NAME

mcx ctty – compute betweenness centrality for network nodes or network edges

SYNOPSIS

mcx ctty [options] [matrix-file]

mcxctty is not in actual fact a program. This manual page documents the behaviour and options of the mcx program when invoked in mode *ctty*. The options **-h**, **--apropos**, **--version**, **-set**, **--nop**, **-progress** <*num*> are accessible in all **mcx** modes. They are described in the **mcx** manual page.

mcx ctty **[-abc** <fname> (*specify label input*)**]** **[-imx** <fname> (*specify matrix input*)**]** **[-extent** <int> (*only consider paths of length at most* <*int*>)**]** **[--edge** (*compute edge betweenness centrality*)**]** **[-o** <fname> (*output file name*)**]** **[-tab** <fname> (*use tab file*)**]** **[-t** <int> (*use* <*int*> *threads*)**]** **[-J** <intJ> (*a total of* <*intJ*> *jobs are used*)**]** **[-j** <intj> (*this job has index* <*intj*>)**]** **[-h** (*print synopsis, exit*)**]** **[--apropos** (*print synopsis, exit*)**]** **[--version** (*print version, exit*)**]**

DESCRIPTION

mcx ctty computes betweenness centrality for all nodes or all edges in a graph, using the between centrality update algorithm from [1]. The program is not yet able to take into account edge weights.

The input graph/matrix, if specified with the **-imx** option, has to be in mcl matrix/graph format. You can use label input instead by using the **-abc** option. Refer to **mcxio(5)** for a description of these two input formats. By default **mcx diameter** reads from STDIN *and expects matrix format*. To specify label input from STDIN use **-abc -**.

OPTIONS

-abc <fname> (*label input*)
The file name for input that is in label format.

-imx <fname> (*input matrix*)
The file name for input. STDIN is assumed if not specified.

-o <fname> (*output file name*)
The name of the file to write output to.

-extent <int> (*only consider paths of length at most* <*int*>)
This option will lead to different results. Results will still be informative however, being representative for the local context in which nodes reside. It does probably not make sense to use values smaller than four or five.

--edge (*compute edge betweenness centrality*)
The output will be a matrix rather than a labeled list of values. If the program is split into multiple jobs with the **-j** and **-J** options, the resulting outputs have to be collated using `mcx collect --add-matrix` followed by the respective output file names for each of the jobs.

-tab <fname> (*use tab file*)
This option causes the output to be printed with the labels found in the tab file. With **-abc** this option will, additionally, construct a graph only on the labels found in the tab file. If this option is used in conjunction with **-imx** the tab domain and the matrix domain are required to be identical.

-t <int> (*use* <*int*> *threads*)
-J <intJ> (*a total of* <*intJ*> *jobs are used*)

-j <intj> (*this job has index <intj>*)

Computing centrality scores in a graph is time-intensive for large graphs. If you have multiple CPUs available consider using as many threads. Additionally it is possible to spread the computation over multiple jobs/machines. These three options are described in the **clm-protocols** manual page. The following set of options, if given to as many commands, defines three jobs, each running four threads.

```
-t 4 -J 3 -j 0 -o out.0
-t 4 -J 3 -j 1 -o out.1
-t 4 -J 3 -j 2 -o out.2
```

For node centrality (the default mode) the output can then be collected with

```
mcx collect --add-column -o out.all out.[0-2]
```

For edge centrality (invoked with **--edge**) the output should be collected with

```
mcx collect --add-matrix -o out.all out.[0-2]
```

This collection step is only necessary if the **-J** was used with a value greater than one. In particular, if threads were used but jobs were not, there will be a single output corresponding to the end result.

REFERENCES

[1] Ulrik Brandes, *A Faster Algorithm for Betweenness Centrality*. Journal of Mathematical Sociology 25(2): 163-177, (2001). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.2024

SEE ALSO

mcxio(5), and **mclfamily(7)** for an overview of all the documentation and the utilities in the mcl family.