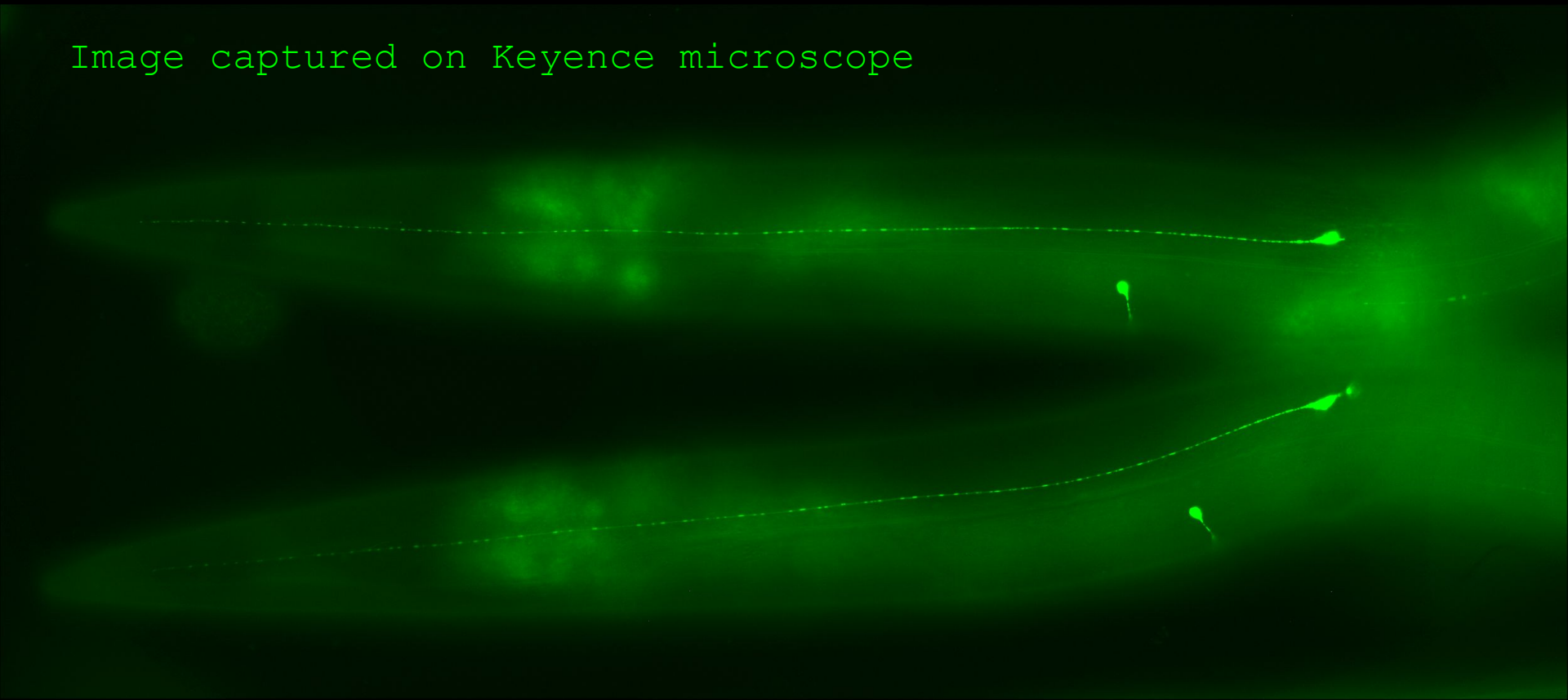


Goals

- Extract information about puncta position, intensity, width etc. from still images of touch receptor neurons in *C. elegans*.
- Quantify differences in these parameters between WT and mutant strains.
- Marker used - `mec-17p::mNeonGreen::MEC-4`

