chromatogram_custom

September 1, 2022

[ ]:
```
from chromatography import load_snapshot_file
```

[ ]:
```
data = load_snapshot_file('./raw_data/bin_text/2020-08-bin-test.txt')

data['Sample Name'].unique()
```

[ ]:
```
array(['20200617-630034_S1', '20200617-630034_S2', '20200617-630034_S3',
       '20200617-630034_S4', '20200617-630034_S5', '20200617-630034_S6',
       'IM054_S1', 'IM054_S2', 'IM054_S3', 'IM054_S4', 'IM054_S5',
       'IM054_S6'], dtype=object)
```

[ ]:
```
import pandas as pd


def format_data(data, sample_name=None):
    '''Return data each size and heigh by row
        | Allele1 | Allel2 | Size1 | Size 2 | Height1 | Height2 |

        ------------------------------
        | Allele  | Size  | Height   |
        ------------------------------
          Allele1 | Size1 | Height 1 |
          Allele2 | Size2 | Height 2 |
    '''
    # selected colums
    cols = ['Marker', 'Allele 1', 'Allele 2', 'Size 1', 'Size 2', 'Height 1',␣
 ↪'Height 2']

    if sample_name:
        tmp = data[data['Sample Name'] == sample_name]
    else:
        tmp = data.copy()

    # reformat table to long format

    df = pd.wide_to_long(tmp[cols],
                        stubnames=['Allele', 'Size', "Height"],
                        i='Marker',
                        j = 'AlleleNumber', sep=" ").reset_index().
↪sort_values('Marker')
    df = df[pd.notna(df.Size)].sort_values(['Marker', 'AlleleNumber']).
↪reset_index(drop=True)

    return df
```

[ ]:
```
format_data(data, '20200617-630034_S1')
```

[ ]:
```
         Marker  AlleleNumber Allele   Size  Height
0    CYP2D6_001             1      G  30.73  2846.0
1    CYP2D6_002             1      C  34.57  1207.0
2    CYP2D6_002             2      T  36.89  2205.0
3    CYP2D6_003             1      T  40.67  1644.0
4    CYP2D6_004             1      G  43.91  4716.0
5    CYP2D6_005             1      G  48.26  1600.0
6    CYP2D6_006             1      G  52.72  3252.0
7    CYP2D6_007             1      G  55.09  3712.0
8    CYP2D6_008             1      C  59.64  1911.0
9    CYP2D6_008             2      T  60.57  2802.0
10   CYP2D6_009             1      G  66.09  3366.0
11   CYP2D6_010             1      A  71.96  5408.0
```

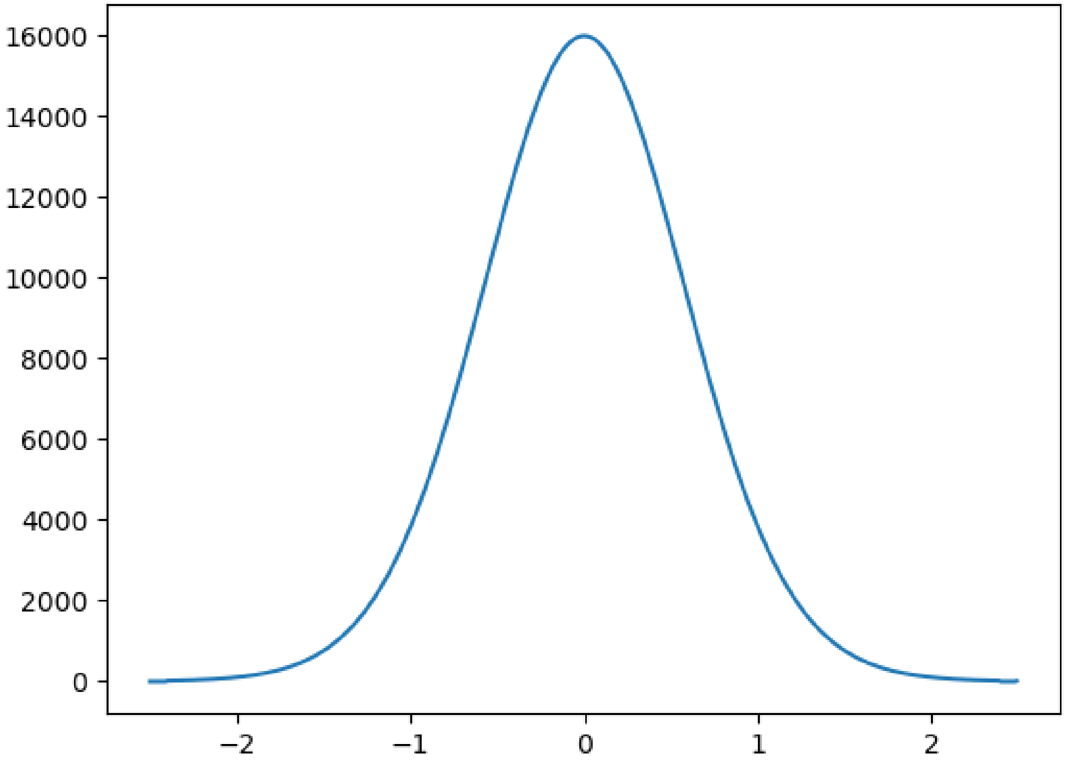
[ ]:
```
from turtle import width
import numpy as np
def generate_intensity(size, height):

    '''Generate intensity base on height of each allele'''

    # generate width based on height
    if height < 15000:
        range = 2.2

    else:
        range = 2.5

    width = np.arange(-1*range,range,0.01)

    intensity = height*np.exp(-15*np.log(1+(width**2)/10))

    # set 100 first and last intensity = 0
    intensity[0] = 0
    intensity[-1] = 0

    return width, intensity
```

