Analysis

May 6, 2024

[1]:
```
import scipy.io
import numpy as np
import networkx as nx
import os
import pandas as pd
from sklearn.cluster import SpectralClustering
# import community  # Install using: pip install python-louvain
import matplotlib.pyplot as plt
import re
```

[2]:
```
excel_file = 'fungal_networks_data/list_of_fungal_networks.xlsx'
fnd_df = pd.read_excel(excel_file, index_col='code')

filename = 'Pp_M_Tokyo_U_N_26h_1.txt'
substrate = fnd_df['substrate']
fnd_df
```

```
/opt/conda/lib/python3.10/site-packages/openpyxl/worksheet/_read_only.py:79:
UserWarning: Unknown extension is not supported and will be removed
  for idx, row in parser.parse():
```

[2]:

code	species	size	resources	grazing	interaction	time point	repeat	include	substrate	-K2	comments
Pp_M_Tokyo_U_N_26h_1	Pp	M	Tokyo	U	N	26h	1	Y	A	-k2	NaN
Pp_M_Tokyo_U_N_26h_2	Pp	M	Tokyo	U	N	26h					
Pp_M_Tokyo_U_N_26h_3	Pp	M	Tokyo	U	N	26h					
Pp_M_Tokyo_U_N_26h_4	Pp	M	Tokyo	U	N	26h					
Pp_M_Tokyo_U_N_26h_5	Pp	M	Tokyo	U	N	26h					
…	…	…	…	…	…	…	…	…	…	…	…
Pv_M_I_U_Hf_18d_3	Pv	M	I	U	Hf	18d					
Pv_M_I_U_Hf_22d_3	Pv	M	I	U	Hf	22d					
Pv_M_I_U_Hf_26d_3	Pv	M	I	U	Hf	26d					
Pv_M_I_U_Hf_38d_3	Pv	M	I	U	Hf	38d					
Pv_M_I_U_Hf_48d_3	Pv	M	I	U	Hf	48d					

```
Pp_M_Tokyo_U_N_26h_2       2       Y         A   -k2      NaN
Pp_M_Tokyo_U_N_26h_3       3     NaN         A  full      NaN
Pp_M_Tokyo_U_N_26h_4       4       Y         A   -k2      NaN
Pp_M_Tokyo_U_N_26h_5       5     NaN         A  full      NaN
…                        …      …        …    …       …
Pv_M_I_U_Hf_18d_3          3       Y         S   -k2      NaN
Pv_M_I_U_Hf_22d_3          3       Y         S   -k2      NaN
Pv_M_I_U_Hf_26d_3          3       Y         S   -k2      NaN
Pv_M_I_U_Hf_38d_3          3       Y         S   -k2      NaN
Pv_M_I_U_Hf_48d_3          3       Y         S   -k2      NaN

                                                                citation
code
Pp_M_Tokyo_U_N_26h_1  Tero, A., Takagi, S., Saigusa, T., Ito, K., Be…
Pp_M_Tokyo_U_N_26h_2  Tero, A., Takagi, S., Saigusa, T., Ito, K., Be…
Pp_M_Tokyo_U_N_26h_3  Tero, A., Takagi, S., Saigusa, T., Ito, K., Be…
Pp_M_Tokyo_U_N_26h_4  Tero, A., Takagi, S., Saigusa, T., Ito, K., Be…
Pp_M_Tokyo_U_N_26h_5  Tero, A., Takagi, S., Saigusa, T., Ito, K., Be…
…                                                                  …
Pv_M_I_U_Hf_18d_3     Rotheray, T.D., Jones, T.H., Fricker, M.D. and…
Pv_M_I_U_Hf_22d_3     Rotheray, T.D., Jones, T.H., Fricker, M.D. and…
Pv_M_I_U_Hf_26d_3     Rotheray, T.D., Jones, T.H., Fricker, M.D. and…
Pv_M_I_U_Hf_38d_3     Rotheray, T.D., Jones, T.H., Fricker, M.D. and…
Pv_M_I_U_Hf_48d_3     Rotheray, T.D., Jones, T.H., Fricker, M.D. and…

[518 rows x 12 columns]
```

[3]:
```python
class TreeNode:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

color_dict = {
    'A': 'blue',
    'B': 'red',
    'S': 'green',
}

node1 = TreeNode('Pp_M_Tokyo_U_N_26h_1.txt')
color_dict[fnd_df.at[node1.value[:-4], 'substrate']]
```

[3]:
```
'blue'
```

[4]:
```python
##############################################################################################
#                                        ALL FUNCTIONS
##############################################################################################
```

```python
def get_graph_from_file(file, flag):
    if flag:
        print(f'Reading the network file: {file}\n')

    return nx.read_weighted_edgelist(file, nodetype=str)


def graph_properties(G, flag, file):
    '''
    Parameters:
        G (graph)
        flag (int): flag=1 to print graph stats, 0 else
    '''

    n = len(G.nodes())
    m = len(G.edges())
    C = nx.transitivity(G)
    degrees = nx.degree(G)
    kis = [k for _,k in degrees]
    kstd = np.std(kis)
    kmean = np.mean(kis)
    max_edges = n * (n - 1) / 2
    edge_density = m / max_edges
    if nx.is_connected(G):
        average_shortest_path_length = nx.average_shortest_path_length(G,␣
 ↪weight='weight')
    else :
        average_shortest_path_length = 0

    betweenness = nx.betweenness_centrality(G, weight='weight')
    non_zero_betweenness = [value for value in betweenness.values() if value !=␣
 ↪0]
    avg_betweenness_centrality = np.mean(non_zero_betweenness)

    total_weight = sum(data['weight'] for _, _, data in G.edges(data=True))
    average_edge_weight = total_weight / G.number_of_edges()

    if flag:
        print("Number of nodes (n):", n)
        print("Number of edges (m):", m)
        print("Clustering Coefficient (C):", C)
        print("Mean Degree (kmean):", kmean)
        print("Degree STD (kstd):", kstd)
        print("Edge density (D):", edge_density)
        print("Average shortest path length: ", average_shortest_path_length)
        print("Average betweeness centrality: ", avg_betweenness_centrality)
```

```
        print("Average edge weight: ", average_edge_weight)
        print("\n")

    return [C,kstd,kmean,edge_density,average_shortest_path_length,
            avg_betweenness_centrality,average_edge_weight, file]


def get_all_vectors(directory, flag):
    '''
    Parameters:
        directory path (string)
        flag (int): flag=1 to print graph stats, 0 else
    '''
    files = os.listdir(directory)  # Slicing should be done after calling␣
 ↪listdir()
    all_graphs = []
    all_graphs_properties = []

    flag_count = 0
    files = list(set(files))

    for file in files:
        file_pathway = directory+file
        if flag_count >= 5:
            flag = 0
            # break

        if os.path.isfile(file_pathway):
            G = get_graph_from_file(file_pathway, flag)
            p = graph_properties(G, flag, file)
            all_graphs.append(G)
            all_graphs_properties.append(p)
            flag_count += 1

    return all_graphs_properties

def euclidean_distance(point1, point2):
    return np.linalg.norm(np.array(point1) - np.array(point2))


def lower_distance_matrix(df):
    """
    Calculate the lower half of the distance matrix for vectors stored in a␣
 ↪DataFrame using Euclidean

    Args:
        df (pandas.DataFrame): DataFrame containing vectors as rows.
```

```
    Returns:
        pandas.DataFrame: Lower half of the distance matrix.
    """

    num_rows = df.shape[0]
    distance_matrix = np.zeros((num_rows, num_rows))

    # Calculate pairwise distances
    for i in range(num_rows):
        for j in range(i):
            distance_matrix[i, j] = euclidean_distance(df.iloc[i], df.iloc[j])

    # Convert to DataFrame with filename as index
    distance_df = pd.DataFrame(distance_matrix, index=df.index, columns=df.
 ↪index)

    return distance_df
```

