

Figure_5

February 14, 2024

```
[1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
from matplotlib.patches import Patch
from pycirclize import Circos, sector
from pycirclize.parser import Matrix
from itertools import permutations

[2]: table_5_A = pd.read_excel("Table_5.xlsx",sheet_name="Fig5 panel a")
table_5_B = pd.read_excel("Table_5.xlsx",sheet_name="Fig 5 panel b")
table_5_C = pd.read_excel("Table_5.xlsx",sheet_name="Fig5 panel c")

[3]: GCFs = []
pGCFs= []
for i in range(len(table_5_C)):
    GCFs.append("GCFs")
    pGCFs.append("pGCFs")

table_5_C["GCFs"] = GCFs
table_5_C["pGCFs"] = pGCFs

table_5_C_1 = table_5_C[["REDgroup", "Number of actual GCFs", "GCFs"]].
    ↪rename(columns={"GCFs" : "groups", "Number of actual GCFs" : "Number of_
    ↪GCFs"})

table_5_C_2 = table_5_C[["REDgroup", "Number of potential GCFs", "pGCFs"]].
    ↪rename(columns={"pGCFs" : "groups", "Number of potential GCFs" : "Number of_
    ↪GCFs"})

new_table_5_C = pd.concat((table_5_C_1, table_5_C_2))

[4]: # table_5_B = table_5_B.replace({"Streptomyces_B":"SB","Streptomyces_C":
    ↪"SC","Streptomyces_D":"SD"})
```

```
[5]: # my_liste = [ (('Actinobacteriota', 7603.0, 24012.0), ('Actinobacteriota', 40421.0, 24012.0)),
#               (('Streptomyces', 126.0, 6034.0), ('Streptomyces', 11942.0, 6034.0)),
#               (('Streptomyces', 32.0, 126.0), ('SB', 311.0, 217.0)),
#               (('Actinobacteriota', 309.0, 7603.0), ('Streptomyces', 19236.0, 11942.0)),
#               (('Actinobacteriota', 93.0, 309.0), ('SB', 527.0, 311.0)),
#               (('Actinobacteriota', 36.0, 93.0), ('SC', 156.0, 99.0)),
#               (('Actinobacteriota', 0.0, 36.0), ('SD', 92.0, 56.0)),
#               (('Streptomyces', 17.0, 32.0), ('SC', 99.0, 84.0)),
#               (('Streptomyces', 0.0, 17.0), ('SD', 56.0, 39.0)),
#               (('SD', 0.0, 16.0), ('SD', 32.0, 16.0)),
#               (('SC', 0.0, 1.0), ('SD', 33.0, 32.0)),
#               (('SC', 1.0, 40.0), ('SC', 79.0, 40.0)),
#               (('SB', 0.0, 6.0), ('SD', 39.0, 33.0)),
#               (('SB', 6.0, 11.0), ('SC', 84.0, 79.0)),
#               (('SB', 11.0, 114.0), ('SB', 217.0, 114.0))
# ]
```

```
[6]: # bosliste = []
# sorted_list = ["Streptomyces", "SB", "SC", "SD", "Actinobacteriota"]

# for permutation in permutations(sorted_list):

#     matrix = Matrix.parse_fromto_table(table_5_B)

#     ax = Circos.initialize_from_matrix(matrix,
#                                       order=permutation,
#                                       space=3,
#                                       r_lim=(99,100),
#                                       cmap="tab10",
#                                       link_kws=dict(direction=0,r1=97,r2=97)
#                                       )
#     bosliste.append(ax)
```

```
[7]: # table_5_B = table_5_B.rename({"source taxon": "from", "target taxon":
#     "to", "Number of shared GCFs": "value"}, axis="columns")
```

```
[8]: # my_liste = [ (('Actinobacteriota', 7603.0, 24012.0), ('Actinobacteriota', 40421.0, 24012.0)),
#               (('Streptomyces', 126.0, 6034.0), ('Streptomyces', 11942.0, 6034.0)),
#               (('Streptomyces', 32.0, 126.0), ('SB', 311.0, 217.0)),
#               (('Actinobacteriota', 309.0, 7603.0), ('Streptomyces', 19236.0, 11942.0)),
#               (('Actinobacteriota', 93.0, 309.0), ('SB', 527.0, 311.0)),
#               (('Actinobacteriota', 36.0, 93.0), ('SC', 156.0, 99.0)),
#               (('Actinobacteriota', 0.0, 36.0), ('SD', 92.0, 56.0)),
#               (('Streptomyces', 17.0, 32.0), ('SC', 99.0, 84.0)),
#               (('Streptomyces', 0.0, 17.0), ('SD', 56.0, 39.0)),
```

```
# (('SD', 0.0, 16.0), ('SD', 32.0, 16.0)),
# (('SC', 0.0, 1.0), ('SD', 33.0, 32.0)),
# (('SC', 1.0, 40.0), ('SC', 79.0, 40.0)),
# (('SB', 0.0, 6.0), ('SD', 39.0, 33.0)),
# (('SB', 6.0, 11.0), ('SC', 84.0, 79.0)),
# (('SB', 11.0, 114.0), ('SB', 217.0, 114.0))
# ]
```

```
[9]: # bosliste = []
# sorted_list = ['SD', 'SC', 'SB', 'Streptomyces', 'Actinobacteriota']

# for permutation in permutations(sorted_list):

#     matrix = Matrix.parse_from_table(table_5_B)
#     mylst = np.random.shuffle(matrix.all_names)
#     matrix = matrix.sort(mylst)
#     ax = Circos.initialize_from_matrix(matrix,
#                                         start=-150, end=210,
#                                         space=3,
#                                         r_lim=(99, 100),
#                                         cmap="tab10",
#                                         link_kws=dict(direction=0, r1=97, r2=97)
#                                         )
#     bosliste.append(ax)
```

```
[10]: # for i, all_ax in enumerate(bosliste):
#     all_ax.savefig(f"fig{i}.png", dpi=21)
```

```
[11]: # matrix_df = pd.DataFrame(matrix_data, index=row_names, columns=col_names)

# circos = Circos.initialize_from_matrix(
#     second_Figure.transpose(),
#     space=5,
#     cmap="RdPu",
#     label_kws=dict(size=12),
# )

# circos.plotfig();
```

