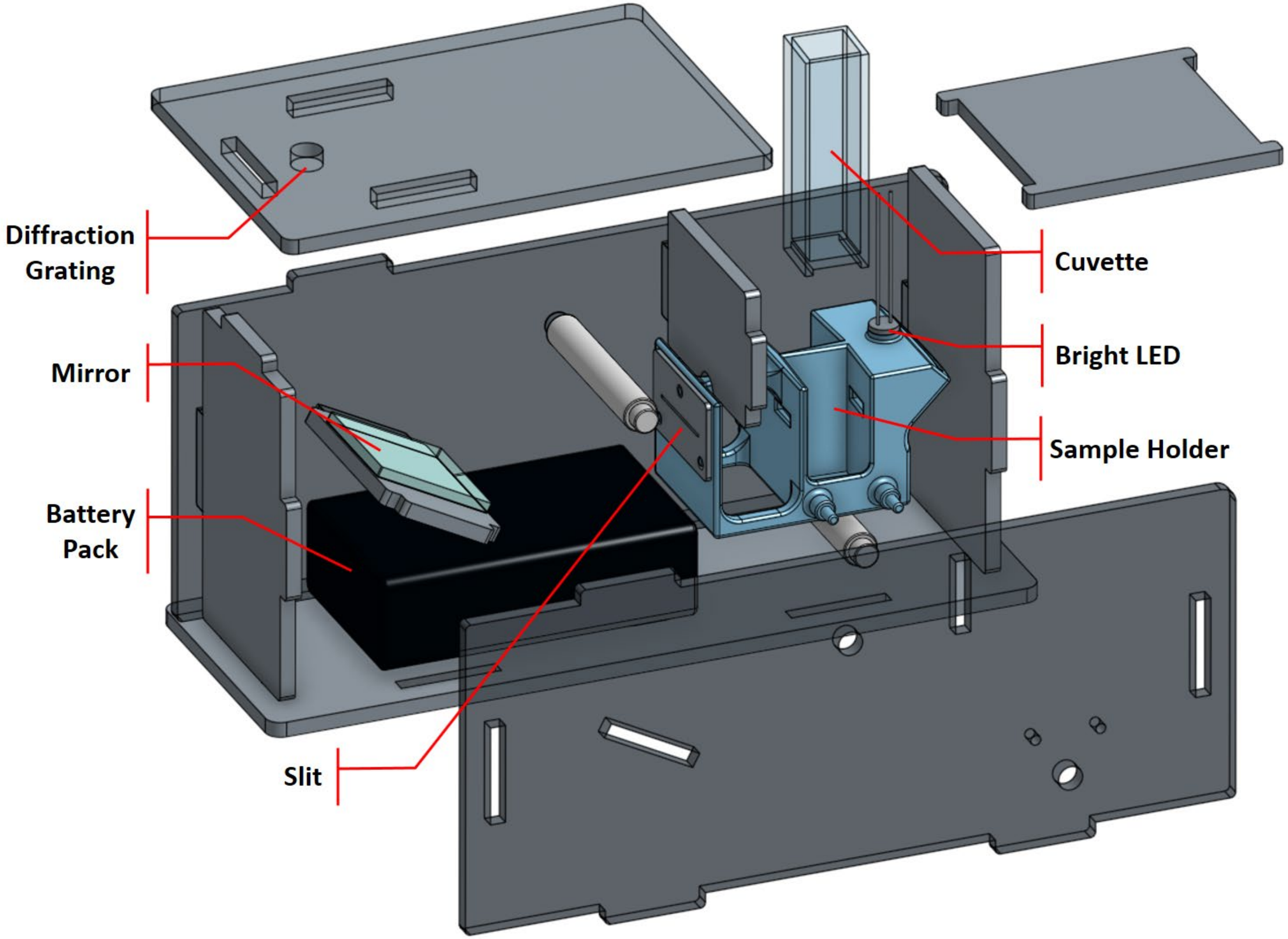
EOSpec Image Analysis

JFeng

GitHub repo: /jianshengfeng/EOSpec_v01

EOSpec / EOS1

- EOSpec / EOS1: an open-source spectrometer designed for measuring nitrate concentration in water
- The EOSpec Hardware: DIY spectrometer, requires 3D printer (FormLabs), laser cutter (Epilog), and supplies



EOSpec / EOS1

- Operating Procedure (if a calibration is available):
  - ✓ Take a water sample, mix reagents.
  - ✓ Take a spectra picture using the EOSpec.
  - ✓ Run software to calculate nitrate concentration.

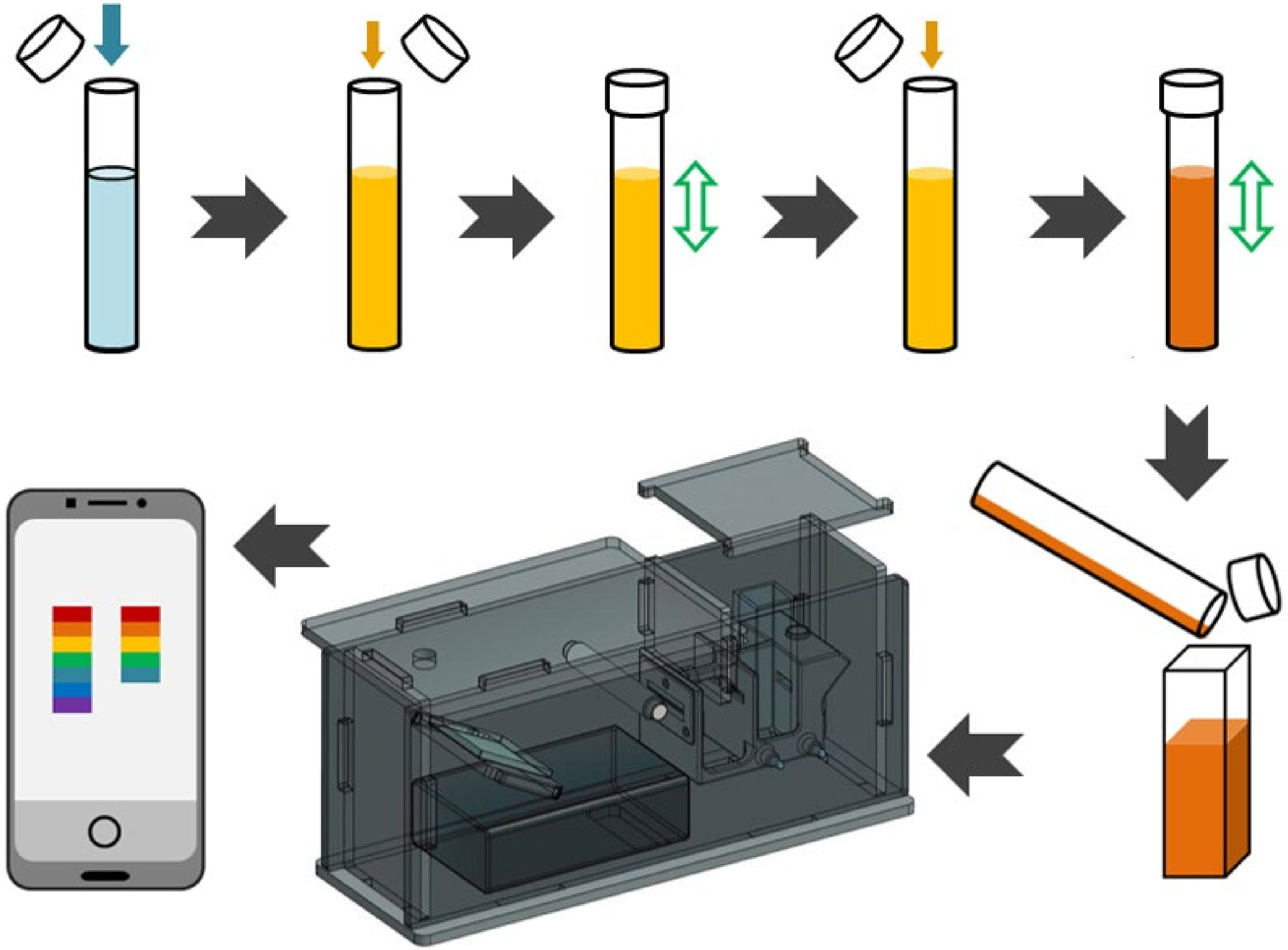
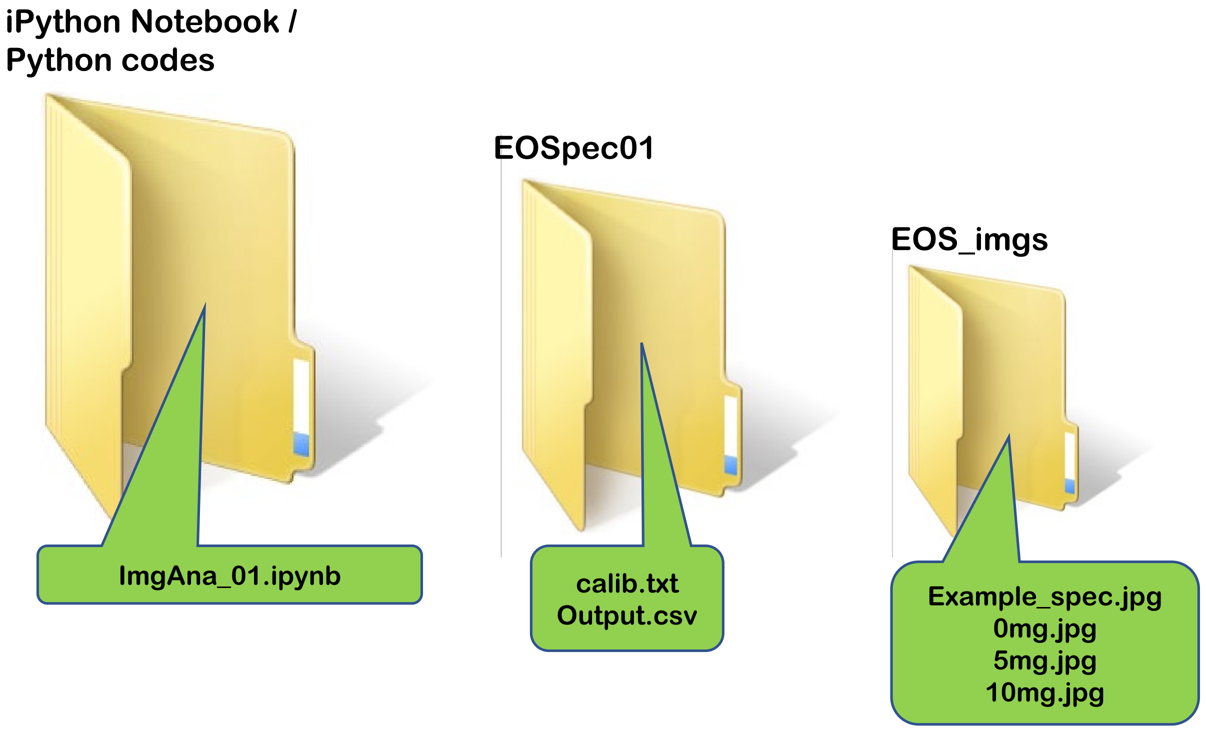