[ ]:

```
import matplotlib.pyplot as plt
x, y = generate_intensity(30, 16000)

plt.plot(x, y)
```

[ ]:

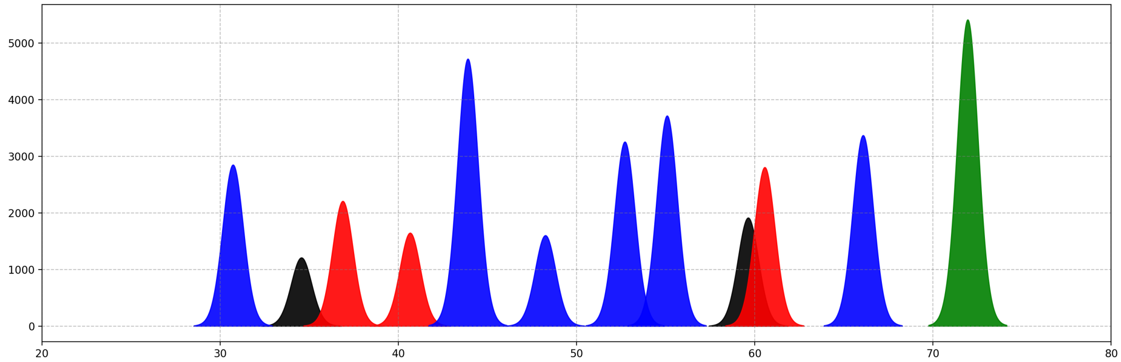
```
[<matplotlib.lines.Line2D at 0x7f6e6c0969d0>]
```



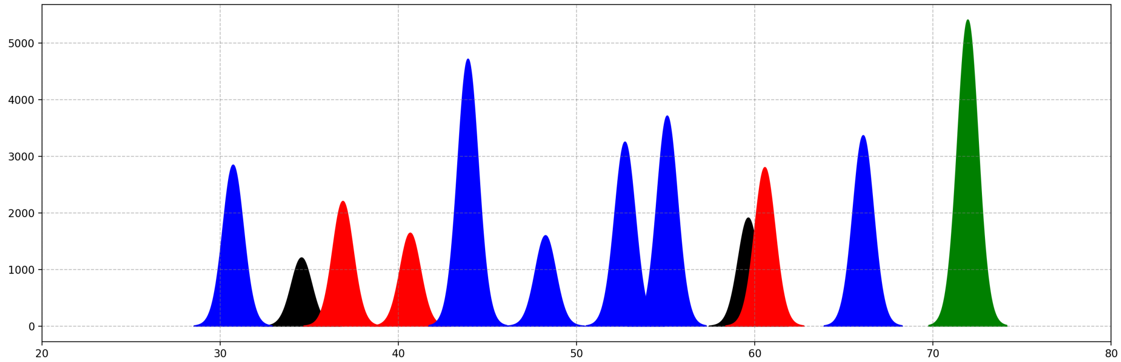
[ ]:

```
ex = format_data(data, '20200617-630034_S1')

def plot(data, xlim =(20,80),
         figsize =(15,5),
         dpi = 200,
         color_map = {'A':'green', 'G':'blue', 'T':'red', 'C':'black'},
         alpha = 0.9):

    fig, ax = plt.subplots(figsize=figsize, dpi=dpi)
    ax.set_xlim(*xlim)
    ax.grid(linestyle='--', alpha = 0.5, color ='gray')

    for idx, row in data.sort_values('Height').iterrows():

        width, y = generate_intensity(row.Size, row.Height)
        x = width + row.Size

        ax.fill(x, y, color=color_map.get(row.Allele), alpha=alpha)

    plt.tight_layout()
    plt.show()
```

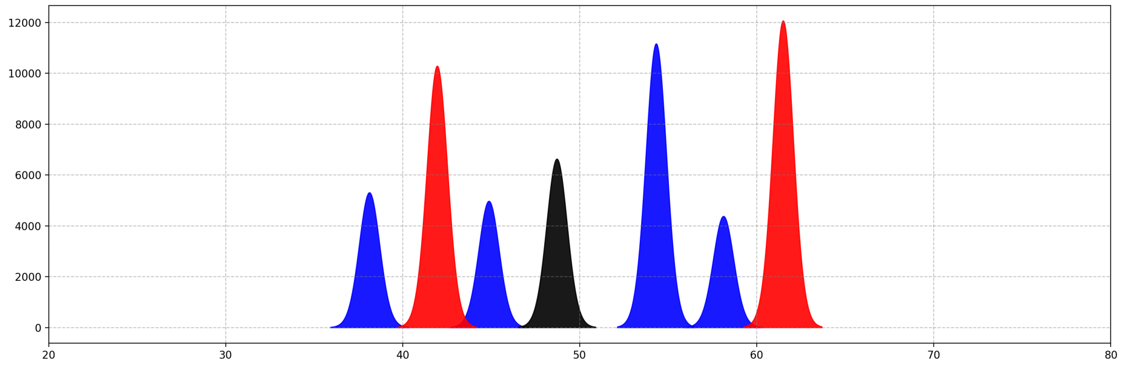
```
[ ]: plot(ex)
```



```
[ ]: plot(ex, alpha=1)
```



```
[ ]: plot(format_data(data, 'IM054_S3'))
```


12000
10000
8000
6000
4000
2000
0
20
30
40
50
60
70
80