NAME

mcxarray − Transform array data to MCL matrices

SYNOPSIS

mcxarray [options]

mcxarray **[-data** fname (*input data file*)**]**
[-imx fname (*input matrix file*)**]**
[-co num (*(absolute) cutoff for output values (required)*)**]**
[-skipr <num> (*skip <num> data rows*)**]**
[-skipc <num> (*skip <num> data columns*)**]**
[-o fname (*output file fname*)**]**
[--text-table (*write output in full text table format*)**]**
[-write-tab <fname> (*write row labels to file*)**]**
[-l <num> (*take labels from column <num>*)**]**

[--pearson (*use Pearson correlation (default)*)**]**
[--spearman (*use Spearman rank correlation*)**]**
[--dot (*use dot product*)**]**
[--cosine (*use cosine (similarity)*)**]**
[--slow-cosine (*use cosine(0.5 alpha) (similarity)*)**]**

[--angle (*use angle between vectors (note: a metric distance)*)**]**
[--acute-angle (*use acute angle between vectors*)**]**
[--angle-norm (*use normalised angle between vectors (by pi)*)**]**
[--acute-angle-norm (*use normalised acute angle between vectors (by pi/2)*)**]**
[--sine (*use sine (note: a metric distance)*)**]**
[--slow-sine (*use sine(0.5 alpha) (note: a metric distance)*)**]**
[--euclid (*use Euclidean distance between vectors*)**]**
[--max (*use L-oo, aka Chebyshev distance*)**]**
[--taxi (*use L-1, aka taxi, aka city-block distance*)**]**
[-minkowski <num> (*use Minkowski distance with power <num>*)**]**
[-fp <mode> (*use fingerprint measure*)**]**

[-digits <num> (*output precision*)**]**
[--write-binary (*write output in binary format*)**]**
[-t <int> (*use <int> threads*)**]**
[-J <intJ> (*a total of <intJ> jobs are used*)**]**
[-j <intj> (*this job has index <intj>*)**]**
[-start <int> (*start at column <int> inclusive*)**]**
[-end <int> (*end at column <int> EXclusive*)**]**
[--transpose-data (*work with the transposed data matrix*)**]**
[--rank-transform (*rank transform the data first*)**]**
[-tf spec (*transform result network*)**]**
[-table-tf spec (*transform input table before processing*)**]**
[-n mode (*normalize input*)**]**
[--zero-as-na (*treat zeroes as missing data*)**]**
[--sparse (*do not store zero values*)**]**
[-write-data <fname> (*write data to file*)**]**
[-write-na <fname> (*write NA matrix to file*)**]**
[--job-info (*print index ranges for this job*)**]**
[--help (*print this help*)**]**
[-h (*print this help*)**]**
[--version (*print version information*)**]**

DESCRIPTION

mcxarray can either read a flat file containing array data (**-data**) or a matrix file satisfying the mcl input format (**-imx**). In the former case it will by default work with the rows as the data vectors. In the latter case it will by default work with the columns as the data vectors (note that mcl matrices are presented as a listing of columns). This can be changed for both using the **--transpose-data** option.

The input data may contain missing data in the form of empty columns, NA values (not available/applicable), or NaN values (not a number). The program keeps track of these, and when computing the correlation between two rows or columns ignores all positions where any one of the two has missing data.

OPTIONS

-data fname (*input data file*)
Specify the data file containing the expression values. It should be tab-separated.

-imx fname (*input matrix file*)
The expression values are read from a file in mcl matrix format.

--pearson (*use Pearson correlation (default)*)
--spearman (*use Spearman rank correlation*)
--cosine (*use cosine*)
--slow-cosine (*use cosine(0.5 alpha) (similarity)*)
--dot (*use the dot product*)
All these measures express the level of similarity or correlation between two vectors. Note that the dot product is not normalised and should only be used with very good reason. A few more similarity measures are provided by the fingerprint option **-fp** described below.

-fp <mode> (*specify fingerprint measure*)
Fingerprints are used to define an entity in terms of it having or not having certain traits. This means that a fingerprint can be represented by a boolean vector, and a set of fingerprints can be represented by an array of such vectors. In the presence of many traits and entities the dimensions of such a matrix can grow large. The sparse storage employed by MCL-edge is ideally suited to this, and mcxarray is ideally suited to the computation of all pairwise comparisons between such fingerprints. Currently mcxarray supports five different types of fingerprint, described below. Given two fingerprints, the number of traits unique to the first is denoted by *a*, the number unique to the second is denoted by *b*, and the number that they have in common is denoted by *c*.

hamming
The Hamming distance, defined as *a*+*b*.

tanimoto
The Tanimoto similarity measure, *c*/(*a*+*b*+*c*).

cosine
The cosine similarity measure, *c*/sqrt((*a*+*c*)*(*b*+*c*)).

meet
Simply the number of shared traits, identical to *c*.

cover
A normalised and non-symmetric similarity measure, representing the fraction of traits shared relative to the number of traits by a single entity. This gives the value *c*/(*a*+*c*) in one direction, and the value *c*/(*b*+*c*) in the other.