Image Analysis Software

- Python version: 2.7.15 | Anaconda
- ImgAna_csv_output: an open-source iPythonNotebook script for EOSpec image analysis _ csv output
- Folder Structure: …\iPythonNotebook\EOSpec01\EOS_imgs\
    - \iPythonNotebook\: folder containing Python/iPythonNotebook scripts
    - \EOSpec01\: folder containing calibration files specific to a spectrometer (hardware)
    - \EOS_imgs\: folder containing spec pics



iPython Notebook /
Python codes

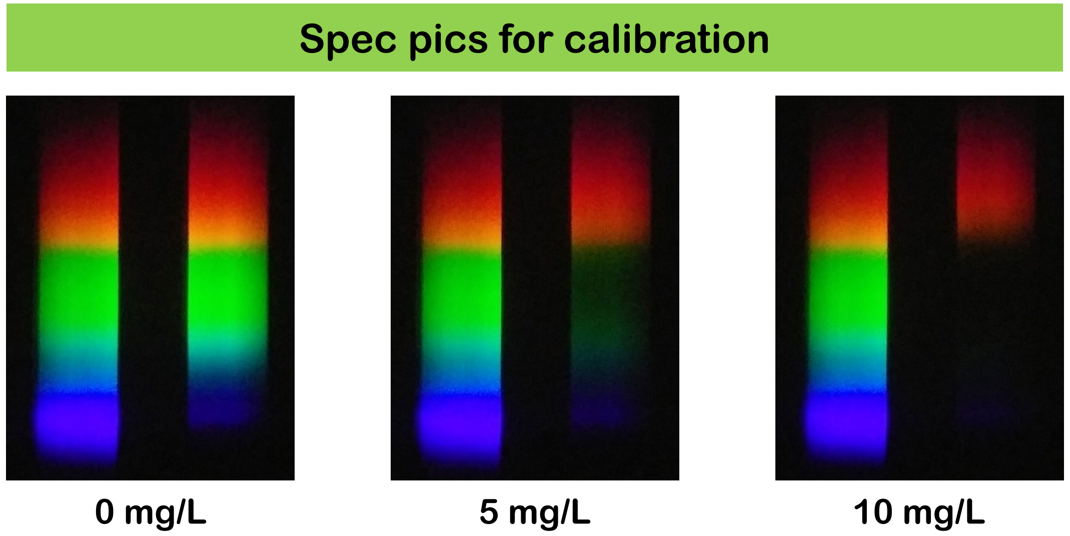
ImgAna_01.ipynb

EOSpec01

calib.txt
Output.csv

EOS_imgs

Example_spec.jpg
0mg.jpg
5mg.jpg
10mg.jpg



Spec pics for calibration

0 mg/L

5 mg/L

10 mg/L

Function for Displaying Images and Making Figures

- Function fig_out() – *see iPythonNotebook script*
- Images in Python are represented by NumPy arrays, can be int (8-bit) or float (0.0 – 1.0), Height × Width × 3(RGB)
- First try a simple figure with 8 "pixels", int or float work the same:

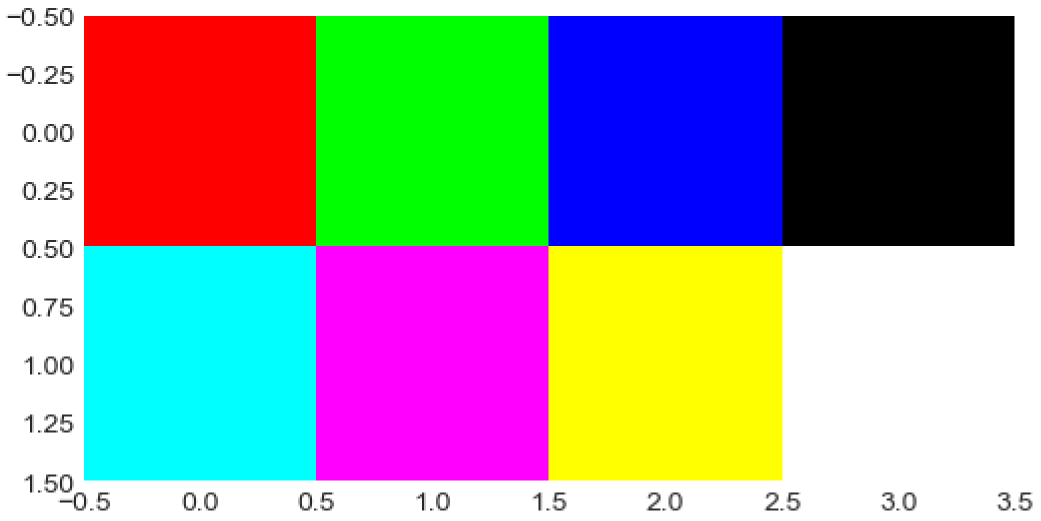
Red	Green	Blue	Black
Cyan	Purple	Yellow	White

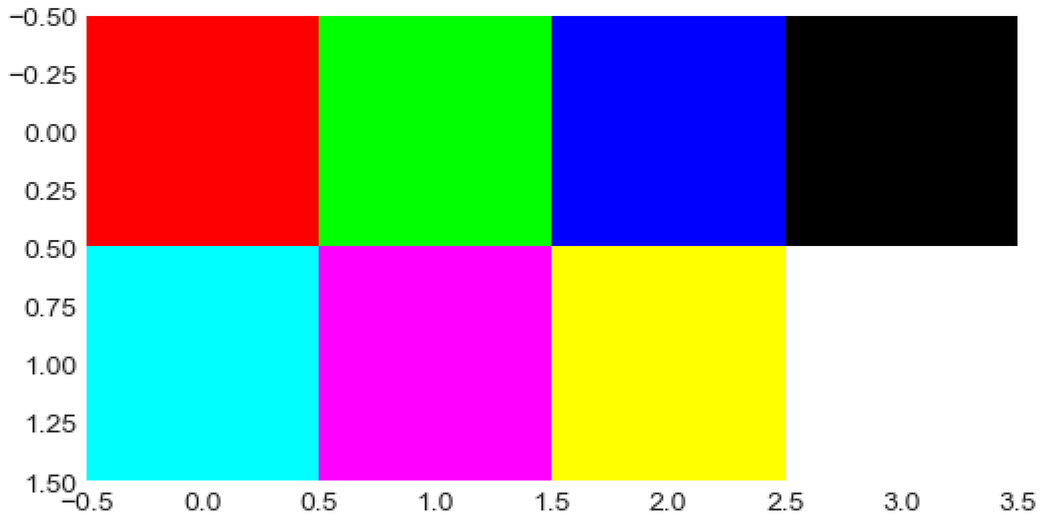
xi8 (int array)			
255, 0, 0	0, 255, 0	0, 0, 255	0, 0, 0
0, 255, 255	255, 0, 255	255, 255, 0	255, 255, 255

