##

Instructions for the C3D code, Release 1

##

Author: Xiaolin Xiao (xiaolin.xiao@csc.mrc.ac.uk)

Date: October 2013

Website:

http://www.csc.mrc.ac.uk/Research/Groups/IB/IntegrativeGenomicsMedicine

The C3D source code used in the paper "*Multi-tissue analysis of co-expression networks by Higher-Order Generalized Singular Value Decomposition identifies functionally coherent transcriptional modules*" by Xiao et al. is designed to identify both common and differential cluster patterns across multiple general matrices. C3D works on two or more matrices. The input matrices can represent different kind of genomic data (gene expression data, co-expression data, protein data, etc.) or other complex network data (e.g., information network data, social network data, etc.). There is no limit in the size of the input matrices. For additional details on the C3D method please refer to Xiao et al. PLoS Genetics 2013.

##

0. Copyright notices

##

##

1. Contents of the C3DmatR1.zip file

##

The source code that implements the C3D algorithm is written in Matlab and requires both Matlab (version r2011 or higher) and R (version 2.15.0 or higher) to be installed.
The following R package needs to be installed:
fdrtool (version 1.2 or higher)
[http://cran.r-project.org/web/packages/fdrtool/index.html]

The Matlab code has been tested with different Matlab distributions (r2011, r2012 and r2013) under Mac and Linux/Unix Systems.

To install – unzip the C3DmatR1.zip file and move the /C3DmatR1 folder to the Matlab folder (usually located in ~/Documents/Matlab).

Source code and example data files are compressed into "C3DmatR1.zip", which includes the following files:

C3D_r1.m
validate_trans.m
hogsvd_trans.m
e_trans.m
fast_pinv.m
fdrqval.r
Instructions for C3D.pdf
/data
/data5k

/data
This directory contains an example of input data files that are needed to run the C3D code:
(1) a gene (nodes) ID file ("geneid.txt"), containing the list of gene (node) identifiers present in the input data matrices (gene node IDs MUST be unique)

(2) a network ID file ("datasets_list.txt"), containing the list of the input data matrices (i.e., file names)
(3) 5 INPUT MATRIX FILES (e.g., "condition1.txt" to "condition5.txt"), which are input data matrices. These are simulated gene expression data in 5 conditions (500 genes x 30 observations). For additional details please refer to: Xiao et al. PloS Genetics 2013).

/data5k
This directory contains an example of input data files that are needed to run the C3D code:
(1) a gene (nodes) ID file ("geneid.txt"), containing the list of gene (node) identifiers present in the input data matrices (gene node IDs MUST be unique)
(2) a network ID file ("datasets_list.txt"), containing the list of the input data matrices (i.e., file names)
(3) 7 INPUT MATRIX FILES (e.g., "condition1.txt" to "condition7.txt"), which are input data matrices. These are simulated gene expression data in 7 conditions (5,000 genes x 30 observations). For additional details please refer to: Xiao et al. PloS Genetics 2013).

##

2. Preparing the input data files

##

All input files MUST be placed in the same directory and MUST be saved as Tab Delimited Text (.txt) files. The required input files are:

1. "datasets_list.txt"
2. "geneid.txt"
3. input matrix files ("condition1.txt", "condition2.txt", "condition3.txt", etc.)

1. **LIST OF INPUT DATA:** each input matrix file MUST be listed within the "datasets_list.txt" file. Each input matrix file name MUST be unique and it is indicated in a separate row. Example:

“condition1.txt”
“condition2.txt”
“condition3.txt”
(etc.)

2. **LIST OF IDENTIFIERS:** the complete list of identifiers (proteins, genes, etc.) MUST be provided within a file named "geneid.txt". The gene (node) IDs can be string characters or numbers. Each gene ID MUST be unique and MUST be placed individually in one row of the gene ID file.
   Example:

   Gene1
   Gene2
   Gene3
   (etc.)