[5]:
```
# A Quick Implementation of UPGMA (Unweighted Pair Group Method with Arithmetic␣
 ↪Mean)

# lowest_cell:
#   Locates the smallest cell in the table
def lowest_cell(table):
    # Set default to infinity
    min_cell = float("inf")
    x, y = -1, -1

    # Go through every cell, looking for the lowest
    for i in range(len(table)):
        for j in range(len(table[i])):
            if table[i][j] < min_cell:
                min_cell = table[i][j]
                x, y = i, j

    # Return the x, y co-ordinate of cell
    return x, y


# join_labels:
#   Combines two labels in a list of labels
def join_labels(labels, a, b):
    # Swap if the indices are not ordered
    if b < a:
        a, b = b, a
```

```
    # Join the labels in the first index
    labels[a] = [labels[a],labels[b]]

    # Remove the (now redundant) label in the second index
    del labels[b]

def join_labels(labels, a, b):
    # Swap if the indices are not ordered
    if b < a:
        a, b = b, a

    # Join the labels in the first index
    labels[a] = [labels[a],labels[b]]

    # Remove the (now redundant) label in the second index
    del labels[b]



# join_table:
#   Joins the entries of a table on the cell (a, b) by averaging their data␣
 ↪entries
def join_table(table, a, b):
    # Swap if the indices are not ordered
    if b < a:
        a, b = b, a

    # For the lower index, reconstruct the entire row (A, i), where i < A
    row = []
    for i in range(0, a):
        row.append((table[a][i] + table[b][i])/2)
    table[a] = row

    # Then, reconstruct the entire column (i, A), where i > A
    #   Note: Since the matrix is lower triangular, row b only contains values␣
 ↪for indices < b
    for i in range(a+1, b):
        table[i][a] = (table[i][a]+table[b][i])/2

    #   We get the rest of the values from row i
    for i in range(b+1, len(table)):
        table[i][a] = (table[i][a]+table[i][b])/2
        # Remove the (now redundant) second index column entry
        del table[i][b]

    # Remove the (now redundant) second index row
    del table[b]
```

```
# UPGMA:
#   Runs the UPGMA algorithm on a labelled table
def UPGMA(table, labels):
    # Until all labels have been joined...
    while len(labels) > 1:
        # Locate lowest cell in the table
        x, y = lowest_cell(table)

        # Join the table on the cell co-ordinates
        join_table(table, x, y)

        # Update the labels accordingly
        join_labels(labels, x, y)

    # Return the final label
    return labels[0]


## A test using an example calculation from http://www.nmsr.org/upgma.htm

# alpha_labels:
#   Makes labels from a starting letter to an ending letter
def alpha_labels(start, end):
    labels = []
    for i in range(ord(start), ord(end)+1):
        labels.append(i)
    return labels

def lower_distance_to_M_data(lower_distance_df):
    """
    Convert lower distance DataFrame to M_data format for UPGMA.

    Args:
        lower_distance_df (pandas.DataFrame): Lower half of the distance matrix.

    Returns:
        list: M_data list for UPGMA.
    """
    M_data = []
    num_rows = lower_distance_df.shape[0]
    for i in range(num_rows):
        row_data = []
        for j in range(i):
            row_data.append(lower_distance_df.iloc[i, j])
        M_data.append(row_data)
```

```
    return M_data

class TreeNode:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

def list_to_binary_tree(lst):
    if isinstance(lst, str):
        return TreeNode(lst)

    root = TreeNode(None)
    root.left = list_to_binary_tree(lst[0])

    if len(lst) == 2:
        root.right = list_to_binary_tree(lst[1])

    return root


def get_node_color(node):
    if node.value is not None:
        print(node.value[:-4])
        return color_dict[fnd_df.at[node.value[:-4], 'substrate']]
    else:
        return 'black'

def plot_tree(node, x, y, spacing, ax=None):
    if ax is None:
        fig, ax = plt.subplots()
    if node is not None:
        ax.plot(x, y, 'o', markersize=12, color='black')  # Plot node
        ax.text(x, y, str(node.value), verticalalignment='center',␣
 ↪horizontalalignment='center')  # Add node value as text
        if node.left is not None:
            ax.plot([x, x - spacing], [y, y - 1], '-k')  # Plot left edge
            plot_tree(node.left, x - spacing, y - 1, spacing / 2, ax)  #␣
 ↪Recursively plot left subtree
        if node.right is not None:
            ax.plot([x, x + spacing], [y, y - 1], '-k')  # Plot right edge
            plot_tree(node.right, x + spacing, y - 1, spacing / 2, ax)  #␣
 ↪Recursively plot right subtree

def replace_with_ln(x):
    if not isinstance(x, str):
        if x > 10:
```

```
            return np.log(x)
        return x
    else:
        return x
```

[5]:
```
####### Helpers for clustering #######

# def louvain_clustering(G, flag):
#     partition = community.best_partition(G)
#     num_clusters = len(set(partition.values())) # Count the number of unique␣
 ↪community IDs\

#     if flag:
#         # Draw the graph with nodes colored by community
#         pos = nx.spring_layout(G)  # Positions for all nodes
#         plt.figure(figsize=(8, 6))

#         # Draw nodes, coloring them by community
#         node_colors = [partition[n] for n in G.nodes()]
#         nx.draw(G, pos, node_color=node_colors, cmap=plt.cm.tab10,␣
 ↪node_size=20)

#         plt.title("Graph with Nodes Colored by Community (Louvain Method)")
#         plt.show()

#     return partition, num_clusters
```

[6]:
```
# G = nx.karate_club_graph
```

[7]:
```
# fname1 = 'fungal_networks_data/Conductance_Text_Files/Sc_M_I_U_N_42d_1.txt'
# G = get_graph_from_file(fname1, fname1)
```

[8]:
```
#########    TEST HELPER FUNCTIONS FOR ONE FILE    #########

# partition, num_clusters = louvain_clustering(G, 1)
# graph_properties(G, 1, fname1)
# print(f'Number of clusters = {num_clusters}')
```

[ ]:
```
##############################################################################
##############################################################################
##############################################################################
##############################################################################
#########################            CONDUCTANCE          ␣
 ↪#########################
##############################################################################
##############################################################################
##############################################################################
```

```
##############################################################################################
```

[68]:
```
##################     GET ALL GRAPHS AND PROPERTIES     ##################
flag = 0
directory = 'fungal_networks_data/Conductance_Text_Files/'
# gmatrix = get_all_vectors(directory, flag)
gmatrix = pd.read_csv('vectors.csv')
column_names = ['C', 'kstd', 'kmean',␣
 ↪'edge_density','average_shortest_path_length','avg_betweenness_centrality','average_edge_we
df = pd.DataFrame(gmatrix, columns=column_names)
df.set_index('file', inplace=True)

df
```

[68]:

file	C	kstd	kmean	edge_density	\
Rb_M_I_Fc-H_N_20d_1.txt	0.079430	1.627588	2.256983	0.012680	
Pv_M_I+4R_U_N_39d_3.txt	0.061126	1.249432	2.233796	0.002588	
Rb_M_I_U_N_18d_2.txt	0.069333	1.999792	2.423841	0.008053	
Pv_L_I+4xR_U_N_57d_1.txt	0.074269	0.996752	2.641404	0.001855	
Pv_M_I_U_N_31d_1.txt	0.041149	1.175326	2.202294	0.001149	
…	…	…	…	…	
Pv_M_I_U_Hf_22d_2.txt	0.078322	1.431390	2.581844	0.001847	
Pv_M_I+4R_U_N_42d_1.txt	0.052120	1.243313	2.237611	0.001423	
Pv_M_I+R_U_N_18d_1.txt	0.115268	1.382307	2.978304	0.005886	
Pv_M_I_U_N_18d_3.txt	0.046584	1.176709	2.147410	0.004286	
Pv_M_5xI_U_N_22d_3.txt	0.083542	1.122883	2.555283	0.002094	

file	average_shortest_path_length	\
Rb_M_I_Fc-H_N_20d_1.txt	0.000000	
Pv_M_I+4R_U_N_39d_3.txt	0.007653	
Rb_M_I_U_N_18d_2.txt	22159.387399	
Pv_L_I+4xR_U_N_57d_1.txt	0.000000	
Pv_M_I_U_N_31d_1.txt	0.016257	
…	…	
Pv_M_I_U_Hf_22d_2.txt	0.117154	
Pv_M_I+4R_U_N_42d_1.txt	0.017854	
Pv_M_I+R_U_N_18d_1.txt	0.099048	
Pv_M_I_U_N_18d_3.txt	0.007177	
Pv_M_5xI_U_N_22d_3.txt	0.000000	

file	avg_betweenness_centrality	average_edge_weight
Rb_M_I_Fc-H_N_20d_1.txt	0.072076	4255.600195
Pv_M_I+4R_U_N_39d_3.txt	0.027278	0.000411
Rb_M_I_U_N_18d_2.txt	0.056738	3140.542753

```
Pv_L_I+4xR_U_N_57d_1.txt                0.029337              0.006161
Pv_M_I_U_N_31d_1.txt                    0.026421              0.000415
…                                            …                     …
Pv_M_I_U_Hf_22d_2.txt                   0.027449              0.008630
Pv_M_I+4R_U_N_42d_1.txt                 0.032957              0.000658
Pv_M_I+R_U_N_18d_1.txt                  0.040881              0.017027
Pv_M_I_U_N_18d_3.txt                    0.050437              0.000368
Pv_M_5xI_U_N_22d_3.txt                  0.030584              0.002475

[270 rows x 7 columns]
```

[11]:
```
df.to_csv('vectors.csv', index=True)
```

[75]:
```
df = pd.read_csv('distances.csv')

# Apply the function to all elements of the DataFrame
df_ln = df.applymap(replace_with_ln)
df_ln.to_csv('distances_ln.csv', index=True)
```

[69]:
```
###################     GET DISTANCE MATRIX     ###################
lower_distance_df = lower_distance_matrix(df)
# print(lower_distance_df.to_string(float_format=lambda x: "%.2f" % x))

###################     FINISHED     ###################
```

[44]:
```
n = lower_distance_df.shape[0]
symmetric_matrix = np.zeros((n, n))
for i in range(n):
    for j in range(i + 1):
        symmetric_matrix[i, j] = lower_distance_df.iloc[i, j]
        symmetric_matrix[j, i] = lower_distance_df.iloc[i, j]

# Convert symmetric matrix to DataFrame
symmetric_distance_df = pd.DataFrame(symmetric_matrix, index=lower_distance_df.
 ↪index, columns=lower_distance_df.columns)

# Display the symmetric distance DataFrame
symmetric_distance_df
```

