CODENSE v1.0

INTRODUCTION

Given a relation graph dataset, $D=\{G_1,G_2,\ldots G_n\}$, where $G_i=(V,E_i)$,

Definition 1 (Support) The support of a graph g is the number of graphs (in D) where g is a subgraph, written support(g). A graph is frequent if its support is greater than a minimum support threshold.

Definition 2 (Summary Graph) The summary graph of D is an unweighted graph $G=(V, \hat{E})$ where an edge $e(u,v)$ connects vertices u and v ($u, v \hat{I} V$) if $\sum_i |e_i(u,v)| \geq k$, where k is a user-defined support threshold

Definition 3 (Edge Support Vector) The support vector of an edge e, written $w(e)$, is of length n where n is the number of graphs. The i-th element of $w(e)$ corresponds to the weight of edge e in the i-th graph.

Definition 4 (Second-Order Graph) The second-order graph is an unweighted graph $S=(V´V, E_S)$ where the vertex set of S is the edge set of G, and an edge connects vertices u and v if the similarity between the corresponding edge support vectors $w(u)$ and $w(v)$ is greater than a threshold.

In reality, if G_i is large and dense, S will be impractically large. Therefore, to achieve efficiency, in this paper we construct S each time only for a subgraph of the summary graph G. In contrast to the second-order graph, we term the original graphs G_i the first-order graphs.

Definition 5 (Coherent Graph) A subgraph $sub(G)$ is coherent if all the edges of $sub(G)$ have support higher than k and the second-order graph of $sub(G)$ is dense.

CODENSE is short for Mining Coherent Dense Subgraphs. By simplifying the problem of identifying coherent dense subgraphs across n graphs into a problem of identifying dense subgraphs in two special graphs: the summary graph and the second-order graph, CODENSE can efficiently mine frequent coherent dense subgraphs across large numbers of massive graphs. CODENSE is scalable in the number and size of the input graphs, flexible in mining either weighted or unweighted graphs, and adjustable in terms of exact or approximate pattern mining. The algorithm behind it is described in the related paper (see REFERENCES).

PLATFORMS

CONDENSE has been developed and tested on Linux(Debian and Redhat) using gcc2.95, and should be able to run on most UNIX systems.

USAGE

coden [command-line options] <input-files>

COMMAND-LINE OPTIONS

-m k(run_mode)

There are 3 running modes available for CODENSE. Valid k values are:
(1) k=0, is to find all clusters
(2) k=1 is to find all clusters containing gene x

(3) k=2 is to find all clusters for supervised clustering. Here by supervised clustering, CODENSE will mine the coherent subgraphs from positive graphs under the constraint that these subgraphs are not coherent in the negative graphs.

-i str(inputfile)

The path and name of the input overlapped frequency graph file, which is in the matrix format or edge format.

-n k(gene_num)

This parameter specifies the gene number from the input file when the input file is in the matrix format. In other words, this argument is equal to the dimension of the input matrix file.

-v str(support vector file)

The path and name of the support vector file.

-p k(supprot vector number in the support vector file)

This parameter specifies the number of support vectors in the support vector file.

-l k(sv_length)

This parameter specifies the size of one support vector. It is also the number of datasets(networks) used for generating the input overlapped frequency graph.

-t str(ttablefile_name)

This parameter specifies the t table used to assess the p-value of the second order correlation.

-o str(outputfile_name_prefix)

This is the prefix of the output clusters file. Thus the output file containing the first order clusters would be outputfile_name_prefixFO, while the final output file containing the second order clusters would be outputfile_name_prefixSO.

-g k(min_graph_size)

This parameter specifies the minimum node number requirement of the output subgraph. The range of the minimum size of frequent dense subgraph generated is from zero to infinity. In practice, this number varies depending on the study object. In our study, because we want to only output clusters which are homogenous, and we assume that clusters containing hundreds or thousands genes has small chance to be homogenous, while clusters containing one or two genes could not be named homogenous, we set this parameter as 4~6. We also set the default as 5.

