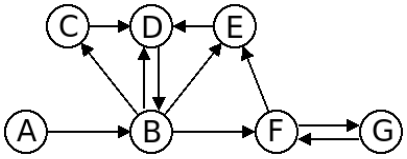
Homework #2
PageRank, Communities and Subgraph Patterns
Due: May ~~16th~~ 23rd, 2021

- The assignment should be delivered digitally by email. Your message should be sent to **pribeiro@dcc.fc.up.pt** with subject *"[ARSI HWK2] FirstName LastName StudentNumber"*
- Your delivery should be a **zip file**, containing a **PDF report with the answers** and all additional files that were used for producing those answers
- This homework is to be made **individually**. You can collaborate with more students by talking about the exercises, but **you should not copy answers or code** and you should do you own writeup.
- Please **acknowledge any help you got** and state any references you consulted (including internet pages) and any other students with whom you talked about the exercises.
- Answers should be **submitted until 23:59 of the due date**. Up to 24h of delay will get you a 25% penalty. 24h to 48h of delay will get you a 50% penalty. After 48h your work will not be counted.

(always explain how you reached each answer, so that I can understand your train of thought)

Link Analysis and PageRank

1. Draw a network in which one node has a very **high value of PageRank**, although the same node has **low closeness and betweenness centrality** (don't forget to point out the node).
2. The ***damping factor*** in PageRank (parameter β, in slides) controls how of often we follow one of the links of the current node *vs* going to an arbitrary node on the network.
   (a) What does $\beta = 0$ mean? What would happen to the PageRank values in that case? Why?
   (b) What does $\beta = 1$ mean? Can you explain a possible problem with using that value?
3. **Implement a program** (in any programming language) for manually computing the (normalized) PageRank values of a small network using **power iterations** (the "flow" mode). Attach the program to your homework submission with a very short description on how it works.
4. Use your program to **compute the PageRank values** of the following network (with $\beta = 0.85$). **Show the values of all nodes for each iteration** until the computation converges.



5. Use the program to do computations varying the β parameter from 0.0 to 1.0 in steps of 0.05 and:
   (a) Show in a plot the **number of iterations needed until convergence is reached as you change** β. Can you explain what is happening?
   (b) Show in a plot the different **PageRank values of all nodes as your change** β. Can you divide the nodes into different curve behaviors? Can you explain what is happening?

Community Discovery

6. For this exercise you will be asked to analyze a set of networks depicting the **"social networks"** (character co-occurrences in a scene) of 5 well known **movies** (you can download gml files here: movies.zip):

   - Blade Runner (1982)
   - Pulp Fiction (1995)
   - Star Wars: Episode V - The Empire Strikes Back (1980)
   - The Godfather: part II (1974)
   - The Lord of the Rings: The Return of the King (2003)

   Start by opening the files on a text editor to see how they internally look like.

   (a) Using Gephi, networkx (both can read networks in the gml format) or any other platform/library, you should run **Louvain Algorithm** to find the best possible communities and **create a table showing**: name of the movie, number of nodes and edges, number of communities found and modularity for those communities. Give a brief comment on which networks seem to present **community structure**, and why.

   (b) Choose **any two of the movies** and produce **visualizations for the networks**, labeling the nodes with their character names, using colors to represent communities and the size of the nodes to represent PageRank values. Try to make the picture as **aesthetically pleasing** as possible, reinforcing the community structure (and explain how you created the layout). Give a brief **informal description on the meaning of the communities** in the context of the movie (are they what you were expecting? are they meaningful? choose movies that you are familiar with)

   (c) **Implement a program** (in any programming language) for manually computing the (normalized) **modularity of a network when given a partition**. Test it on one movie of your choice and the partitions you produced on the previous questions (and report if the value seems ok). Attach the program to your homework submission with a very short description on how it works.

   The modularity can be computed as:

$$Modularity = \frac{1}{2m}\left(\sum_{i,j \in V}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(C_i, C_j)\right)$$

   Where A is the adjacency matrix of the graph, C_i is the community to which node i belongs, k_i is the degree of node i, m is the total number of edges and V is the set of nodes.

   (d) **Implement** (in any programming language) a **simple greedy agglomerative algorithm**: start with each node being a separated community and then do sucessive iterations in which you try all possible changes for one node (that is, for each node $i \in V$, try changing its community to all possible communities $j \in C$), and apply the change that produces the best gain in modularity (if there is ties, choose any possible). Attach the program to your homework submission with a very short description on how it works.

   Make a **plot showing the modularity increase** as you are making more iterations until you reach you a "local maximum", and report the communities you found (as a visualization), comparing them to the communities found previously.

   (e) Using your previous program as a basis, explain how could you obtain a **larger quantity of communities**? And how could you obtain **less communities**?

