Data Visualization Project

Three alternative visualizations of the same artificial data shall be re-created. All three visualizations show the same fictitious genomic annotations together with fictitious RNA binding protein data. The visualizations are an example for RNA binding protein signals as well as the genomic annotations. Recreate each of the shown figures. Two different datasets are provided for this task:

10_project_data_annotations.csv

10_project_data_signals.csv

The 10_project_data_annotations.csv file contains fictitious genomic information as visualized in all bottom panels of the example plots. Each horizontal line represents a transcript. A transcript can contain multiple exons (grey rectangles). Transcripts can be located on the '+' or on the '-' strand of the DNA.

10_project_data_signals.csv contains fictitious signals of four RNA binding proteins (P1, P2, P3, P4).

NOTE - The data is not important, it's just a dummy data. The main thing was the representation.

Type-1

In [22]:
```
#IMPORTING ALL NECESSARY LIBRARIRES
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from matplotlib.patches import Rectangle
import matplotlib.gridspec as gridspec
import pandas as pd
import numpy as np

#to download in pdf format
!pip install Pyppeteer
!pyppeteer-install
```

```
Requirement already satisfied: Pyppeteer in c:\users\ameyb\anaconda3\lib\site-packages
(1.0.2)
Requirement already satisfied: appdirs<2.0.0,>=1.4.3 in c:\users\ameyb\anaconda3\lib\sit
e-packages (from Pyppeteer) (1.4.4)
Requirement already satisfied: certifi>=2021 in c:\users\ameyb\anaconda3\lib\site-packag
es (from Pyppeteer) (2023.5.7)
Requirement already satisfied: importlib-metadata>=1.4 in c:\users\ameyb\anaconda3\lib\s
ite-packages (from Pyppeteer) (6.0.0)
Requirement already satisfied: pyee<9.0.0,>=8.1.0 in c:\users\ameyb\anaconda3\lib\site-p
ackages (from Pyppeteer) (8.2.2)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in c:\users\ameyb\anaconda3\lib\site-
packages (from Pyppeteer) (4.65.0)
Requirement already satisfied: urllib3<2.0.0,>=1.25.8 in c:\users\ameyb\anaconda3\lib\si
te-packages (from Pyppeteer) (1.26.16)
Requirement already satisfied: websockets<11.0,>=10.0 in c:\users\ameyb\anaconda3\lib\si
te-packages (from Pyppeteer) (10.4)
Requirement already satisfied: zipp>=0.5 in c:\users\ameyb\anaconda3\lib\site-packages
(from importlib-metadata>=1.4->Pyppeteer) (3.11.0)
Requirement already satisfied: colorama in c:\users\ameyb\anaconda3\lib\site-packages (f
rom tqdm<5.0.0,>=4.42.1->Pyppeteer) (0.4.6)
chromium is already installed.
```

In [23]:
```
#READING DATA
df = pd.read_csv("D:/LSI Proj/datavis_final_project_23/10_project_data_annotation.csv")
df
```

Out[23]:

	name	type	start	stop	strand
0	geneA	transcript	2000	7764	+
1	geneA	exon	2700	5100	+
2	geneA	exon	6000	6800	+
3	geneB	transcript	9000	12720	-
4	geneB	exon	9900	10100	-
5	geneB	exon	11000	11500	-
6	geneB	exon	11900	12450	-
7	geneC	transcript	14850	18000	+
8	geneC	exon	15700	17090	+

In [24]:
```
#READING DATA
rbp = pd.read_csv("D:/LSI Proj/datavis_final_project_23/10_project_data_signals.csv")
rbp
```

Out[24]:

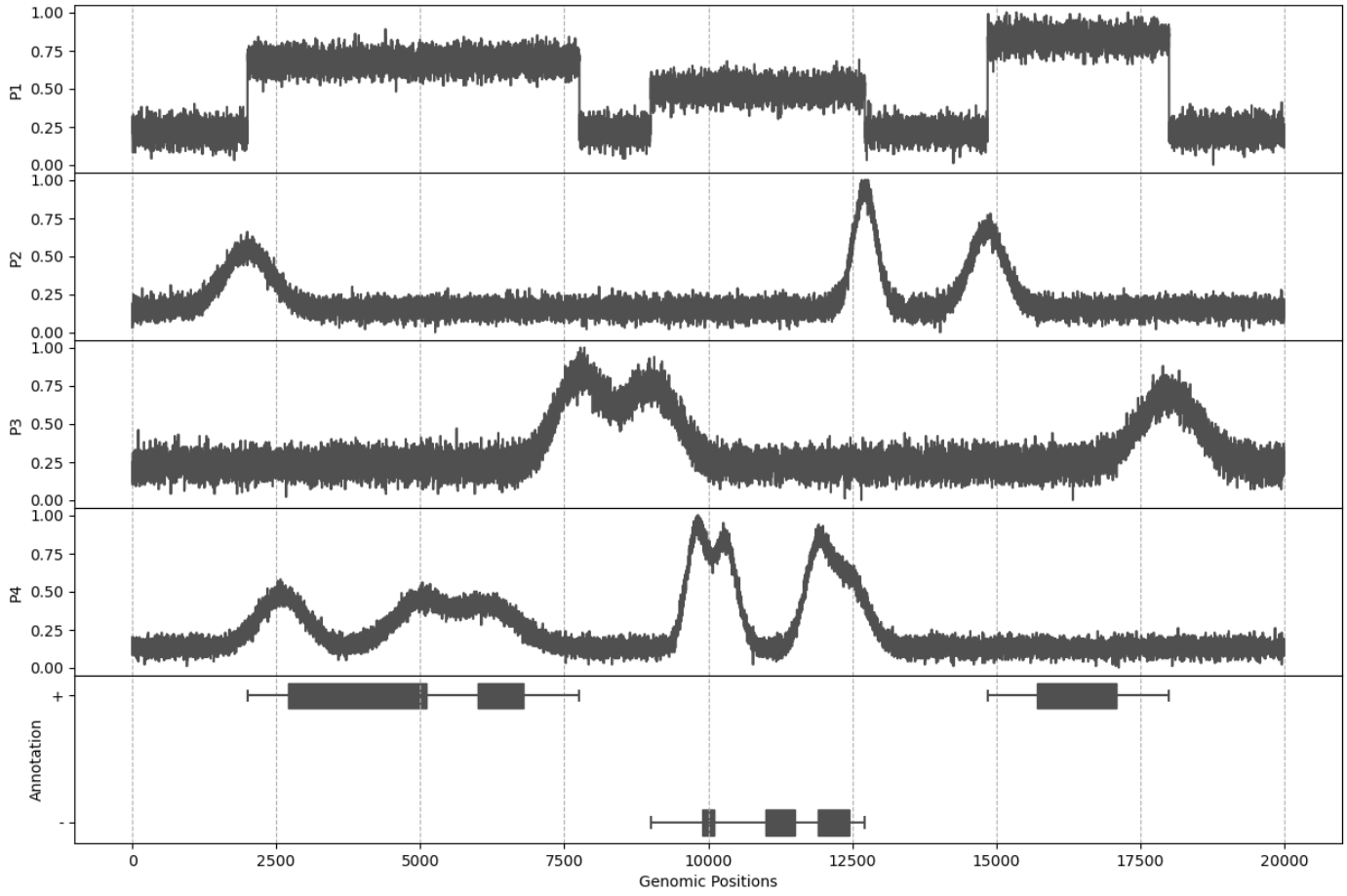
	P1	P2	P3	P4
0	0.28	0.14	0.19	0.19
1	0.30	0.16	0.17	0.20
2	0.26	0.13	0.20	0.12
3	0.21	0.13	0.25	0.15
4	0.31	0.03	0.24	0.20
...	...	...	...	...
19995	0.27	0.13	0.27	0.13
19996	0.24	0.16	0.20	0.09
19997	0.23	0.19	0.18	0.14
19998	0.25	0.10	0.17	0.09
19999	0.21	0.16	0.29	0.09

