Bioinformatics III

Prof. Dr. Volkhard Helms — Saarland University
Ruslan Akulenko, Maryam Nazarieh, Duy Nguyen, Thorsten Will — Chair for Computational Biology
Winter Semester 2014/2015

Exercise Sheet 3

Due: November 14, 2014 13:15

Submit your solutions on paper, hand-written or printed at the *beginning* of the lecture or in building E2.1, Room 3.02. Alternatively you may send an email with a single PDF attachment, via mail to thorsten.will@bioinformatik.uni-saarland.de.

Network Communities & Real Interaction Networks

In this assignment you will deal with network communities which are defined as regions of the network that have more internal connections than connections to the rest of the network. Your task is it to use the algorithm of *Radicchi et al.* to identify the communities of a given network.

Exercise 3.1: Network Communities (50pts)

(a) **Edge-clustering coefficient**

The edge-clustering coefficient $\tilde{C}^{(3)}_{i,j}$ of a link between nodes i and j is defined as the ratio of the actual number of triangles $z^{(3)}_{i,j}$ to which the link between i and j contributes and the number of possible triangles, determined by the minimum of the degrees k_i and k_j of the two nodes i and j:

$$\tilde{C}^{(3)}_{i,j} = \frac{z^{(3)}_{i,j} + 1}{\min[k_i - 1, k_j - 1]}$$

If one of the nodes has a degree of 1, then $\tilde{C}^{(3)}_{i,j}$ is infinite. What is the maximal *finite* value that the edge-clustering coefficient can take? For which configuration does this occur? Give an example!

(b) **Determine communities**

To determine the communities of the supplied network given in `GoT.txt` (found in the additional materials), perform the following steps:

(1) **Decomposition of the network** (by programming)

As explained in the lecture, iteratively delete the links with the smallest $\tilde{C}^{(3)}_{i,j}$:

i. Read in the network file.
ii. Calculate the edge-clustering coefficient $\tilde{C}^{(3)}_{i,j}$ for each link.
iii. Find the link with the smallest $\tilde{C}^{(3)}_{i,j}$ and delete it from the network (or mark it). Store the link.
iv. Repeat from (ii) until there is no link left.

Give the links that you deleted from the network in (iii) by printing the names of the two nodes and their current edge-clustering coefficient in the order of their deletion. Of course, add the output to the PDF/sheet that you hand in. Implement this part as a script or class-based, there are no specifications you need to adjust to.

(2) **Build communities and the dendrogram** (by pen and paper)

There are two criteria for a community (see *Radicchi et al., 2004*):

i. In a *community in a strong sense* every single member of the subgraph V has more links to the inside of the community (k^{in}) than to the outside (k^{out}):

$$k_i^{\text{in}}(V) > k_i^{\text{out}}(V) \qquad \forall i \in V$$

ii. In a *community in a weak sense* the total number of links inside the subgraph V is bigger than to the outside:

$$\sum_{i \in V} k_i^{\text{in}} > \sum_{i \in V} k_i^{\text{out}}$$

Use the links deleted in (1) in reverse order, i.e., the link that was deleted last is now used first to construct the communities. To do so, take one link after the other and check if they have nodes in common with the already included links. During this composition stage you do not need to keep track of the links, but only of the nodes that belong to the same subgraph:

i. If the latest link is disjoint from the already processed links, then start a new subgraph (=list of nodes of this subgraph) from this one.

ii. If the latest link has a single node in common with one of the existing subgraphs, then add the other node of this link to that (list of the nodes of the) subgraph, too.

iii. If the two nodes of the latest link belong to two different subgraphs, then join the two subgraphs to form a single one from them. Highlight the two lists of nodes that are joined in this step.

Finally, when the last link is added, you should end up with a single graph that contains all nodes of the network and a listing of the subgraphs just before they were joined to form bigger ones.

