NetOnZeroDXC Software Package

v1.0

User Manual

April 2019

Contents

General Overview & License

NetOnZeroDXC (*network assessment based on zero delay cross correlation*) is a package that provides a set of software tools to identify links between elements, or "nodes", of a possible network structure, where each node is associated to a time series. The core of the algorithm is the assessment of zero-delay cross-correlation between time series.

The algorithm was originally published in A. Perinelli, D. E. Chiari and L. Ricci, *Correlation in brain networks at different time scale resolution*, Chaos **28**(6):063127, 2018. This paper is henceforth referred to as **"main reference"**. Citing also this reference in manuscripts describing works that rely on this package is appreciated.

This package consists of two graphical user interface (GUI) apps, **`netOnZeroDXC_analysis`** and **`netOnZeroDXC_merge`**, and two command line programs, **`netOnZeroDXC_diagram`**, **`netOnZeroDXC_efficiency`**.

Functions devoted to computation are defined in a shared source file. Both the GUI apps and the command line programs are provided so that users can optimize their analysis pipeline according to their needs and available hardware facilities. The GUI apps provide previews of the results thus allowing for quick evaluation of the analysis outcomes.

This package, all the included source code, documentation and examples are released under the GNU General Public License (GPL) v3. A copy of the license is provided in the package root folder.

The source code is written in C++. The GUI is built on the **wxWidgets library**, released under the *wxWindows Library Licence*. Part of the numerical algorithms rely on functions provided by the **GNU Scientific Libraries** (GSL), released under GPL. For the Windows version of the package, binaries for the libraries are statically linked in the executable files included within the package. On Linux or Mac OS the package has to be compiled from the source code; installation of the libraries is required. The wxWidgets library is not required for the compilation of the command line programs.

Authors

Alessio Perinelli and Leonardo Ricci

Department of Physics, University of Trento

`I-38123` Trento, Italy

alessio.perinelli@unitn.it

leonardo.ricci@unitn.it

Websites

Package repository: `https://github.com/LeonardoRicci/netOnZeroDXC`

wxWidgets library: `https://www.wxwidgets.org/`

GNU Scientific Libraries: `https://www.gnu.org/software/gsl/`

Package Installation

Windows

For Windows users we provide a 64bit installer that installs the package without requiring any additional software. Please refer to the README file in the corresponding `/install/Windows` folder before installing the package.

The installation folder is added to the path registry key, so that the command line programs can be launched in a shell from any directory.

The compilation from the package starting from source code is also possible. This operation requires a copy of the GNU Scientific Libraries binaries for the target compiler as well as the library headers. Concerning the wxWidgets library, the official wxWdigets website provides the requested binaries.

Linux

Linux users have to compile the package on their platform.

As a preliminary step, the necessary libraries have to be installed. A complete list of the required packages and how to install them can be found in the README file under `/install/Linux.` Both the GSL and wxWidgets can either be installed as a software package via a package manager, or downloaded and compiled from the source code. The former method is described in the README file contained in `/install/Linux`.

Once the necessary libraries are installed, the package can be compiled via a Makefile. The related file, contained in the `/install/Linux/build` folder, is expected to work on any standard Linux installation, although it might be necessary to trim it on the target platform by changing some of the parameters listed at the top of the Makefile. The Makefile also provides two optional recipes to “install” the programs, namely linking or copying them to the `/usr/bin` directory.

Mac OS

Mac users have to compile the package on their platform.

MacOS-specific files are not provided. Nevertheless, once the wxWidgets and the GSL libraries are installed from the corresponding official websites (or compiled from source code), a suitable adaptation of the Linux Makefile is expected to successfully install the package.