x8 (float array)			
1.0, 0.0, 0.0	0.0, 1.0, 0.0	0.0, 0.0, 1.0	0.0, 0.0, 0.0
0.0, 1.0, 1.0	1.0, 0.0, 1.0	1.0, 1.0, 0.0	1.0, 1.0, 1.0

```
xi8 = np.array([[[255,0,0], [0,255,0], [0,0,255], [0,0,0]],
                [[0,255,255], [255,0,255], [255,255,0], [255,255,255]]])
```

```
x8 = np.array([[[1.0, 0.0, 0.0], [0.0, 1.0, 0.0], [0.0, 0.0, 1.0], [0.0, 0.0, 0.0]],
               [[0.0, 1.0, 1.0], [1.0, 0.0, 1.0], [1.0, 1.0, 0.0], [1.0, 1.0, 1.0]]])
```





Reading in and Displaying a Spec Pic

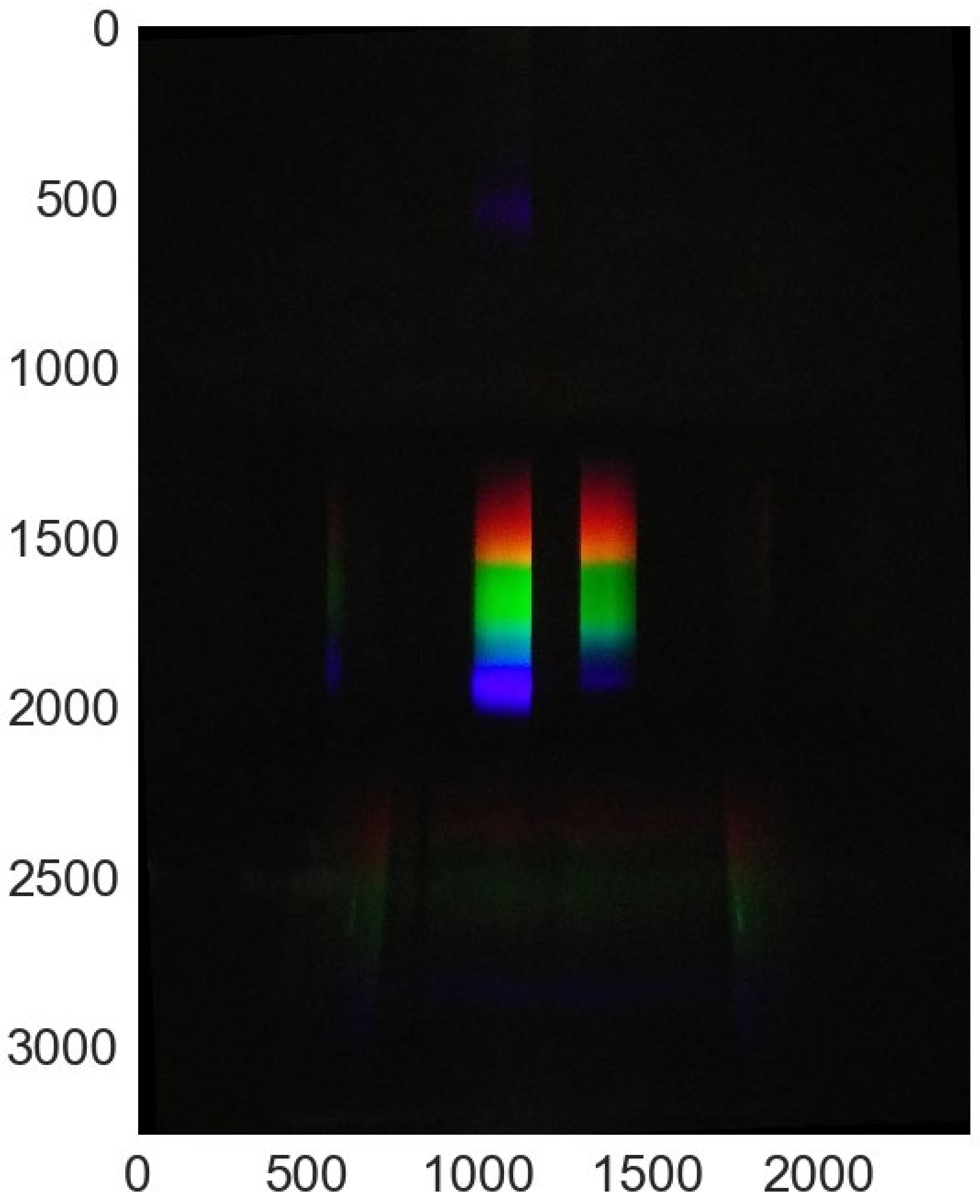
- Read in an image to a NumPy array:
  - JPG’s are often int (8-bit) arrays
  - PNG’s are often float arrays
- Convert the input image array to a float array (if not already)

```
img_file = "EOSpec01/EOS_imgs/example_spec.jpg"
x_img = pp.imread( img_file )
np.amax(xi)
>>>  255
```

```
x_img = xi.astype( float )/255.0
fig_out( x_img, 200 )
```

```
np.shape(x_img)
>>>  (3264L, 2448L, 3L)
```

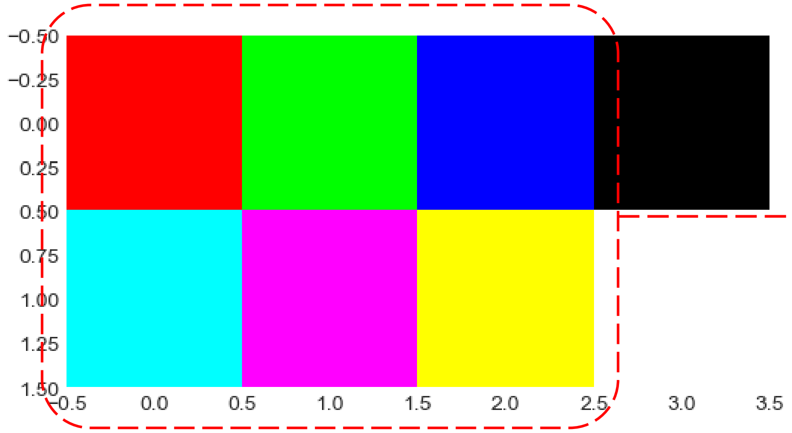
```
x_img.dtype
>>>  dtype(‘float64’)
```



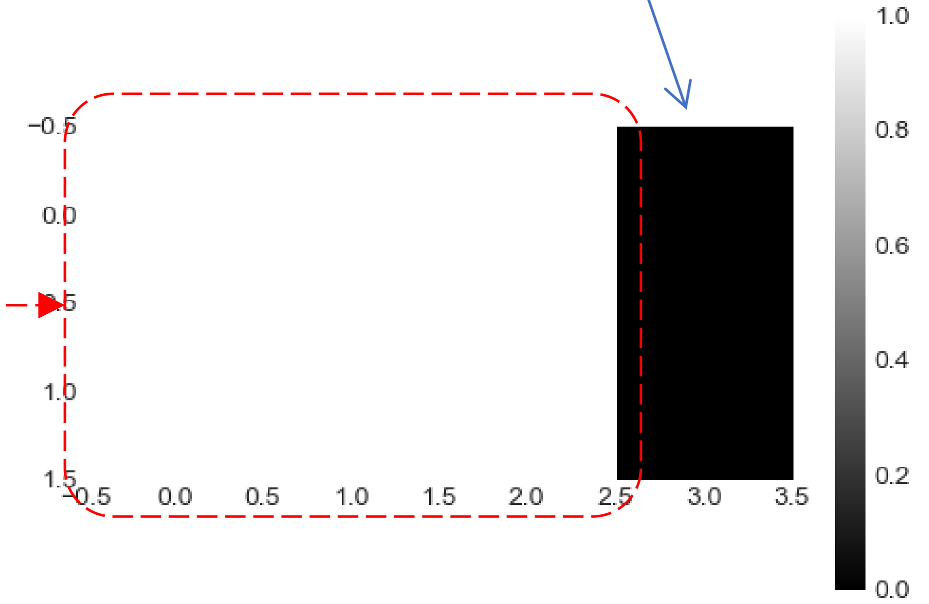
Function for Calculating Heat Map

- Function cal_heatmap() – *see iPythonNotebook script*
- The color_diff_sum metric ($|R - G| + |R - B| + |G - B|$) is used to rank the colorfulness/vibrancy of the pixels. It highlights bright colors while suppresses white, black, and grayscale ($R = G = B$), as demonstrated below:

```
cal_heatmap( x8, True )
>>>  array([[1., 1., 1., 0.],
            [1., 1., 1., 0.]])
```



