Network comparison and visualization with wiring diagram

You are given three social networks and three food webs in Pajek format (edge list and LNA formats are also available).

- Zachary karate club network (34 nodes)
- Davis southern women network (32 nodes)
- Lusseau bottlenose dolphins network (62 nodes)

- Cypress Wetlands food web (wet) (128 nodes)
- Cypress Wetlands food web (dry) (128 nodes)
- Little Rock Lake food web (183 nodes)

Consider different approaches for comparing networks introduced in lectures. These include comparing networks by different metrics or statistics, graph edit distance, graphlet degree distribution agreement, portrait divergence, D-measure etc. You can implement selected approaches by yourself, browse your network library for existing implementations or use the code provided below.

- Simplified D-measure: simplified_dmeasure.py
- Network portrait divergence: portrait_divergence.py
- Graphlet distribution agreement: graphlet_aggrement.py

Note that the last script requires a working installation of the orca algorithm for counting graphlet orbits.

I. Networks and models comparison

(code) Compare given real networks between each other and plot their dissimilarity or distances with a heat map. How similar are networks of different type? For instance, are social networks more similar to each other than to food webs? Does the answer depend on the selected measure of dissimilarity or distance?

(code) Compare networks also to small synthetic graphs such as Erdös-Rényi random graphs, Barabási-Albert scale-free graphs and Watts-Strogatz small-world graphs with $n = 500$ nodes and $m = 1500$ edges. How similar are real networks to synthetic graphs? How do synthetic graphs compare between each other?

II. Network visualization with wiring diagram

(code) Browse NetworkX or other library for implementations of network layout algorithms and methods for network visualization. Try to visualize selected social network or food web with a wiring diagram. For networks that are either too large or dense, you might consider first revealing some network mesoscopic structure (e.g., communities, k-cores) and rather visualize that.