

EXCAVATOR

Software for gene expression data analysis

Manual

Overallview

EXCAVATOR is a computer software for gene expression data clustering. It employs a set of unique clustering algorithms developed by the Protein Informatics Group of Oak Ridge National Laboratory. Unlike any existing gene expression analysis tools, EXCAVATOR represents a set of gene expression data as a minimum spanning tree (MST), a concept from the graph theory. We have rigorously proved that the MST representation, though simple in its structure, did not lose no essential information for the purpose of the data clustering. Through this representation, we have reduced a multiple-dimensional data clustering problem to a tree partitioning problem, a much simpler problem computationally. The MST representation facilitates

- efficient implementations of clustering algorithms with guaranteed mathematical properties, including global optimality;
- strong capabilities in handling data clusters with complex cluster boundaries; and
- strong capabilities in overcoming problems caused by background noise.

EXCAVATOR provides the following features data clustering under various definitions of *distances* and clustering algorithms, data-constrained clustering, automatic selection of the most plausible number of clusters in a data set, removal of background noise, identification of gene with similar expression profiles to a set of specified *seed genes*, and comparison of clustering results using different clustering parameters and algorithms.

General Guidance

EXCAVATOR can run from any Unix/Linux shell as a stand-alone package without any interface or third-party software. It has been tested extensively on Sun, DEC, and Linux PC. Running EXCAVATOR from a Unix shell (rather than from a GUI) provides a convenient way to handle multiple data sets and multiple runs automatically by using scripts. All the options for the program can be specified through flags on the command line. Multiple flags without conflicts can be used at the same time, and their orders in the command line do not matter. For more detail, please see our publications

Ying Xu, Victor Olman, and Dong Xu. Clustering Gene Expression Data Using a Graph-Theoretic Approach An Application of Minimum Spanning Trees. *Bioinformatics*. In press.

Ying Xu, Victor Olman, and Dong Xu. Minimum Spanning Trees for Gene Expression Data Clustering. In *Proceedings of the 12th International Conference on Genome Informatics (GIW)*, edited by S. Miyano, R. Shamir and T. Takagi. Accepted.

Installation

To install EXCAVATOR on Unix, all you need is a GCC compiler. After you get the source code (header.h and excavator.c), compile an executable by doing

```
gcc -O3 -o excavator excavator.c -lm
```

You can save the executable (excavator) in a directory of your Unix command path, e.g., /usr/local/bin. You may also save it elsewhere and put an alias to it in your .cshrc or .login file, e.g.,

```
alias excavator /home/john/software/excavator/excavator
```

Inputs

The gene expression profiles can be saved in an input file. The input file is always the last argument on a command line. Each line in the input file represents the expression profile for one gene, except lines starting with "#" or "REMARK", which will be ignored as comments. EXCAVATOR can read the input data in three formats, where a flag "-input n" will let the program know which one to use

Entries separated by tabs with annotation (default, see an example)

- GeneId annotations data-1 data-2 ...
- "GeneId" is a string without containing any space or tab. The annotation can be any strings without containing any tab (space is fine). Use "-input 1" or no "-input" flag for the input format, e.g.,
- excavator -input 1 stanford.dat
- or
- excavator stanford.dat

Entries separated by tabs without annotation (see an example)

- GeneId data-1 data-2 ...
- Use "-input 2" for the input format, e.g.,
- excavator -input 2 CEFH.dat

Entries separated by space without annotation (see an example)

- GeneId data-1 data-2 ...
- Use "-input 3" for the input format, e.g.,
- excavator -input 3 CEFH-s.dat

The default setup assumes that the data in the input file are log ratios of gene expression levels. If the data have been applied by log, then use "-log" flag, i.e.,

```
excavator -log input.data
```

Sometimes, the gene expression data may not be complete for all data points in every gene. For the first two formats of the data, one can indicate a missing data point by two consecutive tabs (see an example). You can use the "-miss n" flag to specify how you want to replace the missing data

- "-miss 1" or no "-miss" flag (default) use 0 as the log ratio of the gene expression level.
- "-miss 2" use the average over other genes at the same column of the data series.
- "-miss 3" use the average over all the other known data points of the same gene.
- "-miss 4" use the average over two neighboring known data points of the same gene.

The default assumes that data of all columns in the input file will be used. In this case, EXCAVATOR reads the first line, and then determines the number of data points for each gene (N-genes). If the number of data points for a following line is different from "N-genes", this line will be ignored (a warning message will be printed). If you only want to use the first "NAttribute" data points (excluding the GeneId and annotation), you can apply

```
excavator -nattribute NAttribute input.data
```

If you like to ignore certain columns of the data points, you can use the "-ignore" flag followed by the column numbers (separated by comma), e.g.,

```
excavator -ignore 3,7,23 input.data
```

Similarity Measure

The similarity measure represents how to calculate the distance between gene expression profiles. EXCAVATOR uses the "-dist n" flag for the options of similarity measure

- "-dist 1" or no "-dist" flag (default)(1 - correlation coefficient).
- "-dist 11"(1 - square of correlation coefficient).
- "-dist 12"(1 - absolute value of correlation coefficient).
- "-dist 2"Euclidean distance.
- "-dist 21"square of Euclidean distance.
- "-dist 3"sine square of the angle between two vectors.

