

Methods for Color Analysis in Bio-medical Images (an ImageJ/Fiji based tutorial + python codes)

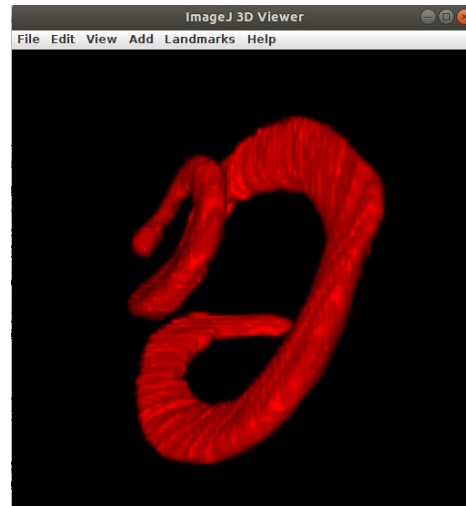
Resources: <http://www.math-info.univ-paris5.fr/~lomn/Cours/CV/BME/HistoPatho>
including the lecture about digital histopathology

(file *TPColor.pdf* and software *fiji-linux64.Core.tar.gz* if needed otherwise install from <http://fiji.sc/Fiji> and *Data*)

A. Introduction : Color image is a 3 channels image, a kind of 3D function $f(x,y)=(R,G,B)$. Hence the mathematical framework can be 3-dimensional vectorial function of 2 spatial variables (x,y) : <https://towardsdatascience.com/step-by-step-the-math-behind-neural-networks-d002440227fb> if you are interested in the math behind it.

Remember that over the 3D tutorial, we started by visualizing a 3D biological sample with the 3D viewer provided by FiJi. Do it again using the same sample *Bat Cochlea Volume* image. *Menu Plugins/3D Viewer*.

Then, test a python 3D viewer using the OpenGL package. The code and the 3D file data are available there <http://helios.mi.parisdescartes.fr/~lomn/Cours/CV/BME/Python/>.



```
$python 3Dbunny.py bunny.obj
```

You will need to install the following packages to make it run (*pygame*, *OpenGL* and *pywavefront*) by the following commands

```
$pip install PyOpenGL
```

see <https://pypi.org/project/PyOpenGL/>

```
$sudo apt-get install python3-pip
```

```
$sudo pip3 install --upgrade pip
```

