import pandas as pd

# 从Excel文件加载数据

excel\_file = "CADVis\_Author.xlsx" # 替换为您的Excel文件路径

df = pd.read\_excel(excel\_file)

# 提取所有作者的列表

all\_authors = df["Fullname"].tolist()

# 计算共同作者的数量

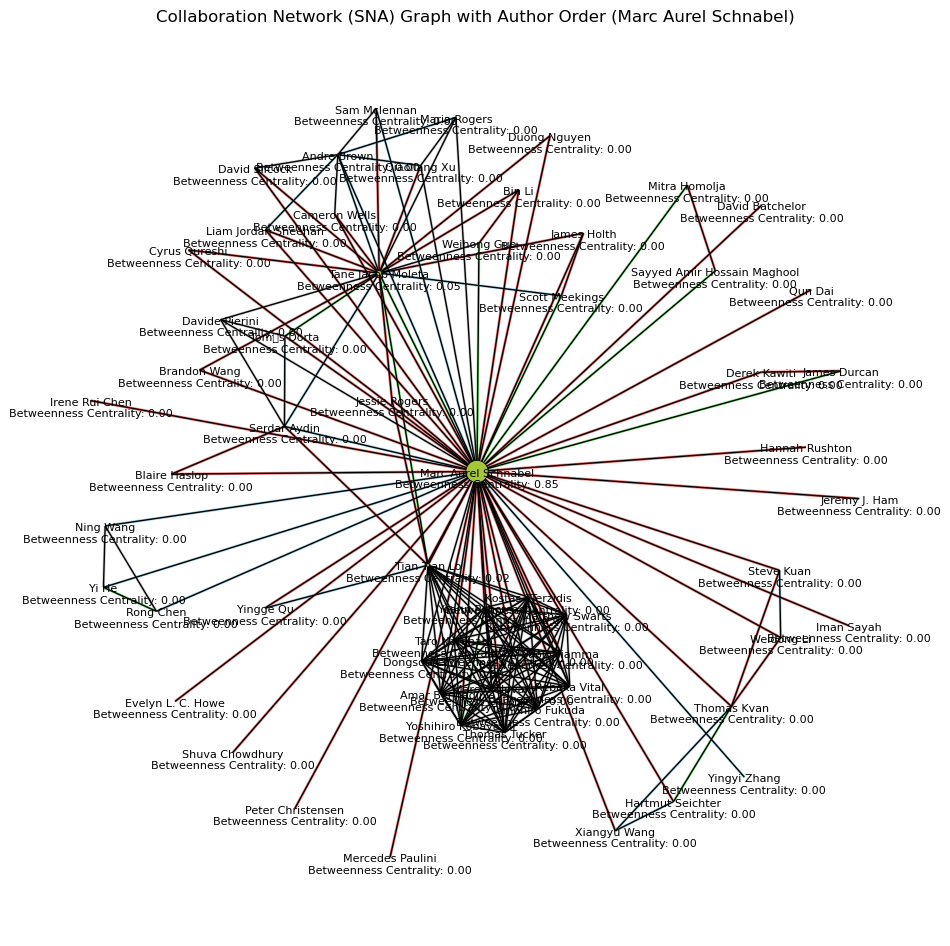
common\_authors = len(set(all\_authors))

# 输出结果

print(f"共有 {common\_authors} 位共同作者在数据集中。")

共有 3141 位共同作者在数据集中。

介數中心性程式碼



import networkx as nx

import pandas as pd

import matplotlib.pyplot as plt

# 创建一个空的无向图

G = nx.Graph()

# 从Excel文件加载数据

excel\_file = "CADVis\_Author.xlsx" # 替换为您的Excel文件路径

df = pd.read\_excel(excel\_file)

# 根据相同的 work\_id 创建节点和边

for work\_id, group in df.groupby("work\_id"):

authors = group["Fullname"].tolist()

order = group["Order"].tolist()

# 添加节点（作者）

for author in authors:

