Package ‘CLAG’

February 19, 2015

Type  Package
Title  An unsupervised non hierarchical clustering algorithm handling biological data
Version  2.18.1
Date  2013-09-13
Author  Linda DIB, Raphael CHAMPEIMONT, Alessandra CARBONE
Maintainer  Raphael CHAMPEIMONT <raphael.champeimont@upmc.fr>
Description  CLAG (for CLusters AGgregation) is an unsupervised non hierarchical clustering algo-
rithm designed to cluster a large variety of biological data and to provide a clustered ma-
trix and numerical values indicating cluster strength. CLAG clusterizes correlation matri-
ces for residues in protein families, gene-expression and miRNA data related to various can-
cer types, sets of species described by multidimensional vectors of characters, binary matri-
ces. It does not ask to all data points to cluster and it converges yielding the same re-
sult at each run. Its simplicity and speed allows it to run on reasonably large datasets.
License  BSD_2_clause + file LICENSE | GPL (>= 2)
LazyLoad  yes
Depends  R (>= 2.14)
Suggests  clue (>= 0.3-46)
URL  http://clag.r-forge.r-project.org/
SystemRequirements  perl
NeedsCompilation  no
Repository  CRAN
Date/Publication  2013-09-13 12:29:19

R topics documented:

BREAST ................................................................. 2
CLAG.clust .............................................................. 2
compareClusterings .................................................... 4
DIM128 ................................................................. 5
Example data set provided with CLAG (breast)

Description

Example data set provided with CLAG. This data set contains microarray data from breast tumor cells, from 20 samples and 378 genes. The data sets are described in the article (see below).

Value

The returned value is a data frame.

References

CLAG: an unsupervised non hierarchical clustering algorithm handling biological data, Linda Dib, Alessandra Carbone, BMC Bioinformatics 2012, 13:194

Run cluster analysis with CLAG

Description

This function computes clusters with the CLAG algorithm. CLAG is an unsupervised non hierarchical clustering algorithm, especially adapted for data sets with many variables, like is often the case with biological data. It does not ask for all data points to cluster. To understand what the parameters are, please refer to the reference article (see below). This function returns what is called "key aggregates" in the article.

Usage

CLAG.clust(M, delta=0.05, threshold=0, analysisType=1, normalization="affine-global", rowIds=NULL, colIds=NULL, verbose=FALSE, keepTempFiles=FALSE)
Arguments

\textbf{M} \hspace{1cm} A matrix or data frame where rows are points to cluster and columns are variables.

\textbf{delta} \hspace{1cm} Delta is a parameter which influences how CLAG decides whether two values are near or not. This parameter is ignored if we are analyzing a discrete matrix (if analysisType=2). Examples: 0.05, 0.1, 0.2

\textbf{threshold} \hspace{1cm} Threshold to apply to the environment score (if analysisType is 1 or 2) or to both symmetric and environment scores (if analysisType is 3). Examples: 0, 0.5, 1

\textbf{analysisType} \hspace{1cm} Set to 1 for a general matrix where rows are points and columns are real variables. Set to 2 for the case when the variables are discrete (can be strings, integers or boolean). Set to 3 if the variables are real and are themselves a (strict or not) superset of the points AND you want to take in account the symmetric score. In that case the matching between rows and columns is specified using the rowIds and colIds parameters.

\textbf{normalization} \hspace{1cm} Only relevant for real variables (analysisType is 1 or 3). A string in "affine-global", "affine-column", "rank-column". CLAG works by first computing a global distribution for all values in the matrix, assumed to be in [0,1]. By default, it performs a single affine transform on all values on the matrix ("affine-global"), that is the minimum value in the matrix is mapped to 0 while the maximum is mapped to 1. While this is relevant if variables are in comparable units, it is inappropriate if they represent very different magnitudes. For this latter case, two other normalization methods are provided. The first, "affine-column", does the same affine transform but independently for every column, that is it maps minima of columns to 0 and maxima of columns to 1. This might be inappropriate if distribution shapes are very different between columns. The second method, "rank-column", replaces values by their ranks in the specific column, therefore making every columns distribution uniform.

\textbf{rowIds} \hspace{1cm} When analysisType=3, used to specify integer ids for rows. Its length must be equal to the number of rows in the matrix. Not necessary for a square matrix.

\textbf{colIds} \hspace{1cm} When analysisType=3, used to specify integer ids for columns. Its length must be equal to the number of columns in the matrix. If not given, it is assumed to be 1:nrow(M).

\textbf{verbose} \hspace{1cm} Display the underlying CLAG program output during computation.

\textbf{keepTempFiles} \hspace{1cm} Keep temporary files created by CLAG execution (useful for debugging).

Value

The returned value is a list with several members:

\textbf{nclusters} \hspace{1cm} Number of final clusters found by CLAG (called "key aggregates" in the article)

\textbf{cluster} \hspace{1cm} A vector of integers indicating the cluster id (from 1 to nclusters) to which each point is allocated. Value 0 means the point is not in any cluster (there may or may not be such points).

\textbf{firstEnvScore} \hspace{1cm} Environmental score of the first before-aggregation cluster in each aggregated cluster.
compareClusterings

lastEnvScore  Environmental score of the last before-aggregation cluster in each aggregated cluster.
firstSymScore  Symmetric score of the first before-aggregation cluster in each aggregated cluster. Only when analysisType=3.
lastSymScore  Symmetric score of the last before-aggregation cluster in each aggregated cluster. Only when analysisType=3.
A  The input matrix normalized with the method chosen. (except when analyzing discrete variables)

Members delta, threshold, analysisType, M, rowIds and colIds contain the original arguments given to CLAG.