Example raw image

Image captured on Keyence microscope



Pre-processing

Manually draw a line along the neurite on ImageJ



Pre-processing

Straighten the line using built-in function in ImageJ



This straightened image is the input for the Python script

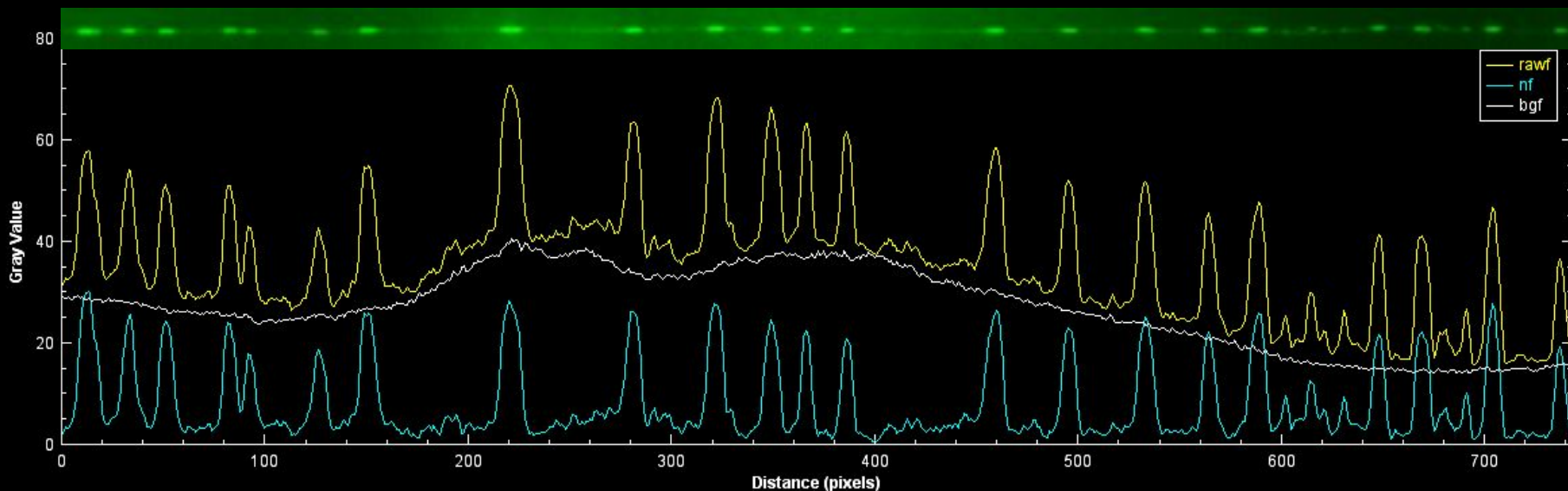
```
img = imageio.imread(fpath+x)[:,:,:1]      #import image and store it in a list of lists
```



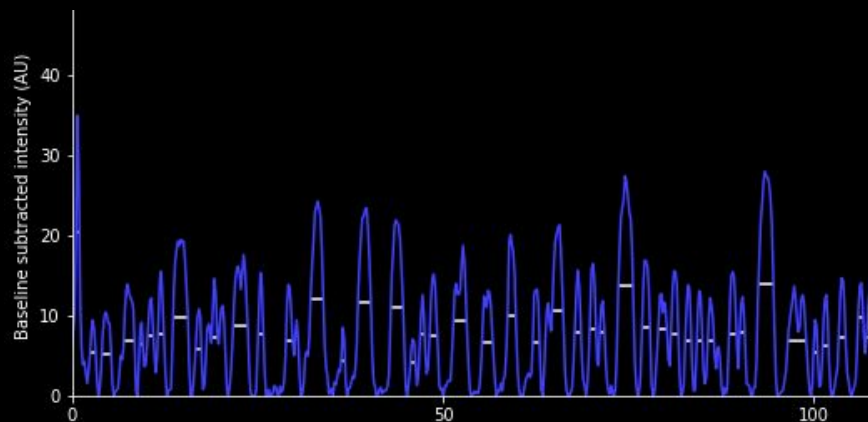
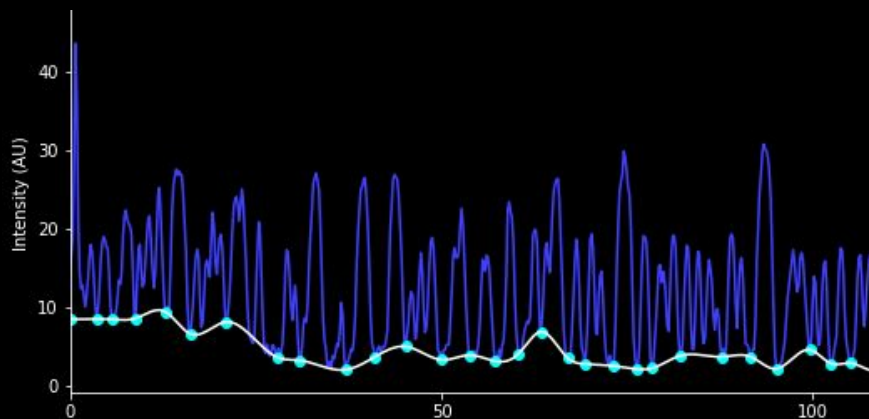
Zoomed in view of the above image