G.add\_node(author) # 添加节点（作者）

# 添加带有作者顺序的边（合作关系）

for i in range(len(authors)):

for j in range(i + 1, len(authors)):

G.add\_edge(authors[i], authors[j], order=order[i]) # 使用order[i]来表示作者顺序

# 创建一个子图，只包括与Marc Aurel Schnabel相关的节点和边

subgraph = G.subgraph(["Marc Aurel Schnabel"] + list(G.neighbors("Marc Aurel Schnabel")))

# 计算节点的介数中心性

betweenness\_centrality = nx.betweenness\_centrality(subgraph)

# 计算节点的大小，根据介数中心性调整

node\_size = [300 \* betweenness\_centrality[node] for node in subgraph.nodes()]

# 可视化SNA图

plt.figure(figsize=(12, 12))

# 自定义布局

pos = nx.spring\_layout(subgraph, seed=42)

# 绘制边，根据作者顺序添加不同颜色的边（浅色系）

for edge in subgraph.edges(data=True):

order = edge[2]["order"]

if order == 1:

edge\_color = "lightcoral" # 浅红色

elif order == 2:

edge\_color = "lightblue" # 浅蓝色

elif order == 3:

edge\_color = "lightgreen" # 浅绿色

else:

edge\_color = "lightgray" # 浅灰色

nx.draw\_networkx\_edges(subgraph, pos, edgelist=[(edge[0], edge[1])], edge\_color=edge\_color, width=2)

# 绘制节点，节点大小基于介数中心性，节点颜色为芥末色

nx.draw(subgraph, pos, node\_size=node\_size, node\_color='#A4C639', with\_labels=False, edgecolors='black', linewidths=0.5)

# 绘制节点标签（介数中心性），但调整它们的位置以避免重叠，并设置文字颜色为深黑色

label\_pos = {k: (v[0], v[1] - 0.02) for k, v in pos.items()}

labels = {node: f"{node}\nBetweenness Centrality: {betweenness\_centrality[node]:.2f}" for node in subgraph.nodes()}

nx.draw\_networkx\_labels(subgraph, label\_pos, labels=labels, font\_size=8, font\_color='#000000')

plt.title("Collaboration Network (SNA) Graph with Author Order (Marc Aurel Schnabel)")

plt.axis('off') # 关闭坐标轴

plt.show()

**轉成JSON檔**

import networkx as nx

import pandas as pd

import json

# 创建一个空的无向图

G = nx.Graph()

# 从Excel文件加载数据

excel\_file = "CADVis\_Author.xlsx" # 替换为您的Excel文件路径

df = pd.read\_excel(excel\_file)

# 根据相同的 work\_id 创建节点和边

for work\_id, group in df.groupby("work\_id"):

authors = group["Fullname"].tolist()

order = group["Order"].tolist()

# 添加节点（作者）

for author in authors:

G.add\_node(author) # 添加节点（作者）

# 添加带有作者顺序的边（合作关系）

for i in range(len(authors)):

for j in range(i + 1, len(authors)):

G.add\_edge(authors[i], authors[j], order=order[i]) # 使用order[i]来表示作者顺序

# 创建一个子图，只包括与Marc Aurel Schnabel相关的节点和边

subgraph = G.subgraph(["Marc Aurel Schnabel"] + list(G.neighbors("Marc Aurel Schnabel")))

# 计算节点的介数中心性

betweenness\_centrality = nx.betweenness\_centrality(subgraph)

# Create a dictionary to store the graph data

graph\_data = {

"nodes": [],

"links": []

}

# Add nodes to the dictionary

for node in subgraph.nodes():

graph\_data["nodes"].append({

"id": node,

"betweenness\_centrality": betweenness\_centrality[node]

})

# Add links to the dictionary

for edge in subgraph.edges(data=True):

order = edge[2]["order"]

graph\_data["links"].append({

"source": edge[0],

"target": edge[1],

"order": order

})

