

can you explain this figure

QuickBot

The figure displays a comparison of CDR H3 hydrophobic scores between the "Head" and "Stem" regions.

The "Stem" region exhibits a significantly higher (less negative) hydrophobic score than the "Head" region ($p=0.001$).

The plots are box plots with individual data points overlaid.

can you write python code to make similar plot with same style as the picture.

QuickBot

python

◇ Collapse □ Run Save Copy

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 from scipy import stats
4
5 # Sample data (replace with your actual data)
6 np.random.seed(42)
7 head_data = np.random.normal(-2.5, 0.8, 100)
8 stem_data = np.random.normal(-1.0, 0.8, 100)
9
10 # Create the figure and axes
11 fig, ax = plt.subplots(figsize=(6, 6))
```

```

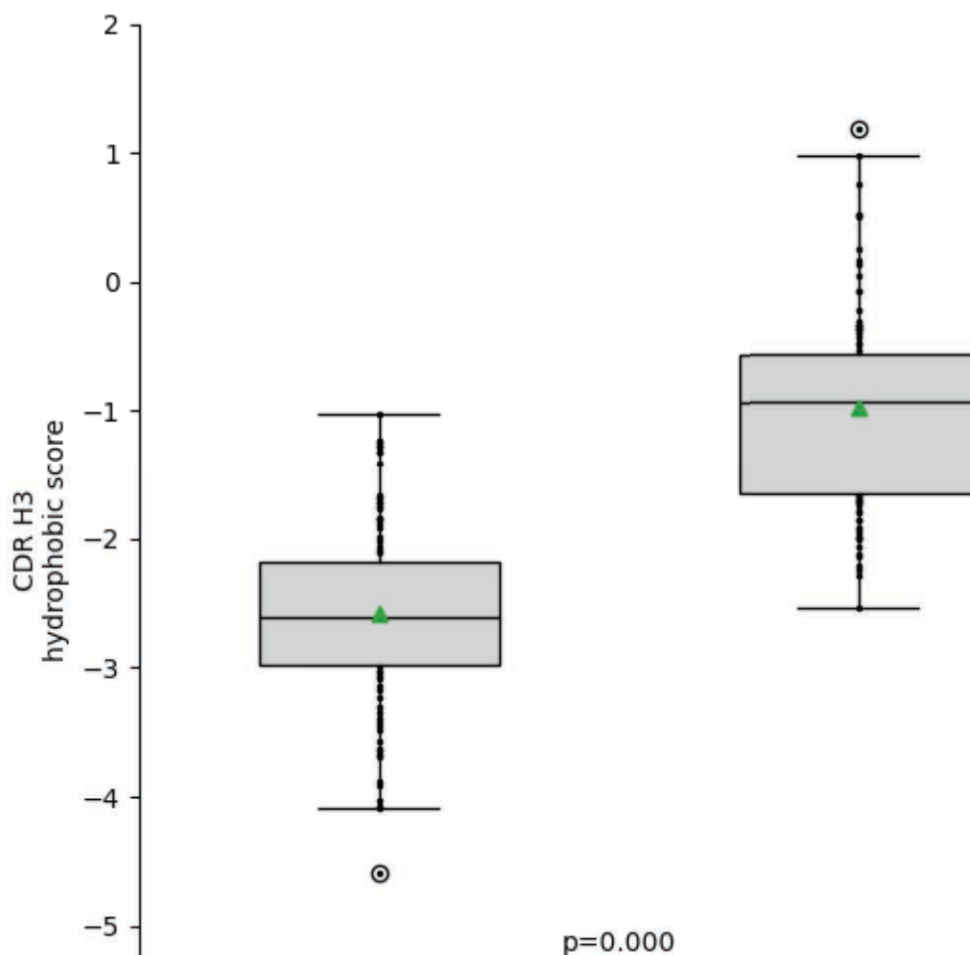
15     capprops=dict(color='black'), whiskerprops=dict(color='black'),
16     bp2 = ax.boxplot(stem_data, positions=[1], widths=0.5, patch_artist=True, showcaps=True,
17                     capprops=dict(color='black'), whiskerprops=dict(color='black'),
18
19     # Add individual data points as scatter plots
20     ax.scatter(np.zeros(len(head_data)), head_data, color='black', marker='.', s=10)
21     ax.scatter(np.ones(len(stem_data)), stem_data, color='black', marker='.', s=10)
22
23     # Set axis labels and title
24     ax.set_xticks([0, 1])
25     ax.set_xticklabels(['Head', 'Stem'])
26     ax.set_ylabel('CDR H3\nhydrophobic score')
27
28     # Add p-value
29     t_statistic, p_value = stats.ttest_ind(head_data, stem_data)
30     ax.text(0.5, 0.1, f'p={p_value:.3f}', ha='center', transform=ax.transAxes)
31
32     # Remove top and right spines