--sine (*use sine (note: a metric distance)*)
--slow-sine (*use sine(0.5 alpha) (note: a metric distance)*)
--angle (*use angle between vectors (note: a metric distance)*)
--acute-angle (*use acute angle between vectors*)
--angle-norm (*use normalised angle between vectors (by pi)*)
--acute-angle-norm (*use normalised acute angle between vectors (by pi/2)*)
--euclid (*use Euclidean distance between vectors*)
--max (*use L-oo, aka Chebyshev distance*)
--taxi (*use L-1, aka taxi, aka city-block, aka Manhattan distance*)
-minkowski <num> (*use Minkowski distance with power <num>*)
All these measures express the level of dissimilarity or distance between two vectors.

-skipr <num> (*skip <num> data rows*)
Skip the first *<num>* data rows.

-skipc <num> (*skip <num> data columns*)
Ignore the first *<num>* data columns.

-l <num> (*take labels from column <num>*)
Specifies to construct a tab of labels from this data column. The tab can be written to file using **-write-tab** *fname*.

-write-tab <fname> (*write row labels to file*)
Write a tab file. In the simple case where the labels are in the first data column it is sufficient to issue **-skipc 1**. If more data columns need to be skipped one must explicitly specify the data column to take labels from with **-l** *l*.

-t <int> (*use <int> threads*)
-J <intJ> (*a total of <intJ> jobs are used*)
-j <intj> (*this job has index <intj>*)
Computing all pairwise correlations is time-intensive for large input. 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
```

The output can then be collected with

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

--job-info (*print index ranges for this job*)
-start <int> (*start at column <int> inclusive*)
-end <int> (*end at column <int> EXclusive*)
--job-info can be used to list the set of column ranges to be processed by the job as a result of the command line options **-t**, **-J**, and **-j**. If a job has failed, this option can be used to manually split those ranges into finer chunks, each to be processed as a new sub-job specified with **-start** and **-end**. With the latter two options, it is impossible to use parallelization of any kind (i.e. any of the **-t**, **-J**, and **-j** options).

-o fname (*output file fname*)
Output file name.

--text-table (*write output in full text table format*)
The output will be written in tabular format rather than native **mcl-edge** format.

-digits <num> (*output precision*)
Specify the precision to use in native interchange format.

--write-binary (*write output in binary format*)
Write output matrices in native binary format.

-co num (*(absolute) cutoff for output values*)
Output values of magnitude smaller than *num* are removed (set to zero). Thus, negative values are removed only if their positive counterpart is smaller than *num*.

--transpose-data (*work with the transpose*)
Work with the transpose of the input data matrix.

--rank-transform (*rank transform the data first*)
The data is rank-transformed prior to the computation of pairwise measures.

-write-data <fname> (*write data to file*)
This writes the data that was read in to file. If **--spearman** is specified the data will be rank-transformed.

-write-na <fname> (*write NA matrix to file*)
This writes all positions for which no data was found to file, in native mcl matrix format.

--zero-as-na (*treat zeroes as missing data*)
This option can be useful when reading data with the **-imx** option, for example after it has been loaded from label input by **mcxload**. An example case is the processing of a large number of probe rankings, where not all rankings contain all probe names. The rankings can be loaded using **mcxload** with a tab file containing all probe names. Probes that are present in the ranking are given a positive ordinal number reflecting the ranking, and probes that are absent are implicitly given the value zero. With the present option mcxarray will handle the correlation computation in a reasonable way.

--sparse (*do not store zero data value*)
With this option internal calculations are performed on compressed data where zeroes are not stored. This can be useful when the input data is very large.

-n mode (*normalization mode*)
If *mode* is set to **z** the data will be normalized based on z-score. No other modes are currently supported.

-tf spec (*transform result network*)
-table-tf spec (*transform input table before processing*)
The transformation syntax is described in **mcxio(5)**.

--help (*print help*)
-h (*print help*)

--version (*print version information*)

AUTHOR

Stijn van Dongen.

SEE ALSO

mcl(1), **mclfaq(7)**, and **mclfamily(7)** for an overview of all the documentation and the utilities in the mcl family.