BCLUST -- A program to assess reliability of gene clusters from expression data by using consensus tree and bootstrap resampling method

Introduction

This program is developed in the lab of Hongyu Zhao in Department of Epidemiology and Public Health of Yale University School of Medicine. It was written by Kui Zhang who is a postdoctoral associate in Hongyu's lab. The program is free for academic use and not permitted for commercial purpose under any circumstance.

BCLUST is a program to investigate the reliability of gene clusters derived from large-scale gene expression data by using hierarchical clustering algorithms. Because with the rapid development of microarray technologies, many of statistical procedures, most notably a variety of clustering algorithms, have been successfully applied to analyze such kind of data. But in essentially all published studies, the observed gene expression levels are treated as if they were an accurate measure of the true expression level, and effects of variations and uncertainties in measured gene expression levels across samples and experiments have been seldom addressed. In the context of hierarchical clustering algorithms, we introduce a statistical resampling method to assess the reliability of gene clusters identified from any hierarchical clustering method. Using the clustering trees constructed from the resampled data sets, we can evaluate the confidence value for each node in the observed clustering tree. A majority-rule consensus tree can be obtained showing clustering that only occur in a majority of resampled trees.

BCLUST reads a file that contains large-scale gene expression row data and another file that contains corresponding error data first, then use these data and bootstrap method to resample many gene expression data sets. After this moment, the original tree and a set of resampled tree will be derived by using the same clustering algorithm, then it constructs a consensus tree by carrying out a family of consensus tree methods called the ML methods (Margush and McMorris, 1981). Basically the consensus tree consists of monophyletic groups that occur as often as possible in the data. If a group occurs in more than 50% of all the resampled trees it will definitely appear in the consensus tree. At the same time, the confidence value at the nodes in the original tree is also evaluated. These values can assess the reliability of these gene clusters. The more details of these methods were described in the references.

The main Menu

To run **BCLUST**, you need a compiled copy of the program, at least one input data file and one output file. This program is written by Visual C++, so you may run it under

Windows 95, 98 or NT operating system. On running the program, you will be asked to choose an input file first. Then you will be asked if the error data file exits. If such file does not exist, you should give a positive real number to indicate the scale of data error. At the last, you will be requested to give the name of an output file.

Once these file names and choices is given, you will see the central menu of the program, which looks like this:

Here are the settings:

(0) The name of expression file is : test_exp.dat
(1) The name of error file is : test_err.dat
(3) The format of input file is : has row names, has column names
(3) The sample size is : 500
(4) Preprocess data : Yes
(5) Preprocess data by : by row, by meancenter
(6) Distance method is : correlation coefficient
(7) Clustering method is : average
(8) The name of output file is : test_out.dat
(9) The format of output file is : New_Hampshire
Do you want to accept these options? (Yes or No)
Type Y or N or the number (0-9) to change the option:

These are the parameters that control reading data, resampling method, computing pair-wise distance, evaluating hierarchical clustering tree, formatting the tree in the output file. You can either accept them as they are, in which case you can type "Y" to the question and press the Return or Enter key, or you can answer "N" if you want to change one, or simply type the digit corresponding to the option you want to change. In fact, if you answer "N" it will just immediately ask you for what number anyway.

When you answer "Y" on running the program, the program will do the following jobs sequentially: reading the gene expression data from the input file, reading error data when it is applicable, deriving the original clustering tree, resampling data by using appropriate methods and then constructing the resampling clustering tree, evaluating the confidence values for original tree and constructing consensus tree. At last, all the results will be written into an output file. The program will take some several seconds to hours to do them, depending on the data size and the resampling size.

When you answer "N" or a digit to change an option, it will go back to the above menu after you set it, allow you to change more options or parameters, and go through the whole process again. The easiest way to learn the meaning of these options and parameters is to try them by yourself. Below the options will be described one by one; you may prefer to read them carefully unless you are familiar with all of them.

Input, Output and Options

The option (0) is to specify the input file that contains large-scale gene expression row data. Once you input a string of characters to represent the name of file, BCLUST will check if it exists. If this file does not exist or can not be opened to read, the program will let user to input it again.