33     ax.spines['top'].set_visible(False)
34     ax.spines['right'].set_visible(False)
35
36     # Adjust y-axis limits
37     ax.set_ylim([-6, 2])
38
39     # Show the plot
40     plt.tight_layout()
41     plt.show()

```

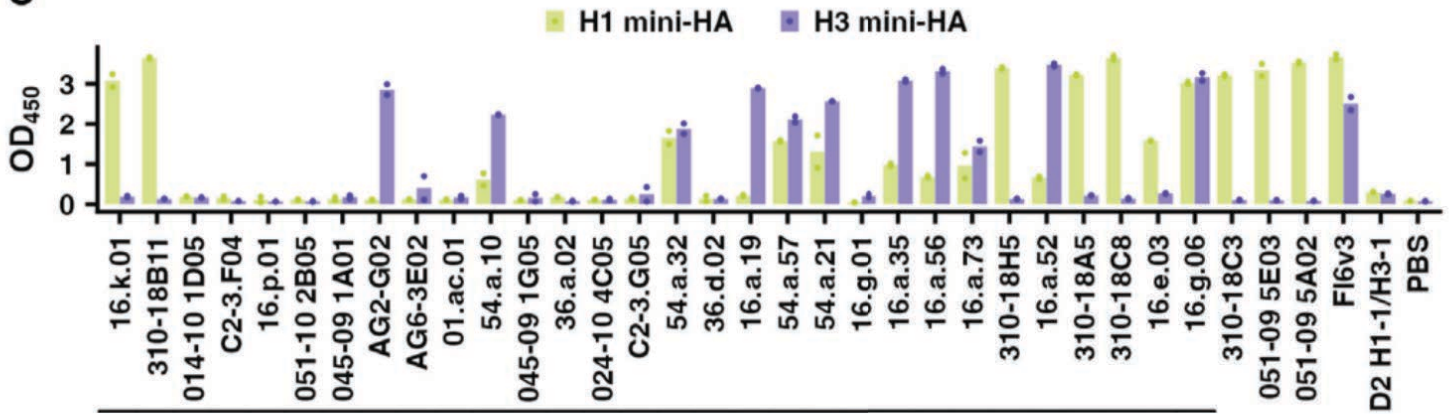
STDOUT/STDERR

Matplotlib is building the font cache; this may take a moment.

RESULT



C



can you write python code to make similar plot with the same style

QuickBot

python

[Collapse](#) [Run](#) [Save](#) [Copy](#)

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # Sample data (replace with your actual data)
5 np.random.seed(42)
6 groups = ['310-18B11', '014-10-18F04', 'C2-3.F05', '16.P.01', '051-10-2B05', '04
7           'AG6-3E02', '01.ac.10', '54.a.10', '045-09-1G05', '30.4C.02', '24-1C.0
8           'C2-3.G05', '54.d.02', '30.4C.05', '54.a.57', '16.g.01', '30.4.35', '1
9           '16.a.73', '310-18H5', '310-18A52', '310-18C8', '310-18E03', '310-18C3
10          '051-09-5E03', '051-09-5A03', 'FIGV3', 'D2 H1-1/H3-1', 'PBS']
11 num_groups = len(groups)
12
13 # Generate random data for each group
14 data1 = np.random.rand(num_groups) * 1.5
15 data2 = np.random.rand(num_groups) * 1.5
16
17 # Create the figure and axes
18 fig, ax = plt.subplots(figsize=(16, 6))
19
20 # Bar plot for data1 (H1 mini-HA)
21 ax.bar(np.arange(num_groups), data1, color='lightgrey', width=0.4)
22
23 # Bar plot for data2 (H3 mini-HA)
24 ax.bar(np.arange(num_groups) + 0.4, data2, color='lightsteelblue', width=0.4)
25
26 # Set x-axis labels
27 ax.set_xticks(np.arange(num_groups) + 0.2)
28 ax.set_xticklabels(groups, rotation=90)
29
30 # Set y-axis label
31 ax.set_ylabel('OD')
32
33 # Adjust x-axis limits
34 ax.set_xlim([0.5, num_groups + 0.5])

