

User's guide

November 18, 2016

Thanks for downloading this package. You just got a selected collection of clustering algorithms, a visualization method and the consensus clustering algorithm.

1 Compiling

The program comes along with a file called `compile.sh`.

Please 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`, please type:

```
chmod 744 compile.sh
```

which makes the script executable and try again with:

```
./compile.sh.
```

After a short while, you will see a new folder, `bin`. Please do not move it! The python script `select.py` will call the binary files located in `bin`, and everything will work fine as long as `select.py`, `methods.py` and `bin` will be in the same folder as they are now.

2 Inputs

Let's go ahead and type:

```
python select.py
```

There are different arguments that you need to provide. Blue flags are optional, red ones are needed.

- **-n [network file]**

[network file] should be in the edge list format, where every edge is in a different row, as:

```
node1 node2
```

If the network is weighted:

node1 node2 w_{12}

node ids should be non negative integers, but they don't have to be consecutive and can start from whatever number. Weights should be positive. For directed network “**node1 node2**” means **node1** links to **node2**. For undirected, the order does not matter and links do not need to be repeated.

- **-p** [program number] Select an algorithm from the list which also appears when you type

`python select.py`

0: oslom undirected

1: oslom directed

2: infomap undirected

3: infomap directed

4: louvain

5: label propagation method

6: hierarchical infomap undirected

7: hierarchical infomap directed

8: modularity optimization (simulated annealing zero temperature)

- **-f** [folder] folder is a directory where all the results and some temporary files will be written. If the folder already exists, the script will not run, please remove the folder beforehand.
- **-c** [consensus runs] This sets the number of partitions for the consensus algorithm. If you set it to 1, the consensus algorithm is skipped.
- **-l** [label file] label file is used just for the visualizations. It provides the names of the nodes (strings). The format is

“**name of node1**” **node1**

where **node1** is an integer (the id of node1 in the network file). The use of “ ” is recommended although the program should be flexible enough to figure out what you meant anyway.

- **-t** [threshold] threshold is a float number for pruning the consensus matrix. Default value is 0.5.

- **-slices** [slice_file] This option is to find a consensus from partitions computed from different networks. For instance, if you have temporal networks $G_1, G_2, G_3, G_4 \dots$, and you want to find a median partition for G_1, G_2, G_3 , `slice_file` should look like:

```
path_to_file.G_1
path_to_file.G_2
path_to_file.G_3
```

The program will collect [consensus runs] partitions for each slice and find the consensus among all these partitions. The last network in `slice_file` will be used for the visualization in the folder `results.consensus`. Please try the example:

```
python select.py -slices datasets/slice_file -p 4 -f slice_test -c 5
```

3 Outputs

The results will be in the folder name you specify.

Together with some temporary files you can discard, you will see different folders as `results_1`, `results_2`, `results_3` ..., one for each run (the number depends on how many runs you set) and `results_consensus`. Each folder will contain files like `tp`, `tp1` ... which reports the partition (each row is a different module) for the different hierarchical levels, `tp` is the bottom level. `cvis_0.gdf`, `cvis_1.gdf` ... are files which can be loaded by gephi: `cvis_0.gdf` is the most coarse grained while the original network is in the file with the the highest index.

NB. If you are running the consensus method with a hierarchical method, the consensus matrix will be constructed only for the bottom level partition. For instance, `results_consensus/tp1` is computed from the consensus matrix of the bottom level partitions `results_/tp`, `results_2/tp`, and not from `results_/tp1`, `results_2/tp1`.

4 Skipping select.py

You can run the bin files directly. This is faster if you are just interested in those or you want to specify more parameters. Run the algorithm you like without arguments to get instructions.

5 References

If you used the consensus algorithm for your research, please cite this paper:

A. Lancichinetti, S Fortunato

Consensus clustering in complex networks

Scientific Reports 2, 336 doi:10.1038/srep00336 (2012)

In addition, you should cite the proper reference for each method you used:

Oslo

A. Lancichinetti, F. Radicchi, J.J. Ramasco and S. Fortunato *Finding statistically significant communities in networks* PLoS One 6, e18961 (2011)

Infomap

Rosvall, M. and Bergstrom, C. T. *Maps of random walks on complex networks reveal community structure.* Proc. Natl. Acad. Sci. USA 105, 11181123 (2008).

Louvain

V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre *Fast unfolding of community hierarchies in large networks.* J. Stat. Mech. 2008 (10): P10008

Label Propagation Method

Raghavan, U. N., Albert, R. and Kumara, S. *Near linear time algorithm to detect community structures in large-scale networks.* Phys. Rev. E 76, 036106 (2007).

Hierarchical Infomap

Martin Rosvall and Carl T. Bergstrom *Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems.* PLoS ONE 6(4): e18209 (2011).

Modularity Optimization (Simulated Annealing)

Sales-Pardo, M., Guimer, R., Moreira, A. A. and Amaral, L. A. N *Extracting the hierarchical organization of complex systems.* Proc. Natl. Acad. Sci. USA 104, 1522415229 (2007).

6 Licence, Credits and Contacts

All the implementations are Andrea Lancichinetti's except to Infomap and Hierarchical Infomap which are Martin Rosvall's and can be also found from:

<http://www.tp.umu.se/~rosvall/code.html>

I would also appreciate if you like to include a URL link to download the code in your paper.

All the codes GPL licensed.

Comments and feedbacks are highly welcome. Do not hesitate to drop me an email: arg.lanci@gmail.com

Special thanks to Paolo Bajardi for pushing me to arrange this small repository.

7 Contribute!

If you developed a clustering algorithm and would like it to be included here, please write me. I will be glad to add other methods.