# Save the JSON data to a file

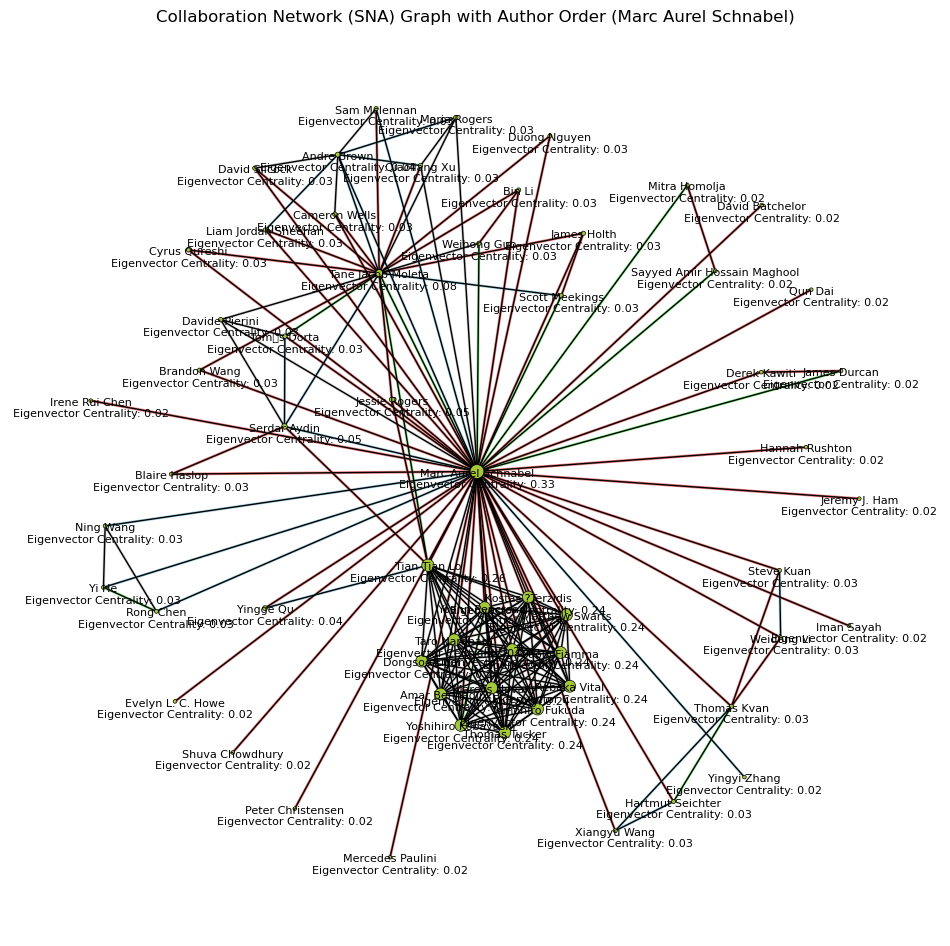
json\_file = "graph\_data.json"

with open(json\_file, "w") as f:

json.dump(graph\_data, f, indent=4)

print(f"Graph data has been saved to {json\_file}")

特徵中心性程式碼



import networkx as nx

import pandas as pd

import matplotlib.pyplot as plt

# 创建一个空的无向图

G = nx.Graph()

# 从Excel文件加载数据

excel\_file = "CADVis\_Author.xlsx" # 替换为您的Excel文件路径

df = pd.read\_excel(excel\_file)

# 根据相同的 work\_id 创建节点和边

for work\_id, group in df.groupby("work\_id"):

authors = group["Fullname"].tolist()

order = group["Order"].tolist()

# 添加节点（作者）

for author in authors:

G.add\_node(author) # 添加节点（作者）

# 添加带有作者顺序的边（合作关系）

for i in range(len(authors)):

for j in range(i + 1, len(authors)):