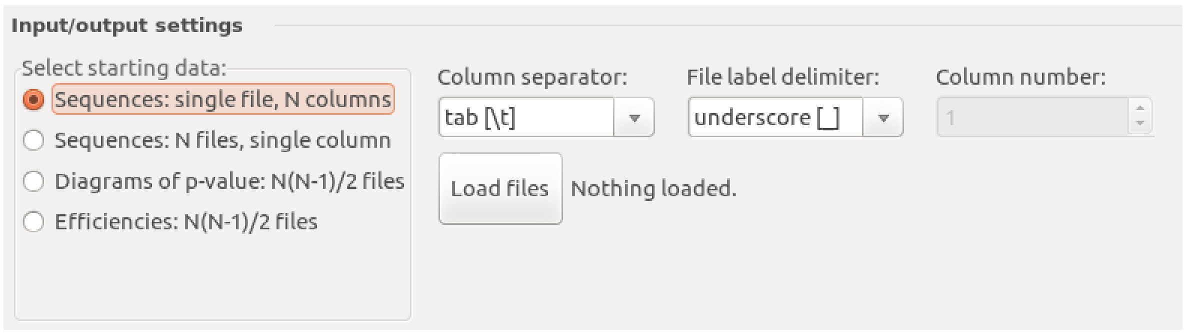
GUI App: `netOnZeroDXC_analysis`

The app is fed by a set of time series, each corresponding to a “candidate node”, and delivers in output the matrix of time scales at which correlations, if any, between nodes are observed. It is also possible to start from any intermediate stage of the analysis by loading the corresponding data. Similarly, the app can be run up to any intermediate stage of the pipeline.

Loading data

Starting data can be either raw sequences, diagrams of p value, or efficiency functions. The app only accepts ASCII files. The correct column separator has to be selected; a wrong assignment of the column separator might produce meaningless results.

The choice of the column separator, the label delimiter (if necessary) and the column number (if necessary) has to be made before the loading operation.


Input/output settings

Select starting data:
- [✓ Sequences: single file, N columns]
- Sequences: N files, single column
- Diagrams of p-value: N(N-1)/2 files
- Efficiencies: N(N-1)/2 files

Column separator: tab [\t]

File label delimiter: underscore [_]

Column number: 1

Load files

Nothing loaded.

Panel to load data.

In the following, the different input modes are briefly described, and the requirements on the file format and naming are highlighted.

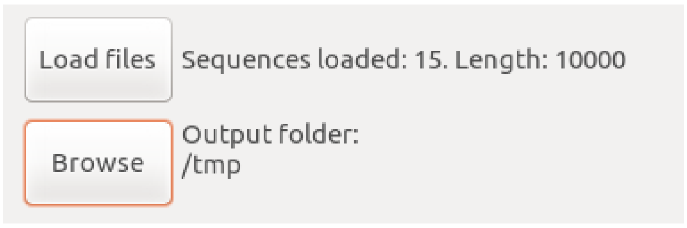
- **Sequences, single file.** After selecting a file, the program reads its content, interpreting each column as a time series associated to a different node. Columns are expected to be separated by the **separator character** selected in the corresponding list box (multiple consecutive separators are parsed as a single separator). The number N of columns determines the number of nodes. No requirements concern the filename.
- **Sequences, multiple files.** Within each file, the column corresponding to the **column number** provided in the corresponding numeric control is loaded as a sequence corresponding to a node. Columns are expected to be separated by the **separator character** selected in the corresponding list box (multiple consecutive separators are parsed as a single separator). The number N of files determines the number of nodes. Each file name is expected to end with a **delimiter** and a label (for example “name_label.dat” or “name-label.csv”) that is read and stored for the following steps.
- **Diagrams of p value.** Within each file, all rows and columns are read as a two-dimensional array of data corresponding to a single p value diagram. Columns are expected to be separated by the **separator character** selected in the corresponding list. If the analysis is carried out on a number N of nodes, there must be exactly N(N-1)/2 diagrams of p value. Each file name is expected to end with two **delimiters** and two labels (for example

"name_l1_l2.dat" or "name-0-1.csv") that are read and stored for the following steps. All pairs of labels have to appear exactly once.

