Protocol for extracting basic color metrics from Images in ImageJ/Fiji

October 2021
By CF Strock

Background Information on Colorspaces:

- **RGB** = **R**ed **G**reen **B**lue
  Commonly used for color in digital cameras, computer display, etc.
  Starts with black and as colors are added, the amount of emitted light is increased (*i.e.* additive color) = **R**ed, **G**reen and **B**lue « **channels** » with pixel values for every pixel in each channel

  For example, for a 24bit color image:
  white = R255, G255, B255
  black = R0, G0, B0.
  Pure **red** = **R255**, **G**0, **B**0
  Pure **green** = **R**0, **G255**, **B**0
  Pure **blue** = **R**0, **G**0, **B255**.
  All other colors are displayed by varying these pixel values.

- **CIE Lab** = Lightness (**L**) and two color components (**a, b**) which work in an opposing way. Attempts to approximate human vision, therefore has a larger color gamut.
  L* = 0 = black
  L* =100 = white
  a* (position between green and red) = negative values are green, positive values are red
  b* (position between blue and yellow) = negative values are blue, positive values are yellow

- **HSB/HSV** = **H**ue **S**aturation **B**rightness/Value
  Each color shade (**hue**) has a value.
  Saturation is equivalent to intensity,
  Brightness/Value relate to lightness. Work in combination. Relates to RGB values.

Download ImageJ (Fiji)

Download ImageJ (Fiji) for free here: https://imagej.net/software/fiji/

Basic color analysis macro available for download here:
https://doi.org/10.5281/zenodo.5595203

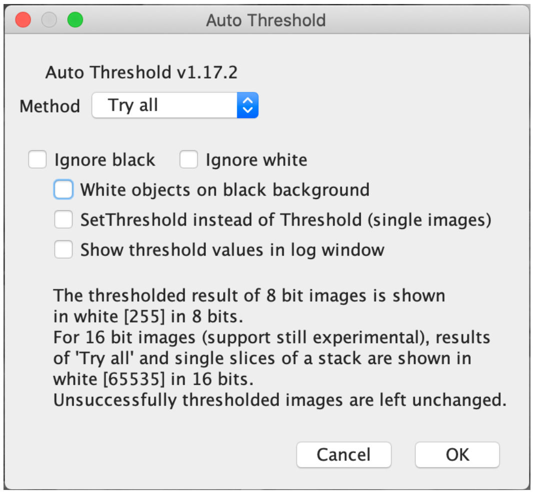
Extracting summary statistics of channels for each color space using a macro

Using a macro helps to automate a job in ImageJ and ensures reproducibility of our work. This macro automates the extraction of summary statistics from component channels of three colorspaces (RGB, Lab, and HSB) for each image. This macro functions to 1) open an image from a directory folder, 2) threshold the image to segment out the object of interest from the background then "clear" the background so it is completely black, 3) separate the image into the component channels for each color space, 4) extract summary statistics for each channel in each colorspace, 5) add these summary statistics for each channel of each colorspace (RGB, Lab, and HSB) to a .csv file.

Before you start, a note on thresholding: The supplied macro (*Whole Color Macro_All.components.ijm*) is parameterized to use color thresholding to segment out the object of interest from the background. The thresholding parameters in the supplied macro will need to be adjusted for your specific image set (lines 37-75 in macro)! Thresholding parameterization provided in the downloadable macro is based on the RGB colorspace, but thresholding based on other color spaces may be a better option for other image sets. Before running this macro, determine the best colorspace and parameters for thresholding the image set you are working with and replace the code in the macro with the correct parameters for your images at lines 35-75.

