NAME

mcxquery − compute simple graph statistics

SYNOPSIS

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

mcxquery **[-abc** <fname> (*specify label input*)**]** **[-imx** <fname> (*specify matrix input*)**]** **[-o** <fname> (*output file name*)**]** **[-tab** <fname> (*use tab file*)**]** **[--node-attr** (*output node degree and weight attributes*)**]** **[-vary-threshold** <start/end/step> (*analyze graph at similarity cutoffs*)**]** **[-vary-knn** <start/end/step> (*analyze graph for varying k-NN*)**]** **[-vary-ceil** <start/end/step> (*analyze graph for varying ceil reductions*)**]** **[--no-legend** (*do not output explanatory legend*)**]** **[--reduce** (*use reduced matrix*)**]** **[--test-metric** (*test whether graph distance is metric*)**]** **[--test-cycle** (*test whether graph contains cycles*)**]** **[-test-cycle** <num> (*test cycles, report cycles*)**]** **[--vary-correlation** (*analyze graph at correlation cutoffs*)**]** **[--clcf** (*include clustering coefficient analysis*)**]** **[--eff** (*include efficiency criterion*)**]** **[-div** <num> (*cluster size separating value*)**]** **[--dim** (*report native format and dimensions*)**]** **[--values** (*output all arc entries/weights, unsorted*)**]** **[--values-sorted** (*output all entries/weights, sorted*)**]** **[-values-hist** <nbins|start/end/nbins> (*weight histogram*)**]** **[-degrees-hist** <step> (*degrees histogram*)**]** **[--output-table** (*output logical tab separated table without key*)**]** **[-t** <num> (*number of threads to use*)**]** **[-icl** <fname> (*input clustering*)**]** **[-tf** spec (*apply tf-spec to input matrix*)**]** **[-h** (*print synopsis, exit*)**]** **[--apropos** (*print synopsis, exit*)**]** **[--version** (*print version, exit*)**]**

DESCRIPTION

The default **mcxquery** output is a list of summary statistics for each node. These are its node degree, the mean, minimum, maximum and median edge weight. If supplied with a clustering, the output will additionally list the cluster size and cluster label for each node.

Additionally, **mcxquery** can be used to analyse a graph at different similarity cutoffs or at varying parameters of edge reduction strategies such as mutual nearest neighbour reduction. Attributes supplied across different thresholds are the number of connected components, the number of singletons, and statistics (median, average, iqr) on node degrees and edge weights. Typically this is done on a graph constructed using a very permissive threshold. For example, one can create a graph from array expression data using **mcxarray** with a very low pearson correlation cutoff such as 0.5 Then **mcxquery** can be used to analyze the graph at increasingly stringent thresholds of 0.50, 0.55, 0.60 .. 0.95.

Other tasks that **mcxquery** be used for include:

- Produce a histogram of edge weights.
- Produce a histogram of edge node degrees.
- Output all edge weights.
- Test whether the graph weight encodes a metric (for edge weights that encode distances rather than similarites).
- Test whether the graph has a cycle.

OPTIONS

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

-imx <fname> (*input matrix*)
The file name for input that is in mcl native matrix format.

-o <fname> (*output file name*)

Set the name of the file where output should be written to.

-tab <fname> (*use tab file*)
This option causes the output to be printed with the labels found in the tab file.

--dim (*report native format and dimensions*)
This will report the matrix format (either interchange or binary) and the matrix dimensions. For a graph the two reported dimensions should be equal.

--values (*output all entries/weights, unsorted*)
--values-sorted (*output all entries/weights, sorted*)
-values-hist <start/end/nbins> (*output weight histogram*)
-values-hist <nbins> (*output weight histogram*)
-degrees-hist <nbins> (*degrees histogram*)
These options are fairly self-documenting. The result of both **-edges-hist** and **-degrees-hist** is a tab separated table of bin offsets and bin counts. When using **-edges-hist** *<nbins>* the program will create a histogram ranging from the smallest to the largest edge weight.

--output-table (*output logical tab separated table without key*)
This option causes table output such as provided by **--vary-correlation** to be output in a logical tab-separated format rather than pretty-printed.

-vary-threshold <start/end/nbins> (*analyze graphs at similarity cutoffs*)
The graph is analysed at different edge weight thresholds, going from **<start>** to **<end>** in **<nbins>** steps.

--vary-correlation (*analyze graphs at correlation cutoffs*)
This instructs **mcxquery** to use a threshold list suitable for use with graphs in which the edge weight similarities are correlations. The list starts at 0.2 and ends at 1.0 using increments of 0.05. If a different start or increment is required it can be achieved by using the **-vary-threshold** option. For example, a start of 0.10 and an increment of 0.02 are obtained by issuing **-vary-threshold :.1/1.0/45**.

defopt{--no-legend}{do not output explanatory legend}
For a fully parseable output format use **--output-table**.

--clcf (*include clustering coefficient analysis*)
--eff (*include efficiency criterion*)
These options can be used to compute additional characteristics in the analysis of thresholded graphs with **--vary-correlation** and **-vary-threshold**. For large graphs these are relatively time-consuming to compute. More information and a reference for the efficiency criterion can be found in **clminfo(1)**.

-vary-knn <start/end/step> (*analyze graphs for varying k-NN*)
-vary-ceil <start/end/step> (*analyze graphs for varying ceil reductions*)
--reduce (*use reduced matrix*)
These options cause analysis of a graph as it is subjected to reductions across a range of parameters. Refer to **mcxio(5)** for a description of these reductions. The analyses starts at the *end* argument, and progresses towards the *start* argument using decrements of size *step*. By default the reduction is always computed relative to the start matrix, i.e. the input matrix after **-tf** transformations have optionally been applied. Specifying **--reduce** causes this to change so that each new reduction is calculated relative to the reduction just computed.

For graphs with ties among edge weights it may be useful to use **-tf '#tug()'**. This will add small perturbations to the edge weights and have the effect of breaking ties. By default perturbations are computed using the cosine between the vectors of neighbours of the two nodes incident to an edge. This can be changed to a random perturbation with **-tf '#rug()'**.

--test-cycle (*test whether graph contains cycles*)
-test-cycle <num> (*test cycles, report cyclees*)
Test whether the input graph contains cycles. With the second option nodes that are part of a cycle are output, up to a maximum of *<num>* nodes. Use *<num>*=**-1** to output all such nodes.

--test-metric (*test whether graph distance is metric*)
This tests all possible triangle relationships.

-div <num> (*cluster size separating value*)
When analyzing graphs at different thresholds with one of the options above, **mcxquery** reports the percentage of nodes contained in clusters not exceeding a specified size, by default 3. This number can be changed using the **-div** option.

-tf <tf-spec> (*transform input matrix values*)
Transform the input matrix values according to the syntax described in **mcxio(5)**.

-t <num> (*number of threads to use*)
This has an effect only when using the **-vary-knn** option, and is only useful on multi-CPU machines.

--node-attr (*output node degree and weight attributes*)
Output is in the form of a tab separated file. The option **-icl** can be used in conjuction.

-icl <fname> (*input clustering*)
Output for each node the size of the cluster it is in. This option can be used in conjunction with **--node-attr**.

SEE ALSO

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