- **Efficiency functions.** Within each file, the first two columns are read. Columns are expected to be separated by the **separator character** selected in the corresponding list. The first column is expected to contain the time scales (in some arbitrary units) that have to be identical for all loaded files. The second columns is expected to contain the corresponding efficiency (a real number between 0 and 1). All other columns are discarded. If the analysis is carried out on a number N of nodes, there has to be exactly N(N-1)/2 efficiency functions. Each filename is expected to end with two **delimiters** and two labels (for example "name_l1_l2.dat" or "name-0-1.csv") that are read and stored for the following steps. All pairs of labels has to appear exactly once.

By default, the app stores the results in the same directory as the loaded data. However, the output directory can be set by means of the "Browse" button.

The "Clear" button on the bottom right deletes all loaded data and resets the pipeline.



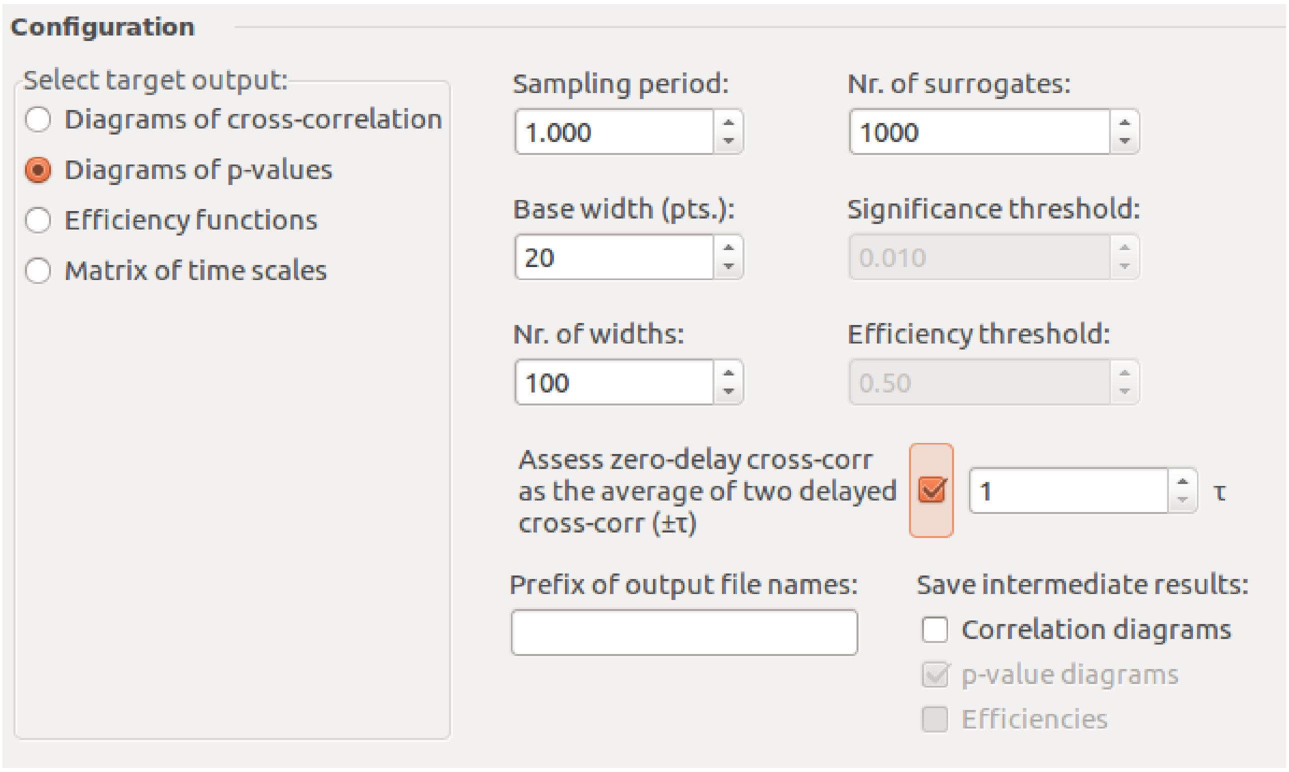
Sequences loaded: 15. Length: 10000

Output folder:
/tmp

Appearance of the load panel upon a successful load operation. The "Browse" button allows to change the output directory.