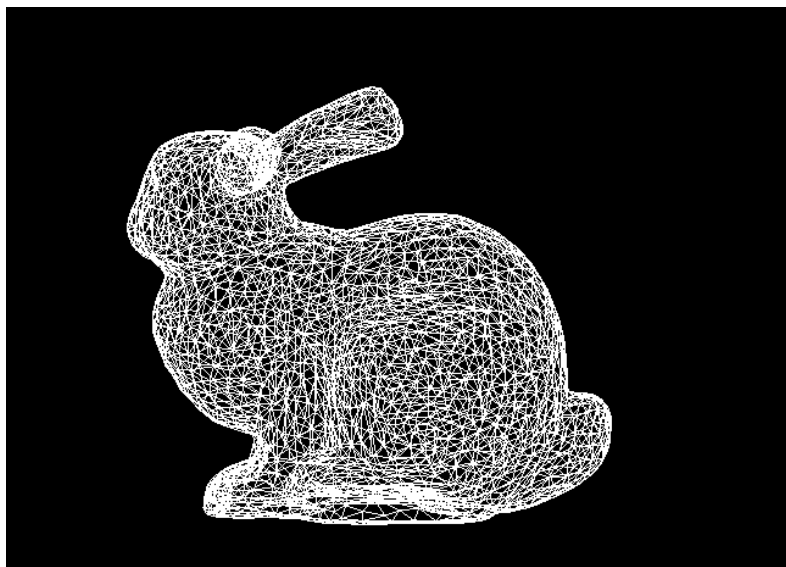
```
$sudo pip3 install pygame
```

see <https://www.pygame.org/wiki/GettingStarted>

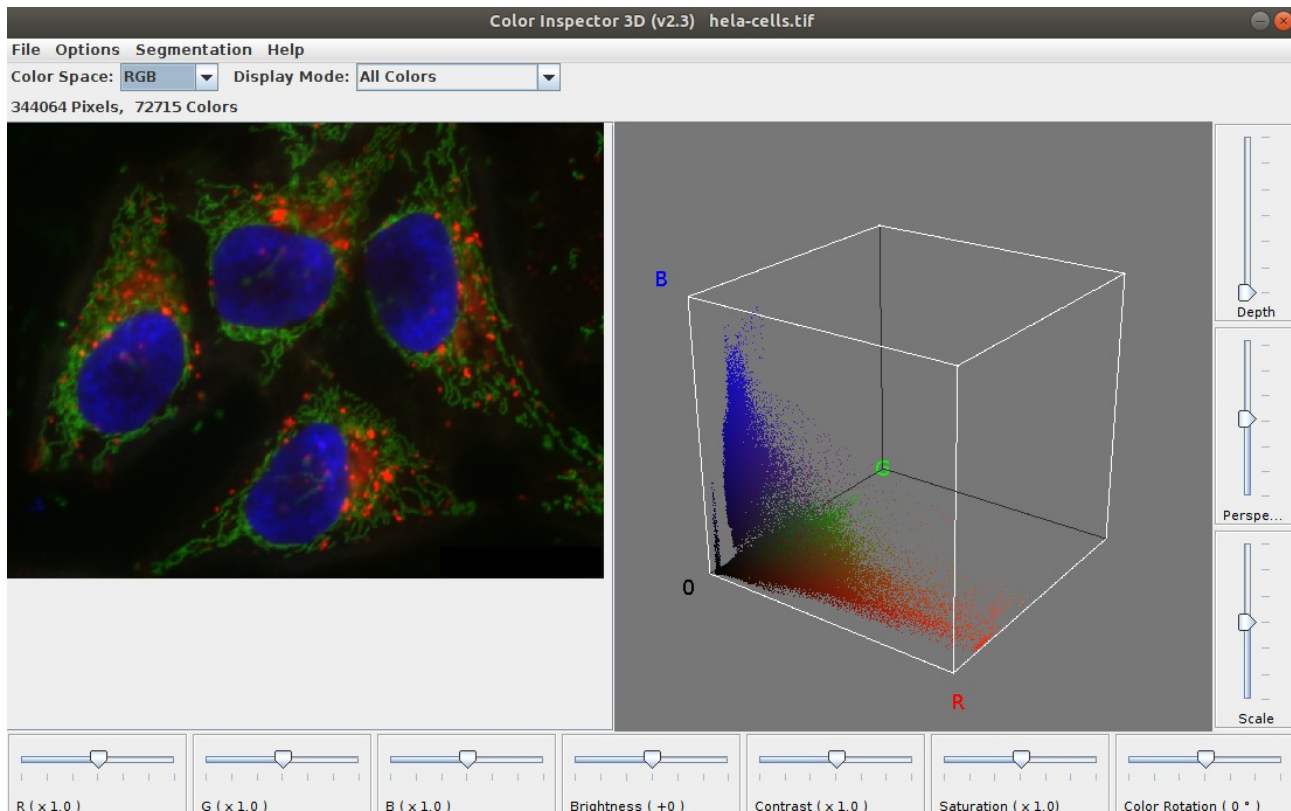
```
$pip install pywavefront
```

see <https://pypi.org/project/PyWavefront/>

You should observe a rotating 3D bunny mesh.