The option (1) sets the error of the gene expression data. Initially, you will be asked if you have an error file. When you answer "Y" for this question, the program will ask you to provide the name of this file. You must give a valid file which it can be opened to read. Sometimes, such variability of the gene expression measurement can be estimated from the replicated experiments. However, even if we do not know it, we also can study the patterns for possible variation in the data to help us understand the reliability of clusters. So when you select "N" about this question, a scale of error, which is a positive float number, must be given. Under this circumstance, we assume that the standard error S for the ith gene in the jth experiment is proportional to the observed expression measure X. This proportion is just described by the scale of error.

The option (2) controls the format of input files. Generally, a file containing of row data is a text file. An example shows below:

Nrow	**Ncol**			
	Exp1	**Exp2**	**Exp3**	**…**
Gene1	**Data**	**Data**	**Data**	**…**
Gene2	**Data**	**Data**	**Data**	**…**
Gene3	**Data**	**Data**	**Data**	**…**
…	**…**	**…**	**…**	**…**

Table 1: An example of the input file.

The cells in green must appear in the file, although they must be real number. The cells in red are optional for columns/rows. In this file, the "Nrow" is an integer number, which is the number of rows in the data, especially it is the number of genes in the analysis. The "Ncol" is an integer number, which is the number of columns in the data, especially it is the number of different experiments in the analysis. The "GeneId" is character string to describe each gene, which will be used in display. The "ExpId" is a text description of each experiment that will be used in the display. The "Data" is the real number for a single gene in a single experiment and the missing values are not acceptable. For the output purpose, "GeneId" and "ExpId" can be any string of characters, except blanks, colons, semicolons, parentheses, and square brackets. The length of them can not be greater than 60.

Sometimes, the name of genes or the name of experiments is not available, so we offer this option to format the file. When you select this option, a submenu will appear on the screen, it looks like this:

Please select one of the following formats of the input file:

1: has row names, has column names
2: has row names, no column names
3: no row names, has column names
4: no row names, no column names

At this moment, you can select one of them that fit your file format. Initially, this option is "has row names, has column names". At the same time, the error file, if it is applicable, must have the same format with the expression data file.

The option (3) specifies the sample size when we resample data sets. Initially, this is 500 times. This number must be a positive integer. When the sample size is larger, the more reliable and precision of result will be obtained. But it also will cost more computer memory and running time. From our experience, the small number of sample size, for example 200, can get comparable or similar results with the relatively large sample size, for example1000.

The option (4) determines whether the data is preprocessed before we construct gene clusters by using them. Once you want to preprocess them, two submenus will appear on the screen one by one, they look like this:

Do you want to preprocess data by:
1: by row
2: by column

Please select one of the following methods to preprocess data:
1: logarithmic transformation
2: normalization
3: mean center
4: median center
5: maximum normalization

You can select one of the five methods to preprocess data across rows or columns. The default of this option is not to preprocess the data.

The option (5), if the option (4) is turned on, specifies the methods of preprocessing data. The two submenus are just same with above. The first sub-option controls to process data across by rows or columns. The second specifies the method to process data.

The option (6) specifies the method to compute the pairwise distances between objects in the data. The choices for the distance are "correlation coefficient", "euclidean", "maximum", "manhattan" and "binary". The "correlation coefficient" distances are equal to one minus their correlation coefficient. The "euclidean" distances are root sum-of-squares of their differences. The "maximum" distances are the maximum differences between objects. The "manhattan" distances are the sum of their absolute differences. The "binary" distances are the proportion of non-zeros that two vectors do not have in common (the number of occurrences of a zero and a one, or a one and a zero divided by the number of times at least one vector has a one). Both these metric methods have applied to the clustering algorithms.

Once you want to change it, a submenu will appear on the screen, it looks like this:

Please select one of the following methods to compute the pairwise distance:

1: correlation coefficient
2: euclidean
3: maximum
4: manhattan
5: binary

The option (7) specifies the method to perform hierarchical clustering based on a distance or similarity structure. Currently choices of methods are "average", "compact (complete linkage)" and "connected (single linkage)". In the hierarchical clustering algorithms, at each stage the two "nearest" clusters are combined to form one bigger cluster (initially each cluster contains a single point). For the method "connected" the distance between two clusters is the minimum distance between an object in the first cluster and an object in the second cluster. In the method "average" the distance between clusters is the average of the distances between the objects in one cluster and the objects in the other cluster. The largest distance between an object in one cluster and an object in the other cluster is used in the method "compact". Long thin clusters are typically created with "connected", and more spherical clusters are formed with "compact". You can choose one of them through a submenu, which looks like this:

Please select one of the following methods to clustering objects:

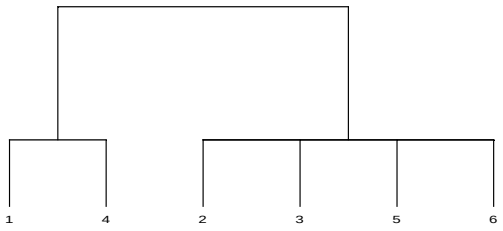
1: average
2: compact (complete linkage)
3: connected (single linkage)

The option (8) will ask you to provide an output file. The program will ask you to input a character string as the name of output name. When this file has been existed in the computer, it will be replaced by it without any warning. So before you perform this option, please pay attention to avoid corrupting the important files in your computer.

The option (9) specifies the output tree form rather than the file format. Because the results derived from hierarchical clustering algorithms can be represented as a tree. The output file is a text file, which contains some parameters used in the program and a list of trees. Currently the choices for tree form are "S-plus format" and "New Hampshire format".

By the S-plus format, a tree consists three components: the first is "merge" which a matrix that row *i* describes the merging of clusters at step *i* of the clustering. The positive integer number in the first column describes the number of objects and sub-clusters that merge in this step. The integer number in the other columns specifies the merged objects and sub clusters. If the element *j* in the other columns is negative, then object -j was merged at this stage. If *j* is positive, then the merge was with the cluster formed at the (earlier) stage *j* of the algorithm. The second item is "height", which is a vector that represents the distance between clusters merged at the successive stages. The length of this vector is equal to the number of rows in the "merge" matrix. The third component is "order", which is a vector whose length is equal to the number of rows in the data file,

will give a permutation of the original objects suitable for plotting, in the sense that a cluster plot using this ordering will not have crossings of the branches. In the *n* hierarchical cluster displays, a decision is needed at each merge to specify which subtree should go on the left and which on the right. The default algorithm in **BCLUST** is to order the subtrees so that the tighter cluster is on the left (the last merge of the left subtree is at a lower value than the last merge of the right subtree). Individuals are the tightest clusters possible, and merges involving two individuals place them in order by their observation number. For example, if we have such a tree:



By using this option, the tree in the output file will be:

The number of nodes is : 3
The node of the tree is :
2 -1 -4
4 -2 -3 -5 -6
2 1 2
The height of the tree node is :
…
…
…
The object order of the tree is :
1 4 2 3 5 6

The New Hampshire Standard for representing trees in computer-readable form makes use of the correspondence between trees and nested parentheses, noticed in 1857 by the famous English mathematician Arthur Cayley. If we have above tooted tree, then in the tree file it is represented by the following string of printable characters, starting at the beginning of the file:

((1,4),(2,3,5,6));

The tree ends with a semicolon. Interior nodes are represented by a pair of matched parentheses. Between them are representations of the nodes that are immediately descended from that node, separated by commas. Objects are represented by their names. A name can be any string of printable characters except blanks, colons, semicolons,

parentheses, and square brackets. Branch lengths can be incorporated into a tree by putting a real number, with or without decimal point, after a node and preceded by a colon. This represents the length of the branch immediately below that node. Thus the above tree might have lengths represented as:

((1,4):2.05,(2,3,5,6):3.1):4.32;

Many programs, especially some tree drawing software, can use the New Hampshire Standard form.

You can choose one of them through a submenu, which looks like this:

Please select one of the following formats of the output file:

1: S-plus format

2: New Hampshire format

Examples

To help users to understand these options better, we give an example bellow. The input files include an expression file named test_exp.txt and an error file named test_err.txt, which are from Wen et al. (1998). The output files are test_out_splus.txt and test_out_nh.txt.

Although this program does not provide options to draw the tree, but users can find many software to display it by using the output file.

References

Felesenstein J(1985) *Confidence limits on phylogenies: an approach using the bootstrap*. **Evolution** 39: 783-791

Margush T, McMorris FR (1983) *Consensus n-trees.* **Bulltein of Mathematical Biology** 43:239-244

Zhang K, Zhao H (2000) *Assessing reliability of gene clusters from gene expression data.* **Functional and Integrative Genomics** 1:156-173

Wen X, Fuhrman S, Michaels GS, Carr DB, Smith S, Barker JL, Somogyi R (1998) *Large-scale temporal gene expression mapping of central nervous system development.* **PNAS** 95:334-339