20000 rows × 4 columns

In [25]:
```
pd.options.mode.chained_assignment = None

#Plotting for RNA binding proteins from P1 to P4
fig, axs = plt.subplots(5, 1, figsize=(15, 10), sharex=True)
fig.subplots_adjust(hspace=0)
axs[0].plot(rbp.P1, c= "#505050")
axs[0].set_yticks(np.arange(0,1.25,0.25))
axs[0].set_ylabel("P1")
axs[0].grid(True, which ='major', axis='x', linestyle='--')
axs[1].plot(rbp.P2, c= "#505050")
axs[1].set_yticks(np.arange(0,1.25,0.25))
axs[1].set_ylabel("P2")
axs[1].grid(True, which ='major', axis='x', linestyle='--')
axs[2].plot(rbp.P3, c= "#505050")
```

```
axs[2].set_yticks(np.arange(0,1.25,0.25))
axs[2].set_ylabel("P3")
axs[2].grid(True, which ='major', axis='x', linestyle='--')
axs[3].plot(rbp.P4, c= "#505050")
axs[3].set_yticks(np.arange(0,1.25,0.25))
axs[3].set_ylabel("P4")
axs[3].grid(True, which ='major', axis='x', linestyle='--')


exon_df = df[df['type'] == 'exon']
exon_df["y"] = np.where(exon_df['strand']=='+',1,0)
exon_df['width']= exon_df['stop']-exon_df['start']


#choosing the rows with transcript type and creating different column y to store strand
trans_df = df[df['type'] == 'transcript']
trans_df["y"] = np.where(trans_df['strand']=='+',1,0)


#plotting the horizontal lines for the transcript
axs[4].hlines(trans_df["y"].iloc[0], trans_df["start"].iloc[0], trans_df["stop"].iloc[0]
axs[4].hlines(trans_df["y"].iloc[1], trans_df["start"].iloc[1], trans_df["stop"].iloc[1]
axs[4].hlines(trans_df["y"].iloc[2], trans_df["start"].iloc[2], trans_df["stop"].iloc[2]


#plotting the vertical lines for ends of the transcript
axs[4].vlines(trans_df["start"].iloc[0], trans_df["y"].iloc[0]-0.05, trans_df["y"].iloc[
axs[4].vlines(trans_df["stop"].iloc[0], trans_df["y"].iloc[0]-0.05, trans_df["y"].iloc[0
axs[4].vlines(trans_df["start"].iloc[1], trans_df["y"].iloc[1]-0.05, trans_df["y"].iloc[
axs[4].vlines(trans_df["stop"].iloc[1], trans_df["y"].iloc[1]-0.05, trans_df["y"].iloc[1
axs[4].vlines(trans_df["start"].iloc[2], trans_df["y"].iloc[2]-0.05, trans_df["y"].iloc[
axs[4].vlines(trans_df["stop"].iloc[2], trans_df["y"].iloc[2]-0.05, trans_df["y"].iloc[2


#using Rectangle patch for the denotion of exon
for i in range(len(exon_df)):
    rectangle= Rectangle((exon_df["start"].iloc[i],exon_df['y'].iloc[i]-0.2/2),width=exo
    axs[4].add_patch(rectangle)
axs[4].grid(True, which ='major', axis='x', linestyle='--')
axs[4].set_yticks([0,1])
axs[4].set_yticklabels(["-", "+"])
axs[4].set_ylabel("Annotation")
axs[4].set_xlabel("Genomic Positions")
```