B. Color Representation.

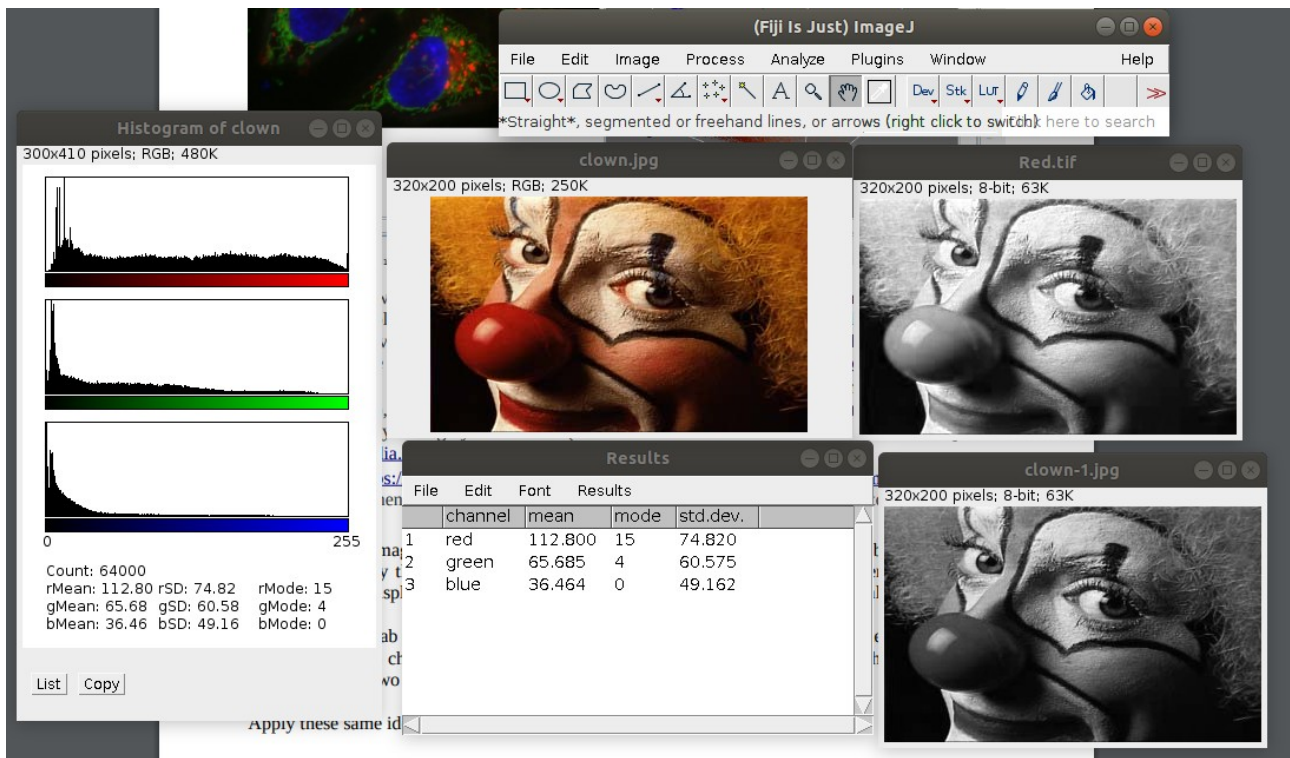


Remember that we opened the **HeLa Cells** sample and that we played with the *Plugins/Color Inspector 3D* tool. With that plugin we clearly see the 3D space in which digital colors live. It adopts the additive principal of human vision where three receptors add up blue, red and green channels to create our color perception (https://en.wikipedia.org/wiki/Color_space).

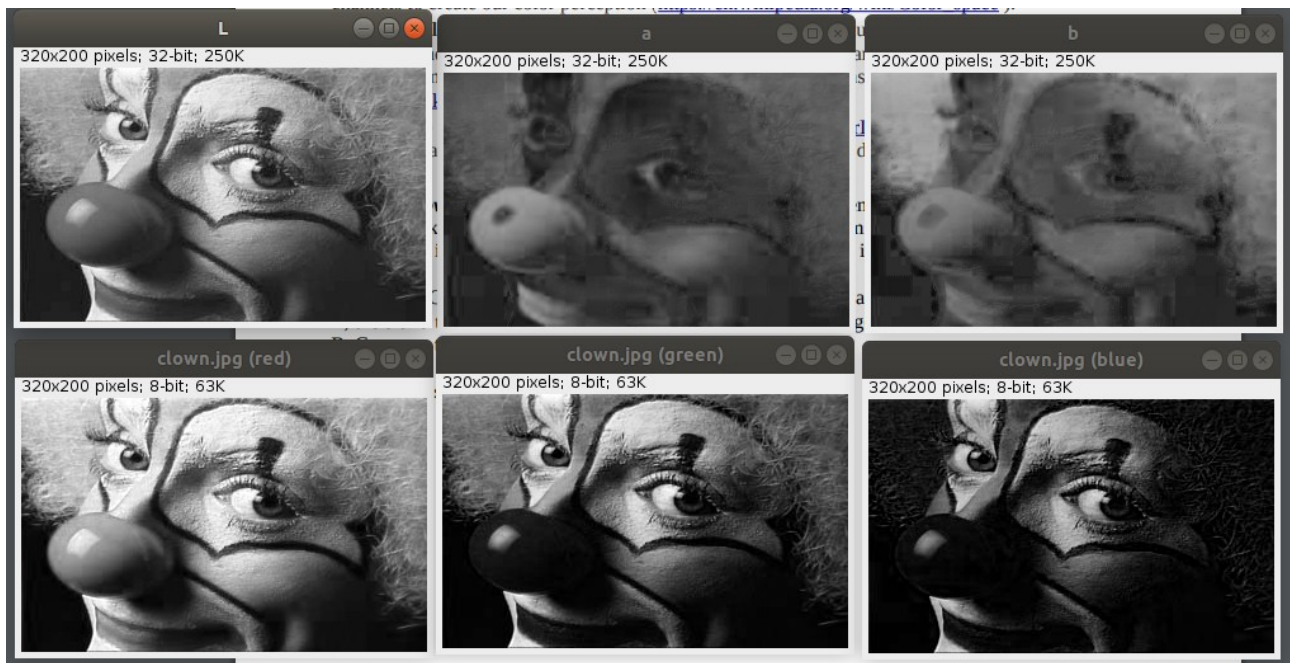
Open the **Clown** image sample. Spend time to explore the plugins : play on the R, G, B level (bottom window), the contrast, brightness etc. It is like you are modifying the color histogram instead of merely the grey level one (or luminance for instance in the Lab color space : https://en.wikipedia.org/wiki/CIELAB_color_space).