To determine the best thresholding parameters for your images:

1. You can test how different automatic thresholding methods from imageJ work on your images by:
    a. opening a representative image from your set in imageJ; go to Image > Type > 8-bit (now your image will appear greyscale and is ready for auto thresholding)
    b. go to Image > Adjust > Auto Threshold > for method select “Try all” and check the box if your objects of interest appear as white on a black background > select “OK”



Auto Threshold v1.17.2

Method: Try all

□ Ignore black □ Ignore white

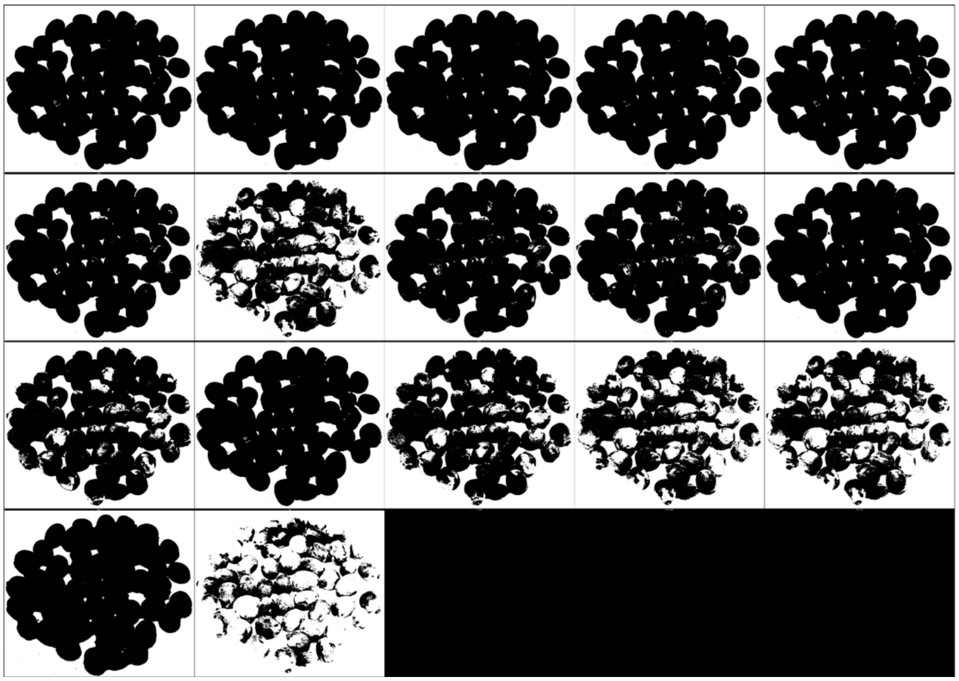
□ White objects on black background

□ SetThreshold instead of Threshold (single images)

□ Show threshold values in log window

The thresholded result of 8 bit images is shown in white [255] in 8 bits.
For 16 bit images (support still experimental), results of 'Try all' and single slices of a stack are shown in white [65535] in 16 bits.
Unsuccessfully thresholded images are left unchanged.

c. Output will appear as a matrix if binary images with results from all 17 methods, allowing the user to explore how different algorithms perform in segmenting their object of interest from the background (see image below). Results from each method will be presented from left to right, top to bottom in the following order: Default, Huang, Huang2, Intermodes, IsoData, Li, MaxEntropy, Mean, MinError(I), Minimum, Moments, Otsu, Percentile, RenyiEntropy, Shanbhag, Triangle, Yen. Further information on these methods and the algorithms behind them can be found at: https://imagej.net/plugins/auto-threshold



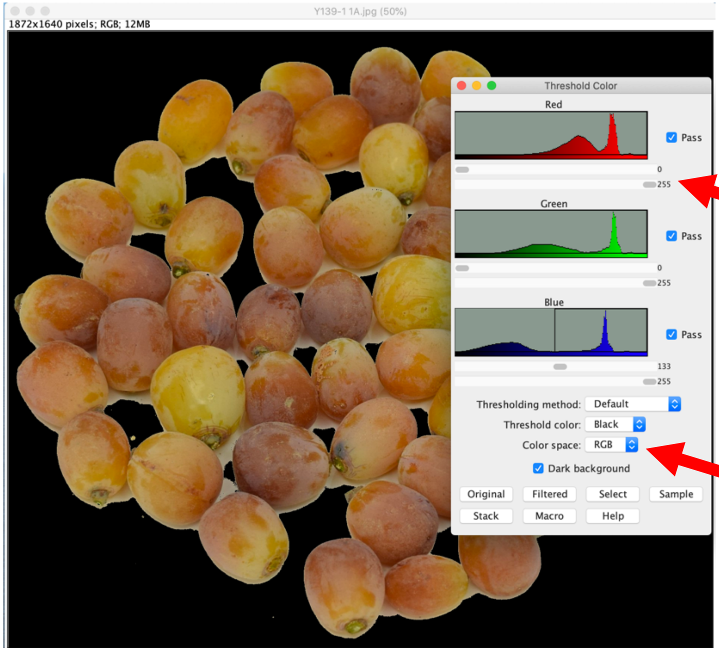
d. To obtain the macro script to run the auto thresholding method that works best for your images, open a new image in imageJ, then go to Plugins > Macros > Record (a window titled “Recorder” should open)

