BIO206 2021

Report 2. Network analysis of a hunter-gatherer social network.

Download *motes.xlxs* with data on social interactions from a real population of hunter-gatherers. It contains ID1 and ID2 from each dyad, and dyadic weights.

Import this dataset into R to produce an ***undirected*** network (using *sna/network*), including the information on edge weights.

*(Note: when importing, **ALREADY change class of ID1 and ID2 to character**, otherwise network cannot be created. Weights should be **integers** or **numeric**).*

1) **Visualisation (starting with sna/network):**

a) Plot the network (in ***undirected*** or 'graph' mode, here and everywhere in this report)

b) Now plot it again, but with edges reflecting their weights (hint: multiply the term in *edge.lwd* by 0.05 for better visualisation)

2) **Main network properties**: calculate

a) Size

b) Density

c) Number of components

d) Diameter

e) Mean path length

f) Transitivity

3) **Node features**: calculate

a) Mean degree

b) Plot a histogram with degree distribution

4) **Centrality measures**

a) Make a table with the three centrality measures for each subject

b) Who is/are the most central individual(s)?

c) Briefly discuss why centrality measures did/did not differ in this network

(NOTE: there may be difference between the subject name (a character, but which in this example is also a number: "1", "2") and the subject number.

Subject name, which may be used as labels in plots etc, follows a character sequence: "0", "1", "10", "11"...

Subject number, which is used for functions, follow sequence 1, 2 , 3, 4. This will be clear when you producing the tables, calculate centralities etc).

5) Plot network identifying any **cutpoints**

(NOTE: again, be aware of distinction between name (shown in the plot as the label) and number (output of function cutpoints).

6) **Coreness**:

(NOTE: don't forget to switch to igraph, and to create an igraph object with package Intergraph)

a) What is the largest and smallest coreness values in this network?

b) Create a plot where each k-core is shown with a different colour

(NOTE: you will need to define an attribute V(yournetwork)$colour for this to work! See lecture code)

7) **Community detection**:

a) Detect communities using those two algorithms:

cluster_walktrap()

cluster_louvain()

and plot the two networks with the identified subgroups

b) Do algorithms produce different classifications?

8) **Model networks**:

a) Create a ***random graph network*** with same size and density as your real network (with p=density of the real network, or m=total number of edges of the real network) and **a small-world network,** with *nei* = (mean degree of real network)/2, and p (randomness) = 0.05.

b) Make a table comparing *diameter*, *mean path length (mean_disttance())* and *transitivity* in the real, random graph, and the small-world network

c) Plot the distribution of degrees and transitivity of all nodes in the four networks

d) What do you conclude? Is the real shared knowledge similar or more similar to any of the two model networks?