

# 510\_HW1\_AD

October 2, 2024

## PHYS 510 Homework 1

*Aiden Dillon*

Reading assignment: <https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1001745>

```
[32]: import numpy as np
import matplotlib.pyplot as plt
from skimage import io

[5]: Image_A_png = plt.imread('AuLait_gray.png')
Image_A_tif = plt.imread('AuLait_gray.tif')

[19]: fig, ax = plt.subplots(1, 2, figsize=(15, 12))

ax[0].imshow(Image_A_png, cmap='gray')
ax[0].axis('off')

ax[1].imshow(Image_A_tif, cmap='gray')
ax[1].axis('off')

plt.subplots_adjust(wspace=0.1)
plt.show()
```



The images look qualitatively the same to the eye, but the arrays creating the images are different. The .png has intensity values from 0 to 1, while the .tif has intensity values from 0 to 255. This

is because the png array is composed of 32-bit floats (decimals) and the tiff array is composed of unsigned 8-bit integers.

```
[42]: print('.png min:', np.min(Image_A_png), '|| .tif min:', np.min(Image_A_tif))
      print('.png max:', np.max(Image_A_png), '|| .tif max:', np.max(Image_A_tif))
      print('.png shape:', Image_A_png.shape, '|| .tif shape:', Image_A_tif.shape)
      print('.png datatype:', Image_A_png.dtype, '|| .tif datatype:', Image_A_tif.
            ↪dtype)
```

```
.png min: 0.011764706 || .tif min: 3
.png max: 1.0 || .tif max: 255
.png shape: (1271, 2009) || .tif shape: (1271, 2009)
.png datatype: float32 || .tif datatype: uint8
```

```
[43]: plt.imsave('AuLait_png_out.tif', Image_A_png, format='TIFF')
      io.imsave('AuLait_png_out_sk.tif', Image_A_png)
```

```
[48]: Image_A_out_plt = io.imread('AuLait_png_out.tif')
      Image_A_out_io = io.imread('AuLait_png_out_sk.tif')

      fig, ax = plt.subplots(1, 2, figsize=(15, 12))

      ax[0].imshow(Image_A_out_plt, cmap = 'gray')
      ax[0].axis('off')

      ax[1].imshow(Image_A_out_io, cmap = 'gray')
      ax[1].axis('off')

      plt.subplots_adjust(wspace=0.1)
      plt.show()
```



```
[49]: print('.plt min:', np.min(Image_A_out_plt), '|| .io min:', np.
      ↪min(Image_A_out_io))
      print('.plt max:', np.max(Image_A_out_plt), '|| .io max:', np.
      ↪max(Image_A_out_io))
```

```
print('.plt shape:', Image_A_out_plt.shape, '|| .io shape:', Image_A_out_io.
      ↪shape)
print('.plt datatype:', Image_A_out_plt.dtype, '|| .io datatype:',
      ↪Image_A_out_io.dtype)
```

```
.plt min: 1 || .io min: 0.011764706
.plt max: 255 || .io max: 1.0
.plt shape: (1271, 2009, 4) || .io shape: (1271, 2009)
.plt datatype: uint8 || .io datatype: float32
```

It seems that when saving with matplotlib, it saves the image in color, with RGB values and intensity. It looks like it adds a colormap by default to the image, which I don't like because the original image was grayscale. The intensity scale is 0 to 255 and dtype uint8 which is light. The array is 3D, saving the RGB values and an alpha channel.

Sci-kitlearn saves it as a grayscale image, but it uses a 0 to 1 float32 dtype. It is just a 2D alpha level array.

```
[55]: print(Image_A_out_plt, Image_A_out_io)
```

```
[[[ 56  86 139 255]
   [ 54  90 140 255]
   [ 52  94 141 255]
   ...
   [ 57  85 139 255]
   [ 57  85 139 255]
   [ 56  86 139 255]]

[[[ 56  86 139 255]
   [ 54  90 140 255]
   [ 52  94 141 255]
   ...
   [ 56  86 139 255]
   [ 56  86 139 255]
   [ 57  85 139 255]]

[[[ 57  85 139 255]
   [ 55  89 140 255]
   [ 53  93 140 255]
   ...
   [ 55  88 140 255]
   [ 56  87 140 255]
   [ 56  86 139 255]]

...

[[[ 44 113 142 255]
   [ 44 113 142 255]
   [ 44 112 142 255]
```

```

...
[ 31 163 134 255]
[ 31 163 134 255]
[ 31 161 135 255]]

[[ 43 115 142 255]
 [ 44 114 142 255]
 [ 44 113 142 255]
...
 [ 31 163 134 255]
 [ 31 163 134 255]
 [ 30 160 135 255]]

[[ 43 115 142 255]
 [ 44 114 142 255]
 [ 44 112 142 255]
...
 [ 31 162 134 255]
 [ 31 162 134 255]
 [ 31 161 135 255]]] [[0.27450982 0.2901961 0.30588236 ... 0.27058825
0.27058825 0.27450982]
[0.27450982 0.2901961 0.30588236 ... 0.27450982 0.27450982 0.27058825]
[0.27058825 0.28627452 0.3019608 ... 0.28235295 0.2784314 0.27450982]
...
[0.38039216 0.38039216 0.3764706 ... 0.58431375 0.58431375 0.5764706 ]
[0.3882353 0.38431373 0.38039216 ... 0.58431375 0.58431375 0.57254905]
[0.3882353 0.38431373 0.3764706 ... 0.5803922 0.5803922 0.5764706 ]]

```