e. Now go to Image > Type > 8-bit; then go to Image > Adjust > Auto Threshold > for “method” select the method that worked best for your image and hit “OK”. You should now see the code that will parameterize the thresholding settings for your images appear in the “Recorder” window. This code will include the following (replace “**PREFERED METHOD**” with the method that worked best for your images):

run(“8-bit”);
run(“Auto Threshold”, “method= **PREFERED METHOD** *white”);*

f. Copy and paste this to replace code at lines 37-75 in the supplied macro (*Whole Color Macro_All.components.ijm*)

2. **Alternatively**, you can customize your own color thresholding parameters by opening a representative image in imageJ; go to Image > Adjust > Color Thresholding
    a. select the colorspace you would like to use to threshold the image (HSB, RGB, Lab, YUV)
    b. Use the sliding controls beneath each of the channels that comprise the selected colorspace to set the appropriate values that most effectively separate the object of interest from the background.

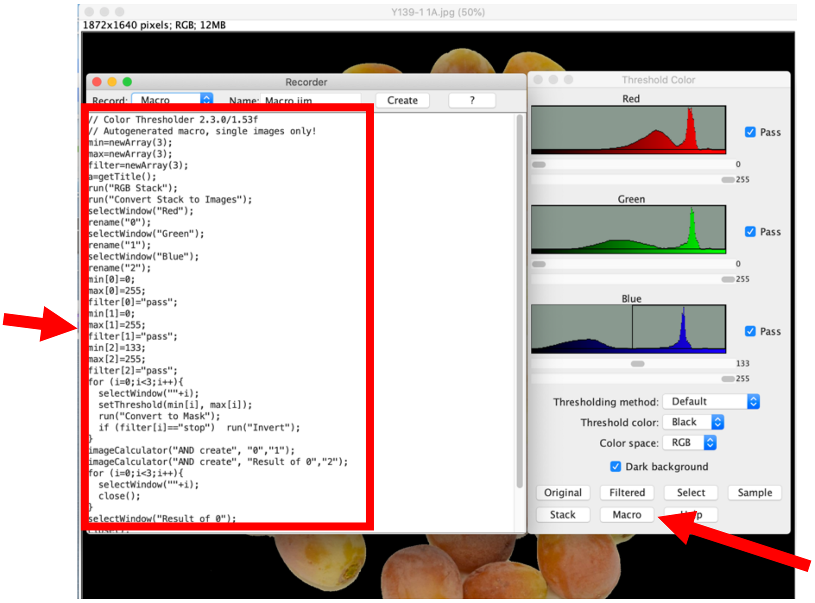


Red: ☑ Pass 0 255
Green: ☑ Pass 0 255
Blue: ☑ Pass 133 255
Thresholding method: Default
Threshold color: Black
Color space: RGB
☑ Dark background

Adjust channel parameters here

Select preferred colorspace here

    c. To obtain the macro script to run the thresholding parameters that work best for you images, go to Plugins > Macros > Record (a window titled “Recorder” should open)
    d. In the “Threshold Color” window you used to select the most effective colorspace and channel parameters to segment your object from the background, select the “Macro” button. You should now see the code that will parameterize the thresholding settings for your images appear in the “Recorder” window. Copy and paste this to replace code at lines 37-75 in the supplied macro (*Whole Color Macro_All.components.ijm*)



☑ Pass (Red: 0, 255)
☑ Pass (Green: 0, 255)
☑ Pass (Blue: 133, 255)

Thresholding method: Default
Threshold color: Black
Color space: RGB
☑ Dark background