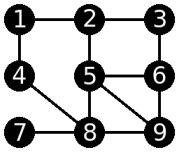
Network Motifs

7. For this part of your homework it is highly advisable that you use the **gtrieScanner** tool. You should download, unzip and compile this version: `gtrieScanner_src_01.zip` (it is the same as the version online with a newly added "-raw" option to help you on the homework plus some pre-computed g-tries)

   Your first task is to be able to compile the source code. You will need a C++ compiler and make tools. If you have Linux you can simply use `g++` and `make` available on any common distribution. If you use Windows we suggest you use WSL or Cygwin to have a shell with Linux-like functionality.

   Counting subgraphs

   (a) Consider the following undirected network:

   The frequency (number of occurrences of size 3) of subgraphs of size 3 in this network is:

Subgraph	Frequency
	18
	2

   You could obtain these results by running (for instance) one of the the following commands:

   ```
   ./gtrieScanner -s 3 -m esu -g network.txt -f simple
   ./gtrieScanner -s 3 -m gtrie undir3.gt -g network.txt -f simple
   ```

   supposing that `network.txt` is a text file containing the description of the network as an adjacency list: one line per edge, each line containing two integers separated by a space, the endpoints of the respective edge (the file should have 12 lines, the first of which could be `1 2`, for example).

   Your task here is to determine the number of occurrences of all subgraphs of size 4 in this network. You should put in the report a table like the one shown above (the html version of the output is "broken", so you should produce your own images of the subgraphs)

   A bit of math: subgraphs in purely random networks

   (b) Imagine you have a $G_{n,p}$ undirected Erdős–Rényi random network. What is its expected number of triangles ( )? And what about the expected number of chains ( )? Justify your answer.

   Note that you can test your theory by generating Erdős–Rényi networks and counting the subgraphs using gtrieScanner, but your answer should be stated as formulas involving n and p.

Back to empirical findings: uncovering motifs in bacteria

(c) Your task is now to find some network motifs of the transcriptional regulation directed network of the bacteria *Escherichia coli*. Start by downloading the network as a weighted adjacency list: `ecoli.txt` (each line is an edge in the format *start_node end_node weight*)

This directed network is ready for being fed to gtrieScanner. For example you could run:

```
./gtrieScanner -s 3 -d -m gtrie dir3.gt -g ecoli.txt
```

This would compute the frequency of all possible 13 types of size 3 subgraphs, and it should show you that the most frequent one is the following, appearing 250 times:

Now, if you add the "-r n" option, it should produce n networks with the same degree sequence and it will you show you how often each subgraph appears on it. For example:

```
./gtrieScanner -s 3 -m gtrie dir3.gt -d -g ecoli.txt -r 500 -raw
```

Check the results and report on what is the more overrepresented subgraph, including its z-score (Z), frequency on the original network ($real$), average number of occurrences ($avgR$) and standard deviation ($stdevR$)) on the randomized networks.

The z-score of subgraph i is computed as $Z_i = (real_i - avgR_i)/stdevR_i$ as in Milo et al. 2004).

Notice how the most frequent subgraph is not the most significant one. Check if your very simplistic analysis is consistent with the known literature (Milo et al. 2002) (Shen-Orr et al. 2002), that is, if the motif you found is also reported (**what is the name given to this motif?**)

Characterizing families of networks using motifs

Start by carefully reading the following paper:

Milo et al. **"Superfamilies of evolved and designed networks."** Science 303.5663 (2004)

The idea here is to perform a very similar analysis, even using some of the same networks!

(d) Start by downloading this set of 9 directed networks: `networks.zip` (inside the zip there is a `README.txt` explaining what is each network). **Use gtrieScanner to compute motif fingerprints of all networks. You should produce and include in the report the following:**

- Plot(s) showing the (normalized) significance profile (SP) of all 13 directed motifs of size 3 for each network. Try to expose the similarity between groups of networks. It should be clear to which subgraph corresponds each data point (ex: see figure 1 of the paper).
- One heat map of 9×9 cells showing the correlation between the SPs of all pairs of networks (ex: see figure 2 of the paper).
- A visual description of the main characteristic motifs of each group of networks (that is, you should draw them). Can you give an interpretation on why are they so significant?

You should use at least 100 random networks for each original network and you can opt to ignore subgraphs that occur only once in the original network (attributing a z-score of zero to them). For normalizing the z-scores use the suggested formula: $SP_i = Z_i/\sqrt{(\sum Z_i^2)}$

For the heat map you can use any sofware. R and Python have several possible packages, but even Excel or LibreOffice will suffice (use range conditional formatting). You even have some possible online alternatives. If you know about it, you can even use a clustering algorithm to produce a dendrogram showcasing the relationship between the families of networks.

(e) **Your task is to find the "family" of the three "unknown" networks** given in `unknown.zip`

You should justify your answer by computing and plotting their motif significance profiles and by adding them to the previous heatmap. Each network will clearly belong to one of the groups discovered on the previous question.