Explore the <https://home2.htw-berlin.de/~barthel/ImageJ/ColorInspector/help.htm> documentation extensively and then play with the color space (top left), and the display mode (histogram).

Over the **clown** image again, use the Image/Type/RGBStack menu. What do we obtain ? Save then on your disk only the red channel of the RGB image as a luminance image then (a grey level). Think about it. Display as well the color histogram of the Clown image (Menu Analyze).



Go into the CIELab space (Image/Color/RGB to CIELab). Display the three channels separately the L, the a and the b channels. Split as well the original RGB image into the three channels R, G and B. Compare the two decompositions. Think about it.



Apply these same ideas to the *Mandrill* image.

We go back to the *HeLa Cells* sample. This sample uses the idea of color deconvolution and composite image that we will explore later on for brightfield microscopic stained histopathological images.

For instance in the RGB fluorescent space, put the blue channel at full power. (For instance use Image/Adjust/ColorBalance over the third Blue Channel. You can also play with the Color threshold tool). Display the 3D color histograms (with adjust color channel again if you use the composite image). #Not easy to switch from the composite image to the RGB image

Do the same operation in the RGB space (Press 'I'). It means that out of the composite image where each channel is overlaid we extract the various fluorescent staining by working directly in the RGB original channels (Press Shift z). Here is an extract of the Show Info menu (or press I). Please execute the commands in the text below to understand the various channels of the composite image, the RGB channels and the meaning of color deconvolution in that case.

This is a 16-bits/channel composite color image of HeLa cells with red lysosomes, green mitochondria and blue nucleus. Image courtesy of Tony Collins, creator of the ImageJ for Microscopy collection of plugins at <<http://www.macbiophotonics.ca/imagej/>>.

The scale of 19.42 pixels per micrometer is not accurate. Use the slider under the image to select the channel to be processed or analyzed.

*Note how the color of the subtitle changes when you switch to a different channel. **Open the "Channels" tool by pressing shift-z (Image>Color>Channels Tool) to enable and disable channels → For instance change to the RGB channel instead of the composite image.***

Use the B&C tool (press shift-c) to adjust the brightness and contrast of the current channel.

Change the channel color by selecting a LUT from the Image>Lookup Tables menu.

Press shift-f (Image>Overlay>Flatten) to convert to 24-bit RGB.

Use File>Save As>Tiff to save as a 48-bit TIFF.

The text in the lower right corner of the image is a non-destructive overlay that can be removed by using the Image>Overlay>Remove Overlay command. This text was copied to the image using the Edit>Copy to Image Info command in the ImageJ text editor.

(Fiji Is Just) ImageJ 2.0.0-rc-69/1.53a; Java 1.8.0_172 [64-bit]; Linux 4.15.0-1101-oem; 48MB of 8635MB (<1%)

Title: hela-cells.tif

Width: 34.6034 μm (672)

Height: 26.3645 μm (512)

Size: 2MB

Resolution: 19.4201 pixels per μm

Voxel size: 0.0515x0.0515x0.0515 μm^3

Bits per pixel: 16 (unsigned)

C. Color Deconvolution

We will study a little bit more this idea of colour deconvolution and color representation with the paper by Ruifrok AC, Johnston DA. [Quantification of histological staining by color deconvolution. Anal Quant Cytol Histol 23: 291-299, 2001.](#)

An explanation of this article and the colour_deconvolution plugin in ImageJ/Fiji (Menu Image/Color) can be read there :<https://biii.eu/colour-deconvolution> and <https://blog.bham.ac.uk/intellimic/g-landini-software/colour-deconvolution-2/>

A corresponding code in python is given in **color.py** and **color_decon.py**.