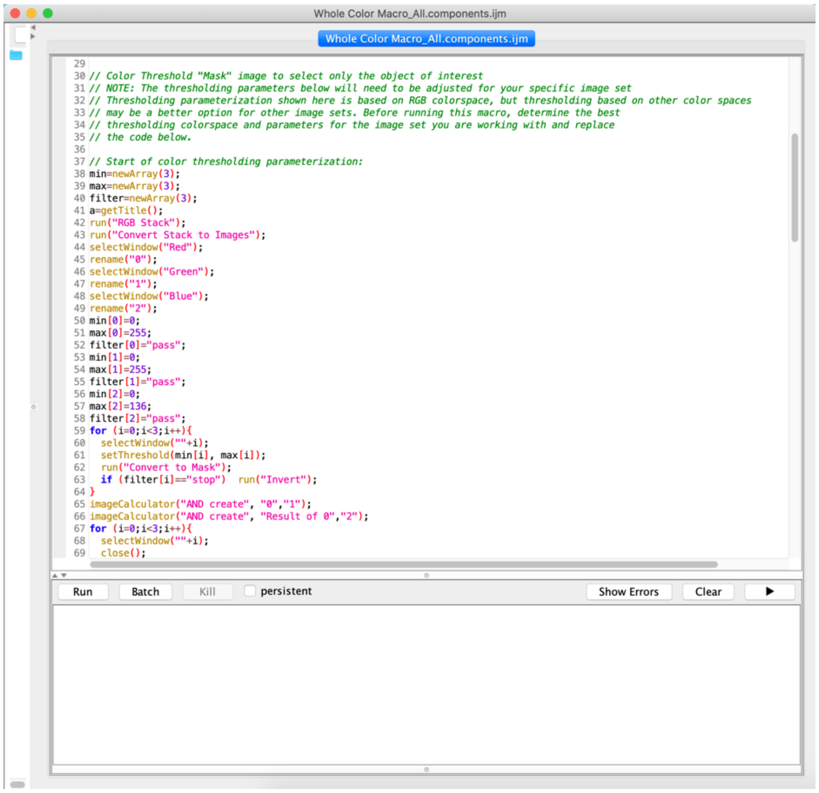
```
// Color Thresholder 2.3.0/1.53f
// Autogenerated macro, single images only!
min=newArray(3);
max=newArray(3);
filter=newArray(3);
a=getTitle();
run("RGB Stack");
run("Convert Stack to Images");
selectWindow("Red");
rename("0");
selectWindow("Green");
rename("1");
selectWindow("Blue");
rename("2");
min[0]=0;
max[0]=255;
filter[0]="pass";
min[1]=0;
max[1]=255;
filter[1]="pass";
min[2]=133;
max[2]=255;
filter[2]="pass";
for (i=0;i<3;i++){
  selectWindow(""+i);
  setThreshold(min[i], max[i]);
  run("Convert to Mask");
  if (filter[i]=="stop")  run("Invert");
}
imageCalculator("AND create", "0","1");
imageCalculator("AND create", "Result of 0","2");
for (i=0;i<3;i++){
  selectWindow(""+i);
  close();
}
selectWindow("Result of 0");
```

macro script provided here

Select macro to get macro script

To edit the macro and apply your own thresholding parameters:

1. Open ImageJ; go to Plugins > Macros > Edit > navigate to and select location of macro (*Whole Color Macro_All.components.ijm*) on computer and select "Open"
2. Replace lines 37-75 in the supplied macro (*Whole Color Macro_All.components.ijm*), with the code for the parameters that best threshold out the object of interest in your image set.



Whole Color Macro_All.components.ijm
Whole Color Macro_All.components.ijm
Run
Batch
Kill
Show Errors
Clear
▶