Subtracting background from the neurite

```
n = img[8:13, 0:]          #extract rows to use for neurite
bg = np.concatenate((img[0:6, 0:], img[15: , 0:])) #extract rows to use for background
rawf = np.mean(n, axis=0)   #calculate average raw neurite fluorescence
bgf = np.mean(bg, axis=0)   #calculate average background fluorescence
nf = rawf - bgf             #calculate background subtracted neurite fluorescence
```



Estimating the baseline fluorescence



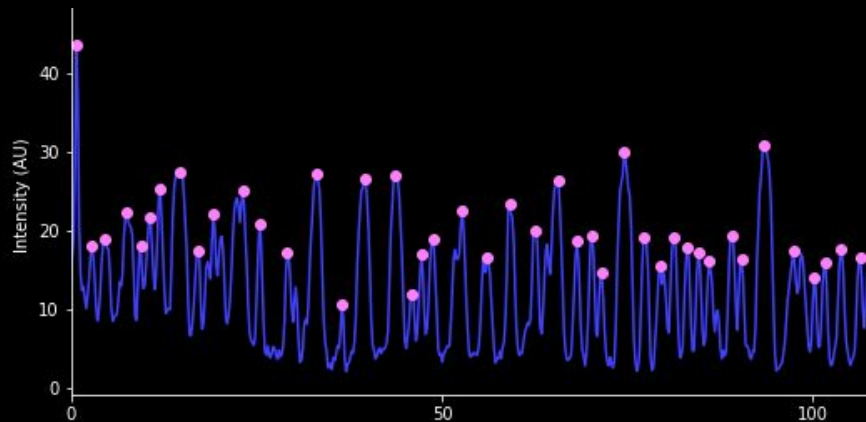
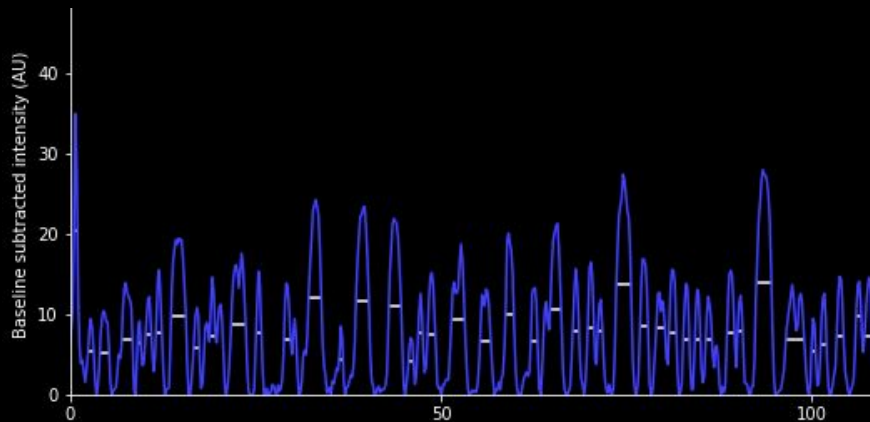
```
#finding baseline
minsp = np.concatenate((0, argrelmin(nf, order=10)[0], imsize[1]), axis=None)
minsh = np.concatenate((nf[minsp[1]], [nf[i] for i in minsp[1:-1]], nf[minsp[-2]]), axis=None)

#set negative minsh values to zero
for i in range(len(minsh)):
    if minsh[i]<0: minsh[i]=0
cs = CubicSpline(minsp, minsh, extrapolate=False)(d)    #fit cubic spline through minima points

#baseline subtracted trace with no negative values
bsub = nf-cs
for i in range(len(bsub)):
    if bsub[i]<0: bsub[i]=0
```

Finding peaks from baseline-subtracted trace

```
#find peaks
peaks = find_peaks(bsub, height=0, prominence=5, width=0, rel_height=0.5)
pd = peaks[0]*mu_per_px
pmi = [nf[i] for i in peaks[0]]
```



Example traces for different strains

