NAME

clm info2 – compute performance measures for graphs and clusterings.

clminfo2 is not in actual fact a program. This manual page documents the behaviour and options of the clm program when invoked in mode *info2*. The options **-h**, **--apropos**, **--version**, **-set**, **--nop** are accessible in all **clm** modes. They are described in the **clm** manual page.

SYNOPSIS

clm info2 [options] <graph file> <cluster file> <cluster file>*

clm info2 [-o fname (*write to file* **fname)] [-pi f (***apply inflation beforehand***)] [--list (***list efficiency for all nodes***)] [-tf spec (***apply tf-spec to input matrix***)] [-cl-ceil <num> (***skip clusters of size exceeding* **<num>)] [-cat-max <num> (***do at most* **<num>** *tree levels***)] [-cl-tree fname (***expect file with nested clusterings***)] [-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***)] <matrix file> <cluster file> <cluster file>***

DESCRIPTION

clm info2 is a streamlined and updated version of **clm info**. The latter outputs a key-value format listing a number of measures. In contrast, **clm info2** only outputs the so-called efficiency criterion, a quality index for networks and clusterings. This criterion can be generated for each node independently with the **--list** option, indicating how well a clustering captures the neighbour distribution of a given node.

clm info2 can utilise threading and job dispatching. This may be useful when dealing with very large graphs.

Multiple clusterings can be supplied on the command-line. Output is tabular, each row corresponding with a clustering in the ordering as supplied on the command line. Multiple columns will result only if node-wise output is induced with **--list**. By default a single number is produced for each individual clustering: the mean of all node-wise scores for that clustering.

The **efficiency** factor is described in [1] (see the **REFERENCES** section). It tries to balance the dual aims of capturing a lot of edges or edge weights and keeping the cluster footprint or area fraction small. The efficiency number has several appealing mathematical properties, cf. [1].

OPTIONS

-o fname (*output file name*)

-pi f (*apply inflation beforehand*)
Apply inflation to the graph matrix and compute the performance measures for the result.

-tf <tf-spec> (*transform input matrix values*)
shared_defopt{-tf}

--list (*list efficiency for all nodes*)
The efficiency scores for all nodes are given on a single line. Each clustering specified corresponds to a single line.

-cl-tree fname (*expect file with nested clusterings (cone format)*)
-cl-ceil <num> (*skip (nested) clusters of size exceeding <num>*)
The specified file should contain a hierarchy of nested clusterings such as generated by **mclcm**. The output is then in a special format, undocumented but easy to understand. Its purpose is to help cherrypick a single clustering from a tree, in conjunction with the slightly experimental and undocumented program **mlmfifofum**.

The measure that is used is very slow to compute for large clusters, and generally it will be outside any interesting range (i.e. it will be small). Use **-cl-ceil** to skip clusters exceeding the specified size – **clm info** will directly proceed to subclusters if they exist.

-cat-max num (*do at most num levels*)
This only has effect when used with **-cl-tree**. **clm info** will start at the most fine-grained level, working upwards.

-t <int> (*use <int> threads*)
-j <intj> (*this job has index <intj>*)
-J <intJ> (*a total of <intJ> jobs are used*)
For very large graphs (millions of nodes) and clusterings with large clusters it may be helpful to allow this program to use multiple CPUs. Additionally it is possible to spread the computation over multiple jobs/machines. These three options are described in the **clmprotocols** 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
```

The output can then be collected with

```
clxdo add_table out.[0-2]
```

AUTHOR

Stijn van Dongen.

SEE ALSO

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

REFERENCES

[1] Stijn van Dongen. *Performance criteria for graph clustering and Markov cluster experiments*. Technical Report INS-R0012, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, May 2000.
http://www.cwi.nl/ftp/CWIreports/INS/INS-R0012.ps.Z