Out[25]:
```
Text(0.5, 0, 'Genomic Positions')
```



Type-2

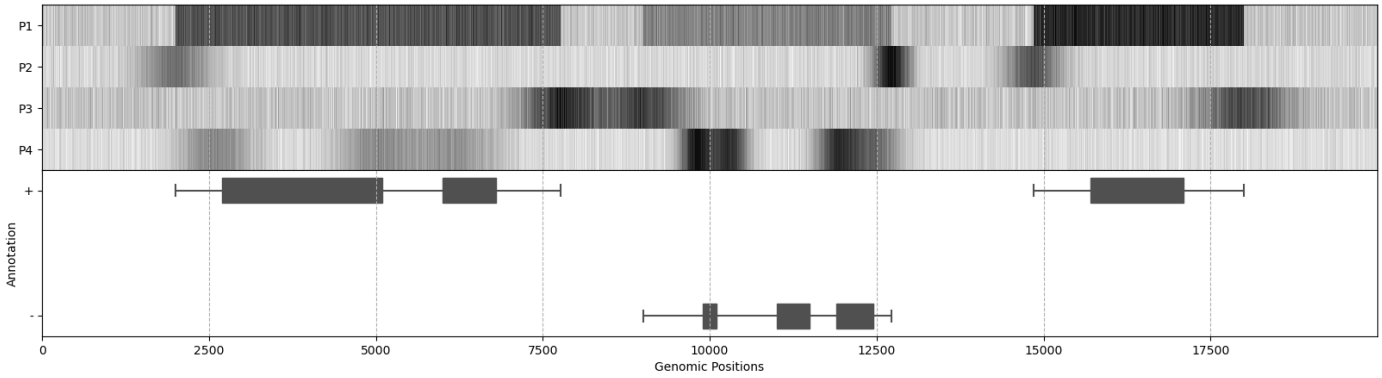
In [26]:

```
#creating 2 rows and 1 column for plotting
fig, axs = plt.subplots(2,1, sharex=True)
fig.set_size_inches(20,5)
#merging the two plots
fig.subplots_adjust(hspace=0)

#creating variables for the pcolormesh function
x = np.arange(0, len(rbp["P1"]))
y = np.arange(0, len(rbp.columns))
c = [rbp.P4, rbp.P3, rbp.P2, rbp.P1]

#plotting the 1st map with pcolormesh
axs[0].pcolormesh(x, y, c, shading = "nearest", cmap = "gray_r")
axs[0].set_yticks([0,1,2,3])
axs[0].set_yticklabels(["P4", "P3", "P2", "P1"])
axs[0].grid(True, which='major', axis='x', linestyle='--')


axs[1].hlines(trans_df["y"].iloc[0], trans_df["start"].iloc[0], trans_df["stop"].iloc[0]
axs[1].hlines(trans_df["y"].iloc[1], trans_df["start"].iloc[1], trans_df["stop"].iloc[1]
axs[1].hlines(trans_df["y"].iloc[2], trans_df["start"].iloc[2], trans_df["stop"].iloc[2]

axs[1].vlines(trans_df["start"].iloc[0], trans_df["y"].iloc[0]-0.05, trans_df["y"].iloc[
axs[1].vlines(trans_df["stop"].iloc[0], trans_df["y"].iloc[0]-0.05, trans_df["y"].iloc[0
axs[1].vlines(trans_df["start"].iloc[1], trans_df["y"].iloc[1]-0.05, trans_df["y"].iloc[
axs[1].vlines(trans_df["stop"].iloc[1], trans_df["y"].iloc[1]-0.05, trans_df["y"].iloc[1
axs[1].vlines(trans_df["start"].iloc[2], trans_df["y"].iloc[2]-0.05, trans_df["y"].iloc[
axs[1].vlines(trans_df["stop"].iloc[2], trans_df["y"].iloc[2]-0.05, trans_df["y"].iloc[2


for i in range(len(exon_df)):
```

```
    rectangle= Rectangle((exon_df["start"].iloc[i],exon_df['y'].iloc[i]-0.2/2),width=exo
    axs[1].add_patch(rectangle)
axs[1].grid(True, which ='major', axis='x', linestyle='--')
axs[1].set_yticks([0,1])
axs[1].set_yticklabels(["-", "+"])
axs[1].set_ylabel("Annotation")
axs[1].set_xlabel("Genomic Positions")
```