Colorful pixels are highlighted



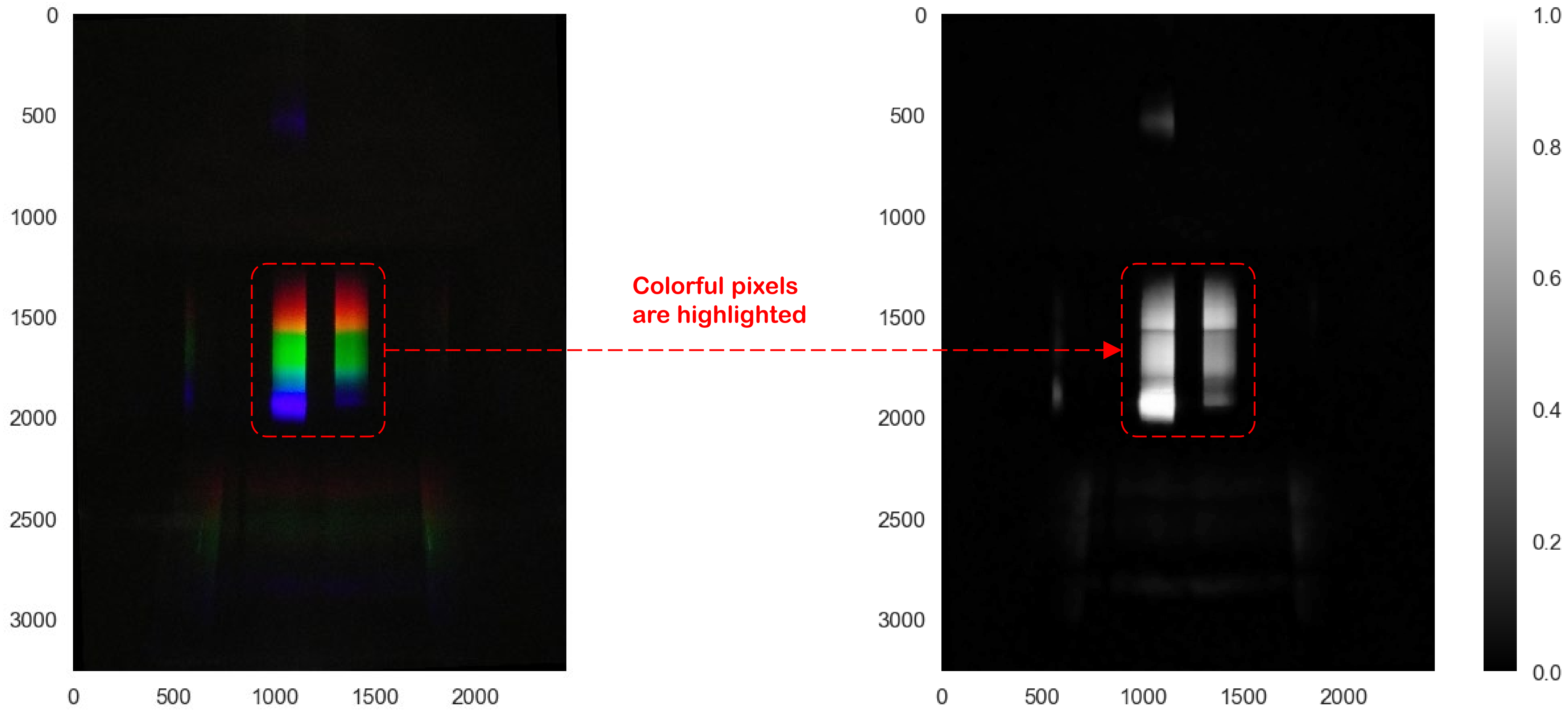
Black, white, grayscale pixels are suppressed

Highlighting the Spectra

- Function fig_out() – *see iPythonNotebook script*
- First try a simple figure (represented by an array): can be int (8-bit) or float (0.0 – 1.0), Height × Width × 3(RGB)

```
fig_out( x_img, 120 )
```

```
hm = cal_heatmap( x_img, True )
```



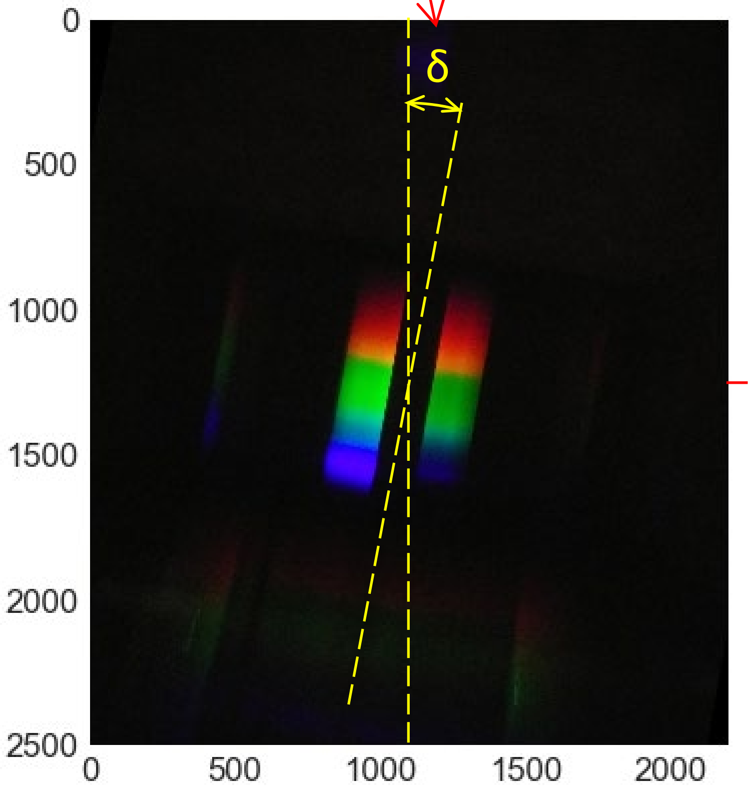
Colorful pixels are highlighted

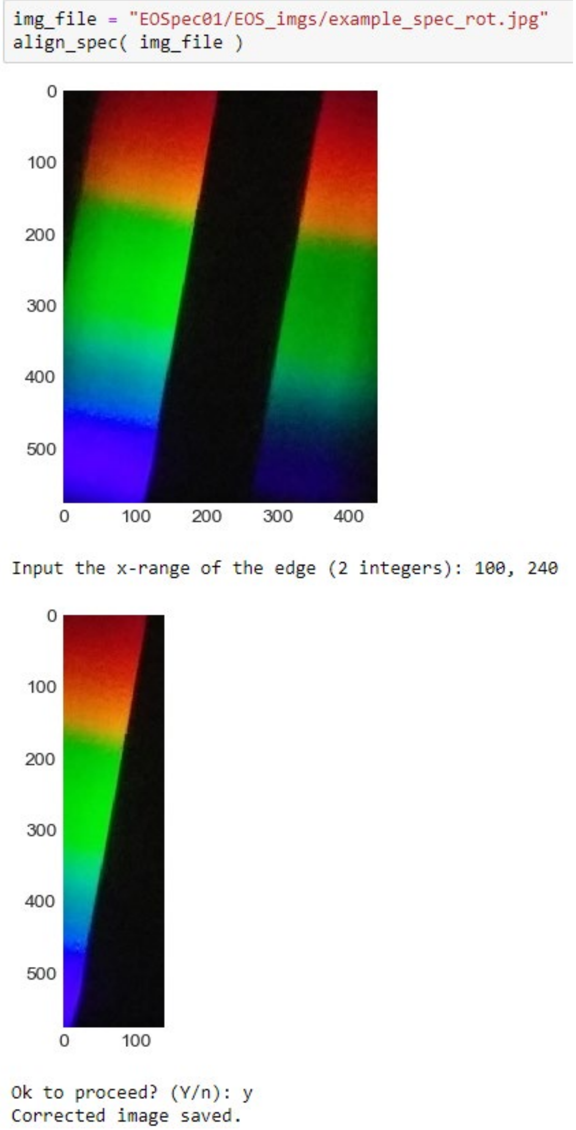
Correcting Alignment of the Spectra

- If the phone camera is not aligned with the EOSpec, the spectra will be tilted. This misalignment should be corrected.
- Misalignment of the spectra can be caused by factors involving construction of the EOSpec, and that cannot be corrected by this procedure
- Function align_spec() – *see iPythonNotebook script*
- Alignment corrected image is automatically saved

```
img_file = "EOSpec01/EOS_imgs/example_spec_rot.jpg"
align_spec( img_file )
>>>  Input the x-range of the edge (2 integers): 100, 240
>>>  Ok to proceed? (Y/n): y
>>>  9.73052940904
>>>  Corrected image save
```