[44]:
```
file                    Rb_M_I_Fc-H_N_20d_1.txt  Pv_M_I+4R_U_N_39d_3.txt  \
file
Rb_M_I_Fc-H_N_20d_1.txt                 0.000000              4255.599801
Pv_M_I+4R_U_N_39d_3.txt              4255.599801                 0.000000
Rb_M_I_U_N_18d_2.txt                22187.424438             22380.820306
Pv_L_I+4xR_U_N_57d_1.txt             4255.594098                 0.479854
Pv_M_I_U_N_31d_1.txt                 4255.599805                 0.083426
…                                           …                        …
```

```
Pv_M_I_U_Hf_22d_2.txt                4255.591584                 0.408167
Pv_M_I+4R_U_N_42d_1.txt              4255.599555                 0.016457
Pv_M_I+R_U_N_18d_1.txt               4255.583238                 0.764005
Pv_M_I_U_N_18d_3.txt                 4255.599853                 0.116198
Pv_M_5xI_U_N_22d_3.txt               4255.597761                 0.346330

file                      Rb_M_I_U_N_18d_2.txt  Pv_L_I+4xR_U_N_57d_1.txt  \
file
Rb_M_I_Fc-H_N_20d_1.txt          22187.424438               4255.594098
Pv_M_I+4R_U_N_39d_3.txt          22380.820306                  0.479854
Rb_M_I_U_N_18d_2.txt                 0.000000              22380.827086
Pv_L_I+4xR_U_N_57d_1.txt         22380.827086                  0.000000
Pv_M_I_U_N_31d_1.txt             22380.811789                  0.475510
…                                       …                         …
Pv_M_I_U_Hf_22d_2.txt            22380.710729                  0.454102
Pv_M_I+4R_U_N_42d_1.txt          22380.810171                  0.474019
Pv_M_I+R_U_N_18d_1.txt           22380.727485                  0.523367
Pv_M_I_U_N_18d_3.txt             22380.820786                  0.526988
Pv_M_5xI_U_N_22d_3.txt           22380.827597                  0.153059

file                      Pv_M_I_U_N_31d_1.txt  Rb_M_I_U_N_26d_3.txt  \
file
Rb_M_I_Fc-H_N_20d_1.txt           4255.599805          13109.480470
Pv_M_I+4R_U_N_39d_3.txt              0.083426          13047.500922
Rb_M_I_U_N_18d_2.txt             22380.811789           9334.383427
Pv_L_I+4xR_U_N_57d_1.txt             0.475510          13047.507640
Pv_M_I_U_N_31d_1.txt                 0.000000          13047.492416
…                                       …                    …
Pv_M_I_U_Hf_22d_2.txt                0.470381          13047.391406
Pv_M_I+4R_U_N_42d_1.txt              0.077687          13047.490797
Pv_M_I+R_U_N_18d_1.txt               0.811103          13047.408071
Pv_M_I_U_N_18d_3.txt                 0.060932          13047.501404
Pv_M_5xI_U_N_22d_3.txt               0.359771          13047.508182

file                      Pp_M_Tokyo_U_N_26h_1.txt  Pv_M_I+4R_U_N_36d_2.txt  \
file
Rb_M_I_Fc-H_N_20d_1.txt               77439.075233               4255.596544
Pv_M_I+4R_U_N_39d_3.txt               77899.735886                  0.152118
Rb_M_I_U_N_18d_2.txt                  55519.328401              22380.772713
Pv_L_I+4xR_U_N_57d_1.txt              77899.742686                  0.344897
Pv_M_I_U_N_31d_1.txt                  77899.727361                  0.175327
…                                            …                         …
Pv_M_I_U_Hf_22d_2.txt                 77899.626277                  0.320894
Pv_M_I+4R_U_N_42d_1.txt               77899.725745                  0.144057
Pv_M_I+R_U_N_18d_1.txt                77899.643080                  0.639594
Pv_M_I_U_N_18d_3.txt                  77899.736364                  0.228910
Pv_M_5xI_U_N_22d_3.txt                77899.743186                  0.210226
```

file	Pv_M_I_U_N_27d_1.txt	Pv_L_I+4xR_U_N_36d_4.txt	…	\
file			…	
Rb_M_I_Fc-H_N_20d_1.txt	4255.599854	4255.593781	…	
Pv_M_I+4R_U_N_39d_3.txt	0.094394	0.283851	…	
Rb_M_I_U_N_18d_2.txt	22380.813432	22380.827038	…	
Pv_L_I+4xR_U_N_57d_1.txt	0.484274	0.216259	…	
Pv_M_I_U_N_31d_1.txt	0.014589	0.304099	…	
…	…	… …		
Pv_M_I_U_Hf_22d_2.txt	0.483552	0.305944	…	
Pv_M_I+4R_U_N_42d_1.txt	0.089558	0.279597	…	
Pv_M_I+R_U_N_18d_1.txt	0.824218	0.535658	…	
Pv_M_I_U_N_18d_3.txt	0.054333	0.357177	…	
Pv_M_5xI_U_N_22d_3.txt	0.370741	0.066788	…	

file	Pp_M_Tokyo_U_N_26h_11.txt	Pv_M_5xI_U_N_33d_3.txt	\
file			
Rb_M_I_Fc-H_N_20d_1.txt	4255.572255	4255.596232	
Pv_M_I+4R_U_N_39d_3.txt	0.922752	0.456321	
Rb_M_I_U_N_18d_2.txt	22380.592636	22380.786547	
Pv_L_I+4xR_U_N_57d_1.txt	0.490105	0.156095	
Pv_M_I_U_N_31d_1.txt	0.926128	0.477296	
…	…	…	
Pv_M_I_U_Hf_22d_2.txt	0.724592	0.313506	
Pv_M_I+4R_U_N_42d_1.txt	0.915231	0.451338	
Pv_M_I+R_U_N_18d_1.txt	0.510405	0.392255	
Pv_M_I_U_N_18d_3.txt	0.977301	0.532227	
Pv_M_5xI_U_N_22d_3.txt	0.597656	0.128384	

file	Pv_M_I_U_Hf_26d_3.txt	Pv_M_I_Fc_Hf_48d_2.txt	\
file			
Rb_M_I_Fc-H_N_20d_1.txt	4255.589776	4255.583939	
Pv_M_I+4R_U_N_39d_3.txt	0.591240	0.450383	
Rb_M_I_U_N_18d_2.txt	22380.690440	22380.825651	
Pv_L_I+4xR_U_N_57d_1.txt	0.542775	0.443469	
Pv_M_I_U_N_31d_1.txt	0.653139	0.514954	
…	…	…	
Pv_M_I_U_Hf_22d_2.txt	0.184921	0.132814	
Pv_M_I+4R_U_N_42d_1.txt	0.589131	0.450742	
Pv_M_I+R_U_N_18d_1.txt	0.269067	0.357774	
Pv_M_I_U_N_18d_3.txt	0.700698	0.561509	
Pv_M_5xI_U_N_22d_3.txt	0.454164	0.328361	

file	Pv_M_I+4R_U_N_27d_1.txt	Pv_M_I_U_Hf_22d_2.txt	\
file			
Rb_M_I_Fc-H_N_20d_1.txt	4255.599709	4255.591584	
Pv_M_I+4R_U_N_39d_3.txt	0.043477	0.408167	

```
Rb_M_I_U_N_18d_2.txt                22380.815728          22380.710729
Pv_L_I+4xR_U_N_57d_1.txt                0.450236              0.454102
Pv_M_I_U_N_31d_1.txt                    0.112891              0.470381
…                                            …                     …
Pv_M_I_U_Hf_22d_2.txt                   0.367124              0.000000
Pv_M_I+4R_U_N_42d_1.txt                 0.042652              0.405599
Pv_M_I+R_U_N_18d_1.txt                  0.720917              0.401932
Pv_M_I_U_N_18d_3.txt                    0.154849              0.517011
Pv_M_5xI_U_N_22d_3.txt                  0.311935              0.331183

file                    Pv_M_I+4R_U_N_42d_1.txt  Pv_M_I+R_U_N_18d_1.txt  \
file
Rb_M_I_Fc-H_N_20d_1.txt             4255.599555           4255.583238
Pv_M_I+4R_U_N_39d_3.txt                0.016457              0.764005
Rb_M_I_U_N_18d_2.txt               22380.810171          22380.727485
Pv_L_I+4xR_U_N_57d_1.txt               0.474019              0.523367
Pv_M_I_U_N_31d_1.txt                   0.077687              0.811103
…                                           …                     …
Pv_M_I_U_Hf_22d_2.txt                  0.405599              0.401932
Pv_M_I+4R_U_N_42d_1.txt                0.000000              0.760839
Pv_M_I+R_U_N_18d_1.txt                 0.760839              0.000000
Pv_M_I_U_N_18d_3.txt                   0.114152              0.863819
Pv_M_5xI_U_N_22d_3.txt                 0.341664              0.507343

file                    Pv_M_I_U_N_18d_3.txt  Pv_M_5xI_U_N_22d_3.txt
file
Rb_M_I_Fc-H_N_20d_1.txt           4255.599853           4255.597761
Pv_M_I+4R_U_N_39d_3.txt              0.116198              0.346330
Rb_M_I_U_N_18d_2.txt             22380.820786          22380.827597
Pv_L_I+4xR_U_N_57d_1.txt             0.526988              0.153059
Pv_M_I_U_N_31d_1.txt                 0.060932              0.359771
…                                         …                     …
Pv_M_I_U_Hf_22d_2.txt                0.517011              0.331183
Pv_M_I+4R_U_N_42d_1.txt              0.114152              0.341664
Pv_M_I+R_U_N_18d_1.txt               0.863819              0.507343
Pv_M_I_U_N_18d_3.txt                 0.000000              0.413615
Pv_M_5xI_U_N_22d_3.txt               0.413615              0.000000

[270 rows x 270 columns]
```

[59]:
```python
def func(row):
    index_name = row.name
    return fnd_df.at[index_name[:-4], 'substrate']

# Apply the function to create a new column
symmetric_distance_df['substrate'] = symmetric_distance_df.apply(func, axis=1)
```

```
symmetric_distance_df.to_csv('distances_substrate.csv', index_label='index',␣
 ↪header=True)
```

[50]:
```
# np.savetxt('distances.csv', symmetric_distance_df, delimiter=',')
# # Assuming df is your DataFrame
# symmetric_distance_df.to_csv('distances.csv', index_label='index',␣
 ↪header=True)
```

[48]:
```
import scipy.io
import numpy as np

# Save the distance matrix and row/column names to a .mat file
scipy.io.savemat('distance_matrix.mat', {'distances': symmetric_distance_df,␣
 ↪'row_names': symmetric_distance_df.index.tolist(),
                                          'column_names': symmetric_distance_df.
 ↪columns.tolist()})
```

[13]:
```
###################     SMALL TEST ON THING       ###################

M_labels = lower_distance_df.index.tolist()
M = lower_distance_to_M_data(lower_distance_df)
lst = UPGMA(M, M_labels)
```

[65]:
```
symmetric_distance_df = symmetric_distance_df.drop(columns=['substrate'])
```

[29]:
```
from scipy.cluster.hierarchy import linkage, dendrogram
import matplotlib.pyplot as plt
from scipy.spatial.distance import squareform


# Your distance matrix
distances = pd.read_csv('distances_ln.csv')
# # distances = distances.drop(df.columns[[1]], axis=1)
dist_sub = distances

dist_sub.set_index('index', inplace=True)

unnamed_columns = [col for col in dist_sub.columns if col.startswith('Unnamed')]
dist_sub = dist_sub.drop(columns=unnamed_columns)

dist_sub
```

[29]:

index	Rb_M_I_Fc-H_N_20d_1.txt	Pv_M_I+4R_U_N_39d_3.txt	\
Rb_M_I_Fc-H_N_20d_1.txt	0.000000	8.355991	
Pv_M_I+4R_U_N_39d_3.txt	8.355991	0.000000	
Rb_M_I_U_N_18d_2.txt	10.007281	10.015960	

