Proximity, Separation and LCC Metrics in Graphs

Introduction

In network medicine, understanding relationships between various node sets (primarily genes) can offer deep insights into the patterns and structure of a PPI (Protein-Protein Interaction) network. This README introduces the concepts of `proximity`, `separation`, and `LCC (Largest Connected Component)`, and provides guidance on executing the related code.

Proximity: Represents the closeness or average distance between two node sets. A lower average distance suggests a higher proximity. **Separation**: Depicts how distant two sets of nodes are from one another. **LCC (Largest Connected Component)**: Represents the largest subgraph in which any two nodes are connected to each other by paths. The significance of LCC lies in its ability to indicate the major component of a network where most crucial interactions occur.

This Python implementation uses precomputed distance matrices to optimize calculations. With precalculated distances between every node pair, the code can rapidly compute proximity and separation.

Getting Started

Prerequisites

- Ensure you have Python installed.
- Install necessary libraries:

  ```
  pip install networkx seaborn matplotlib scipy
  ```

Import NetworkMetrics module and networkx:

```
import random
import networkx
import network_distances.NetworkMetrics as distances
```

Replace with the appropriate path to the `NetworkMetrics` module on your system.

Create the network and sets

As an example, consider a basic network G, and two sets of randomly selected node sets:

```
G = nx.barabasi_albert_graph(1000, m)
nodes = list(G.nodes())
S = random.sample(nodes, 20)
T = random.sample(nodes, 20)
```

Precomputing distances

```
 D = distances.all_pair_distances(G)
```

Save distance matrix as a pickle file

```
 distances.save_distances(D, "distances.pkl")
```

Load distance matrix from a pickle file

```
D = distances.load_distances("distances.pkl")
```

Calculate proximity between S and T, using exact degree preserving randomization and 1000 iterations

```
p = distances.proximity(G, S, T, D, degree_preserving='exact', n_iter=1000)
```

The result is a dictionary with the next attributes:

- ‘d_mu’: The average distance in the randomized samples.
- ‘d_sigma’: The standard deviation of distances in the randomized samples.
- ‘z_score’: The z-score of the actual distance in relation to the randomized samples.
- ‘raw_amspl’: The raw average minimum shortest path length between sets T and S.
- ‘dist’: A list containing distances from each randomization iteration.

To retrieve the z-score value of proximity:

```
print(p['z_score'])
```

Calculate separation between S and T

Calculate the separation between S and T:

```
sep = distances.separation(G, S, T, D)
print(sep)
```

Calculate the z-score significance of the separation value:

```
s = distances.separation_z_score(G, S, T, D, degree_preserving='exact',
         n_iter=1000)
print(s['z_score'])
```

Calculate LCC significance

```
#Let's select a random set of nodes to evaluate:
G1 = G.subgraph(random.sample(nodes, 50))
L = set(G1.nodes())

lcc_data = metrics.lcc_significance(G,L,
         degree_preserving='exact',n_iter=10000)

# Z-score and p-value
size = lcc_data['lcc_size']
z = lcc_data['z_score']
p = lcc_data['p_val']

print(f"LCC-size={size} z-score={z:0.2f} p-value={p:0.2f}")
```

Further reading

An example on the use of the implemented functions is available in the file ‘Example.py’. Consult the full documentation of the appropiate functions in the file ‘NetworkMetrics.pdf’ or ‘NetworkMetrics.md’

References

1. Menche, Jörg, et al. “Uncovering disease-disease relationships through the incomplete interactome.” Science 347.6224 (2015). DOI 10.1126/science.1257601
2. Guney, Emre, et al. “Network-based in silico drug efficacy screening.” Nature Communications 7,1 (2015). DOI 10.1038/ncomms10331

Authors

- Andres Aldana Gonzalez (a.aldana@northeastern.edu)
- Rodrigo Dorantes Gilardi (r.dorantesgilardi@northeastern.edu)

Contact

For further inquiries or additional details, reach out to the authors or consult the given documentation.