```

38

39 # Show the plot

40 plt.tight_layout()

41 plt.show()

Collapse

Run

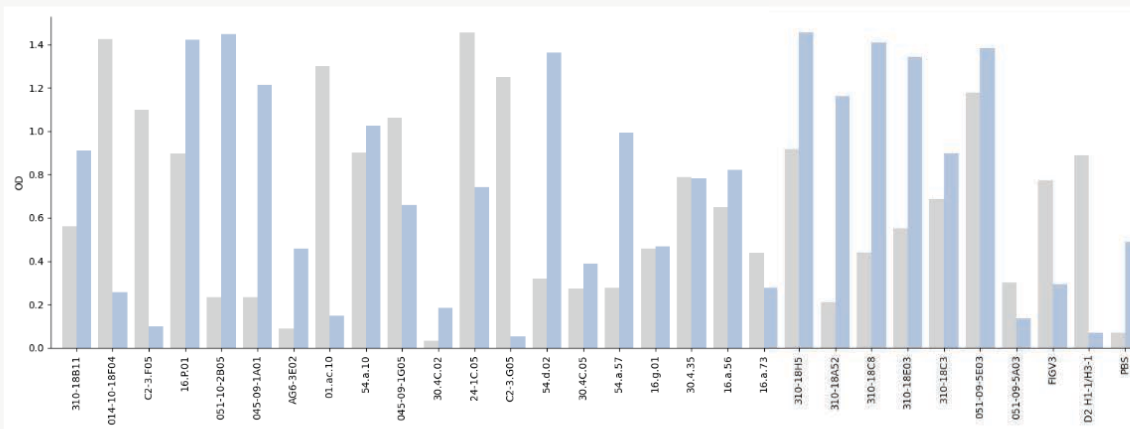
Save

Copy

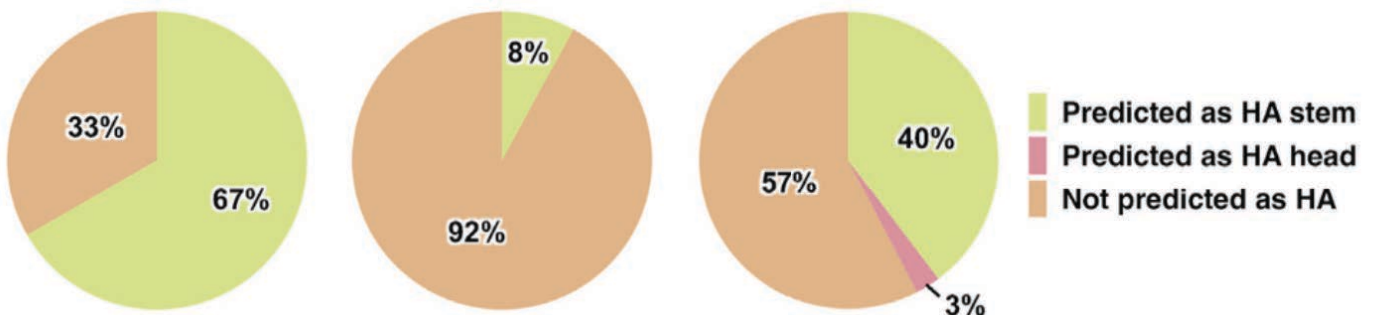
STDOUT/STDERR

Matplotlib is building the font cache; this may take a moment.

RESULT



A 60 central stem Abs (Andrews et al. 2023) 38 anchor stem Abs (Andrews et al. 2023) **B** 4,453 HA unk Abs



can you write python code to make similar plot with same style

QuickBot

python

Collapse

Run

Save

Copy

1 import matplotlib.pyplot as plt

2

3 # Data for the pie charts

4 labels = ['Predicted as HA stem', 'Predicted as HA head', 'Not predicted as HA']

5 sizes1 = [67, 33, 0] # Data for the first pie chart

6 sizes2 = [92, 8, 0] # Data for the second pie chart

7 sizes3 = [57, 40, 3]

8

9 # Colors for the pie charts

10 colors = ['lightgrey', 'lightsteelblue', 'darkseagreen']

11

12 # Create the figure and axes

13 fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(18, 6))