Running the analysis

The configuration panel allows to set the parameters of the analysis.



Configuration

Select target output:
- ○ Diagrams of cross-correlation
- ● Diagrams of p-values
- ○ Efficiency functions
- ○ Matrix of time scales

Sampling period: 1.000

Nr. of surrogates: 1000

Base width (pts.): 20

Significance threshold: 0.010

Nr. of widths: 100

Efficiency threshold: 0.50

Assess zero-delay cross-corr as the average of two delayed cross-corr (±τ) ✓ 1 τ

Prefix of output file names:

Save intermediate results:
- ☐ Correlation diagrams
- ✓ p-value diagrams
- ☐ Efficiencies

Panel to set up the analysis. On the left, the target quantity to compute is chosen via the radio buttons. The parameters of the analysis can be trimmed with the corresponding control boxes. Boxes that are not required for the selected target output are disabled. The settings for the output files are available below.

The meaning of the parameters is described in the **main reference**.

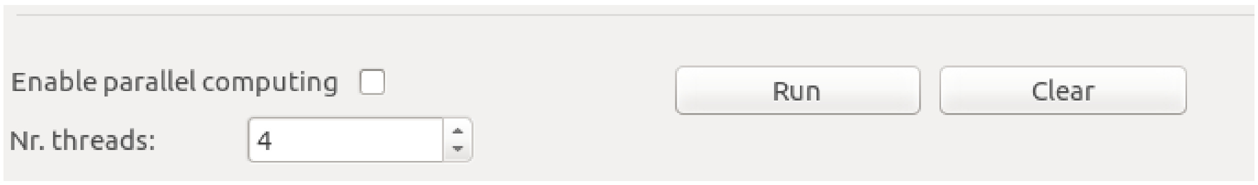
It is possible to save the intermediate results of the analysis by checking the corresponding boxes. The output files are written according to the same format described for data loading. If node labels are not available, integer numbers are used instead.

The direct assessment of zero-delay cross-correlation can be replaced by the average of two cross-correlations in which each of the two time series is alternatively delayed by τ. This option is provided to overcome situations in which noise or other phenomena would cause phony correlations (e.g. *source leakage* in MEG).

Finally, a “prefix” to be appended at the beginning of each filename can be provided.

The names of the output files thus have the formats shown in the following table, where <D> is the selected delimiter (default “_”), and <L1>, <L2> are the labels.

Output data	***File name format***	***Examples***
Correlation diagrams	`<prefix><D>`**`cdiag`**`<D><L1><D><L2>`**`.dat`**	*"foo-cdiag-1-2.dat"* *"cdiag_nodeA_nodeB.dat"*
Diagrams of p value	`<prefix><D>`**`pdiag`**`<D><L1><D><L2>`**`.dat`**	*"foo-pdiag-1-2.dat"* *"pdiag_nodeA_nodeB.dat"*
Efficiency functions	`<prefix><D>`**`eff`**`<D><L1><D><L2>`**`.dat`**	*"foo-eff-1-2.dat"* *"eff_nodeA_nodeB.dat"*
Matrix of time scales	`<prefix><D>`**`matrix.dat`**	*"foo-matrix.dat"* *"matrix.dat"*



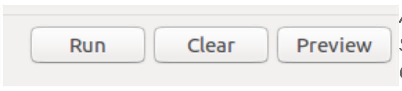
Panel to run the analysis. It is possible to select the number of parallel threads in the case of parallel computing. The "Clear" button deletes all loaded and computed data from memory.