To draw the dendrogram of the network, look at the above choice (iii), the joining of two groups: start from the individual nodes and every time that this happens, connect two subgraphs.

(c) **Visualization of the communities**

In Figure 1 a layout of the network is given. Use it to visually identify communities. Point out **three** communities that are disjunct. Specify for each of the communities whether the weak or the strong criterion applies.

Exercise 3.2: Real interaction networks (50 pts)

BioGRID ("Biological General Repository for Interaction Datasets") is a protein interaction database which, in version 3.2.118 (Nov. 2014), contains data of 787,370 raw protein and genetic interactions from major model organism species compiled from 43,913 publications. The supplement contains this release as a tab-separated file ("`BioGRID.txt`"). The format is documented in the beginning of the file, make yourself familiar with that.
In this exercise you implement the class **BioGRIDReader** which should help you to deal with such data.

(a) The class should read the file in its initialization and store the necessary data in a data structure that simplifies your later queries. For every organism found in the file (as **NCBI taxon identifiers**) one should be able to retrieve all interactions as pairs of **official gene symbols** easily.

(b) Implement **getMostAbundantTaxonIDs(n)** and use it to return the **five** organism with the most interactions annotated in BioGRID as well as their respective number of interactions. Argument why the order is not surprising.

(c) Implement **writeInteractionFile(taxon_id, filename)** to be able to create organism-specific network files that can be used by the GenericNetwork-class. Build a network for human (taxon 9606), determine and plot the corresponding degree distribution. Discuss if it behaves more like a scale-free or a random network.

(d) How big is the human interaction network and which are the **10** proteins with the highest degree? Take one of them as an example and **briefly** explain the biology behind the connectivity.

Have fun!

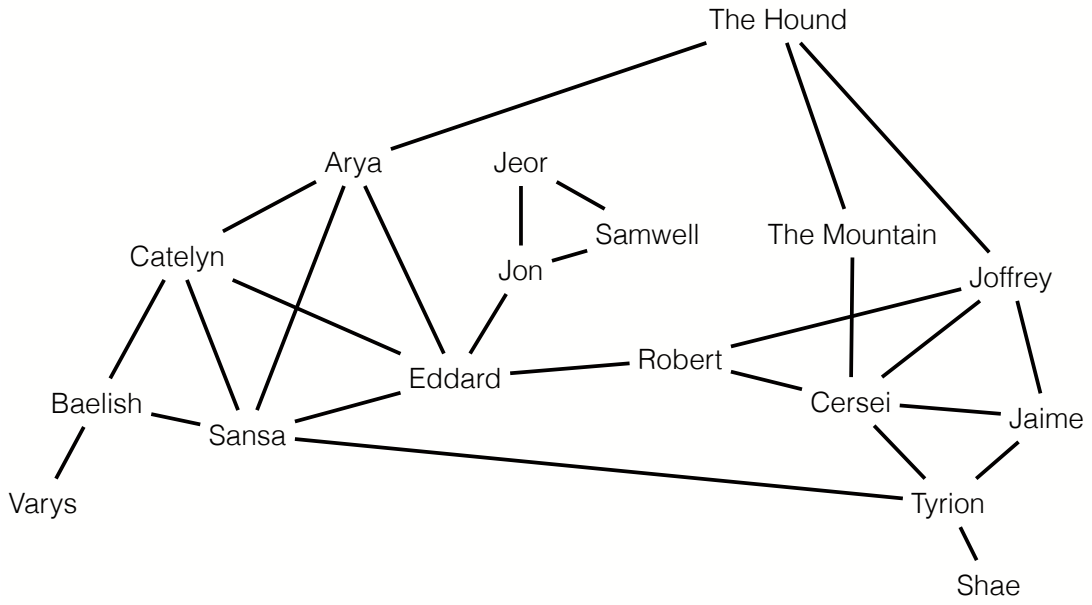

Figure 1: The network in "GoT.txt" visualized.