-e k(bottom_edge_freq)

This argument specifies the minimum edge frequency required to be kept as an edge in the summary graph, also called user-defined support threshold for an edge in the summary graph. This frequency threshold is used for generating the summary graph and is related to the number of the datasets you are using to generate the multiple networks. Because we are using 39 datasets, we set the default as 6. We recommend setting this parameter at least as 10% of the number of datasets used. Default value is 6.

-d f(density_cutoff_order1)

This argument specifies the minimum density requirement for the first order dense subgraph generated during the posterior dense clustering after project back to the summary graph from the second order clusters. Because the density is calculated from the ratio between the edge number and the possible edge number, it is ranging from 0 to 1. 1 represents the complete graph like cliques. Usually, the suggested setting range for the first order graph density threshold is 0.3~0.6, Default value is 0.4.

-q f(density_cutoff_order2)

This argument specifies the minimum density requirement for the second order dense subgraph generated. Because the density is calculated from the ratio between the edge number and the possible edge number, it is ranging from 0 to 1. 1 represents the complete graph like cliques. Usually because the second order graph is very big relative to the first order graph, we recommend setting the density requirement for the second order graph as lower than that of the first order graph. Therefore, and the second order graph density threshold as 0.2~0.5 is recommended here. Default value is 0.4.

-b f(density_cutoff_order3)

This argument specifies the minimum density requirement for the first order dense subgraph generated during the initial dense clustering on the summary graph. Because the density is calculated from the ratio between the edge number and the possible edge number, it is ranging from 0 to 1. 1 represents the complete graph like cliques. Usually, the suggested setting range for the first order graph density threshold is 0.1~0.5, Default value is 0.4.

-s k(the maximum node number of a first order subgraph)

This paprameter specifies the maximum number of nodes in a graph when performing min-cut algorithm instead of normal-cut algorithm. Default value is 80.

-c f(connect perc restoring the condensed cluster)

This argument controls the connectivity percentage requirement for keeping a node when restore a subgraph from a condensed cluster node. Default value is 0.6.

-x k(genex)

This argument specifies the gene (index), the clusters containing which is to be discovered when running modes with run_mode as 0.

–y k(edge_num)

The total edge number when input summary graph prototype is in edge format

-k str(positive datasets)

This argument specifies the positive graphs directly, while indirectly points out the rest graphs are negative when running mode as 2.

-u k(max degrees of freedom for T table)

The maximum degrees of freedom for T table

INPUT-FILES

Before users are preparing the following input files, users should have a defined gene order (gene index). Suppose users graph cover m genes, this gene index should be rang from 0 to m-1. In all the description of this manual, these defined gene indices are actually used to represent genes.

(1) Support vector file

The support vector of an edge is of length sv_length where sv_length is the number of graphs. At the command line, user needs to specify the total number of support vectors in this input support vector file by –p sv_num, and the size of each support vector by –l sv_length.

The format of this support vector file is gene_i_index | gene_j_index | short((correlation in dataset 1)*1000) | short((correlation in dataset 2)*1000) | ... | short((correlation in dataset n)*1000) .In this version, gene_i_index must be specified as larger than gene_j_index. The example of this file is in ~/CONDENSE/data/input/sv_38_f500.txt. In the file sv_38_f500..txt, each support vector has 40 elements, the first two are gene index, the rest are corresponding to the correlation of these two genes in 38 datasets (networks). The first support vector in this file is for gene 1 and gene 0, and because the correlation value has been amplified by 1000 times, it looks like in reality:

1 0 859 -80 406 840 101 448 1100 713 -444 -542 -277 1 100
-516 280 328 436 -327 -116 -64 -454 168 588 651 630 - 74
614 481 358 27 -227 172 457 581 267 -401 0 473 - 91

(2) T table file

The T table file is corresponding to one p value. The format is: degrees of freedom | critical value
For example, corresponding to p value as 0.01:

_____

1	31.82052
2	6.96456
3	4.54070
4	3.74695
5	3.36493
6	3.14267
7	2.99795
8	2.89646
9	2.82144

….