Spectra are tilted wrt the orientation of the spec pic

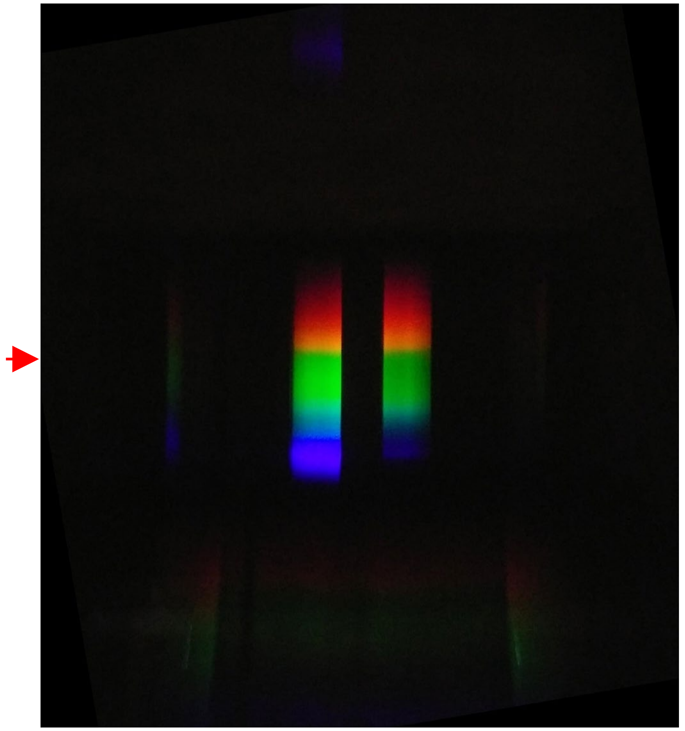




```
img_file = "EOSpec01/EOS_imgs/example_spec_rot.jpg"
align_spec( img_file )
```

```
Input the x-range of the edge (2 integers): 100, 240
```

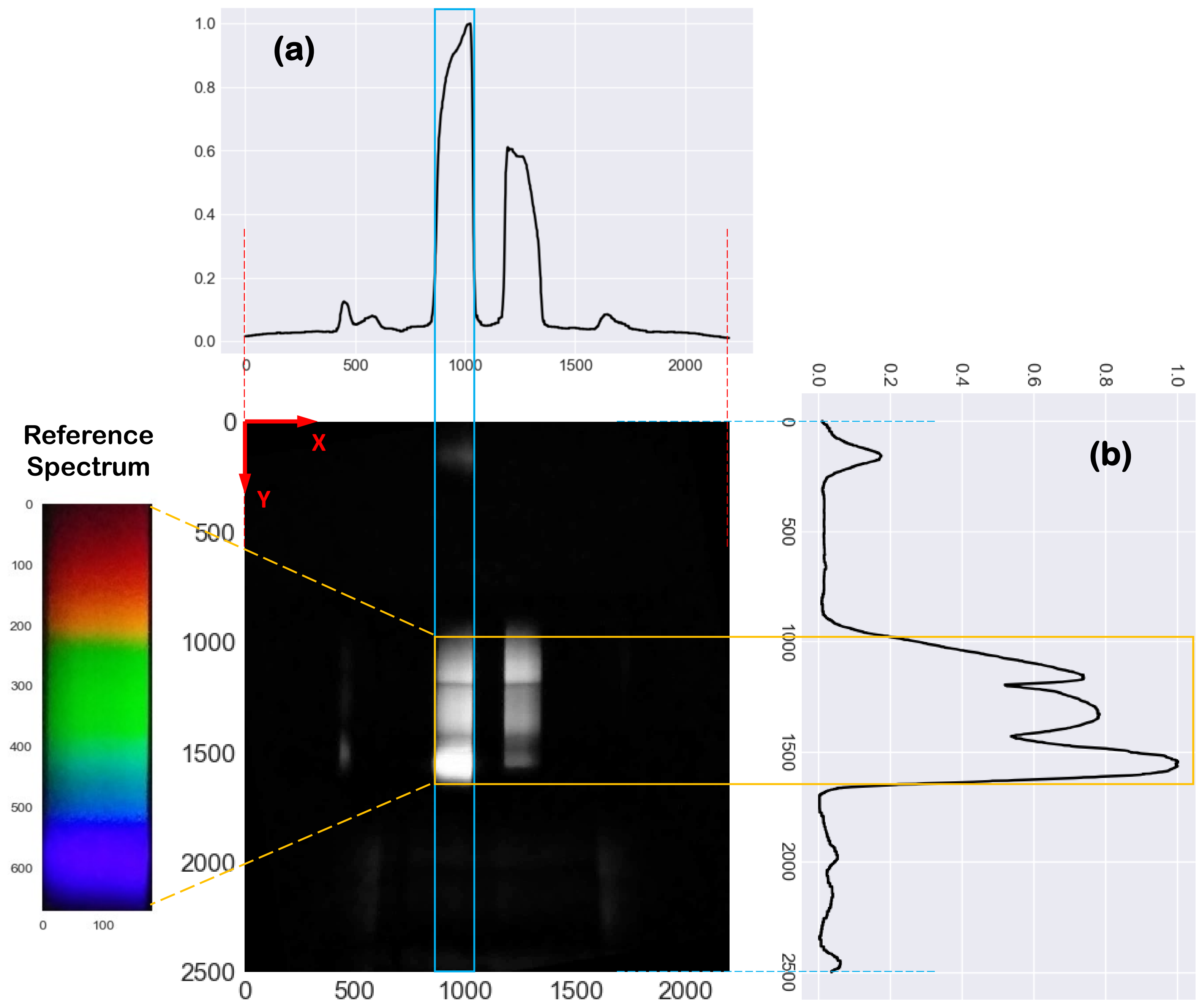
```
Ok to proceed? (Y/n): y
Corrected image saved.
```



Locating the Reference Spectrum

- Function find_ref() – *see iPythonNotebook script*
  1) Collapse along the Y-axis, Fig (a)
  2) Use a preset threshold, *n* = 0.2, to find left and right edges of the reference spectrum
  3) Collapse the vertical band containing the reference spectrum, Fig (b)
  4) Use preset threshold (*n* = 0.2) again to find top and bottom edges of the reference spectrum

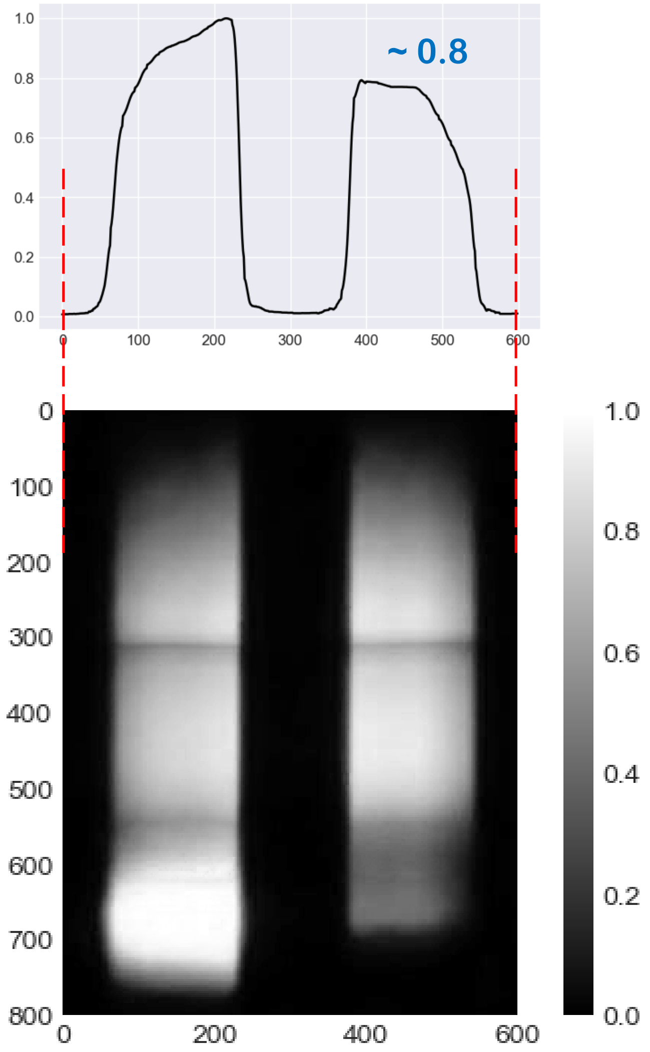
```
img_file = "EOSpec01/EOS_imgs/example_spec_rot_crt.jpg"
xi = pp.imread( img_file )
x_img = xi.astype( float )/255.0
hm = cal_heatmap( x_img, True )
top, btm, lft, rgt, gap = find_ref( hm, True )
```



