;cellstr(get(ihandles(i),'Name'))];
end

if isempty(ilist)
    ilist = cellstr('No Images Open');
end
set(hObject,'String',ilist)
```

Next, under the function runRandomize_Callback add the image passing code, a call to the original function randomize, and the output image passing to figure code

```
%% Image Selection Data Passing
imgList = get(handles.imageList,'String');
listVal = get(handles.imageList,'Value');
imgName = imgList{listVal};

if isequal(imgName,'No Images Open')
    errordlg('No Images Selected','make Hist Error');
    return
end

imgHandle = findobj('Name',imgName);
img = get(imgHandle,'UserData');
bitDepth = str2double(imgName(1:2));
```

```
  %% Call to function
  img_8_rando = randomize(img,bitDepth);


  %% Pass resulting image to figure

 figure('Name',[num2str(bitDepth) ' Bit_Ref Remove_'...
imgName],'NumberTitle','Off',...
                'Tag','imageRefRemove');
 set(gcf,'UserData',img_8_rando);
 imshow(img_8_rando)
 delete(handles.randomizeGUI)
```

Finally, in receptorToolsGUI.m under the function menu_tools_randomize_Callback put

```
randomizeGUI();
```