Clustering Methods

EXCAVATOR offers the following methods for clustering algorithms

- "-h" (default)hierarchical clustering based on the objective function for the selected similarity measure (to optimize the sum of the distance between a gene and the center of its cluster hierarchically).
- "-i"non-hierarchical clustering based on the objective function for the selected similarity measure using an iterative approach (to optimize the sum of the distance between a gene and the center of its cluster iteratively; the clustering result may not reach the global optimal solution for the objective function).
- "-g"non-hierarchical clustering to optimize the sum of the distance between a gene and its best representative gene in the cluster. The globally optimal solution is guaranteed, but it takes much longer time than other methods.
- "-ledge"hierarchical clustering by simply cutting longest edges on the minimum spanning tree. It is the fastest method, but the result may not be desired.
- "-cutoff CutoffDistance -mne MinNumElement"cutting all the edges longer than "CutoffDistance" and remove all the small clusters with less than "MinNumElement" elements.

Number of Clusters

A user can either specify the number of clusters or let EXCAVATOR determine it automatically. To specify the number of clusters, use "-ncluster n" flag, where "n" is the number of clusters, e.g.,

```
excavator -ncluster 3 input.data
```

EXCAVATOR can determine the number of clusters through calculating the objective functions for different numbers of clusters up to "MaxNCluster" cluster, where "MaxNCluster" can be specified by the "-maxncluster MaxNCluster" flag. e.g.,

```
excavator -maxncluster 100 input.data
```

Without either "-ncluster" or "-maxncluster" flag (i.e., default), EXCAVATOR will automatically give the value of "MaxNCluster" (up to 1/3 of the total number of genes, depending on the clustering methods). If you like to see the the objective functions for different numbers of clusters first, you can run

```
excavator -profile input.data
```

which will generate "quality.data" and "diff.data" (see the following).

Constraints

EXCAVATOR allows a user to add constraints so that certain specified genes will stay in the same cluster. The flag "-constraint ConstraintFile" will enforce the constraints. The format of the "ConstraintFile" is that genes in the same line (separated by spaces or tabs) are forced into the same cluster (e.g., see list.cons). For example, by using the file CEFH.cons,

```
excavator -input 2 -cons CEFH.cons -ncluster 4 CEFH.dat
```

will force the genes "eTYE7" and "cRPT3" in the same cluster.

Another related option in conjunction with the "-cutoff CutoffDistance" flag is to cut long edges with distance longer than "CutoffDistance" and then select the clusters which contain the genes specified in "ConstraintFile". In this case, a "-cn" flag is needed, e.g.,

```
excavator -input 2 -cons CEFH.cons -cutoff 0.1 -cn CEFH.dat
```

In this case, only 26 out of the original 68 genes in CEFH.dat will be left and they contain the genes "eTYE7" and "cRPT3" specified in "CEFH.cons". Instead of using the "-cutoff CutoffDistance" flag, you can also choose the number of genes to be left instead of "CutoffDistance" by using the "-ns NumLeft" flag, e.g.,

```
excavator -input 2 -cons CEFH.cons -cn -ns 20 CEFH.dat
```

will leave 20 out of the original 68 genes in CEFH.dat, and they also contain the genes "eTYE7" and "cRPT3" specified in "CEFH.cons". Sometimes the number of genes left may differ a little from what is specified ("NumLeft") due to the constraints.

Output Files

EXCAVATOR may produce the following files, depending on the flags used in the command line

- "cluster.out" the major clustering result file (see an example). The comments in the file are self-explanatory. Each line of gene expression profile gives the "GeneId", annotation, and the data points (without log being added).
- "cluster.tree" the binary tree for the clustering result (see an example). The comments in the file are self-explanatory. Each line after "#Tree links" shows four two-dimensional coordinates. Linking these coordinates will produce the binary tree.
- "MST.data" the minimum spanning tree, where each line shows the distance and the vertices associated with the distance (see an example).
- "quality.data" the optimized assessment function (the second column) versus the number of clusters (the first column) (see an example).
- "diff.data" the transition profile (the second column) as a function of the number of clusters (the first column) (see an example). The peak of this function often indicates "natural" number of clusters.
- "dist-distri.data" the distribution for the length of the minimum spanning tree (see an example). This file can be generated using the "-filter" flag, e.g.,
 - excavator -filter stanford.dat
- "Filter.data" the genes left after the filtering by the "-cutoff" or "-cn" flag (see an example).
- "FLAG" an empty file indicating the calculating is finished properly.

Please note that EXCAVATOR will overwrite existing files with the same names as specified above. Hence, if you want to save the results of your previous run, you need to rename the related files.

Comparing Clustering Results

You can compare two clustering results (in the cluster.out format) on the same data set using the "-comp" file

```
excavator -comp cluster.out cluster.out
```

It will give a value between 0 (most different clustering results) and 1 (identical clustering results).