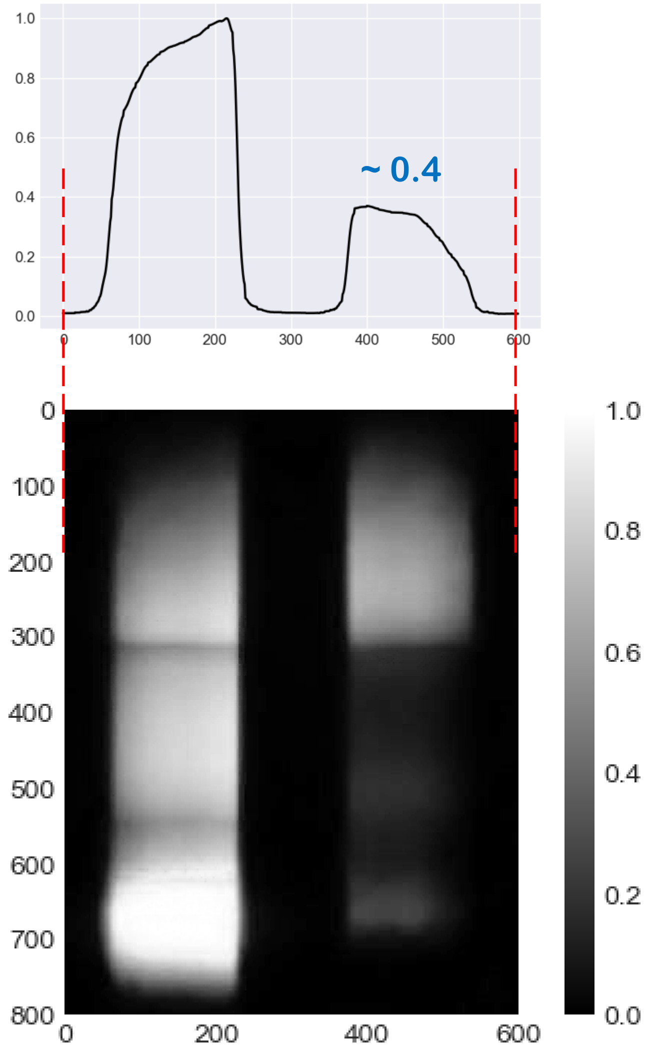
Locating the Sample Spectrum

- Unlike the Reference Spectrum which should be full (using water without any reagent as reference), the Sample Spectrum can be significantly diminished (e.g., for high-concentration samples). For example:

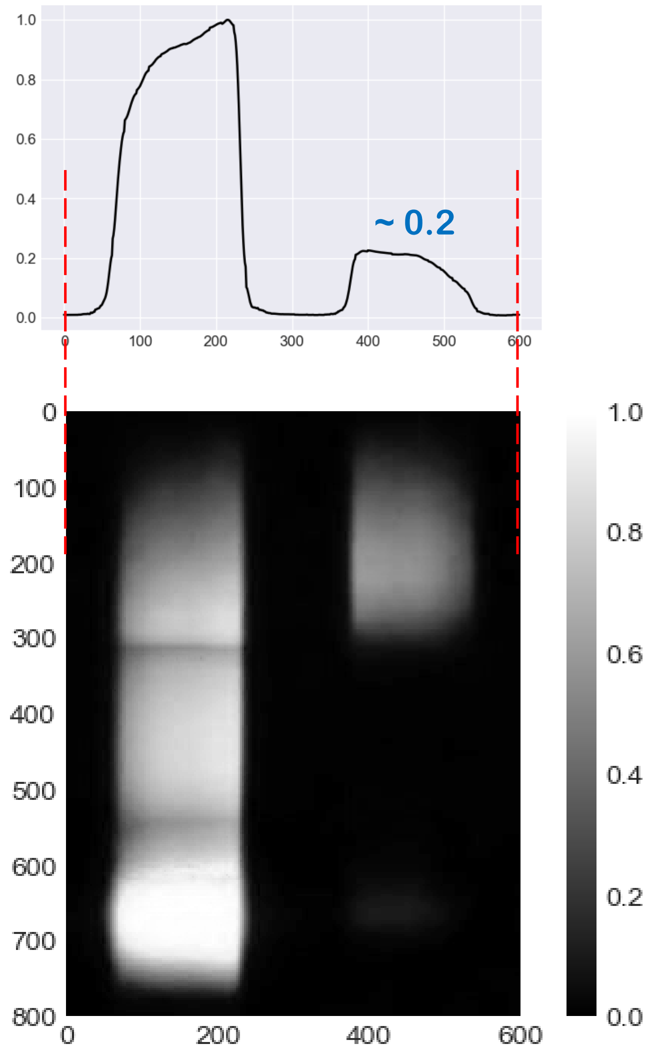
0 mg/mL



5 mg/mL

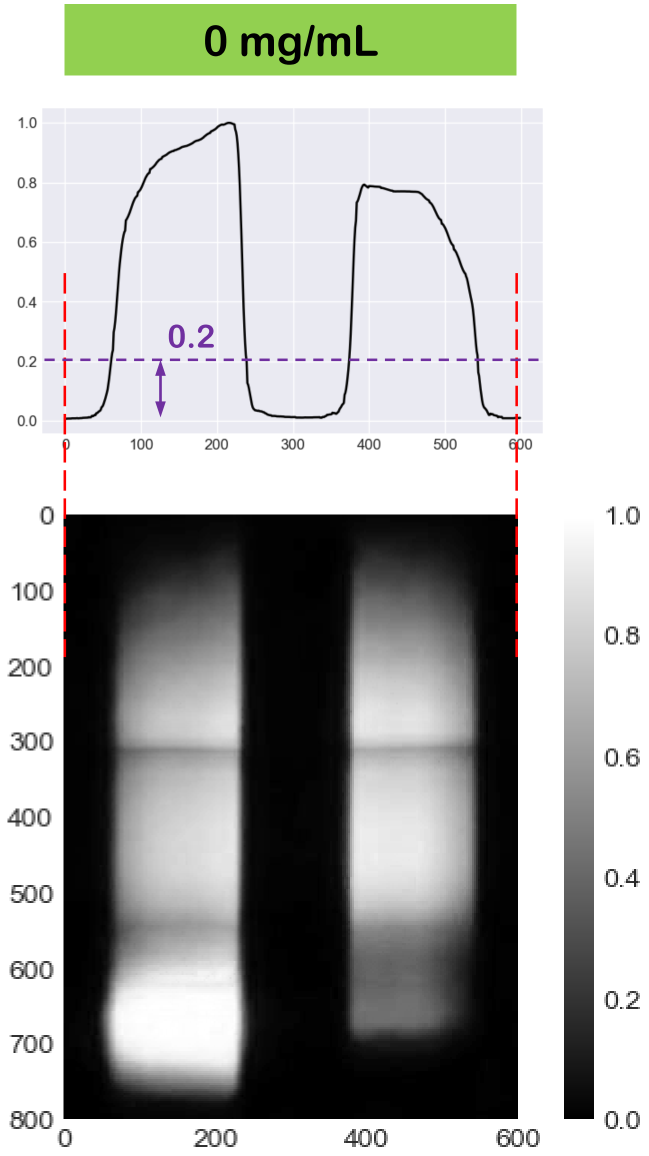


10 mg/mL

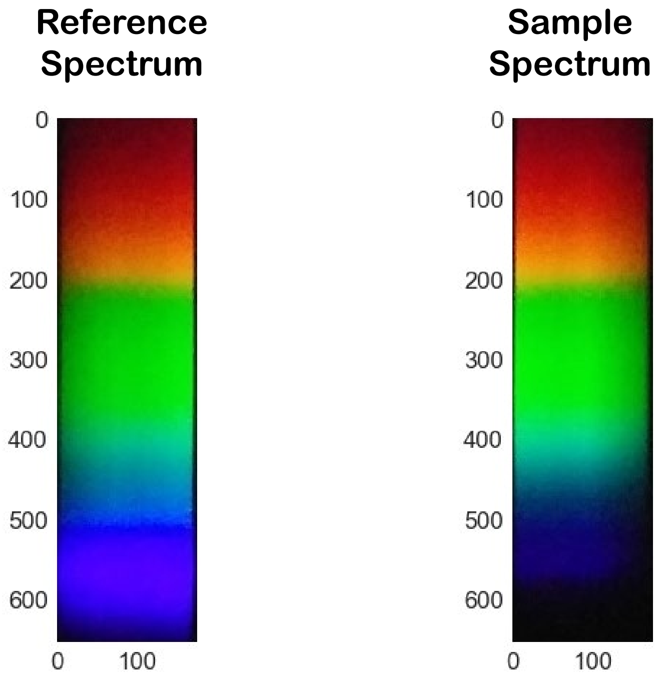


Locating the Sample Spectrum

- **Use the 0 mg/mL (i.e., blank) sample to locate the Sample Spectrum**
- **For the 0 mg/mL spec pic: the reference spec is water (no color), the sample spec is water + reagent (yellow)**
- **Store the gap value in the calibration file**



```
top0, btm0, lft0, rgt0, gap0 = find_ref( hm0, fo=True, find_gap=True )
print top0, btm0, lft0, rgt0, gap0
```



```
cali_file = "EOSpec01/calib.txt"
save_cali( cali_file, gap=gap0 )
```