_____

Therefore, when there are 6 samples in one dataset, the 0.01 critical value from the t-distribution with 6 degrees of freedom is 3.14267.

There are two example files to illustrate the T table.
(a) For p value as 0.01 (~/CONDENSE/data/input/ttableFromMatlabt1p-2.txt)
(b) For p value as 0.001 (~/CONDENSE /data/input/ttableFromMatlabt1p-3.txt).
If you are only going to use the one tail t distribution corresponding to p value as 0.01 and 0.001, you can use the one in the ~/CONDENSE/data/input/ttableFromMatlabt1p-2.txt and ~/CONDENSE/data/input/ttableFromMatlabt1p-3.txt respectively). When these two example files are used, the maximum experiments number is 500.
If you are going to use your own t table file, please specify the maximum degrees of freedom for the option -u in the command line. Otherwise, the user should reset the T_TABLE_LENGTH variable.

(3) Overlapped frequency graph file

The summary graph prototype before using the minimum support threshold cut.

There are two formats. One is matrix format, the other is edge format.
(a) Matrix format
The summary graph prototype is a real symmetric matrix with dimension as gene number × gene number. The intersection of ith gene row and jth gene column is the number of datasets in which this gene pair significant correlated in terms of Jackknife correlation. Or other interested relation frequency defined by user.

If your input summary graph prototype is in the matrix format, you need to specify –n gene number in the command line. The example of this file is in ~/CONDENSE/data/input/summaryG500.txt.

(b) Edge format

The summary graph prototype is a set of weighted edges. The format is:
Node I1 | Node J1 | Weight
Node I2 | Node J2 | Weight
Node I3 | Node J3 | Weight
….
The weight for one edge is the number of datasets in which this gene pair significant correlated in terms of Jackknife correlation. Or other interested relation frequency defined by user.
If your input summary graph prototype is in the matrix format, you need to specify –y edge number in the command line.

OUTPUT-FILES

(1) First order clustering results are in file outputfile_name_prefixFO.
The format is:
Cluster index | node number n in this cluster | edge number m in this cluster | gene 1's index | gene 2's index | ...| gene n's index.

(2) Second order clustering results are in outputfile_name_prefixSO.
The format is:
Cluster index | node number n in this cluster | edge number m in this cluster | gene 1's index | gene 2's index | ...| gene n's index.

(3) Log file outputfile_name_prefix_log
The format is:
1st order cluster num=x
first order cluster index i | number of second order clusters from first order cluster i | number of coherent clusters generated from first order cluster i

EXAMPLES

coden -m run_mode -i myinputfile -n gene number -v svfile -p sv_num -l sv_length -t ttablefile_name -o outputfile prefix -g min_graph_size -e bottom_edge_freq -d density_cutoff_order1 -q density_cut_off_order2 -s the maximum node number of a first order subgraph –c minimum percentage of nodes outside the belonging cluster a node is connected to if this node is to be kept during restoring the condensed cluster when performing MODES -x genex –y the edge number when input summary graph prototype is in edge format –k positive dataset1/dataset2/dataset3/.../datasetm –u maximum degrees of freedom for T table

The initial try could be the following command:

./coden -m 0 -i ../data/input/summaryG500.txt -n 500 -v ../data/input/sv_38_f500.txt -p 13174 -l 38 -t ../data/input/ttableFromMatlabt1p-3.txt -o ../data/output/y38p3g5e4d1q3s80c5try_500 -g 5 -e 4 -d 0.1 -q 0.3 –b 0.4 -s 80 -c 0.5
This generates three files in the ../data/output/ directory: y38p3g5e4d1q3s80c5try_500FO, y38p3g5e4d1q3s80c5try_500SO, and y38p3g5e4d1q3s80c5try_500_log