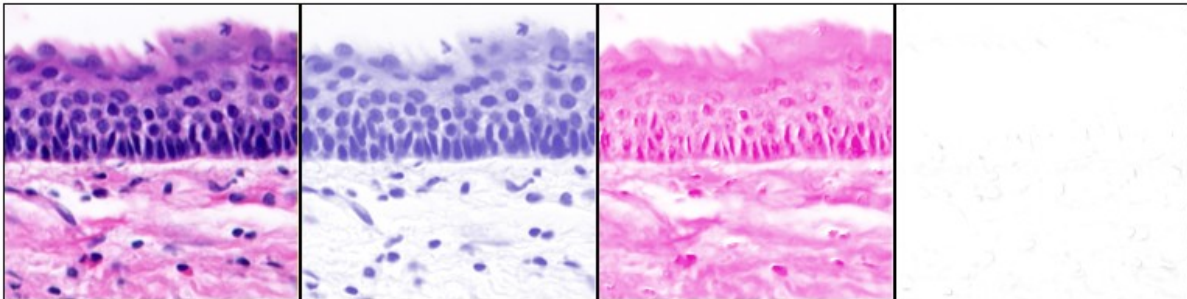
First, with a python command line, reproduce the code step by step using the instructions in **color.py**. Then, you can run the color.py over another image like RNA1.tif

```
$python color.py RNA1.tif
```

This image takes much more time to be processed in python.

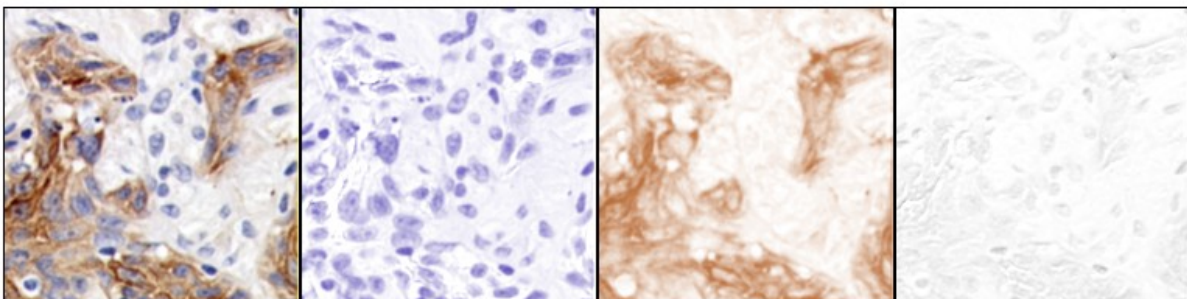
We can modify the matrix of deconvolution in the **color_decon.py** code. So please do your own matrix of deconvolution and observe and analyze it.

Examples



Haematoxylin and Eosin unmixing (using the built-in H&E vectors).

From left to right: original, Haematoxylin, Eosin, virtually empty 3rd (complementary) component (showing that the vectors match the image quite well, except a column of corrupted pixels at the right border of the image).



Haematoxylin and DAB unmixing (using the H DAB built-in vectors).

From left to right: original, Haematoxylin, DAB, 3rd component (the vectors did not perfectly matched the stains in this image, so they should be determined again from single-stained samples).

D. Extra activities :

Working on a 3D cube of color data.

You might have explored the Python example 3D.py in the Repertory **Python** just by running it (check if cv2 is installed ou opencv library for python). This code creates a 3D numpy array and feeds another one with real data from a z-stack the one corresponding to the cochlea volume. The data are in the folder **VolumeData.zip** to unzip in your current working directory.

```
$python 3D.py
```

Spend time understanding the way it works. Then try to answer the questions within the code if not done last time. Then you can use this code to work on 3D cube of colored image by loading each channel as a z slice for instance and then apply filters on each channel before outputting the filtered image in the color space. Or you can try as well to implement different color spaces to understand them better.

Detecting eyes in the mandrill and using the ColorSpace.zip plugin in FiJi.

How to add a plugin in imageJ:

https://imagejdocu.tudor.lu/howto/plugins/how_to_install_a_plugin . The main part is in <http://helios.mi.parisdescartes.fr/~lomn/Cours/CV/TI/Material/Tutorial/Plugins%20Menu.pdf>

Q1. Over the image **DotBlots**, we want to detect the 20 pixels radius circles. For this we need the plugin related to the Hough transform (<https://imagej.nih.gov/ij/plugins/hough-circles.html> improved in https://imagej.net/Hough_Circle_Transform). Install it in FiJi. Use it.

Q2 . Now we need the **ColorSpace.zip** plugin. Install it and try it again with he clown image. We will use the YcbCr space then.. Y captures the luminance and Cr and Cb capture the chromatic information.

Q3. Convert the obtained Stack into 3 images corresponding to the three Y, Cr and Cb channels.

Then try to get the eyes of the mandrill with the Cr channel.

See the document **HoughMandrill.pdf** for the solution.