□ persistent

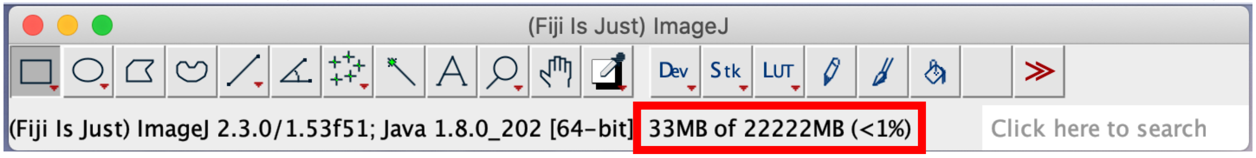
```
29
30 // Color Threshold "Mask" image to select only the object of interest
31 // NOTE: The thresholding parameters below will need to be adjusted for your specific image set
32 // Thresholding parameterization shown here is based on RGB colorspace, but thresholding based on other color spaces
33 // may be a better option for other image sets. Before running this macro, determine the best
34 // thresholding colorspace and parameters for the image set you are working with and replace
35 // the code below.
36
37 // Start of color thresholding parameterization:
38 min=newArray(3);
39 max=newArray(3);
40 filter=newArray(3);
41 a=getTitle();
42 run("RGB Stack");
43 run("Convert Stack to Images");
44 selectWindow("Red");
45 rename("0");
46 selectWindow("Green");
47 rename("1");
48 selectWindow("Blue");
49 rename("2");
50 min[0]=0;
51 max[0]=255;
52 filter[0]="pass";
53 min[1]=0;
54 max[1]=255;
55 filter[1]="pass";
56 min[2]=0;
57 max[2]=136;
58 filter[2]="pass";
59 for (i=0;i<3;i++){
60   selectWindow(""+i);
61   setThreshold(min[i], max[i]);
62   run("Convert to Mask");
63   if (filter[i]=="stop")  run("Invert");
64 }
65 imageCalculator("AND create", "0","1");
66 imageCalculator("AND create", "Result of 0","2");
67 for (i=0;i<3;i++){
68   selectWindow(""+i);
69   close();
```

Basic color analysis macro available for download here:
https://doi.org/10.5281/zenodo.5595203

3. Copy all the images you wish to analyze in a single folder containing only the images. NOTE: This macro will save the segmented image showing what regions were considered in the color analysis over the original image in that folder. If you would like to keep the original images, have them saved in another location as well.

4. Open ImageJ; go to Plugins > Macros > Run > navigate to and select location of macro (*Whole Color Macro_All.components.ijm*) on computer and select "Open"

5. Navigate to the location of your folder containing the images you wish to analyze and select "Open"
   a. TIME: With 20GB of RAM allocated to ImageJ, macro will process a stack of 100 images in ~50 minutes (~30 seconds per image). If you allocate more RAM to imageJ, this time may be reduced.
   b. To modify amount of memory allocated to ImageJ, go to Edit > Options > Memory & Threads.. > at most, the memory you allocate to ImageJ should be at least 1000 MB less than your computer's total RAM
   c. You can confirm how much memory is available when running ImageJ by clicking on the status bar. You will see a "[used] of [max]" memory message.



(Fiji Is Just) ImageJ 2.3.0/1.53f51; Java 1.8.0_202 [64-bit] 33MB of 22222MB (<1%)

6. Macro will run by:
   a. opening the first image in the folder,
   b. thresholding the object from the background,
   c. creating a mask to overlay original image,
   d. create 3 duplicate images for analysis of 1) RGB channels, 2) L*a*b channels and 3) HSB channels
   e. add an ROI on these duplicate images using the mask,
   f. separate the duplicate images into RGB channels, L*a*b channels, and HSB channels
   g. extract the pixel intensities color components from each of the channels in composite images
   h. for each channel (R, G, B, L, a, b, H, S, B), measure the total area of the ROI in pixels, mean value for the ROI, standard deviation of values, modal value, the min and max values. *These data are labeled with the image name they were extracted from and appear in the results in the following row order for each image:

      1st row = Red
      2nd row = Green
      3rd row = Blue

4th row = Lightness
5th row = a* (green to red)
6th row = b* (blue to yellow)
7th row = Hue
8th row = Saturation
9th row = Brightness

i. data are logged to a .csv file titled “*Whole_Color_Measurements.csv*” that will be saved to the folder that contains all the images.

j. the measured image showing what parts of the image were used to extract color components and what was considered “background” (colored in black) will be saved to the same folder the original images were stored in. This macro will overwrite the original images in the folder with the segmented image showing what regions were considered in the color analysis. If you would like to keep the original images, have them saved in another location as well.