\EOSpec01\calib.txt

```
gap, wavelength, intensity, k_fit, b_fit:
136
0
0
0
0
```

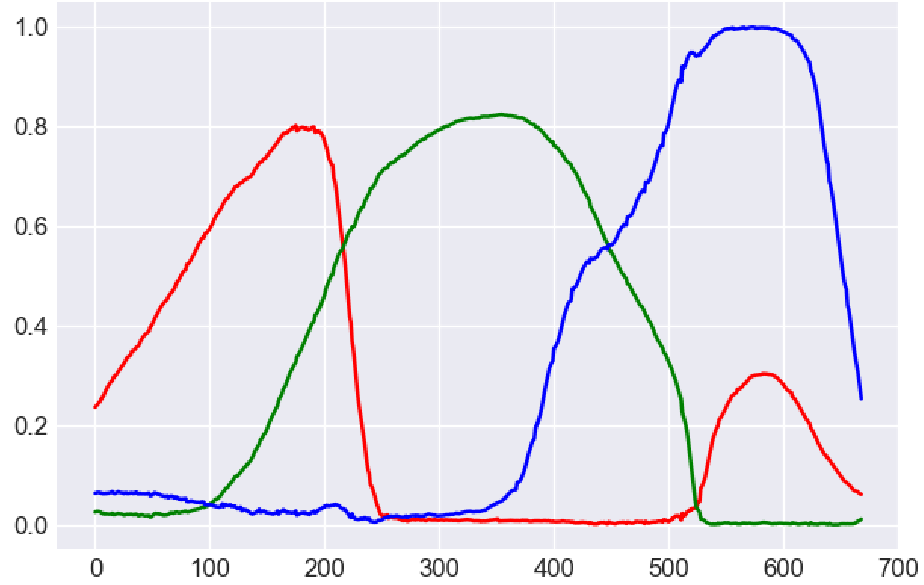
Normalizing the Sample Spectrum

- **Function norm_sam() – *see iPythonNotebook script***
- **Locate the Sample Spectrum using the Reference Spectrum and the gap value**

```
full_result = norm_sam( x0_img, fo=True, csv_out="EOSpec01/output.csv" )
wv, sam_norm = full_result[0]
ref_raw, sam_raw = full_result[1]
other_result = full_result[2]
```

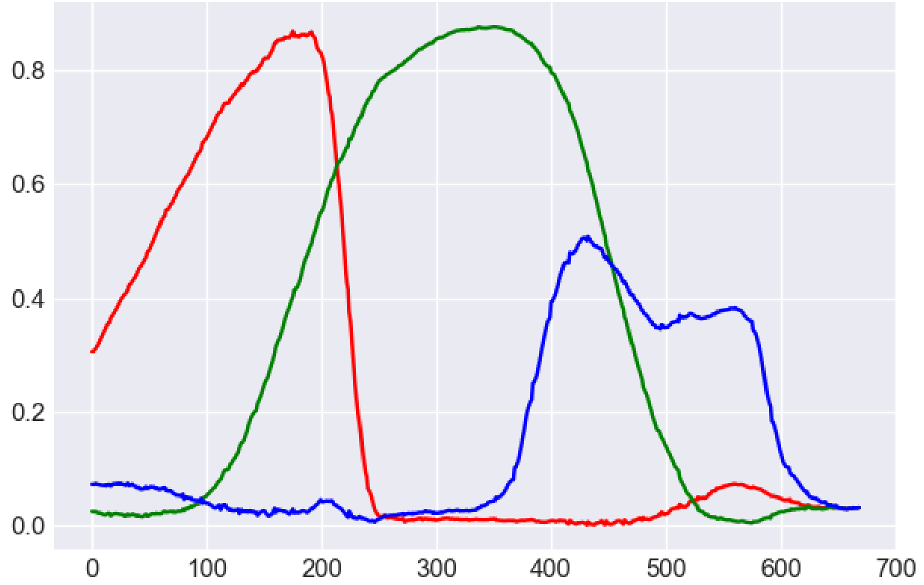
```
fig_out( ref_raw )
```

Ref. Spec., collapsed, raw



```
fig_out( sam_raw )
```

Sam. Spec., collapsed, raw

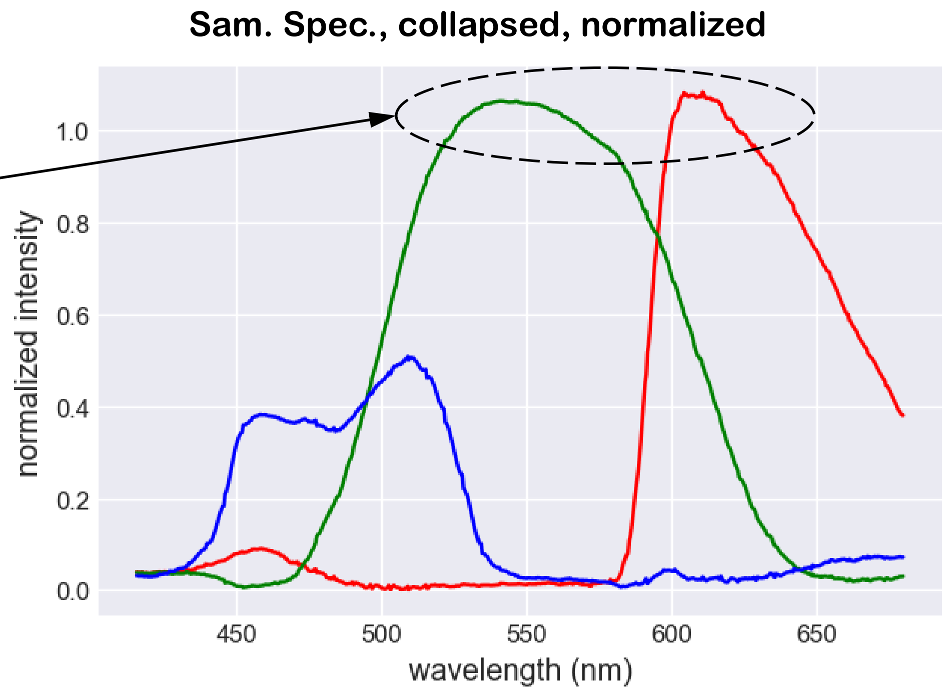


Normalizing the Sample Spectrum

- **Normalize each (RGB) channel in the Sample Spectrum by the max value of the respective channel in the Reference Spectrum**
- **Create csv output**

wavelength	sam_spec_norm_R	sam_spec_norm_G	sam_spec_norm_B
675.1908088	0.440981595	0.023308989	0.070233452
674.7948529	0.447081507	0.025288376	0.071731389
674.3988971	0.450341805	0.026516961	0.072974959
674.0029412	0.450832603	0.025902669	0.072607541
673.6069853	0.462050833	0.020612927	0.070176926
673.2110294	0.464609991	0.022114531	0.071731389
672.8150735	0.468641543	0.023957409	0.073879374
672.4191176	0.473234005	0.024947103	0.073144537
. . .	. . .	. . .	. . .

```
pp.figure( dpi=120 )
pp.style.use( vseaborn-darkgrid" )
pp.plot( wv, sam_norm[:,0], 'r-' )
pp.plot( wv, sam_norm[:,1], 'g-' )
pp.plot( wv, sam_norm[:,2], 'b-' )
pp.xlabel( "wavelength (nm)", size=12 )
pp.ylabel( "normalized intensity", size=12 )
```