Pv_L_I+4xR_U_N_57d_1.txt	8.355990	0.479854
Pv_M_I_U_N_31d_1.txt	8.355991	0.083426
…	…	…
Pv_M_I_U_Hf_22d_2.txt	8.355989	0.408167
Pv_M_I+4R_U_N_42d_1.txt	8.355991	0.016457
Pv_M_I+R_U_N_18d_1.txt	8.355987	0.764005
Pv_M_I_U_N_18d_3.txt	8.355991	0.116198
Pv_M_5xI_U_N_22d_3.txt	8.355991	0.346330

index	Rb_M_I_U_N_18d_2.txt	Pv_L_I+4xR_U_N_57d_1.txt	\
Rb_M_I_Fc-H_N_20d_1.txt	10.007281	8.355990	
Pv_M_I+4R_U_N_39d_3.txt	10.015960	0.479854	
Rb_M_I_U_N_18d_2.txt	0.000000	10.015960	
Pv_L_I+4xR_U_N_57d_1.txt	10.015960	0.000000	
Pv_M_I_U_N_31d_1.txt	10.015959	0.475510	
…	…	…	
Pv_M_I_U_Hf_22d_2.txt	10.015955	0.454102	
Pv_M_I+4R_U_N_42d_1.txt	10.015959	0.474019	
Pv_M_I+R_U_N_18d_1.txt	10.015955	0.523367	
Pv_M_I_U_N_18d_3.txt	10.015960	0.526988	
Pv_M_5xI_U_N_22d_3.txt	10.015960	0.153059	

index	Pv_M_I_U_N_31d_1.txt	Rb_M_I_U_N_26d_3.txt	\
Rb_M_I_Fc-H_N_20d_1.txt	8.355991	9.481091	
Pv_M_I+4R_U_N_39d_3.txt	0.083426	9.476352	
Rb_M_I_U_N_18d_2.txt	10.015959	9.141460	
Pv_L_I+4xR_U_N_57d_1.txt	0.475510	9.476352	
Pv_M_I_U_N_31d_1.txt	0.000000	9.476351	
…	…	…	
Pv_M_I_U_Hf_22d_2.txt	0.470381	9.476344	
Pv_M_I+4R_U_N_42d_1.txt	0.077687	9.476351	
Pv_M_I+R_U_N_18d_1.txt	0.811103	9.476345	
Pv_M_I_U_N_18d_3.txt	0.060932	9.476352	
Pv_M_5xI_U_N_22d_3.txt	0.359771	9.476352	

index	Pp_M_Tokyo_U_N_26h_1.txt	Pv_M_I+4R_U_N_36d_2.txt	\
Rb_M_I_Fc-H_N_20d_1.txt	11.257247	8.355990	
Pv_M_I+4R_U_N_39d_3.txt	11.263178	0.152118	
Rb_M_I_U_N_18d_2.txt	10.924486	10.015958	
Pv_L_I+4xR_U_N_57d_1.txt	11.263178	0.344897	
Pv_M_I_U_N_31d_1.txt	11.263178	0.175327	
…	…	…	
Pv_M_I_U_Hf_22d_2.txt	11.263176	0.320894	
Pv_M_I+4R_U_N_42d_1.txt	11.263178	0.144057	

```
Pv_M_I+R_U_N_18d_1.txt                  11.263177                  0.639594
Pv_M_I_U_N_18d_3.txt                    11.263178                  0.228910
Pv_M_5xI_U_N_22d_3.txt                  11.263178                  0.210226

                          Pv_M_I_U_N_27d_1.txt  Pv_L_I+4xR_U_N_36d_4.txt  … \
index                                                                     …
Rb_M_I_Fc-H_N_20d_1.txt               8.355991                  8.355990  …
Pv_M_I+4R_U_N_39d_3.txt               0.094394                  0.283851  …
Rb_M_I_U_N_18d_2.txt                 10.015959                 10.015960  …
Pv_L_I+4xR_U_N_57d_1.txt              0.484274                  0.216259  …
Pv_M_I_U_N_31d_1.txt                  0.014589                  0.304099  …
…                                        …                         …  …
Pv_M_I_U_Hf_22d_2.txt                 0.483552                  0.305944  …
Pv_M_I+4R_U_N_42d_1.txt               0.089558                  0.279597  …
Pv_M_I+R_U_N_18d_1.txt                0.824218                  0.535658  …
Pv_M_I_U_N_18d_3.txt                  0.054333                  0.357177  …
Pv_M_5xI_U_N_22d_3.txt                0.370741                  0.066788  …

                          Pp_M_Tokyo_U_N_26h_11.txt  Pv_M_5xI_U_N_33d_3.txt  \
index
Rb_M_I_Fc-H_N_20d_1.txt                    8.355985                8.355990
Pv_M_I+4R_U_N_39d_3.txt                    0.922752                0.456321
Rb_M_I_U_N_18d_2.txt                      10.015949               10.015958
Pv_L_I+4xR_U_N_57d_1.txt                   0.490105                0.156095
Pv_M_I_U_N_31d_1.txt                       0.926128                0.477296
…                                             …                       …
Pv_M_I_U_Hf_22d_2.txt                      0.724592                0.313506
Pv_M_I+4R_U_N_42d_1.txt                    0.915231                0.451338
Pv_M_I+R_U_N_18d_1.txt                     0.510405                0.392255
Pv_M_I_U_N_18d_3.txt                       0.977301                0.532227
Pv_M_5xI_U_N_22d_3.txt                     0.597656                0.128384

                          Pv_M_I_U_Hf_26d_3.txt  Pv_M_I_Fc_Hf_48d_2.txt  \
index
Rb_M_I_Fc-H_N_20d_1.txt                8.355989                8.355987
Pv_M_I+4R_U_N_39d_3.txt                0.591240                0.450383
Rb_M_I_U_N_18d_2.txt                  10.015954               10.015960
Pv_L_I+4xR_U_N_57d_1.txt               0.542775                0.443469
Pv_M_I_U_N_31d_1.txt                   0.653139                0.514954
…                                         …                       …
Pv_M_I_U_Hf_22d_2.txt                  0.184921                0.132814
Pv_M_I+4R_U_N_42d_1.txt                0.589131                0.450742
Pv_M_I+R_U_N_18d_1.txt                 0.269067                0.357774
Pv_M_I_U_N_18d_3.txt                   0.700698                0.561509
Pv_M_5xI_U_N_22d_3.txt                 0.454164                0.328361

                          Pv_M_I+4R_U_N_27d_1.txt  Pv_M_I_U_Hf_22d_2.txt  \
```

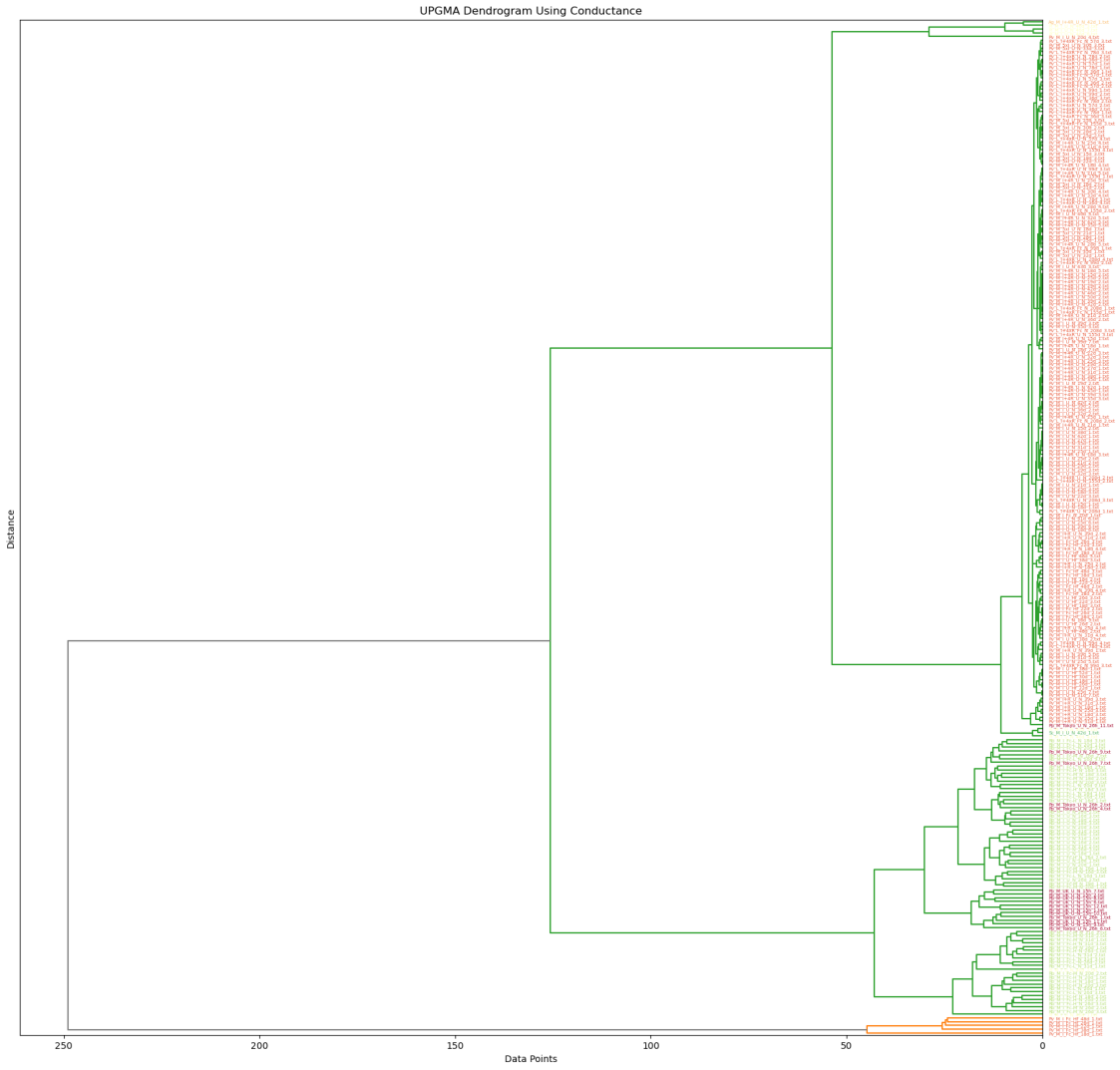
```
index
Rb_M_I_Fc-H_N_20d_1.txt                 8.355991                8.355989
Pv_M_I+4R_U_N_39d_3.txt                 0.043477                0.408167
Rb_M_I_U_N_18d_2.txt                   10.015959               10.015955
Pv_L_I+4xR_U_N_57d_1.txt                0.450236                0.454102
Pv_M_I_U_N_31d_1.txt                    0.112891                0.470381
…                                            …                       …
Pv_M_I_U_Hf_22d_2.txt                   0.367124                0.000000
Pv_M_I+4R_U_N_42d_1.txt                 0.042652                0.405599
Pv_M_I+R_U_N_18d_1.txt                  0.720917                0.401932
Pv_M_I_U_N_18d_3.txt                    0.154849                0.517011
Pv_M_5xI_U_N_22d_3.txt                  0.311935                0.331183

                        Pv_M_I+4R_U_N_42d_1.txt  Pv_M_I+R_U_N_18d_1.txt  \
index
Rb_M_I_Fc-H_N_20d_1.txt                 8.355991                8.355987
Pv_M_I+4R_U_N_39d_3.txt                 0.016457                0.764005
Rb_M_I_U_N_18d_2.txt                   10.015959               10.015955
Pv_L_I+4xR_U_N_57d_1.txt                0.474019                0.523367
Pv_M_I_U_N_31d_1.txt                    0.077687                0.811103
…                                            …                       …
Pv_M_I_U_Hf_22d_2.txt                   0.405599                0.401932
Pv_M_I+4R_U_N_42d_1.txt                 0.000000                0.760839
Pv_M_I+R_U_N_18d_1.txt                  0.760839                0.000000
Pv_M_I_U_N_18d_3.txt                    0.114152                0.863819
Pv_M_5xI_U_N_22d_3.txt                  0.341664                0.507343

                        Pv_M_I_U_N_18d_3.txt  Pv_M_5xI_U_N_22d_3.txt
index
Rb_M_I_Fc-H_N_20d_1.txt             8.355991              8.355991
Pv_M_I+4R_U_N_39d_3.txt             0.116198              0.346330
Rb_M_I_U_N_18d_2.txt               10.015960             10.015960
Pv_L_I+4xR_U_N_57d_1.txt            0.526988              0.153059
Pv_M_I_U_N_31d_1.txt                0.060932              0.359771
…                                        …                     …
Pv_M_I_U_Hf_22d_2.txt               0.517011              0.331183
Pv_M_I+4R_U_N_42d_1.txt             0.114152              0.341664
Pv_M_I+R_U_N_18d_1.txt              0.863819              0.507343
Pv_M_I_U_N_18d_3.txt                0.000000              0.413615
Pv_M_5xI_U_N_22d_3.txt              0.413615              0.000000

[270 rows x 270 columns]
```

