
MODES v1.1
User Manual

References

Mining Coherent Dense Subgraphs Across Massive Biological Networks for Functional Discovery

Haiyan Hu[1], Xifeng Yan[2], Yu Huang[1], Jiawei Han[2], and Xianghong Jasmine Zhou[1]
[1] Program in Molecular and Computational Biology, University of Southern California, Los Angeles, CA 90089, USA
[2] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801

Introduction

MODES stands for Mining Overlapping DENSE Subgraphs. The input graph for MODES is an unweighted graph $\hat{G}=(V, \hat{E})$ where an edge $e(u,v)$ connects vertices u and v $(u, v \in V)$. MODES is developed based on HCS (Mining Highly Connected Subgraphs) (Hartuv & Shamir, 2000), with two new features: (1) MODES is efficient in identifying dense subgraphs; and more importantly, (2) MODES can discover overlapping subgraphs. The algorithm behind it is described in the related paper (see REFERENCES).

Platforms

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

Usage

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

Command-Line Options

-m k(run_mode)

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

-i str(inputfile)

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

-n k(gene_num)

This parameter specifies the gene number from the inputfile, i.e. the dimension of the input matrix file.

-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. Default value is 5.

-e k(bottom_edge_freq)

This argument specifies the minimum edge weight required to be kept as an edge in the input graph. Default value is 6.

-d f(density_cutoff_order1)

This argument specifies the minimum density requirement for the dense subgraph generated. Default value is 0.5.

-s k(the maximum node number to apply min-cut)

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.

Note: The maximum gene num MODESv1.1 can handle is 65535.

Input-Files

The input graph could be in three formats: matrix format, edge format, and another is edge list format.

Note: In the examples below, the symbol “|” represent a Tab separator, and “|_|” represents a space separator.

(a) Matrix format

The input graph prototype is an integer symmetric matrix with dimension as genenumber × 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 ~/MODES/data/input/summaryG500.txt.

(b) Edge format

The input graph is a set of weighted edges. The format is:

Node I1 | Node J1 | Weight
Node I2 | Node J2 | Weight
Node I3 | Node J3 | Weight
….

Since MODESv1.0 is applied on unweighted graph, the weight value is not really used in MODES. Or other interested relation frequency defined by user. If your input graph is in the edge format, you need to specify –y edge number in the command line.

(c) Edge List format

The input graph is a set of edges. The format is:

Node I1 |_| Node J1
Node I2 |_| Node J2
Node I3 |_| Node J3
….

If your input graph is in the edge list format, you need to specify –y edge number in the command line.

Output-Files

The clustering results are in the output file user specified.
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.

EXAMPLES

```
modes -m run_mode -i myinputfile -n genenum -o outputfile -g min_graph_size -e bottom_edge_freq -d density_cutoff_order1 -s the maximum node number of a first order subgraph -c connect perc restoring the condensedcluster -x genex
```

The initial try could be the following command:

```
./modes -m 0 i ../data/input/g1.matrix -n 10 -o ../data/output/g1.matrix.out4 -g 4 -e 1 -d 0.9
```

Example for running mode at 0:

```
modes –m 0 –i myinputfile –n genenum –o myoutputfile –g 5 –e 6 –d 4 –s 80 –c 0.6
```

This will set the minimum output graph size as 5, the edge support threshold as >=6, the dense subgraph cut off as 0.4, and the maximum number of nodes in a graph when performing min-cut algorithm instead of normal-cut algorithm is 80. This will generate the dense subgraph file as myoutputfile.

Example for running mode at 1:

```
modes –m 1 –i myinputfile -n genenum –o myoutputfileprefix –g 5 –e 6 –d 4 –s 80 –c 0.6
–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, and the maximum number of nodes in a graph when performing min-cut algorithm instead of normal-cut algorithm is 80. This will generate the subgraph file containing gene 21 as myoutputfile.

Note

This is MODES version 1.1. Testing hasn't been exhaustive. Feedback and application description are always welcome. Contact xjzhou@usc.edu for bugs and questions about MODES.

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