FIGURE 5 C i,ii

```
[12]: f, ax = plt.subplots(figsize=(2, 12), dpi=400)

new_table_5_C['groups'] = pd.Categorical(new_table_5_C['groups'],
categories=['GCFs', 'pGCFs'])
```

```

ax = sns.barplot(x="Number of GCFs",
                 y="REDgroup",
                 data = new_table_5_C,
                 hue="groups",
                 palette=["#070159", "#a9d4f5"],
                 orient='h',
                 hue_order=["pGCFs", "GCFs"])

ax.invert_yaxis()

ax.xaxis.set_ticks([0, 2000, 4000])

ax.set_ylim([-2, 20.48])

ax.set_ylabel(None)
ax.set_xlabel("Number of GCFs", fontsize=12.9)

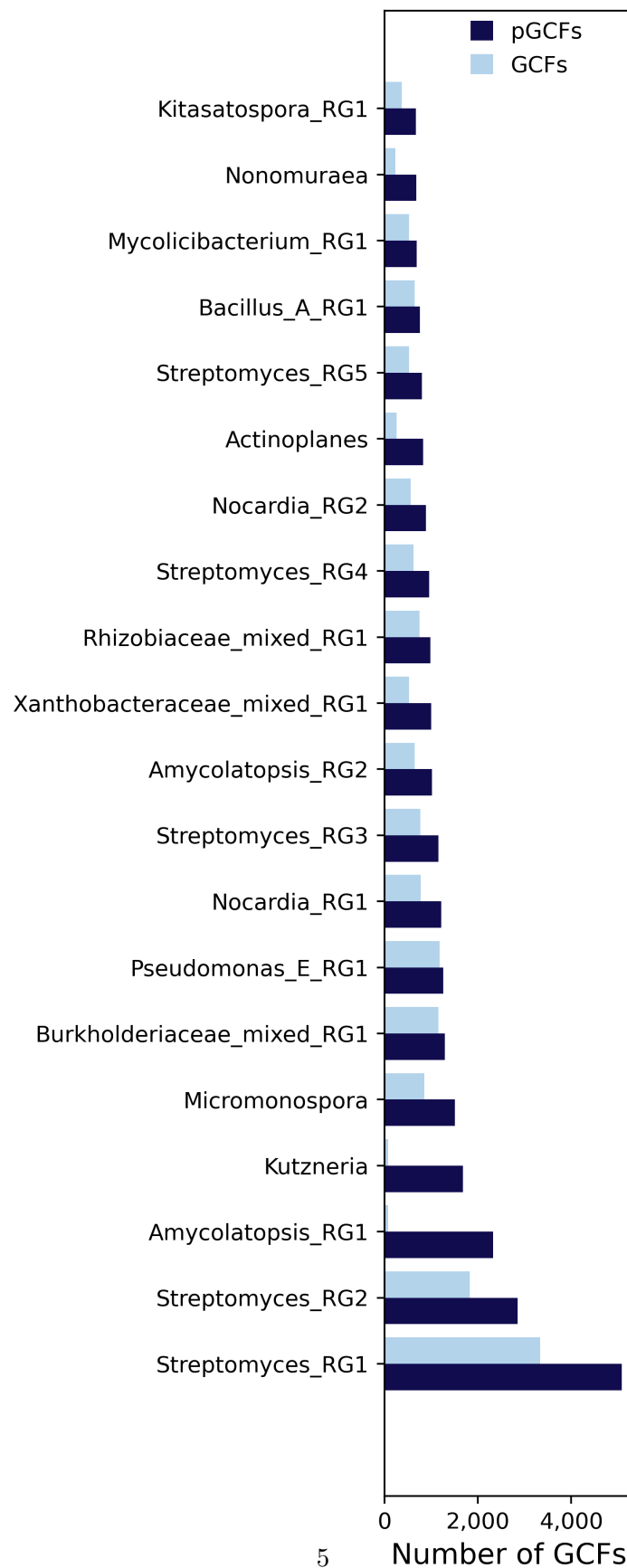
ax.set_xticklabels(["0", "2,000", "4,000"])

sns.move_legend(ax, "lower center", bbox_to_anchor=(.57, 0.942), ncol=1,
               ↪title=None, frameon=False)

ax.set_xticks([0,2000,4000]);

f.savefig('output_figure_5_C_i_Topic_10.png', dpi=400)

