User's guide for cvis - a program to visualize networks using clusters

May 19, 2011

1 Get Started

The program comes along with a file called **compile_all.sh**.

Type:

./compile.sh

from a Unix (MAC) terminal. If you are using Windows, you could still run the program by installing MinGW (Minimalist GNU for Windows, `http://www.mingw.org/`). If you see something like **./compile.sh: Permission denied**, type:

chmod 744 compile.sh

which makes the script executable and try again with:

./compile.sh.

Now try:

./cvis_undir -f network.dat -folder network.dat_oslo_files

2 Output files

If you open the folder **network.dat_cvis** , you will see a few files:

- **cvis_0.gdf** is the most important: it can be opened using gephi (`http://gephi.org/`). You should see two clusters of nodes with the same colors, one white (not assigned) node and one bright overlapping node.
- **cvis_1.gdf** will visualize the same network where nodes belonging to the lowest level are aggregated.
- **node_memberships.dat** reports the number of clusters each node belongs for the different hierarchical levels (from fine levels to coarse ones)
- **modules_1.dat** reports the nodes contained in each module of the first hierarchical level.

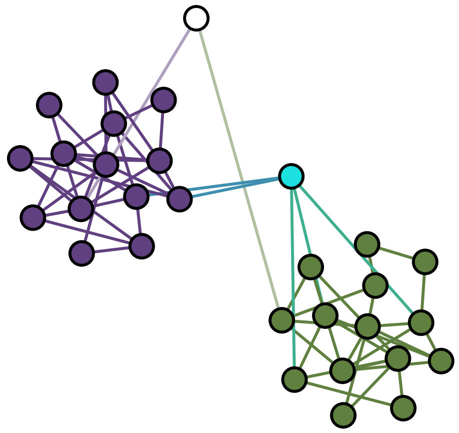

Figure 1: Visualization of **network.dat_cvis/cvis_0.gdf**

3 Input files and Options

To run the program, you first need to run a clustering algorithm and find the modules you want to visualize. The modules can be hierarchical and overlapping.

Choose between **cvis_undir** (undirected networks) and **cvis_dir** (directed networks). Then, you need to select the network and the partitions:

1. **-f network_file**: network_file is expected to contain the edge list, with possible weights.

2. **-folder network_file_folder**: network_file_folder is the folder which contains the partitions. It is expected to contain a file called **tp**, which is the set of (overlapping) modules at the lowest hierarchical level (each row corresponds to a cluster). The higher hierarchical modules are specified in files **short_tp1**, **short_tp2** .... **short_tp1** gives the second hierarchical level modules: the ids refers to the modules written in **tp** indexed in order of appearance (for instance, a row like 0 1 2 means that the first, second and third modules of **tp** are put together at this level). This folder is formatted as the output given by OSLOM [1]

3. **-tree network_tree**: network_tree is a unique file which specifies all the partitions. This option can be used instead of option **-folder**. Here, each row contains the module assignments from the coarse to the fine level. The last word is the node id. This is the same format given by hierarchical infomap [2].

4 Other Options

1. **-labels filename**: filename contains the node labels, which are strings. The format of the file is: string node_id. For an example, look at file **labels_nodeid.dat**. Please note that the node labels should not contain characters like **, -** or **"**, otherwise gephi gets confused.

2. **-opt U [0, 5]**: U is an optimization parameter. Higher values of U take more time and lead to better results. Default value is 2.

3. **-gap U**: this is to decide how far the modules should be. Default is 1.4. Reasonable values should be between 1 and 2.

4. **-singlets U**: this is a parameter used to locate singletons (clusters with one only node). Default is 0.5 (must be U> 0). Bigger values create a layout where singletons occupy more space (more CPU consuming).

5. **-seed U**: seed for the random number generator. Default value is read from file **time_seed.dat**, and is automatically increased every time that file is opened.

5 References

1. **oslom**: `http://www.oslom.org/`

2. **hierarchical infomap**: `http://www.tp.umu.se/~rosvall/code.html`