

Clustering with MCL

[Main](#)[Characteristics](#)[Communities](#)[Speed & memory](#)[Cite/publications](#)[Other](#)

Network analysis

[MCL-edge](#)[Tools](#)[Protocols](#)[News & Views](#)[Questions](#)[License & software](#)

Documentation:

[Overview I](#)[Overview II](#)[Discovery](#)[MCL manual](#)[FAQ](#)

1. [NAME](#)
2. [GETTING STARTED](#)
3. [SYNOPSIS](#)
4. [DESCRIPTION](#)
5. [OPTIONS](#)
6. [PRUNING OPTIONS](#)
7. [IMPLEMENTATION OPTIONS](#)
8. [EXAMPLES](#)
9. [APPLICABILITY](#)
10. [FILES](#)
11. [ENVIRONMENT](#)
12. [DIAGNOSTICS](#)
13. [BUGS](#)
14. [AUTHOR](#)
15. [HISTORY/CREDITS](#)
16. [SEE ALSO](#)
17. [REFERENCES](#)
18. [NOTES](#)

NAME

mcl — The Markov Cluster Algorithm, aka the MCL algorithm.

This program implements **mcl**, a cluster algorithm for graphs. A single parameter controls the granularity of the output clustering, namely the **-i inflation** option described further below. In standard usage of the program this parameter is the only one that may require changing. By default it is set to 2.0 and this is a good way to start. If you want to explore cluster structure in graphs with MCL, vary this parameter to obtain clusterings at different levels of granularity. A good set of starting values is 1.4, 2, 4, and 6.

The program has a rather large set of options. Except for **-i** none affects the clustering method itself. The other options are for a variety of aspects, such as study of the underlying MCL process (i.e. dumping of iterands), network preprocessing (incorporated for efficiency), resource allocation options (for large-scale analyses), output naming and placement, output formatting, setting of verbosity levels, and so on.

Network construction and reduction techniques should not be considered as part of a clustering algorithm. Nevertheless particular techniques may benefit particular methods or applications. In mcl many transformations are accessible through the **-tf** option. It can be used for edge weight transformations and selection, as well as transformations that act on a graph as a whole. It is for example possible to remove edges with weight below 0.7 by issuing **-tf 'gq(0.7)'**, where the quotes are necessary to prevent the shell from interpreting the parentheses. The option accepts more complicated sequences, such as **-tf 'gq(0.7),add(-0.7)'**. This causes all remaining edge weights to be shifted to the range [0-0.3], assuming that the input contains correlations. Many more transformations are supported, as