14


```

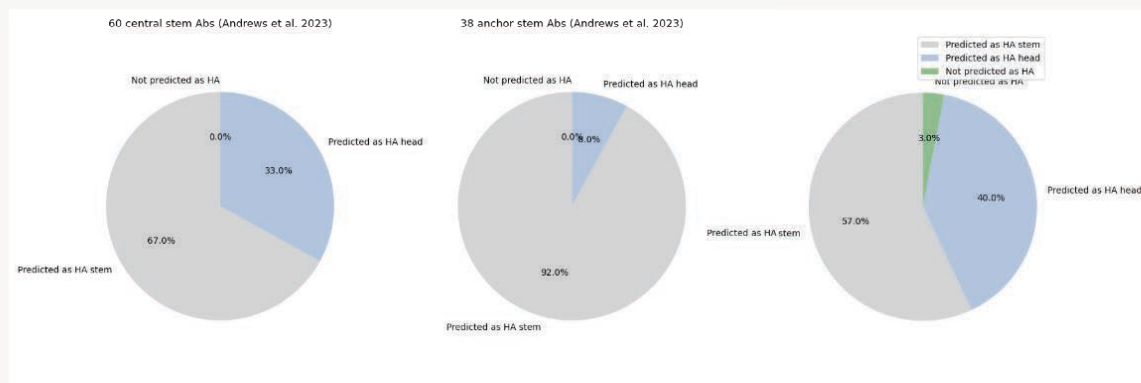
18 ax1.set_title('60 central stem Abs (Andrews et al. 2023)')
19
20 # Second pie chart
21 ax2.pie(sizes2, labels=labels, autopct='%1.1f%%', startangle=90, colors=colors)
22 ax2.axis('equal')
23 ax2.set_title('38 anchor stem Abs (Andrews et al. 2023)')
24
25 # Third pie chart
26 ax3.pie(sizes3, labels=labels, autopct='%1.1f%%', startangle=90, colors=colors)
27 ax3.axis('equal')
28
29 # Add a legend
30 plt.legend(labels, loc="best")
31
32 # Show the plot
33 plt.tight_layout()
34 plt.show()

```

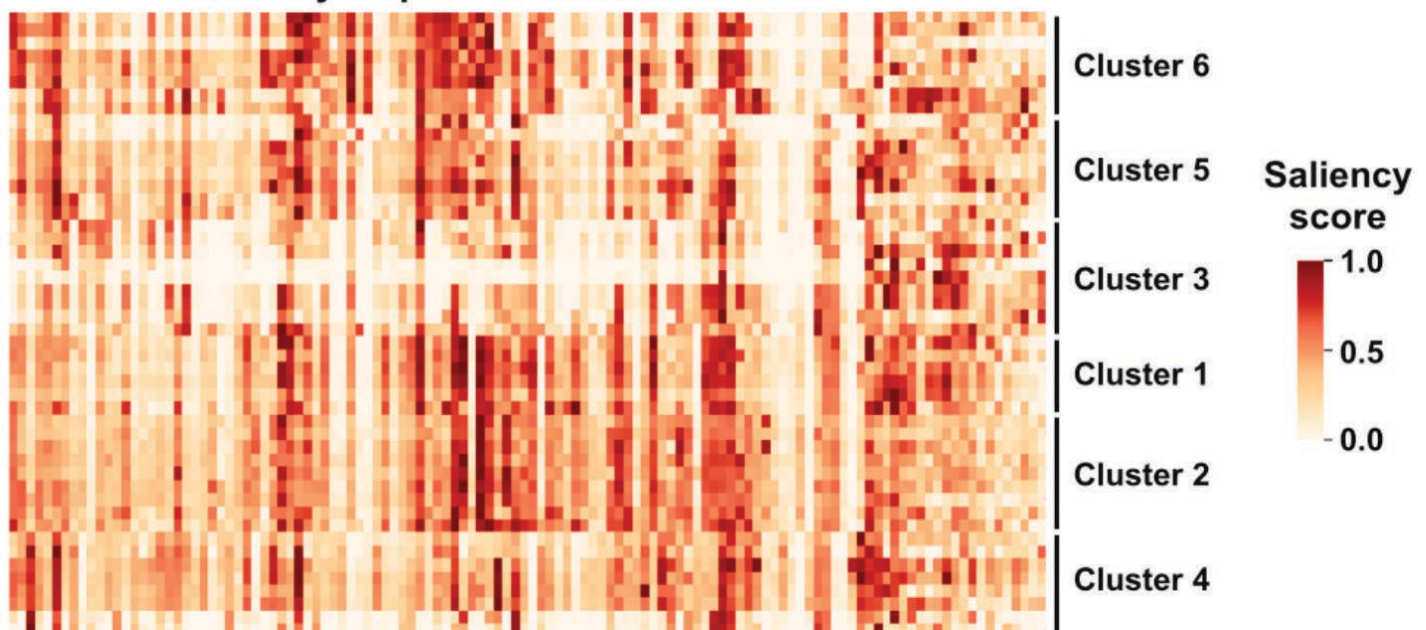
STDOUT/STDERR

Matplotlib is building the font cache; this may take a moment.