[116]:
```
# graph

import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage
```

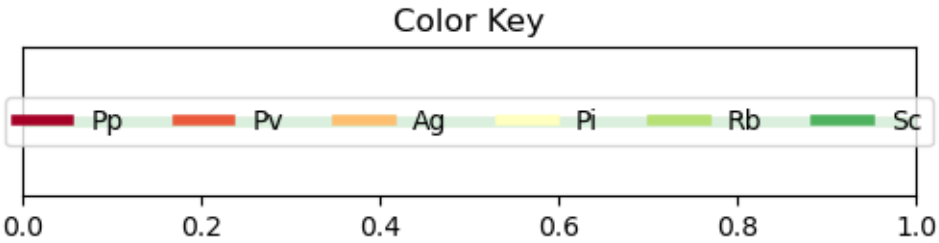
```
color_mapping = {
    'Pp': 0,
    'Pv': 1,
    'Ag': 2,
    'Pi': 3,
    'Rb': 4,
    'Sc': 5,
    'none': 6
}

def get_leaf_color(filename):
    if filename[-4:] == '.txt':
        value = filename[0:2]
        return color_mapping[value]
    else:
        return color_mapping['none']

# Assuming 'df' contains attributes for each leaf node
# Pass the leaf attributes to the color calculation function
leaf_colors = [get_leaf_color(index_name) for index_name in dist_sub.index]

# Your distance matrix
distances = pd.read_csv('conductance_distances_ln.csv')
distances.set_index('index', inplace=True)
distances = distances.drop(distances.columns[[0]], axis=1)

tree = linkage(distances, method='average',optimal_ordering=False)  # Use␣
 ↪'average' method for UPGMA

# Plot dendrogram
plt.figure(figsize=(20, 20))
# dendrogram(tree, labels=distances.index, leaf_rotation=90)
dendrogram(tree, labels=distances.index, leaf_rotation=0, orientation="left",␣
 ↪color_threshold=240, above_threshold_color='grey')
plt.title('UPGMA Dendrogram Using Conductance')
plt.xlabel('Data Points')
plt.ylabel('Distance')
# plt.show()


my_palette = plt.cm.get_cmap('RdYlGn', 7)

ax = plt.gca()
xlbls = ax.get_ymajorticklabels()
num=-1
for lbl in xlbls:
```

```
    val = get_leaf_color(lbl.get_text())
    lbl.set_color(my_palette(val))

# Show the graph
plt.show()

# Define labels for each color
color_labels = ['Pp','Pv','Ag','Pi','Rb','Sc']

# Create a dictionary to map colors to labels
color_map = {my_palette(i): label for i, label in enumerate(color_labels)}

# Display the colors and their corresponding labels
for color, label in color_map.items():
    print(f'{label}: {color}')

# Plot example colors
plt.figure(figsize=(6, 1))
for color, label in color_map.items():
    plt.plot([0, 1], [0, 0], color=color, label=label, linewidth=4)
plt.xlim(0, 1)
plt.ylim(-0.5, 0.5)
plt.yticks([])
plt.legend(loc='center', ncol=len(color_labels))
plt.title('Color Key')
plt.show()
```

```
/tmp/ipykernel_375/3232813409.py:32: ClusterWarning: scipy.cluster: The
symmetric non-negative hollow observation matrix looks suspiciously like an
uncondensed distance matrix
  tree = linkage(distances, method='average',optimal_ordering=False)  # Use
'average' method for UPGMA
```




```
Pp: (0.6470588235294118, 0.0, 0.14901960784313725, 1.0)
Pv: (0.918954248366013, 0.3477124183006535, 0.22614379084967318, 1.0)
Ag: (0.9934640522875817, 0.7477124183006535, 0.4352941176470587, 1.0)
Pi: (1.0, 1.0, 0.7490196078431373, 1.0)
Rb: (0.7176470588235295, 0.8797385620915033, 0.45882352941176474, 1.0)
Sc: (0.3006535947712421, 0.6928104575163401, 0.3633986928104576, 1.0)
```



[78]:
```
##############################################################################
##############################################################################
##############################################################################
##############################################################################
##########################            PATH SCORE           ␣
 ↪#########################
##############################################################################
##############################################################################
##############################################################################
##############################################################################
```

[81]:
```
###################     GET ALL GRAPHS AND PROPERTIES     ###################
flag = 0
directory = 'fungal_networks_data/Path_Score_Text_Files/'
# gmatrix = get_all_vectors(directory, flag)
gmatrix = pd.read_csv('path_score_vectors.csv')
column_names = ['C', 'kstd', 'kmean',␣
 ↪'edge_density','average_shortest_path_length','avg_betweenness_centrality','average_edge_we
psvecs = pd.DataFrame(gmatrix, columns=column_names)
psvecs.set_index('file', inplace=True)

psvecs
```

[81]:

file	C	kstd	kmean	edge_density	\
Pv_M_I+4R_U_N_27d_1.txt	0.144913	0.855159	2.785924	0.008194	
Rb_M_I_Fc-L_N_16d_3.txt	0.136957	1.570601	3.195876	0.033290	
Pv_M_I+4R_U_N_29d_3.txt	0.180583	0.736542	2.647619	0.012668	
Pv_M_I_Fc_Hf_38d_2.txt	0.134267	1.056477	2.988372	0.002899	
Pv_M_I_U_N_15d_1.txt	0.124424	0.581344	2.310078	0.018047	
…	…	…	…	…	
Pv_M_5xI_U_N_21d_2.txt	0.119231	0.766758	2.735043	0.004683	
Pv_M_I_U_N_18d_6.txt	0.095137	1.262376	2.637500	0.016588	
Pv_L_I+4xR_U_N_78d_1.txt	0.088156	0.599863	2.758539	0.003253	
Pv_M_I_U_N_32d_3.txt	0.128259	0.694244	2.622222	0.006491	
Pv_M_I+4R_U_N_31d_1.txt	0.149254	0.846110	2.745205	0.007542	

file	average_shortest_path_length	\
Pv_M_I+4R_U_N_27d_1.txt	0.0	
Rb_M_I_Fc-L_N_16d_3.txt	0.0	
Pv_M_I+4R_U_N_29d_3.txt	0.0	
Pv_M_I_Fc_Hf_38d_2.txt	0.0	
Pv_M_I_U_N_15d_1.txt	0.0	

```
…                                                 …
Pv_M_5xI_U_N_21d_2.txt                            0.0
Pv_M_I_U_N_18d_6.txt                              0.0
Pv_L_I+4xR_U_N_78d_1.txt                          0.0
Pv_M_I_U_N_32d_3.txt                              0.0
Pv_M_I+4R_U_N_31d_1.txt                           0.0
```

file	avg_betweenness_centrality	average_edge_weight
Pv_M_I+4R_U_N_27d_1.txt	0.022698	0.009950
Rb_M_I_Fc-L_N_16d_3.txt	0.053408	0.020824
Pv_M_I+4R_U_N_29d_3.txt	0.035083	0.019230
Pv_M_I_Fc_Hf_38d_2.txt	0.019169	0.003332
Pv_M_I_U_N_15d_1.txt	0.001925	0.027958
…	…	…
Pv_M_5xI_U_N_21d_2.txt	0.029372	0.008983
Pv_M_I_U_N_18d_6.txt	0.037684	0.025245
Pv_L_I+4xR_U_N_78d_1.txt	0.029216	0.005701
Pv_M_I_U_N_32d_3.txt	0.031155	0.010983
Pv_M_I+4R_U_N_31d_1.txt	0.019604	0.009367

