Package ‘MultiNetPy’

February 14, 2025

Type Package

Author Aida Harooni

Maryam Lotfi Shahreza

Mohammadreza Shams

Alireza Firouzi

language Python

Requirements contourpy==1.3.1

cycler==0.12.1

fonttools==4.55.0

kiwisolver==1.4.7

matplotlib==3.9.3

networkx==3.4.2

numpy==2.1.3

packaging==24.2

pandas==2.2.3

pillow==11.0.0

pyparsing==3.2.0

python-dateutil==2.9.0.post0

pytz==2024.2

scipy==1.14.1

seaborn==0.13.2

six==1.16.0

tzdata==2024.2

Description MultiNetPy is a Python package for analyzing multiplex networks. It provides functionalities for importing multiplex network data and computing centrality measures

Github Link: https://github.com/Multinetpy/Multinetpy

Capabilities of MultiNetPy

Import dataset:

mg = gm.Import_Graph.make_graph("YourPath_nodes.txt", "YourPath_nodes.edges", "YourPath_nodes_layers.txt")

Calculate Closeness Centrality:

Description: Calculate closeness centrality of nodes in a layer using networkX with an optional input parameter as weight.

Usage: closeness_centrality(weight = None)

Return type: dictionary

Calculate Aggregated Closeness Centrality:

Usage: closeness_centrality_aggregated(weight = None)

Calculate Weighted Closeness Centrality:

Usage: weighted_CC()

Calculate Betweenness Centrality:

Description: Calculate closeness centrality of nodes in a layer using networkX with an optional input parameter as weight.

Usage: betweenness_centrality (weight = None)

Return type: dictionary

Calculate Aggregated Betweenness Centrality:

Usage: betweenness_centrality_aggregated (weight = None)

Calculate Weighted Betweenness Centrality:

Usage: weighted_BC ()

Kendall Tau:

Description: Used for comparing attained aggregated and weighted result with the other result which is accommodated in an excel file.

Usage: plot_kendall_tau (aggregated_centralities, weighted_centralities, table_ranks)

Rank_Difference:

Description: Used for comparing attained aggregated and weighted result with the other result which is accommodated in an excel file

Usage: Rank_Difference (*table_ranks*, *aggregated_ranks*, *weighted_ranks*)

Return Type: Three values including R1, R2 and R3

Plot Rank Difference:

Usage: plot_rank_difference (R, R2, R3)

Intersection Similarity:

Description: Used for comparing attained aggregated and weighted result with the other result which is accommodated in an excel file

Usage: intersection_similarity (table_ranks, aggregated_ranks, weighted_ranks, max_k=20)

Return Type: Three values including isim_k_values, isim_k2_values, isim_k3_values

Display Intersection Similarity:

Usage: display_isim_table(self, isim_k_values, isim_k2_values, isim_k3_values)

Load table from excel:

Description: File should be contained ID and Order columns

Usage: load_table_ranks_from_excel (file_path)