Out[26]:

```
Text(0.5, 0, 'Genomic Positions')
```



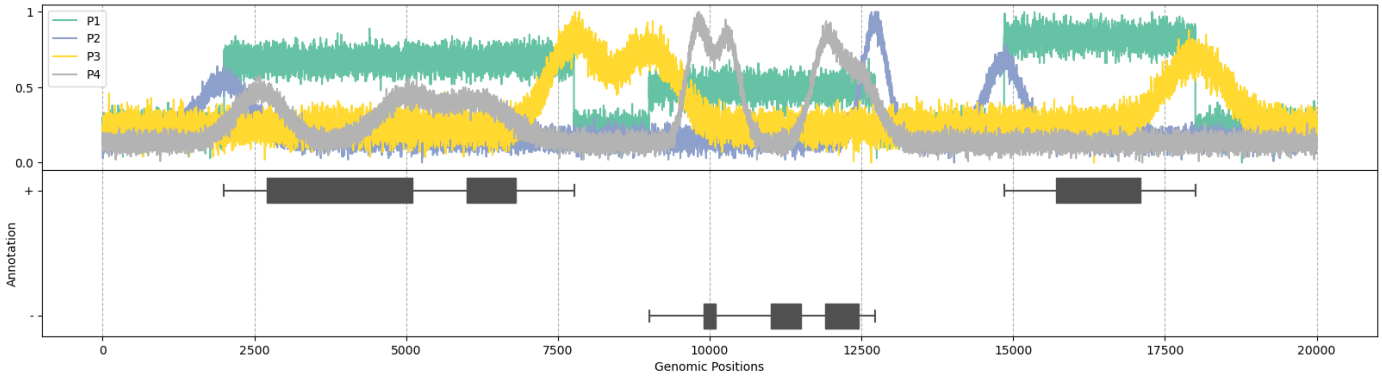
Type-3

In [27]:

```
fig, axs = plt.subplots(2,1, sharex=True)
fig.set_size_inches(20,5)
fig.subplots_adjust(hspace=0)

#creating color list for storing colors
color= ["#66C2A5", "#8DA0CB", "#FFD92F", "#B3B3B3"]
#plotting all rna binding graphs in a single graph
axs[0].plot(rbp.P1, color=color[0], label="P1")
axs[0].plot(rbp.P2, color=color[1], label="P2")
axs[0].plot(rbp.P3, color=color[2], label="P3")
axs[0].plot(rbp.P4, color=color[3], label="P4")
axs[0].grid(True, which ='major', axis='x', linestyle='--')
axs[0].set_yticks(np.arange(0, 1.25, 0.5))
axs[0].set_yticklabels([0.0, 0.5, 1])
axs[0].legend(loc="upper left")


axs[1].hlines(trans_df["y"].iloc[0], trans_df["start"].iloc[0], trans_df["stop"].iloc[0]
axs[1].hlines(trans_df["y"].iloc[1], trans_df["start"].iloc[1], trans_df["stop"].iloc[1]
axs[1].hlines(trans_df["y"].iloc[2], trans_df["start"].iloc[2], trans_df["stop"].iloc[2]

axs[1].vlines(trans_df["start"].iloc[0], trans_df["y"].iloc[0]-0.05, trans_df["y"].iloc[
axs[1].vlines(trans_df["stop"].iloc[0], trans_df["y"].iloc[0]-0.05, trans_df["y"].iloc[0
axs[1].vlines(trans_df["start"].iloc[1], trans_df["y"].iloc[1]-0.05, trans_df["y"].iloc[
axs[1].vlines(trans_df["stop"].iloc[1], trans_df["y"].iloc[1]-0.05, trans_df["y"].iloc[1
axs[1].vlines(trans_df["start"].iloc[2], trans_df["y"].iloc[2]-0.05, trans_df["y"].iloc[
axs[1].vlines(trans_df["stop"].iloc[2], trans_df["y"].iloc[2]-0.05, trans_df["y"].iloc[2


for i in range(len(exon_df)):
    rectangle= Rectangle((exon_df["start"].iloc[i],exon_df['y'].iloc[i]-0.2/2),width=exo
    axs[1].add_patch(rectangle)
axs[1].grid(True, which ='major', axis='x', linestyle='--')
axs[1].set_yticks([0,1])
axs[1].set_yticklabels(["-", "+"])
axs[1].set_ylabel("Annotation")
axs[1].set_xlabel("Genomic Positions")
```

```
Text(0.5, 0, 'Genomic Positions')
```

Out[27]:



Type 4

In this task, two additional plots shall be added to create a figure with multiple panels. Two additional datasets are provided:

10_project_data_scatter.csv contains the data needed to create the shown scatter plot

10_project_data_barplot.csv contains the data needed to create the shown bar plot

In [28]:
```
df2 = pd.read_csv("D:/LSI Proj/datavis_final_project_23/10_project_data_barplot.csv")
df2
```

Out[28]:

	Unnamed: 0	condition_a_sample_1	condition_a_sample_2	control
0	XY	756	619	689
1	XZ	2411	2189	782
2	YX	577	821	689
3	YZ	743	781	719

In [29]:
```
df1 = pd.read_csv("D:/LSI Proj/datavis_final_project_23/10_project_data_scatter.csv")
df1
```