[270 rows x 7 columns]

[80]:
```
# df.to_csv('path_score_vectors.csv', index=True)
```

[88]:
```
ps_lower_dist = lower_distance_matrix(psvecs)
# ps_lower_dist
```

[103]:
```
####################### MAKE SYMETRICAL ######################
n = ps_lower_dist.shape[0]
ps_symetric_dist = np.zeros((n, n))
for i in range(n):
    for j in range(i + 1):
        ps_symetric_dist[i, j] = ps_lower_dist.iloc[i, j]
        ps_symetric_dist[j, i] = ps_lower_dist.iloc[i, j]

# Convert symmetric matrix to DataFrame
ps_symetric_dist = pd.DataFrame(ps_symetric_dist, index=ps_lower_dist.index,␣
 ↪columns=ps_lower_dist.columns)

# Display the symmetric distance DataFrame
ps_symetric_dist
```

[103]:

file	Pv_M_I+4R_U_N_27d_1.txt	Rb_M_I_Fc-L_N_16d_3.txt	\
file			
Pv_M_I+4R_U_N_27d_1.txt	0.000000	0.825634	
Rb_M_I_Fc-L_N_16d_3.txt	0.825634	0.000000	

```
Pv_M_I+4R_U_N_29d_3.txt                 0.186360                 0.999453
Pv_M_I_Fc_Hf_38d_2.txt                  0.285853                 0.556588
Pv_M_I_U_N_15d_1.txt                    0.550161                 1.329045
…                                          …                        …
Pv_M_5xI_U_N_21d_2.txt                  0.105457                 0.927568
Pv_M_I_U_N_18d_6.txt                    0.436877                 0.639594
Pv_L_I+4xR_U_N_78d_1.txt                0.263120                 1.066627
Pv_M_I_U_N_32d_3.txt                    0.230314                 1.048077
Pv_M_I+4R_U_N_31d_1.txt                 0.042060                 0.854447

file                    Pv_M_I+4R_U_N_29d_3.txt  Pv_M_I_Fc_Hf_38d_2.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                 0.186360                 0.285853
Rb_M_I_Fc-L_N_16d_3.txt                 0.999453                 0.556588
Pv_M_I+4R_U_N_29d_3.txt                 0.000000                 0.470338
Pv_M_I_Fc_Hf_38d_2.txt                  0.470338                 0.000000
Pv_M_I_U_N_15d_1.txt                    0.377332                 0.828894
…                                          …                        …
Pv_M_5xI_U_N_21d_2.txt                  0.111899                 0.385329
Pv_M_I_U_N_18d_6.txt                    0.532882                 0.409936
Pv_L_I+4xR_U_N_78d_1.txt                0.199583                 0.513373
Pv_M_I_U_N_32d_3.txt                    0.072756                 0.515295
Pv_M_I+4R_U_N_31d_1.txt                 0.151238                 0.321974

file                    Pv_M_I_U_N_15d_1.txt  Pv_M_I_Fc_Hf_22d_1.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt              0.550161                 0.035963
Rb_M_I_Fc-L_N_16d_3.txt              1.329045                 0.848524
Pv_M_I+4R_U_N_29d_3.txt              0.377332                 0.177133
Pv_M_I_Fc_Hf_38d_2.txt               0.828894                 0.302399
Pv_M_I_U_N_15d_1.txt                 0.000000                 0.545517
…                                       …                        …
Pv_M_5xI_U_N_21d_2.txt               0.465073                 0.087499
Pv_M_I_U_N_18d_6.txt                 0.757071                 0.471237
Pv_L_I+4xR_U_N_78d_1.txt             0.451924                 0.232973
Pv_M_I_U_N_32d_3.txt                 0.333874                 0.217843
Pv_M_I+4R_U_N_31d_1.txt              0.510708                 0.058807

file                    Pv_M_I_Fc_Hf_22d_3.txt  Pv_M_I+R_U_N_25d_2.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                0.510857                 0.683918
Rb_M_I_Fc-L_N_16d_3.txt                0.450957                 0.205435
Pv_M_I+4R_U_N_29d_3.txt                0.650687                 0.847215
Pv_M_I_Fc_Hf_38d_2.txt                 0.362617                 0.446525
Pv_M_I_U_N_15d_1.txt                   0.923710                 1.155070
…                                         …                        …
Pv_M_5xI_U_N_21d_2.txt                 0.600694                 0.784287
```

```
Pv_M_I_U_N_18d_6.txt                  0.192404                0.442198
Pv_L_I+4xR_U_N_78d_1.txt              0.764886                0.937136
Pv_M_I_U_N_32d_3.txt                  0.693097                0.894882
Pv_M_I+4R_U_N_31d_1.txt               0.522942                0.706595

file                      Pv_L_I+4xR_U_N_155d_1.txt  Rb_M_I_Fc-H_N_16d_3.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                   0.224900                 0.876111
Rb_M_I_Fc-L_N_16d_3.txt                   1.044908                 0.278068
Pv_M_I+4R_U_N_29d_3.txt                   0.120490                 1.026358
Pv_M_I_Fc_Hf_38d_2.txt                    0.499113                 0.659316
Pv_M_I_U_N_15d_1.txt                      0.396883                 1.304863
…                                            …                        …
Pv_M_5xI_U_N_21d_2.txt                    0.121736                 0.972901
Pv_M_I_U_N_18d_6.txt                      0.612275                 0.553889
Pv_L_I+4xR_U_N_78d_1.txt                  0.087742                 1.132182
Pv_M_I_U_N_32d_3.txt                      0.088713                 1.073162
Pv_M_I+4R_U_N_31d_1.txt                   0.204685                 0.894013

file                      …  Pv_L_I+4xR_Fc_N_99d_3.txt  \
file                      …
Pv_M_I+4R_U_N_27d_1.txt   …                   0.166148
Rb_M_I_Fc-L_N_16d_3.txt   …                   0.892280
Pv_M_I+4R_U_N_29d_3.txt   …                   0.252102
Pv_M_I_Fc_Hf_38d_2.txt    …                   0.338369
Pv_M_I_U_N_15d_1.txt      …                   0.606389
…                         …                      …
Pv_M_5xI_U_N_21d_2.txt    …                   0.166587
Pv_M_I_U_N_18d_6.txt      …                   0.593193
Pv_L_I+4xR_U_N_78d_1.txt  …                   0.197739
Pv_M_I_U_N_32d_3.txt      …                   0.276801
Pv_M_I+4R_U_N_31d_1.txt   …                   0.190154

file                      Rb_M_I_Fc-H_N_20d_1.txt  Rb_M_I_Fc-M_N_31d_2.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                   0.855142                 0.618249
Rb_M_I_Fc-L_N_16d_3.txt                   0.428068                 1.351946
Pv_M_I+4R_U_N_29d_3.txt                   0.992467                 0.451047
Pv_M_I_Fc_Hf_38d_2.txt                    0.669687                 0.875011
Pv_M_I_U_N_15d_1.txt                      1.257518                 0.329602
…                                            …                        …
Pv_M_5xI_U_N_21d_2.txt                    0.943964                 0.561759
Pv_M_I_U_N_18d_6.txt                      0.562132                 0.825626
Pv_L_I+4xR_U_N_78d_1.txt                  1.096581                 0.571152
Pv_M_I_U_N_32d_3.txt                      1.036648                 0.456839
Pv_M_I+4R_U_N_31d_1.txt                   0.870829                 0.582019
```

file	Pv_M_I+4R_U_N_15d_1.txt	Rb_M_I_Fc-L_N_31d_1.txt	\
file			
Pv_M_I+4R_U_N_27d_1.txt	0.079561	1.455208	
Rb_M_I_Fc-L_N_16d_3.txt	0.797104	0.894267	
Pv_M_I+4R_U_N_29d_3.txt	0.246994	1.568770	
Pv_M_I_Fc_Hf_38d_2.txt	0.244624	1.288085	
Pv_M_I_U_N_15d_1.txt	0.615036	1.767391	
…	…	…	
Pv_M_5xI_U_N_21d_2.txt	0.152573	1.542095	
Pv_M_I_U_N_18d_6.txt	0.475215	1.051327	
Pv_L_I+4xR_U_N_78d_1.txt	0.272600	1.711029	
Pv_M_I_U_N_32d_3.txt	0.285785	1.614433	
Pv_M_I+4R_U_N_31d_1.txt	0.120149	1.462166	

file	Pv_M_5xI_U_N_21d_2.txt	Pv_M_I_U_N_18d_6.txt	\
file			
Pv_M_I+4R_U_N_27d_1.txt	0.105457	0.436877	
Rb_M_I_Fc-L_N_16d_3.txt	0.927568	0.639594	
Pv_M_I+4R_U_N_29d_3.txt	0.111899	0.532882	
Pv_M_I_Fc_Hf_38d_2.txt	0.385329	0.409936	
Pv_M_I_U_N_15d_1.txt	0.465073	0.757071	
…	…	…	
Pv_M_5xI_U_N_21d_2.txt	0.000000	0.506170	
Pv_M_I_U_N_18d_6.txt	0.506170	0.000000	
Pv_L_I+4xR_U_N_78d_1.txt	0.171419	0.673984	
Pv_M_I_U_N_32d_3.txt	0.134457	0.569607	
Pv_M_I+4R_U_N_31d_1.txt	0.086053	0.434129	

file	Pv_L_I+4xR_U_N_78d_1.txt	Pv_M_I_U_N_32d_3.txt	\
file			
Pv_M_I+4R_U_N_27d_1.txt	0.263120	0.230314	
Rb_M_I_Fc-L_N_16d_3.txt	1.066627	1.048077	
Pv_M_I+4R_U_N_29d_3.txt	0.199583	0.072756	
Pv_M_I_Fc_Hf_38d_2.txt	0.513373	0.515295	
Pv_M_I_U_N_15d_1.txt	0.451924	0.333874	
…	…	…	
Pv_M_5xI_U_N_21d_2.txt	0.171419	0.134457	
Pv_M_I_U_N_18d_6.txt	0.673984	0.569607	
Pv_L_I+4xR_U_N_78d_1.txt	0.000000	0.170706	
Pv_M_I_U_N_32d_3.txt	0.170706	0.000000	
Pv_M_I+4R_U_N_31d_1.txt	0.254308	0.196891	

file	Pv_M_I+4R_U_N_31d_1.txt
file	
Pv_M_I+4R_U_N_27d_1.txt	0.042060
Rb_M_I_Fc-L_N_16d_3.txt	0.854447
Pv_M_I+4R_U_N_29d_3.txt	0.151238

```
Pv_M_I_Fc_Hf_38d_2.txt                  0.321974
Pv_M_I_U_N_15d_1.txt                    0.510708
…                                          …
Pv_M_5xI_U_N_21d_2.txt                  0.086053
Pv_M_I_U_N_18d_6.txt                    0.434129
Pv_L_I+4xR_U_N_78d_1.txt                0.254308
Pv_M_I_U_N_32d_3.txt                    0.196891
Pv_M_I+4R_U_N_31d_1.txt                 0.000000

[270 rows x 270 columns]
```