3. **INPUT MATRIX FILES:** these can be named by the user (“condition1.txt”, “condition2.txt”, “condition3.txt”, etc.) and MUST be listed in the "datasets_list.txt" file.

The algorithm allows the INPUT MATRIX FILES to be formatted as (A) raw expression data sets or (B) co-expression data sets.

A. Raw expression matrices MUST have the same ROW size where each row contains the expression profile for a gene (node) (IDs are specified in the "geneid.txt" file). The rows of each data matrix file MUST be ordered according to the gene (node) IDs, as listed in the "geneid.txt" file.

B. Co-expression matrices MUST have the same ROW and COLUMN sizes. Each element of the matrix represents a co-expression measure (e.g. Pearson correlation, Kendall correlation, Mutual Information, etc.) between each pair of genes (nodes), which are listed in the "geneid.txt" file.

The C3D code automatically checks whether (A) or (B) is provided. We suggest to the test the C3D code on the examples of 5 INPUT

MATRIX FILES (simulated expression data), which are provided under the ../data directory. For details of the command usage see Part 3 (Getting started).

##

3. Getting started with the C3D code

##

C3D_r1.m is the main Matlab function used to identify common and differential clusters across multiple input matrix files.

The algorithm takes all matrix files as input and returns the clusters files, the summary files, the log file and other results which are automatically saved in the directory ../data/results/ (see details on the results files below).

The C3D algorithm can be run by typing a single command line within MATLAB:

>> C3D_r1(directory, MER, n.vectors, sim.indicator, *optional arguments*)

The C3D_r1.m Matlab function takes mandatory and optional input arguments as follows:

--MANDATORY INPUT ARGUMENTS

(directory, MER, n.vectors, sim.indicator)

directory: the directory (with complete path) where all the input data files are located;

MER: the misclassification error rate threshold used in the cluster detection (MUST be a float number in [0,1]); (a maximum MER = 0.2 is suggested)

n.vectors: the number of candidate vectors (clusters) to be extracted and tested in the permutation procedure (MUST be a positive integer number). When n.vectors > total number of observations, the total number of observations is used. Please

note that the overall computation time (accounting for the incremental permutations) increases linearly with the number of candidate vectors (clusters) that are selected to be tested by permutations.

sim.indicator: indicator to run simulations to validate clusters and calculate empirical p-values ("1" will run permutations; any other input different from "1" will skip the permutation procedure).

--OPTIONAL INPUT ARGUMENTS

(normalize, correlation, max.perms, min.perms, p.threshold, individual.cluster.quality, overall.cluster.quality, norm)

normalize: specifies how to normalize/scale input data; 'zscore' (z-score for each ROW across COLUMNS) or 'log' (log transformation);
[Default = no normalization/scaling of the input data is performed]

correlation: specifies to compute the correlations between each pair of nodes from the input data matrices, and generate new co-expression matrices that will be automatically passed as input for the C3D algorithm. The user can specify a correlation metric from: 'Pearson', 'Spearman', 'Kendall';
[Default = same inputs as provided by the user (no correlations are calculated)]

max.perms: the maximum number of incremental permutations (MUST be a positive integer >=100)
[Default = 1,000]

min.perms: the minimum number of incremental permutations (MUST be a positive integer >=100)
[Default = 100]

p.threshold: the P-value threshold used in the incremental permutation procedure to stop the permutations (MUST be a float

number in (0,1))
[Default = 0.05]

individual.cluster.quality: specify the individual cluster quality measure used by the algorithm. The user can choose an alternative cluster quality measure (i.e., 'c1'), which corresponds to c*h as detailed in Text S1. Supporting Methods of Xiao et al. Multi-tissue analysis of co-expression networks by Higher-Order Generalized Singular Value Decomposition identifies functionally coherent transcriptional modules. PloS Genetics 2013.
[Default = 'c2']

overall.cluster.quality: specify the overall cluster quality measure used by the algorithm. Other cluster quality measures can be specified by the user ('q1', 'q2', 'q3', 'q4', 'q6', 'q7'). Please refer to Text S1. Supporting Methods of Xiao et al. Multi-tissue analysis of co-expression networks by Higher-Order Generalized Singular Value Decomposition identifies functionally coherent transcriptional modules. PLoS Genetics, for additional details.
[Default = 'q5' (i.e., "product based" cluster quality measure)]

norm: specify to use the absolute values (norm) for each value in the input data matrices: 'abs';
[Default = *original values are used*].