Out[29]:

	x1	x2
0	8.41	5.43
1	9.56	3.92
2	10.83	1.80
3	11.14	2.32
4	11.41	1.41
...	...	...
1995	8.76	1.49
1996	9.89	4.09
1997	10.60	4.36
1998	10.83	2.96
1999	11.52	2.29

2000 rows × 2 columns

In [30]:
```
fig = plt.figure(constrained_layout = True)
fig.set_size_inches(15,12)

#creating the layout for the graphs
gs = fig.add_gridspec(2, 2)
ax1 = fig.add_subplot(gs[0, 0])
ax2 = fig.add_subplot(gs[0, 1])

#plotting x1 and x2 with scatterplot
ax1.scatter(df1["x1"], df1["x2"], s= 170, facecolors="none", edgecolors="black", alpha=0
ax1.set_xticks([8,10,12])
ax1.set_xticklabels([8,12,15])
ax1.set_xlabel("X1")
ax1.set_ylabel("X2")
ax1.grid(True, which ='major', linestyle='dotted', zorder=10)


x = np.arange(len(df2))  # the label locations
width = 0.17  # the width of the bars

#plotting the grouped bar plots
ax2.bar(x-0.2, df2["condition_a_sample_1"], width, color='black', zorder=7, label = "Con
ax2.bar(x, df2["condition_a_sample_2"], width, color='black', zorder=7)
ax2.bar(x+0.2, df2["control"], width, color='red', zorder=7, label = "Control")
ax2.set_xticks([0, 1, 2, 3])
ax2.set_xticklabels(['X ->Y', 'X ->Z', 'Y ->X', 'Y ->Z'])
ax2.set_ylabel("Number of events")
ax2.grid(True, which ='major', axis='y', linestyle='dotted', zorder=0)
ax2.legend()

#creating 2 rows and 1 column for plotting
fig, axs = plt.subplots(2,1, sharex=True)
fig.set_size_inches(20,5)
#merging the two plots
fig.subplots_adjust(hspace=0)

#creating variables for the pcolormesh function
x = np.arange(0, len(rbp["P1"]))
y = np.arange(0, len(rbp.columns))
c = [rbp.P4, rbp.P3, rbp.P2, rbp.P1]

#plotting the 1st map with pcolormesh
axs[0].pcolormesh(x, y, c, shading = "nearest", cmap = "gray_r")
axs[0].set_yticks([0,1,2,3])
axs[0].set_yticklabels(["P4", "P3", "P2", "P1"])
axs[0].grid(True, which='major', axis='x', linestyle='--')


axs[1].hlines(trans_df["y"].iloc[0], trans_df["start"].iloc[0], trans_df["stop"].iloc[0]
axs[1].hlines(trans_df["y"].iloc[1], trans_df["start"].iloc[1], trans_df["stop"].iloc[1]
axs[1].hlines(trans_df["y"].iloc[2], trans_df["start"].iloc[2], trans_df["stop"].iloc[2]

axs[1].vlines(trans_df["start"].iloc[0], trans_df["y"].iloc[0]-0.05, trans_df["y"].iloc[
axs[1].vlines(trans_df["stop"].iloc[0], trans_df["y"].iloc[0]-0.05, trans_df["y"].iloc[0
axs[1].vlines(trans_df["start"].iloc[1], trans_df["y"].iloc[1]-0.05, trans_df["y"].iloc[
axs[1].vlines(trans_df["stop"].iloc[1], trans_df["y"].iloc[1]-0.05, trans_df["y"].iloc[1
axs[1].vlines(trans_df["start"].iloc[2], trans_df["y"].iloc[2]-0.05, trans_df["y"].iloc[
axs[1].vlines(trans_df["stop"].iloc[2], trans_df["y"].iloc[2]-0.05, trans_df["y"].iloc[2


for i in range(len(exon_df)):
    rectangle= Rectangle((exon_df["start"].iloc[i],exon_df['y'].iloc[i]-0.2/2),width=exo
```

```
    axs[1].add_patch(rectangle)
axs[1].grid(True, which ='major', axis='x', linestyle='--')
axs[1].set_yticks([0,1])
axs[1].set_yticklabels(["-", "+"])
axs[1].set_ylabel("Annotation")
axs[1].set_xlabel("Genomic Positions")
```

Out[30]:

```
Text(0.5, 0, 'Genomic Positions')
```