[104]:
```
ps_symetric_dist_2 = ps_symetric_dist[~ps_symetric_dist.index.duplicated()]
ps_symetric_dist_2
```

[104]:
```
file                    Pv_M_I+4R_U_N_27d_1.txt  Rb_M_I_Fc-L_N_16d_3.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                 0.000000                 0.825634
Rb_M_I_Fc-L_N_16d_3.txt                 0.825634                 0.000000
Pv_M_I+4R_U_N_29d_3.txt                 0.186360                 0.999453
Pv_M_I_Fc_Hf_38d_2.txt                  0.285853                 0.556588
Pv_M_I_U_N_15d_1.txt                    0.550161                 1.329045
…                                          …                        …
Pv_M_5xI_U_N_21d_2.txt                  0.105457                 0.927568
Pv_M_I_U_N_18d_6.txt                    0.436877                 0.639594
Pv_L_I+4xR_U_N_78d_1.txt                0.263120                 1.066627
Pv_M_I_U_N_32d_3.txt                    0.230314                 1.048077
Pv_M_I+4R_U_N_31d_1.txt                 0.042060                 0.854447

file                    Pv_M_I+4R_U_N_29d_3.txt  Pv_M_I_Fc_Hf_38d_2.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                 0.186360                 0.285853
Rb_M_I_Fc-L_N_16d_3.txt                 0.999453                 0.556588
Pv_M_I+4R_U_N_29d_3.txt                 0.000000                 0.470338
Pv_M_I_Fc_Hf_38d_2.txt                  0.470338                 0.000000
Pv_M_I_U_N_15d_1.txt                    0.377332                 0.828894
…                                          …                        …
Pv_M_5xI_U_N_21d_2.txt                  0.111899                 0.385329
Pv_M_I_U_N_18d_6.txt                    0.532882                 0.409936
Pv_L_I+4xR_U_N_78d_1.txt                0.199583                 0.513373
Pv_M_I_U_N_32d_3.txt                    0.072756                 0.515295
Pv_M_I+4R_U_N_31d_1.txt                 0.151238                 0.321974

file                    Pv_M_I_U_N_15d_1.txt  Pv_M_I_Fc_Hf_22d_1.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt              0.550161                 0.035963
Rb_M_I_Fc-L_N_16d_3.txt              1.329045                 0.848524
Pv_M_I+4R_U_N_29d_3.txt              0.377332                 0.177133
```

```
Pv_M_I_Fc_Hf_38d_2.txt                0.828894                0.302399
Pv_M_I_U_N_15d_1.txt                  0.000000                0.545517
…                                        …                       …
Pv_M_5xI_U_N_21d_2.txt                0.465073                0.087499
Pv_M_I_U_N_18d_6.txt                  0.757071                0.471237
Pv_L_I+4xR_U_N_78d_1.txt              0.451924                0.232973
Pv_M_I_U_N_32d_3.txt                  0.333874                0.217843
Pv_M_I+4R_U_N_31d_1.txt               0.510708                0.058807

file                     Pv_M_I_Fc_Hf_22d_3.txt  Pv_M_I+R_U_N_25d_2.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                0.510857                0.683918
Rb_M_I_Fc-L_N_16d_3.txt                0.450957                0.205435
Pv_M_I+4R_U_N_29d_3.txt                0.650687                0.847215
Pv_M_I_Fc_Hf_38d_2.txt                 0.362617                0.446525
Pv_M_I_U_N_15d_1.txt                   0.923710                1.155070
…                                         …                       …
Pv_M_5xI_U_N_21d_2.txt                 0.600694                0.784287
Pv_M_I_U_N_18d_6.txt                   0.192404                0.442198
Pv_L_I+4xR_U_N_78d_1.txt               0.764886                0.937136
Pv_M_I_U_N_32d_3.txt                   0.693097                0.894882
Pv_M_I+4R_U_N_31d_1.txt                0.522942                0.706595

file                     Pv_L_I+4xR_U_N_155d_1.txt  Rb_M_I_Fc-H_N_16d_3.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                   0.224900                 0.876111
Rb_M_I_Fc-L_N_16d_3.txt                   1.044908                 0.278068
Pv_M_I+4R_U_N_29d_3.txt                   0.120490                 1.026358
Pv_M_I_Fc_Hf_38d_2.txt                    0.499113                 0.659316
Pv_M_I_U_N_15d_1.txt                      0.396883                 1.304863
…                                            …                        …
Pv_M_5xI_U_N_21d_2.txt                    0.121736                 0.972901
Pv_M_I_U_N_18d_6.txt                      0.612275                 0.553889
Pv_L_I+4xR_U_N_78d_1.txt                  0.087742                 1.132182
Pv_M_I_U_N_32d_3.txt                      0.088713                 1.073162
Pv_M_I+4R_U_N_31d_1.txt                   0.204685                 0.894013

file                     …  Pv_L_I+4xR_Fc_N_99d_3.txt  \
file                     …
Pv_M_I+4R_U_N_27d_1.txt  …                   0.166148
Rb_M_I_Fc-L_N_16d_3.txt  …                   0.892280
Pv_M_I+4R_U_N_29d_3.txt  …                   0.252102
Pv_M_I_Fc_Hf_38d_2.txt   …                   0.338369
Pv_M_I_U_N_15d_1.txt     …                   0.606389
…                        …                      …
Pv_M_5xI_U_N_21d_2.txt   …                   0.166587
Pv_M_I_U_N_18d_6.txt     …                   0.593193
```

```
Pv_L_I+4xR_U_N_78d_1.txt  …                   0.197739
Pv_M_I_U_N_32d_3.txt      …                   0.276801
Pv_M_I+4R_U_N_31d_1.txt   …                   0.190154

file                      Rb_M_I_Fc-H_N_20d_1.txt  Rb_M_I_Fc-M_N_31d_2.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                  0.855142                 0.618249
Rb_M_I_Fc-L_N_16d_3.txt                  0.428068                 1.351946
Pv_M_I+4R_U_N_29d_3.txt                  0.992467                 0.451047
Pv_M_I_Fc_Hf_38d_2.txt                   0.669687                 0.875011
Pv_M_I_U_N_15d_1.txt                     1.257518                 0.329602
…                                           …                        …
Pv_M_5xI_U_N_21d_2.txt                   0.943964                 0.561759
Pv_M_I_U_N_18d_6.txt                     0.562132                 0.825626
Pv_L_I+4xR_U_N_78d_1.txt                 1.096581                 0.571152
Pv_M_I_U_N_32d_3.txt                     1.036648                 0.456839
Pv_M_I+4R_U_N_31d_1.txt                  0.870829                 0.582019

file                      Pv_M_I+4R_U_N_15d_1.txt  Rb_M_I_Fc-L_N_31d_1.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                  0.079561                 1.455208
Rb_M_I_Fc-L_N_16d_3.txt                  0.797104                 0.894267
Pv_M_I+4R_U_N_29d_3.txt                  0.246994                 1.568770
Pv_M_I_Fc_Hf_38d_2.txt                   0.244624                 1.288085
Pv_M_I_U_N_15d_1.txt                     0.615036                 1.767391
…                                           …                        …
Pv_M_5xI_U_N_21d_2.txt                   0.152573                 1.542095
Pv_M_I_U_N_18d_6.txt                     0.475215                 1.051327
Pv_L_I+4xR_U_N_78d_1.txt                 0.272600                 1.711029
Pv_M_I_U_N_32d_3.txt                     0.285785                 1.614433
Pv_M_I+4R_U_N_31d_1.txt                  0.120149                 1.462166

file                      Pv_M_5xI_U_N_21d_2.txt  Pv_M_I_U_N_18d_6.txt  \
file
Pv_M_I+4R_U_N_27d_1.txt                 0.105457              0.436877
Rb_M_I_Fc-L_N_16d_3.txt                 0.927568              0.639594
Pv_M_I+4R_U_N_29d_3.txt                 0.111899              0.532882
Pv_M_I_Fc_Hf_38d_2.txt                  0.385329              0.409936
Pv_M_I_U_N_15d_1.txt                    0.465073              0.757071
…                                          …                     …
Pv_M_5xI_U_N_21d_2.txt                  0.000000              0.506170
Pv_M_I_U_N_18d_6.txt                    0.506170              0.000000
Pv_L_I+4xR_U_N_78d_1.txt                0.171419              0.673984
Pv_M_I_U_N_32d_3.txt                    0.134457              0.569607
Pv_M_I+4R_U_N_31d_1.txt                 0.086053              0.434129

file                      Pv_L_I+4xR_U_N_78d_1.txt  Pv_M_I_U_N_32d_3.txt  \
```

```
file
Pv_M_I+4R_U_N_27d_1.txt                    0.263120              0.230314
Rb_M_I_Fc-L_N_16d_3.txt                    1.066627              1.048077
Pv_M_I+4R_U_N_29d_3.txt                    0.199583              0.072756
Pv_M_I_Fc_Hf_38d_2.txt                     0.513373              0.515295
Pv_M_I_U_N_15d_1.txt                       0.451924              0.333874
…                                               …                     …
Pv_M_5xI_U_N_21d_2.txt                     0.171419              0.134457
Pv_M_I_U_N_18d_6.txt                       0.673984              0.569607
Pv_L_I+4xR_U_N_78d_1.txt                   0.000000              0.170706
Pv_M_I_U_N_32d_3.txt                       0.170706              0.000000
Pv_M_I+4R_U_N_31d_1.txt                    0.254308              0.196891

file                        Pv_M_I+4R_U_N_31d_1.txt
file
Pv_M_I+4R_U_N_27d_1.txt                    0.042060
Rb_M_I_Fc-L_N_16d_3.txt                    0.854447
Pv_M_I+4R_U_N_29d_3.txt                    0.151238
Pv_M_I_Fc_Hf_38d_2.txt                     0.321974
Pv_M_I_U_N_15d_1.txt                       0.510708
…                                               …
Pv_M_5xI_U_N_21d_2.txt                     0.086053
Pv_M_I_U_N_18d_6.txt                       0.434129
Pv_L_I+4xR_U_N_78d_1.txt                   0.254308
Pv_M_I_U_N_32d_3.txt                       0.196891
Pv_M_I+4R_U_N_31d_1.txt                    0.000000

[270 rows x 270 columns]
```