RESULT



Saliency map for HA stem antibodies



can you write python code to make similar plot with same style

python

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # Generate dummy data (replace with your actual data)
5 np.random.seed(42)
6 n_rows = 20 # Number of rows (clusters)
7 n_cols = 30 # Number of columns (features)
8 data = np.random.rand(n_rows, n_cols)
9 data = np.clip(data, 0, 1) # Ensure values are between 0 and 1
10
11 # Create the figure and axes
12 fig, ax = plt.subplots(figsize=(10, 8))
13
14 # Display the heatmap
15 im = ax.imshow(data, cmap="RdBu", aspect="auto")
16
17 # Set axis labels
18 ax.set_xlabel("Features")
19 ax.set_ylabel("Clusters")
20 ax.set_title("Saliency map for HA stem antibodies")
21
22 # Add cluster labels
23 ax.set_yticks(np.arange(n_rows))
24 cluster_labels = ["Cluster 1", "Cluster 2", "Cluster 3", "Cluster 4", "Cluster 5"]
25 ax.set_yticklabels(cluster_labels)
26
27 # Add a colorbar
28 cbar = fig.colorbar(im, ax=ax)
29 cbar.set_label("Saliency\score")
30
31 # Remove ticks and spines
32 ax.set_xticks([])
33 ax.spines["top"].set_visible(False)
34 ax.spines["right"].set_visible(False)
35 ax.spines["left"].set_visible(False)
36 ax.spines["bottom"].set_visible(False)
37
38 # Show the plot
39 plt.tight_layout()
40 plt.show()
```

STDOUT/STDERR

PythonError: Traceback (most recent call last):

File "/lib/python312.zip/_pyodide/_base.py", line 597, in eval_code_async
await CodeRunner(

File "/lib/python312.zip/_pyodide/_base.py", line 411, in run_async
coroutine = eval(self.code, globals, locals)

^^

File "<exec>", line 25, in <module>

File "/lib/python3.12/site-packages/matplotlib/axes/_base.py", line 73, in wrapper
return get_method(self)(*args, **kwargs)

^^

File "/lib/python3.12/site-packages/matplotlib/_api/deprecation.py", line 297, in wrapper
return func(*args, **kwargs)

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "/lib/python3.12/site-packages/matplotlib/axis.py", line 2025, in set_ticklabels

i got error: ValueError: The number of FixedLocator locations (20), usually from a call to set_ticks, does not match the number of labels (6).

QuickBot

```
python ⌵ Collapse ☑ Run Save Copy  
1 import matplotlib.pyplot as plt  
2 import numpy as np  
3  
4 # Generate dummy data (replace with your actual data)  
5 np.random.seed(42)  
6 n_rows = 20 # Number of rows (clusters)  
7 n_cols = 30 # Number of columns (features)  
8 data = np.random.rand(n_rows, n_cols)  
9 data = np.clip(data, 0, 1) # Ensure values are between 0 and 1  
10  
11 # Create the figure and axes  
12 fig, ax = plt.subplots(figsize=(10, 8))  
13  
14 # Display the heatmap  
15 im = ax.imshow(data, cmap="RdBu", aspect="auto")  
16  
17 # Set axis labels  
18 ax.set_xlabel("Features")  
19 ax.set_ylabel("Clusters")  
20 ax.set_title("Saliency map for HA stem antibodies")  
21  
22 # Add cluster labels. Adjust the number of labels to match the actual data size  
23 ax.set_yticks(np.arange(n_rows))  
24 cluster_labels = [f"Cluster {i+1}" for i in range(n_rows)] # Create labels for  
25 ax.set_yticklabels(cluster_labels)  
26  
27 # Add a colorbar  
28 cbar = fig.colorbar(im, ax=ax)  
29 cbar.set_label("Saliency\ncore")  
30  
31 # Remove ticks and spines  
32 ax.set_xticks([])  
33 ax.spines["top"].set_visible(False)  
34 ax.spines["right"].set_visible(False)  
35 ax.spines["left"].set_visible(False)  
36 ax.spines["bottom"].set_visible(False)  
37  
38 # Show the plot  
39 plt.tight_layout()  
40 plt.show()
```

STDOUT/STDERR

Matplotlib is building the font cache; this may take a moment.

RESULT