The most time-consuming step is the evaluation of p value diagrams due to the generation of a large number of surrogate sequences. Computational time can be reduced by exploiting parallel computing.

The analysis can be stopped at any time. Stopping the analysis when the "Enable parallel computing" box is checked might take some time because all threads need to reach a synchronization barrier in order to be aware of the stop.

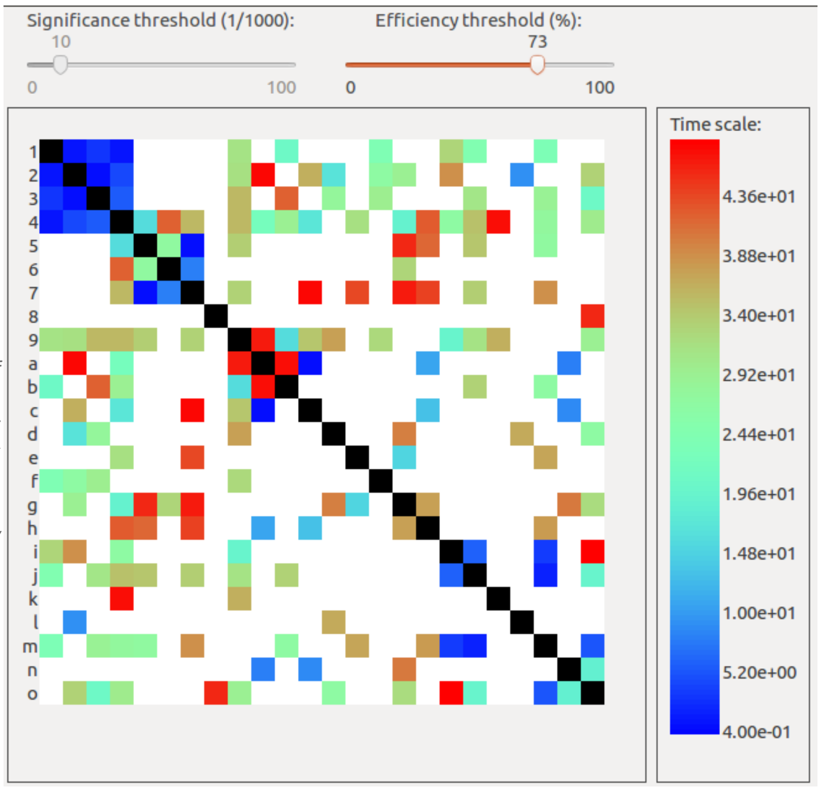
Preview of the results

If a matrix of time scale is available to the app, a preview of the result can be shown.



At the end of the computation, if the target output was set to "Matrix of time scales", a preview of the latter quantity becomes available. The "Preview" button opens a new window to graphically display the matrix.

The preview window shows the matrix of time scales by using a color scale. White pixels correspond to nodes that do not correlate at any time scale. Each matrix row is labeled with the labels associated to the nodes, sorted in alphabetic order. Two sliders are available to adjust the efficiency threshold and, if possible, the significance threshold for p values (the latter is available only if the input consists of p value diagrams or raw sequences).



The preview button opens a window in which the matrix is displayed as a color scale image.

- If p value diagrams are available, the significance threshold can be trimmed by means of the corresponding slider.
- The efficiency threshold can be trimmed by means of the corresponding slider.

Examples

In the "examples" folder, example 1 shows how to carry out a complete analysis by means of the **`netOnZeroDXC_analysis`** app. The input data is a set of time series.

Example 2 shows how the **`netOnZeroDXC_analysis`** app can be used when a set of p value diagrams is available.

For both examples, refer to the README file in the corresponding folder.