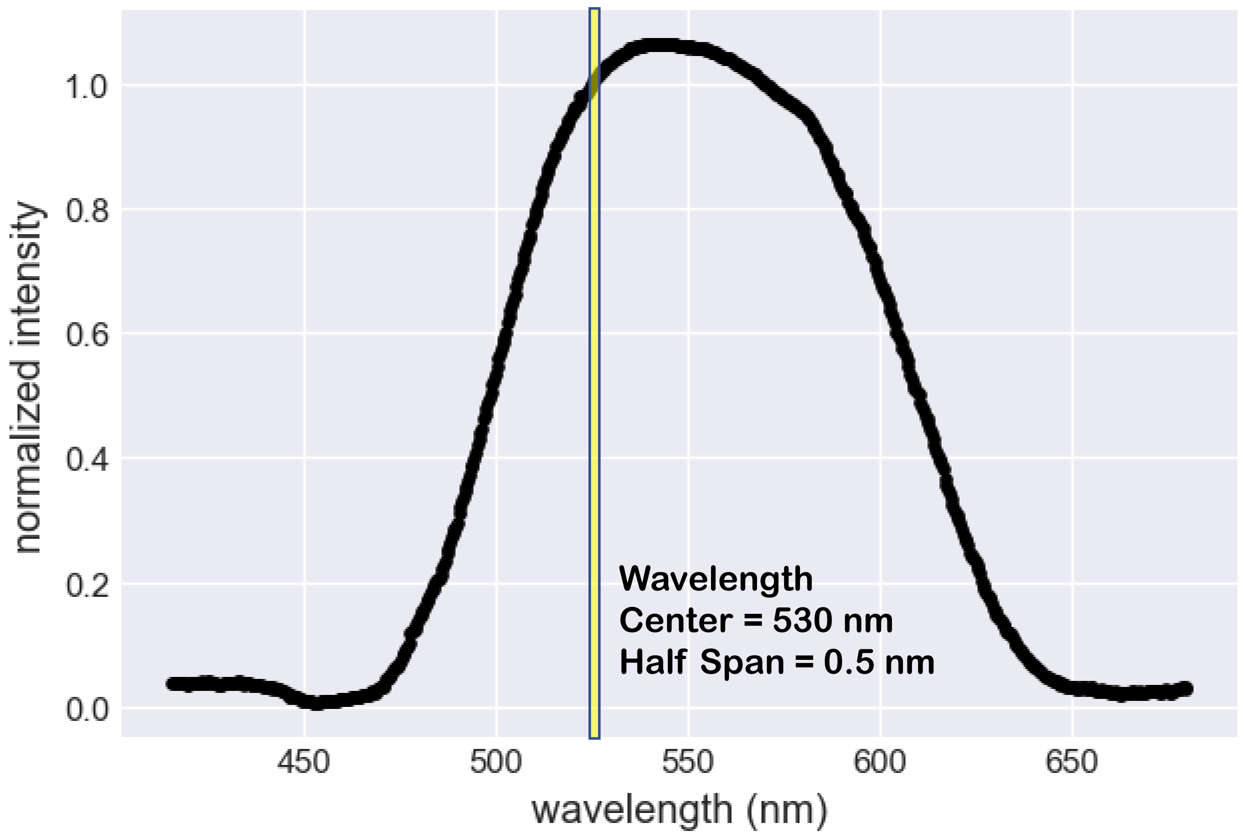
Depending on the construction of the EOSpec, the blank Sam. Spec. intensity maybe higher than the Ref. Spec. intensity, resulting in norm. Sam. Spec. > 1.0

Calculating Average Intensity over a Narrow Band

- Function cal_I() – *see iPythonNotebook script*
- Store wavelength and intensity of the blank in calibration file

```
cal_I( img_file, ch='g', wlc=530., wlhs=.5, fo=True )
>>>  [530.27095588  529.875]
      [1.03579284  1.03322669]
>>>  1.034509764713725
```

```
cali_file = "EOSpec01/calib.txt"
bk0 = [530.0, 1.034509764713725]
save_cali( cali_file, blank=bk0 )
```



Wavelength
Center = 530 nm
Half Span = 0.5 nm

\EOSpec01\calib.txt

```
gap, wavelength, intensity, k_fit, b_fit:
136
530.0
1.03450976471
0
0
```

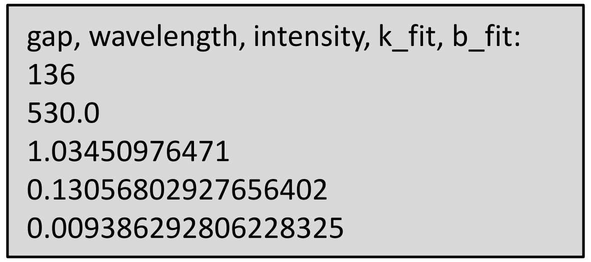
Calibrate and Run Nitrate Tests

- **Function cali_N(), test_N() – *see iPythonNotebook script***
- **Store slope and intercept of the fit in calibration file**

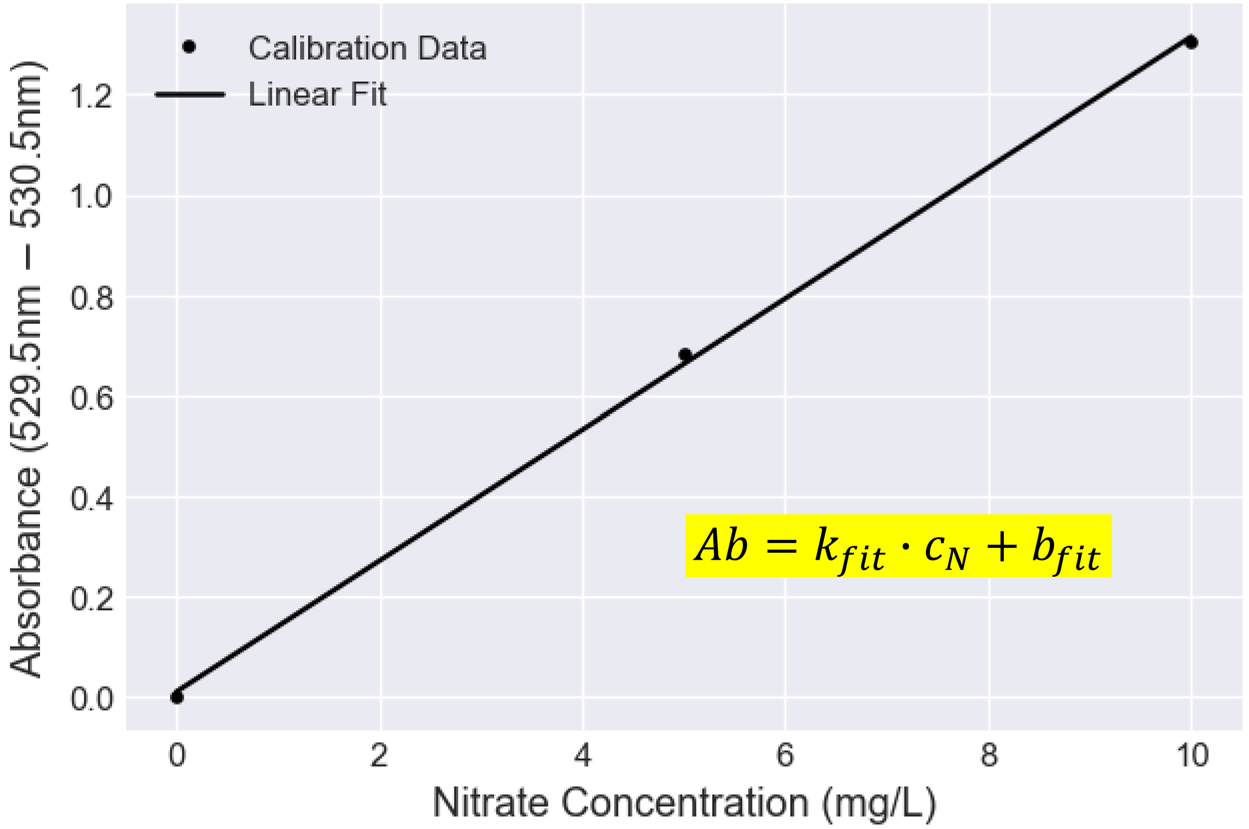
```
imgs = ["EOSpec01/EOS_imgs/0mg.jpg",
          "EOSpec01/EOS_imgs/5mg.jpg",
          "EOSpec01/EOS_imgs/10mg.jpg"]
ncs = [0.0, 5.0, 10.0]
Kb_fit, nc, lgI = cali_N( imgs, ncs, 530. )
```

```
cali_file = "EOSpec01/calib.txt"
save_cali( cali_file, kb=kb_fit )
```

\EOSpec01\calib.txt



```
gap, wavelength, intensity, k_fit, b_fit:
136
530.0
1.03450976471
0.13056802927656402
0.009386292806228325
```



$$Ab = k_{fit} \cdot c_N + b_{fit}$$

```
img_file = "EOSpec01/EOS_imgs/example_spec_rot_crt.jpg"
>>>  Nitrate Concentration: 1.3 mg/L
>>>  1.3017753805295134
```