G.add\_edge(authors[i], authors[j], order=order[i]) # 使用order[i]来表示作者顺序

# 创建一个子图，只包括与Marc Aurel Schnabel相关的节点和边

subgraph = G.subgraph(["Marc Aurel Schnabel"] + list(G.neighbors("Marc Aurel Schnabel")))

# 计算节点的特征向量中心性

eigenvector\_centrality = nx.eigenvector\_centrality(subgraph)

# 计算节点的大小，根据特征向量中心性调整

node\_size = [300 \* eigenvector\_centrality[node] for node in subgraph.nodes()]

# 可视化SNA图

plt.figure(figsize=(12, 12))

# 自定义布局

pos = nx.spring\_layout(subgraph, seed=42)

# 绘制边，根据作者顺序添加不同颜色的边（浅色系）

for edge in subgraph.edges(data=True):

order = edge[2]["order"]

if order == 1:

edge\_color = "lightcoral" # 浅红色

elif order == 2:

edge\_color = "lightblue" # 浅蓝色

elif order == 3:

edge\_color = "lightgreen" # 浅绿色

else:

edge\_color = "lightgray" # 浅灰色

nx.draw\_networkx\_edges(subgraph, pos, edgelist=[(edge[0], edge[1])], edge\_color=edge\_color, width=2)

# 绘制节点，节点大小基于特征向量中心性，节点颜色为芥末色

nx.draw(subgraph, pos, node\_size=node\_size, node\_color='#A4C639', with\_labels=False, edgecolors='black', linewidths=0.5)

# 绘制节点标签（特征向量中心性），但调整它们的位置以避免重叠，并设置文字颜色为深黑色

label\_pos = {k: (v[0], v[1] - 0.02) for k, v in pos.items()}

labels = {node: f"{node}\nEigenvector Centrality: {eigenvector\_centrality[node]:.2f}" for node in subgraph.nodes()}

nx.draw\_networkx\_labels(subgraph, label\_pos, labels=labels, font\_size=8, font\_color='#000000')

plt.title("Collaboration Network (SNA) Graph with Author Order (Marc Aurel Schnabel)")

plt.axis('off') # 关闭坐标轴

plt.show()

**轉成JSON檔**

import networkx as nx

import pandas as pd

import json

# 创建一个空的无向图

G = nx.Graph()

# 从Excel文件加载数据

excel\_file = "CADVis\_Author.xlsx" # 替换为您的Excel文件路径

df = pd.read\_excel(excel\_file)

# 根据相同的 work\_id 创建节点和边

for work\_id, group in df.groupby("work\_id"):

authors = group["Fullname"].tolist()

order = group["Order"].tolist()

# 添加节点（作者）

for author in authors:

G.add\_node(author) # 添加节点（作者）

# 添加带有作者顺序的边（合作关系）

for i in range(len(authors)):

for j in range(i + 1, len(authors)):

G.add\_edge(authors[i], authors[j], order=order[i]) # 使用order[i]来表示作者顺序

# 创建一个子图，只包括与Marc Aurel Schnabel相关的节点和边

subgraph = G.subgraph(["Marc Aurel Schnabel"] + list(G.neighbors("Marc Aurel Schnabel")))

# 计算节点的特征向量中心性

eigenvector\_centrality = nx.eigenvector\_centrality(subgraph)

# Create a dictionary to store the graph data

graph\_data = {

"nodes": [],

"links": []

}

# Add nodes to the dictionary

for node in subgraph.nodes():

graph\_data["nodes"].append({

"id": node,

"eigenvector\_centrality": eigenvector\_centrality[node]

})

# Add links to the dictionary

for edge in subgraph.edges(data=True):

order = edge[2]["order"]

graph\_data["links"].append({

"source": edge[0],

"target": edge[1],

"order": order

})

# Save the JSON data to a file

json\_file = "graph\_data.json"

with open(json\_file, "w") as f:

json.dump(graph\_data, f, indent=4)

print(f"Graph data has been saved to {json\_file}")