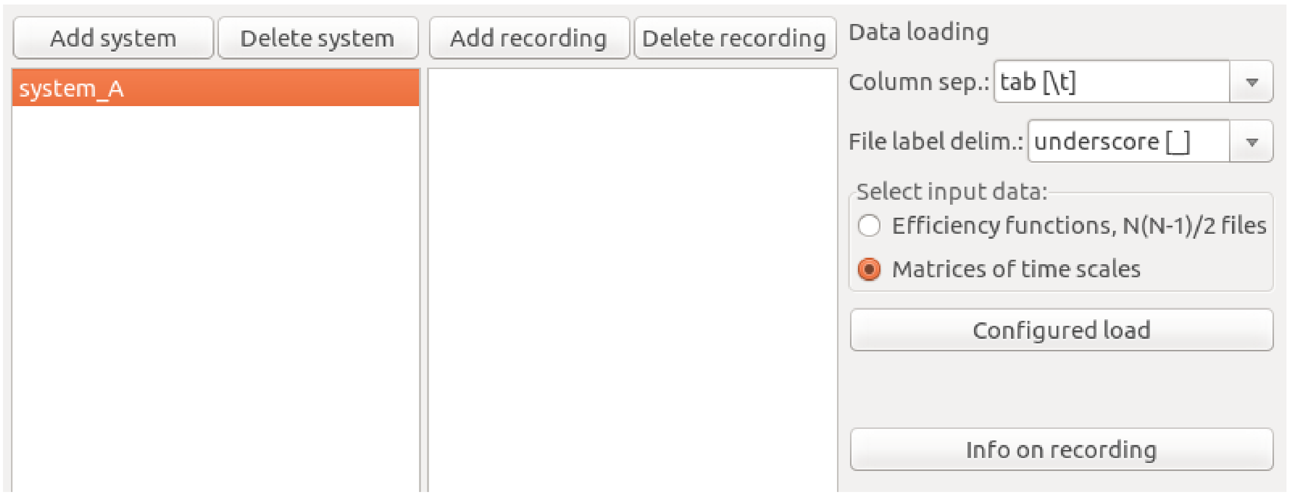
GUI App: `netOnZeroDXC_merge`

This app allows to merge results from different recordings and/or different “systems”.

This app is fed with a set of matrices of time scales, corresponding to different recordings or, alternatively, with a set of efficiency functions. The app produces a single matrix corresponding to the merged information.

Loading data

Systems have to be labeled with unique names. In order to add data, at least one system has to be present.



Column sep.: tab [\t]

File label delim.: underscore [_]

Select input data:

- Efficiency functions, N(N-1)/2 files
- [✓ Matrices of time scales]

Panel to manage recordings and systems.

Input data can be either matrices or efficiency functions. The app only accepts ASCII files. The correct column separator has to be selected. A wrong assignment of the column separator might produce meaningless results.

Data loading is carried out by means of the “Add recording” button.

- If the “Efficiency functions” box is checked, the user has to select a set of files, each corresponding to an efficiency function. Files loaded together are associated to the same recording.
- If the “Matrices of time scales” box is checked, the user can select multiple matrix files, each corresponding to a recording.

Alternatively, a **“Configured load”** mode is available. This mode allows to load data by feeding a configuration file to the app. The configuration file, in which each row corresponds to a recording, has to comply with the following requirements:

- each row of the configuration file has to contain four strings, separated by a valid column separator. Characters after the fourth separator are ignored. Multiple consecutive separators are parsed as a single separator. The four strings have to be in the following order:
  `[system name] [recording name] [mode label] [filename]`
- recordings associated to the same system must be listed consecutively;
- the **mode label** has to be a single character, namely ‘m’, for “matrix mode”, or ‘e’, for “efficiency mode”. The same mode character has to appear in all lines;
- if mode is ‘m’, the fourth string is interpreted as the **filename** of a matrix file. If mode is ‘e’, the fourth string is interpreted as a file, which has to contain a list of names of efficiency files. Both for the ‘m’ and ‘e’ mode, as well as in the file list, filenames have to be expressed as paths relative to the location of the configuration file. For example, if the configuration file is in `/home/analysis/`, and the matrix files are in `/home/analysis/results/`, the configuration file must display filenames as `results/matrix.dat`

