CENG 465
Fall 2013-2014

Assignment #4

Programming Assignment on Biological Network Analysis

Due Date: January 19, 2014, 11:55PM
(submit source code and report via COW)

Clustering a Protein-Protein Interaction Network using Random Walks

In this assignment your goal is to find the top-3 clusters in protein-protein interaction networks using random walks on graphs. The description of the algorithm that you are required to implement is as follow:

1. Using Random Walks (without restarts) find the most central node in the network (i.e., the node with the highest probability of being visited). Let's call this node *h*.
2. By using node *h* as the start node, find the closest nodes to *h* using the Random Walks with restarts method. Use a restart probability of 0.4.
3. By using the node probabilities computed at Step 2, build a cluster around *h*, by adding nodes to the cluster in decreasing order of their proximity to *h* (proximity = steady state probability). Keep expanding the cluster until the cluster density is less than 0.6. Cluster density is computed as: the number of edges in the cluster / number of edges in a clique of same size.
4. After the cluster is built, remove the nodes (and the corresponding edges) of the cluster from the network and repeat from Step 1, until 3 clusters are built.

Use the following network as the protein-protein interaction network:

I2D network at http://www.ceng.metu.edu.tr/~tcan/ceng465_f1314/i2d_swissprot.txt

Analyze the discovered clusters in terms of biological functionality using the tool g:Profiler available at:

http://biit.cs.ut.ee/gprofiler/

Answer the following questions in a short report

a) What are the sizes and densities of the clusters you have found?
b) Which biological functions are significantly represented in these clusters?

Notes:

- The network data is given as a list of undirected edges. Each line of the file represents an edge that connects a pair of tab separated protein ids.
- In order to identify convergence of Random Walk iterations, use an epsilon of 0.0001 (10E-4). In other words, if the Manhattan distance (L1-norm) between the probability distributions of two successive iterations is less than or equal to 0.0001, then you may consider that the random walk process has converged.
- For step 1 of the algorithm, you will implement the random walks without restarts as if restarting at all of the nodes in the network (in order to prevent the problem of

disconnected islands) in every iteration again with restart probability, d=0.4. The initial probability distribution vector can be a vector with 0.4/|V| for every node, where |V| is the number of nodes in the network.

Deliverables:

A short report which contains your answers to the questions above and your source code.

Submission:

Submit the deliverables using the COW system.

Late Submission Policy:

Penalty: 20 points per day.