Exercise afternoon 2, BNF079, fall 2004

The purpose of this exercise is to investigate som basic properties of a network, with the help of simple perl programs that you write yourself. To the aid is available a perl module file with subroutines, described in "Guide to the subrotines:". The module "bnf079_sub1.pm" is available at **http://www.thep.lu.se/ teaching/BNF079.html**. Download the module in the same directory as you main program, and write "use bnf079_sub1;" in your perl script.

Exercise 1: Connected components of a network

The available subroutines make it possible to rather quickly examine the number of connected components in a network, and the size of the components. However, whenever you program something yourself, it is essential to always check your code on a simple example, where you know the answer. To emphasize this, the first part of this exercise is to check that the available subroutine works as it should.

- Construct your own small network in Cytoscape file format (".sif-file"). Make the network disconnected, and make sure you know the number of nodes in its connected components. (You can have a look at your network in Cytoscape, to be sure.)
- Write a perl script that reads the file with your small network, symmetrizes it, and extracts the components. Make a printout of the result to control that it agrees with the network you constructed.

Hopefully, there is no need to debug the subroutines, and you are ready to study the components of a real biological network, for example the one you downloaded in the exercise a previous day.

- Modify your program to extract the distribution of connected components in a real network, available Cytoscape file format. Make a printout with number of components of size 2, number of components of size 3, etc..
- It can be a good idea to develop your printout using your small network. The subroutines are not optimized for speed.

Save the results. When everyone are done with exercise 1, compare results and discuss. How did various networks differ in number of nodes, number of edges, and number of components of different sizes? Can we understand the differences? (The answer may well be "no", this is a discussion where supervisors hope to learn just as much as the students!')
If you have a large network, it can be a good idea to print, *e.g.*, the largest connected component into a new file in Cytoscape format, and work with that in the next exercise.

Exercise 2: Degree distribution

- Write a new perl script that, with help of the subroutines, calculates the degree distribution. Develop your program using a small network where you know the answer. Then apply it to the real network.
- Make the output of your program suitable for plotting with gnuplot (or some other plot program, if you have your own favourite). An example is like this.

# degree	number_of_nodes
1	8
2	5
3	3
4	2
5	3
7	1

  Plot the result. Does it look like a power law distribution?
- For large degrees, it is hard to see what the distribution is, since the number of nodes is either 0 or 1. One way to see it better, is to *bin* together the large, rare, degrees. We are likely to plot our result with a logarithmic x-axis, and then a logarithmic binning is suitable. A simple such binning is to double the bin interval for each new bin. The above table would then look like this:

# degree_interval	average_number_of_nodes
1 - 1	8
2 - 3	4
4 - 7	1.5

  If there were nodes of larger degree, the next interval would be 8 - 15.

  Modify your program to present a binned result, or write a new script that takes your unbinned result file as input and creates a binned result. Plot the result. Does it look like a power law distribution? If so, what is the power?

If time allows, compare your results to others, that used different networks.