Node labels are determined by parsing the efficiency filenames or, in the case of input matrices, by assigning integer numbers to the matrices rows. In the latter case, it is possible to override the default assignment by loading a list of node labels. The node labels file has to contain pairs of `[row index] [label], where` row indexes has to start from 0 or 1, and have to be consecutive integer numbers.

Data validation and previews

The dataset has to be validated before merging the loaded data. The validation process checks the following requirements:

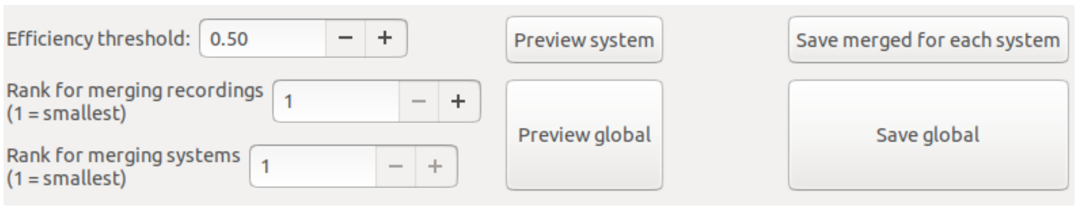
- all systems have the same number of recordings;
- all recordings have the same kind of data loaded (either matrices or efficiencies);
- in all recordings, the same number of nodes is present

The app allows to graphically preview the merged matrix of time scales. The three relevant parameters for evaluating the merged matrix can be trimmed by means of three sliders.

Saving merged results

It is possible to save either a merged matrix for each system, or a single merged matrix for the whole dataset. In the former case, a number of files equal to the number of systems is saved in a selected folder. A “prefix” to be appended at the beginning of each filename can be specified. Filenames are built as `<prefix><D><system name>`**`.dat`** where <D> is the selected delimiter.

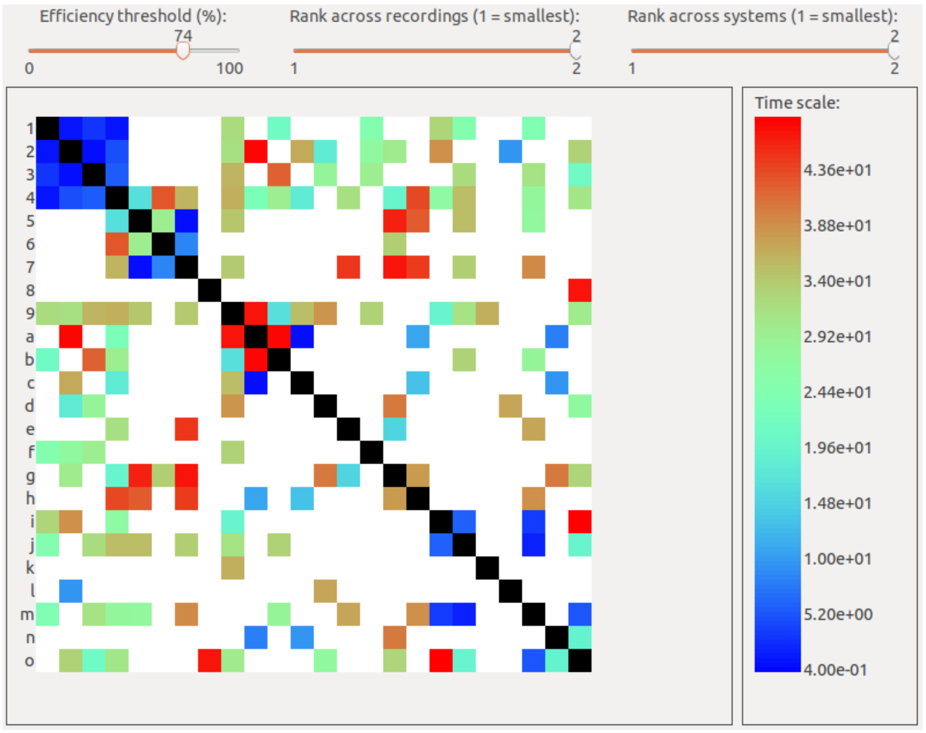
A log file, which contains metadata on the merging process, is also saved in the same directory.