[108]:
```
# ps_symetric_dist.to_csv('path_score_distances_substrate.csv',␣
 ↪index_label='index', header=True)
# Drop the specified column
# to_drop = ['Rb_M_I_Fc-M_N_31d_3.txt']  # Column name to drop
# ps_symetric_dist_2 = ps_symetric_dist_2.drop(columns=to_drop)
# ps_symetric_dist_2 = ps_symetric_dist_2.drop(index=to_drop)
```

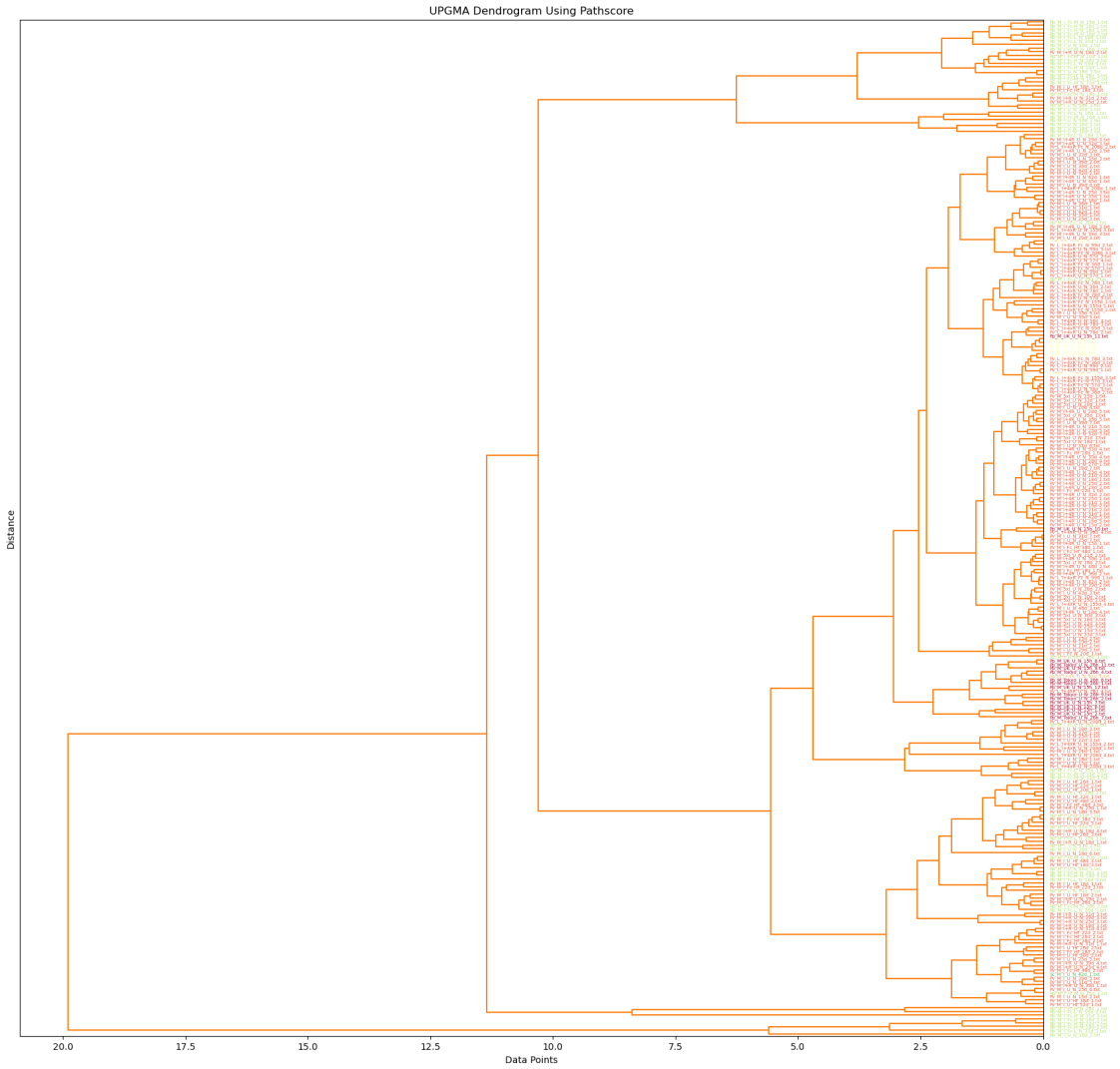
[109]:
```
ps_symetric_dist_2.to_csv('path_score_distances_dropped.csv',␣
 ↪index_label='index', header=True)
```

[115]:
```
# graph
color_mapping = {
    'Pp': 0,
    'Pv': 1,
    'Ag': 2,
    'Pi': 3,
    'Rb': 4,
    'Sc': 5,
```

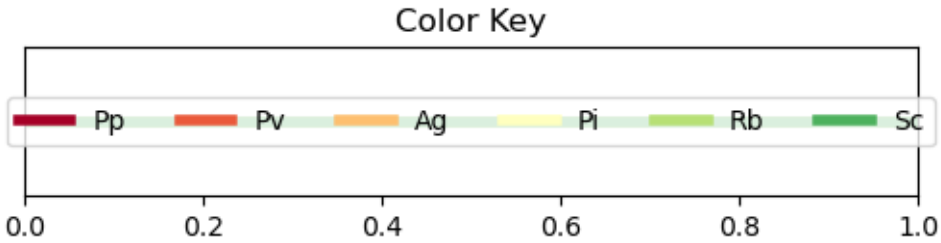
```
    'none': 6
}

def get_leaf_color(filename):
    if filename[-4:] == '.txt':
        value = filename[0:2]
        return color_mapping[value]
    else:
        return color_mapping['none']

# Assuming 'df' contains attributes for each leaf node
# Pass the leaf attributes to the color calculation function
leaf_colors = [get_leaf_color(index_name) for index_name in ps_symetric_dist_2.
 ↪index]

# Your distance matrix
# ps_symetric_dist = pd.read_csv('distances_ln.csv')
# ps_symetric_dist.set_index('index', inplace=True)
# ps_symetric_dist = distances.drop(distances.columns[[0]], axis=1)

tree = linkage(ps_symetric_dist_2, method='average',optimal_ordering=False)  #␣
 ↪Use 'average' method for UPGMA

# Plot dendrogram
plt.figure(figsize=(20, 20))
# dendrogram(tree, labels=distances.index, leaf_rotation=90)
dendrogram(tree, labels=ps_symetric_dist_2.index, leaf_rotation=0,␣
 ↪orientation="left", color_threshold=240, above_threshold_color='grey')
plt.title('UPGMA Dendrogram Using Pathscore')
plt.xlabel('Data Points')
plt.ylabel('Distance')
# plt.show()


my_palette = plt.cm.get_cmap('RdYlGn', 7)

ax = plt.gca()
xlbls = ax.get_ymajorticklabels()
num=-1
for lbl in xlbls:
    val = get_leaf_color(lbl.get_text())
    lbl.set_color(my_palette(val))

# Show the graph
plt.show()

# Define labels for each color
```

```
color_labels = ['Pp','Pv','Ag','Pi','Rb','Sc']

# Create a dictionary to map colors to labels
color_map = {my_palette(i): label for i, label in enumerate(color_labels)}

# Display the colors and their corresponding labels
for color, label in color_map.items():
    print(f'{label}: {color}')

# Plot example colors
plt.figure(figsize=(6, 1))
for color, label in color_map.items():
    plt.plot([0, 1], [0, 0], color=color, label=label, linewidth=4)
plt.xlim(0, 1)
plt.ylim(-0.5, 0.5)
plt.yticks([])
plt.legend(loc='center', ncol=len(color_labels))
plt.title('Color Key')
plt.show()
```

```
/tmp/ipykernel_375/1475335721.py:28: ClusterWarning: scipy.cluster: The
symmetric non-negative hollow observation matrix looks suspiciously like an
uncondensed distance matrix
  tree = linkage(ps_symetric_dist_2, method='average',optimal_ordering=False)  #
Use 'average' method for UPGMA
```




```
Pp: (0.6470588235294118, 0.0, 0.14901960784313725, 1.0)
Pv: (0.918954248366013, 0.3477124183006535, 0.22614379084967318, 1.0)
Ag: (0.9934640522875817, 0.7477124183006535, 0.4352941176470587, 1.0)
Pi: (1.0, 1.0, 0.7490196078431373, 1.0)
Rb: (0.7176470588235295, 0.8797385620915033, 0.45882352941176474, 1.0)
Sc: (0.3006535947712421, 0.6928104575163401, 0.3633986928104576, 1.0)
```



[ ]:
```
jupyter nbconvert --to pdf --TagRemovePreprocessor.enabled=True␣
↪--TagRemovePreprocessor.remove_input_tags='{"graph"}' --output graphs.pdf␣
↪Analysis.ipynb
```