```

33 ax.spines["top"].set_visible(False)
34 ax.spines["right"].set_visible(False)
35 ax.spines["left"].set_visible(False)
36 ax.spines["bottom"].set_visible(False)
37
38 # Show the plot
39 plt.tight_layout()
40 plt.show()

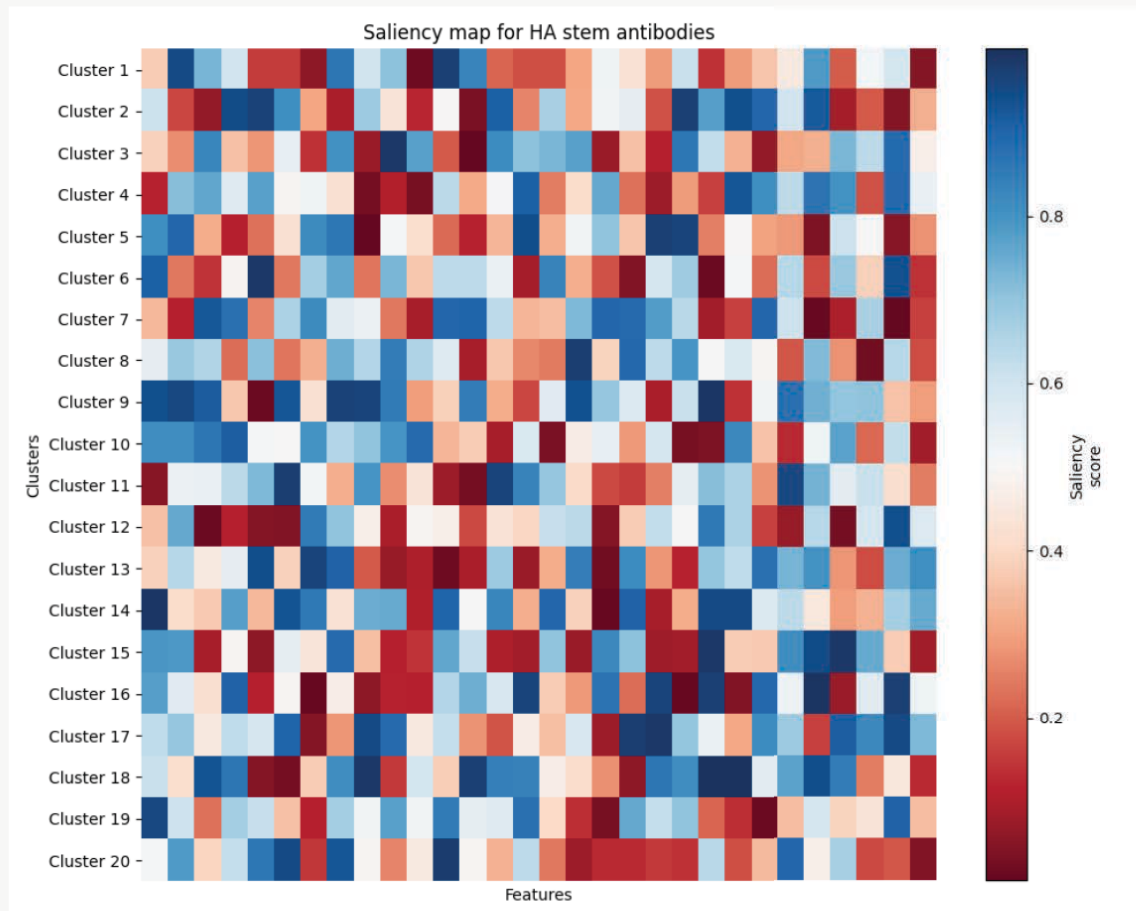
```

[Collapse](#)
[Run](#)
[Save](#)
[Copy](#)

STDOUT/STDERR

Matplotlib is building the font cache; this may take a moment.

RESULT



Reasoning for the fix:

ValueError: The number of FixedLocator locations (20), usually from a call to set_ticks, does not match the number of labels (6).

the number of tick locations (positions on the y-axis where labels are displayed) and the number of labels provided.