Efficiency threshold: 0.50

Rank for merging recordings (1 = smallest) 1

Rank for merging systems (1 = smallest) 1

Preview and save controls. Control boxes for setting the efficiency threshold and the ranks are automatically updated when the corresponding sliders are moved in the preview window.



Examples

In the “examples” folder, example 3 shows how to use the **`netOnZeroDXC_merge`** app to merge the information contained in efficiency functions for two systems, each containing two recordings. Two configuration files, which are used to load either efficiency functions or matrices, are provided in the example folder.

Example 4 shows how to merge the results for a single system with six recordings. For each recording, a matrix is available. Example 4 also shows how to load node labels.

For both examples, refer to the README file in the corresponding folder.

Command Line Program: `netOnZeroDXC_diagram`

This command line program carries out the first step of the analysis pipeline, namely the computation of cross-correlation diagrams and p value diagrams out of two input time series.

Usage

Data can be either piped into the program through standard input or loaded from a file. Similarly, the output diagram is by default written on standard output or, if specified, to a file.

Options are given to the program by means of flags followed, if necessary, by suitable additional strings.

Options list:

Flag	***Behavior***
`-C`	Compute correlation diagram only
`-p`	Compute p value diagram (if both -C and -p are given, -C is ignored)
`-i <fname>`	Load input data from file “fname”, ignoring standard input. “fname” has to be a valid file name without spaces
`-o <fname>`	Write output data to file “fname”, neglecting standard output. “fname” has to be a valid file name without spaces
`-s <@>`	Set column separator by choosing either t (tab), s (space) or c (comma). Other labels are ignored. Tab is used as the default.
`-n <#> <#>`	**Mandatory**. Set column numbers (≥ 1) to be used as input sequences
`-W <#>`	**Mandatory**. Set the number of window widths
`-L <#>`	**Mandatory**. Set base width of window, in samples
`-M <#>`	Set the number of surrogates to generate (default = 100)
`-tau <#>`	Apply the delay of +/-tau points to assess zero-delay cross-correlation as the average of two delayed cross-correlations
`-parallel`	Enable parallel computing
`-h` or `--help`	Print this list of options and exit

If neither `-C` or `-p` are given, `-p` is the default option. The meaning of the different parameters is described in the **main reference**.

Command Line Program: `netOnZeroDXC_efficiency`

This command line program performs the second step of the analysis pipeline, namely the computation of an efficiency function out of an input p value diagram.

Usage

Data can be either piped into the program through standard input or loaded from a file. Similarly, the output diagram is by default written on standard output or, if specified, to a file.

Options are given to the program by means of flags followed, if necessary, by suitable additional strings.

Options list:

Flag	***Behavior***
`-i <fname>`	Load input data from file “fname”, ignoring standard input. “fname” has to be a valid file name without spaces
`-o <fname>`	Write output data to file “fname”, neglecting standard output. “fname” has to be a valid file name without spaces
`-a <%f>`	**Mandatory**. Set the significance p value threshold below which the null hypothesis is rejected, i.e. correlation is deemed to be significant. “%f” has to be a floating point number (e.g. 0.01 for α = 1%).
`-w <%f>`	Set the base window width. Default is 1. “%f” has to be a floating point number.
`-h` or `--help`	Print this list of options and exit