```



```

[13]: legend_elements = [Patch(facecolor='orange', edgecolor='orange', label='NPs')]

f, ax = plt.subplots(figsize=(2, 12), dpi=400)

ax = sns.barplot(x="Number of NPs in NPASS", y="REDgroup", data = table_5_C,
    ↪orient='h',color="#f56d05")

ax.legend(handles=legend_elements, loc="lower center", bbox_to_anchor=(.3, 0.
    ↪966), ncol=1, title=None, frameon=False)

plt.text(250,1.9, "*", ha='center', va='bottom', color="red")
plt.text(250,2.9, "*", ha='center', va='bottom', color="red")
plt.text(250,10.9, "*", ha='center', va='bottom', color="red")
plt.text(250,16.9, "*", ha='center', va='bottom', color="red")
plt.text(250,17.9, "*", ha='center', va='bottom', color="red")
plt.text(250,18.9, "*", ha='center', va='bottom', color="red")

ax.invert_xaxis()
ax.invert_yaxis()

ax.set(yticklabels=[])
ax.set(ylabel=None)

yticks = [i*20/8 for i in range(0,9)]
ax.set_yticks(yticks)

ax.set_xticks([0,500])

ax.text(800, -3.9, "Number of NPs in\nNPASS database", fontsize=12,
    ↪multialignment="left")

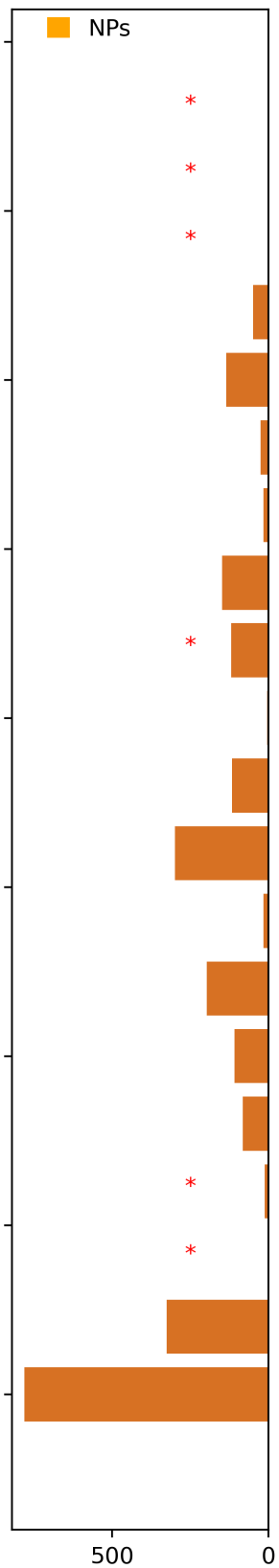
ax.set_ylim([-2, 20.48])

ax.set_xlabel(None)
ax.set_ylabel(None)

plt.show()

f.savefig('output_figure_5_C_ii_Topic_10.png', dpi=400)

```



Number of NPs in
NPASS database

0.0.1 **END OF FIGURE 5** A, B, C_i, C_{ii}