If no values are specified for the optional arguments, the C3D program will automatically use the assigned default values.

The details of the Inputs arguments can be also found by typing "help C3D" in the MATLAB environment.

Please note that the C3D program will check the validity of all the input arguments. When you type in an illegal value for any of the input arguments, the C3D program will be automatically stopped and will return a specific error.

##

An example of C3D analysis of seven input data matrices

##

This example illustrates how to use the C3D program to identify common and differential clusters from the example data sets provided in the ../data sub folder.

There are 5 data set files in the ../data folder: “condition1.txt” – “condition5.txt”, which represent simulated gene expression data in 5 conditions, where each data matrix consists of 500 genes x 30 observations.

Step-by-step use of the C3D package:

1. Download the package C3DmatR1.zip and uncompress the zip file. This will create a folder as detailed in (1. Contents).
2. Start MATLAB and change directory (cd) to the C3DmatR1 folder. (To change directory type the cd() command in MATLAB environment or double click the corresponding folder target in the Current folder window of MATLAB environment.)
3. Then change directory (cd) to the example data folder (../C3DmatR1/data) and obtain the directory(file_path) for the input data sets as follows: within the MATLAB environment type the following command lines

   ```
   clear;
   cd('data');
   pwd;
   file_path=pwd;
   ```

4. Then change directory (cd) back to the source code directory and run C3D as follows:

   ```
   cd ..
   C3D_r1(file_path,0.05,5,1,'abs');
   ```

This command will run the C3D program and save the results into the directory data/results (automatically generated).

All results will be always saved in the data/results folder and overwritten on previously generated results. If you want to re-run the

code with different data files and/or input parameters, we suggest to move or rename the old results/ directory. (Matlab generates a warning message when the results directory already exists).

##

4. Details on the results files

##

All the results files are saved in the directory

/data/results/

This directory is automatically generated by running "C3D_r1.m'.

../data/results/

clusters_summary_*.txt: tab separated txt file including the summary results of the identified clusters as follows:

Column header	Content and information
cluster	cluster name
overall p	overall significance of the cluster in multiple conditions
conditions	conditions where the cluster is detected
overall randomization	number of permutations used for computing overall p
overall cluster quality	cluster quality measure calculated in the conditions where the cluster is detected
individual p in conditionx	significance of the cluster in conditionx
individual randomization in conditionx	number of permutations used for computing individual p in conditionx
individual cluster quality in conditionx	cluster quality measure calculated in condition x
cluster size	number of nodes within each cluster

run_c3d_main_at*.txt: log file

../data/results/fdrcut/**

Number_all_clusters_overlap.txt: tab separated txt file reporting the number of overlapping nodes between each pair of clusters, which is specified in the off-diagonal elements of a symmetric matrix.

Nodes_all_clusters_overlap.txt: tab separated txt file detailing the gene (nodes) ID of the nodes that overlap between different clusters. The diagonal entries "all" denote the cluster overlap with itself.

../data/results/fdrcut/clusters/**

For each cluster reported in “clusters_summary_*.txt” file (first column, cluster), we report:

The gene (nodes) ID [first column]

The misclassification error rate (MER) [second column]

The node position in the input data matrix [third column]

../data/results/fdr*cut/edgelist/

For each cluster reported in “clusters_summary_*.txt” file (first column, cluster) we report the list of edges connecting any node pair and the strength of association (default = covariance between node pairs).

Column 1 and Column 2 denote the nodes (genes) pairs IDs

Column 3 to the last column: strength of association in each condition.