Example for running mode at 0:

coden –m 0 –i myinputfiledir/myinputfilename -n 500 –v ../data/input/sv_38_f500.txt –p 13174 –l 38 –t ../data/input/ttableFromMatlabt1p-3 –o myoutputfileprefix –g 5 –e 6 –d 0.4 –q 0.5 –s 80 -c 0.5 –u 500

This will set the minimum output graph size as 5, the edge support threshold as >=6, the first order dense subgraph cut off as 0.4, the second order dense subgraph cut off as 0.5, the maximum node number of a first order subgraph is 80, and at least 50% of nodes outside the belonging cluster a node should be connected to if this node is to be kept during restoring the condensed cluster when performing MODES. This will generate the first order subgraph file as ~/CONDENSE/data/output/myoutputfileprefixFO, the coherent subgraph file as ~/CONDENSE/data/output/myoutputfileprefixSO.

Example for running mode at 1:

coden –m 1 –i myinputfiledir/myinputfilename -n 500 –v ../data/input/sv_38_f500.txt –p 13174 –l 38 –t ../data/input/ttableFromMatlabt1p-3 –o myoutputfileprefix –g 5 –e 6 –d 0.4 –q 0.5 –s 80 -c 0.5 –x 21

This will set the minimum output graph size as 5, the edge support threshold as >=6, the first order dense subgraph cut off as 0.4, the second order dense subgraph cut off as 0.5, the maximum node number of a first order subgraph is 80, and at least 50% of nodes outside the belonging cluster a node should be connected to if this node is to be kept during restoring the condensed cluster when performing MODES. This will generate the first order subgraph file containing gene 21 as ~/CONDENSE/data/output/myoutputfileprefixFO, the coherent subgraph file containing gene 21 as ~/CONDENSE/data/output/myoutputfileprefixSO.

Example for running mode at 2:

coden –m 2 –i myinputfiledir/myinputfilename -n 500 –v ../data/input/sv_38_f500.txt –p 13174 –l 38 –t ../data/input/ttableFromMatlabt1p-3 –o myoutputfileprefix –g 5 –e 6 –d 0.4 –q 0.5 –s 80 –c 0.5 –k 2/3/4/5/6/7/8/9/11

This will set the minimum output graph size as 5, the edge support threshold as >=6, the first order dense subgraph cut off as 0.4, the second order dense subgraph cut off as 0.5, the maximum node number of a first order subgraph is 80, and at least 50% of nodes outside the belonging cluster a node should be connected to if this node is to be kept

during restoring the condensed cluster when performing MODES. This will generate the first order subgraph file as ~/CONDENSE/data/output/myoutputfileprefixFO, the coherent subgraph as ~/CONDENSE/data/output/myoutputfileprefixSO. These subgraphs will appear frequently in the datasets 2, 3, 4, 5, 6, 7, 8, 9, 11, and not recurrent pattern of the rest datasets out of the total 38 datasets.

NOTE

This is the CODENSE version 1.0. Testing hasn't been exhaustive. Feedback and application description are always welcome. Contact hhu@usc.edu for bugs and questions about CODENSE.

REFERENCES

Mining coherent dense subgraphs across mas-sive biological networks for functional discovery
Haiyan Hu[1], Xifeng Yan[2], Yu Huang[1], Jiawei Han[2], and
Xianghong Jasmine Zhou[1]
1Program in Molecular and Computational Biology, University of Southern California, Los Angeles, CA 90089, USA and 2Department of Computer Science, University of Illinois at Urbana-Champaign, Ur-bana, IL 61801

CONTACTS

Xianghong Jasmine Zhou
Assistant Professor
Program in Molecular and Computational Biology
University of Southern California
Office: DRB291 Phone: 213-740-7055 Fax: 213-740-2437
Email: xjzhou@usc.edu