

# Visualization Options in Fiji

William Giang

2022-11-06

## Table of contents

<b>Selected Visualization Options in Fiji</b>	<b>1</b>
Max intensity projection (typical) . . . . .	1
Adjusting gamma via a look-up table (LUT) . . . . .	2
Adjusting gamma using Visualization Toolset . . . . .	2
Color coding by orientation . . . . .	2
Color coding by radially/non-radially . . . . .	3
Color coding by depth . . . . .	3

## Selected Visualization Options in Fiji

A curated subset of visualization options within Fiji for labmates.

Example dataset: single-channel z-stack with fluorescently tagged keratin-14

We'll cover:

- A common “solution” for seeing fainter structures
- A better solution using gamma for seeing fainter structures
- Color-coding by orientation
- Color-coding by radially
- Color-coding by depth

### Max intensity projection (typical)

First, we'll load a max intensity projection with brightness & contrast settings chosen such that they are pretty much at the min and max values of the image

```
open("C:/Users/will/Documents/2022-11-06_keratin-visualization/MAX_00-keratin-crop_MIP_130_14000.  
orig_name = getTitle();  
setMinAndMax(130, 14000);
```

This 16-bit image has a fair amount of dynamic range, so it's tough to make out the fainter keratin filaments.

One common “solution” is by reducing the maximum value within the Brightness and Contrast window, but then the brighter structures appear blown out.

```
run("Duplicate...", " ");
setMinAndMax(130, 3100);
```

## Adjusting gamma via a look-up table (LUT)

A better solution is to use gamma<sup>1</sup> for adjusting the LUT.

Let's make use of LUTs<sup>2</sup> that incorporate different gamma values.

From left to right: gamma = 0.25, 0.50, 0.75

```
setMinAndMax(130, 14000);

for (i = 25; i < 100; i=i+25){
    run("Duplicate...", " ");
    run("JDM Grays g=0." + i + " ");
}
```

## Adjusting gamma using Visualization Toolset

However, what if you want a gamma of 0.6?

Using the [Visualization Toolset](#) from [@kWolbachia](#), you can freely adjust the gamma on the LUT itself.

By holding the ctrl key and then dragging the mouse side to side, you can easily and quickly find the optimal gamma setting.

## Color coding by orientation

Using [OrientationJ](#), images can be colored by local orientation and coherency.

Left: Original greyscale

Right: Colored in HSB mode where hue is orientation, saturation is coherency, and brightness comes from original image

---

<sup>1</sup>See [here](#) for a nice tweetorial on gamma by [@loicaroyer](#)

<sup>2</sup>Thanks to [JDM\\_LUTs](#) and [NeuroCyto\\_LUTs](#)!

```
selectWindow(orig_name);
run("Duplicate...", " ");
run("OrientationJ Analysis", "tensor=2.0 gradient=0 color-survey=on hsb=on hue=Orientation sat=Co
setMinAndMax(0, 100);
```

## Color coding by radiality/non-radiality

Building on OrientationJ, [@katpyxa](#) created [a macro for splitting an image into radial and non-radial components](#).

Left: Original greyscale

Right: Radial (magenta) and non-radial (cyan) components

## Color coding by depth

ImageJ/Fiji has a “Temporal-Color Code” feature which works fine if eventually you want a Z-projection.

[@katpyxa](#) has a nice [plugin](#) for doing depth color coding without Z-projections and with bonus options.

Astute readers will notice that this coverslip was slightly tilted!

Image: Example of Temporal-Color Code with the Fire LUT and increased contrast

```
open("C:/Users/will/Documents/2022-11-06_keratin-visualization/00-keratin-crop.tif");
run("Temporal-Color Code", "lut=Fire start=1 end=18");
setMinAndMax(0, 100);
```