References

CLAG: an unsupervised non hierarchical clustering algorithm handling biological data, Linda Dib, Alessandra Carbone, BMC Bioinformatics 2012, 13:194

Examples

# Example with real variables
data(DIM128, package="CLAG")
# Take a subset (this is to make the example fast
# but you can use the entire dataset)
M <- DIM128[seq(1, nrow(DIM128), by=20),]
# Run the cluster analysis
RES <- CLAG.clust(M)
# Display points in 2D using a PCA and color them by cluster
# except unclustered points which are left black.
PCA <- prcomp(M)
colors <- c("black", rainbow(RES$ncluster))
plot(PCA$x[,1], PCA$x[,2], col=colors[RES$cluster+1], main=paste(RES$nclusters, "clusters"))

---

compareClusterings  Compare two clusterings of the same elements (and print results)

Description

Same as mapClusterings except that is prints nicely the result instead of just returning it.

Usage

compareClusterings(cl1, cl2, verbose=FALSE, use.solve.LSAP=NULL)
**Arguments**

- `c11` First clustering, see `mapClusterings`.
- `c12` Second clustering.
- `verbose` Display information about tested cases.
- `use.solve.LSAP` Whether to use `solve_LSAP` function from package `clue`. If NULL (the default), `solve_LSAP` will be used only if the package `clue` is installed.

**Examples**

```r
compareClusterings(c(0,1,1,2,2,3,3,3), c(0,3,3,3,1,1,2,2,2), verbose=TRUE)

compareClusterings(c(0,1,1,2,2,3,3,3,0), c(3,1,3,3,1,1,2,0,2,3,4,0), verbose=TRUE)
```

---

**DIM128**

*Example data set provided with CLAG (DIM128)*

**Description**

Example data set provided with CLAG. This is an artificial data set with 1024 observations of 128 variables. The data sets are discribed in the article (see below).

**Value**

The returned value is a data frame.

**References**

CLAG: an unsupervised non hierarchical clustering algorithm handling biological data, Linda Dib, Alessandra Carbone, BMC Bioinformatics 2012, 13:194

---

**DIM128_subset**

*Example data set provided with CLAG (DIM128 subset)*

**Description**

Example data set provided with CLAG. This rows of this matrix are a subset of the rows of DIM128. The data sets are discribed in the article (see below).

**Value**

The returned value is a data frame.

**References**

CLAG: an unsupervised non hierarchical clustering algorithm handling biological data, Linda Dib, Alessandra Carbone, BMC Bioinformatics 2012, 13:194
GLOBINE

Example data set provided with CLAG (globine)

Description

Example data set provided with CLAG. This is a square matrix with coevolution coefficients (sort of correlation measure). The data sets are described in the article (see below).

Value

The returned value is a list with the data frame and row and column ids.

References

CLAG: an unsupervised non hierarchical clustering algorithm handling biological data, Linda Dib, Alessandra Carbone, BMC Bioinformatics 2012, 13:194

mapClusterings

Compare two clusterings of the same elements

Description

This function solves this problem: Given two clusterings of the same elements, how many elements are in a different cluster, when we map for the best the clusters of the first clustering to the second?

The function finds the best mapping between clusters of cl1 with clusters of cl2 (injective in both directions - but not necessarily surjective), that is such that the number of differently clustered elements is minimum.

An element e is identically clustered if one of these conditions is true:

1. it belongs to a cluster i in cl1 and j in cl2 and i is associated with j.
2. e is unclustered in both cl1 and cl2

Differently clustered is the opposite.

Usage

mapClusterings(cl1, cl2, verbose=FALSE, use.solve.LSAP=NULL)

Arguments

c1l A vector of integers such that cl1[i] is the number (>= 1) of the cluster in which i is. 0 is a special value that means the element is not clustered. (This is the format of the cluster field returned by CLAG.clust)
c12 The same for the other clustering.
verbose Display information about tested cases.
use.solve.LSAP Whether to use solve_LSAP function from package clue. If NULL (the default), solve_LSAP will be used only if the package clue is installed.
Details

We propose two methods for computing the number of differently clustered elements: using the solve_LSAP function from package clue or using our branch and bound algorithm. You can choose which one to use by setting the use.solve.LSAP parameter.

The solve_LSAP function is much faster both in practice and theory (it is polynomial) but requires the clue package to be installed. See its own help page for information about how it works.

The branch and bound algorithm we provide works like this: For a cluster $i$ in the first clustering, it tries to associate it with every cluster $j$ not already associated (and also to leave it alone), and for each of this choices explores recursively the choice for cluster $i+1$. In the worst case it runs in exponential time.

The exploration is optimized by:

- considering only choices of $j$ for which at least one element would be common with $i$ (otherwise it would always be better to not associate $i$ at all)
- exploring $j$ other than 0 before 0 (more likely to find solution faster)
- keeping in memory the "best association found", cutting a branch if the clusters already associated at the node already generate more differently clustered elements that in the best association found
- stopping exploration in case a perfect solution ($\text{ndiff}=0$) is found

Value

The returned value is a list with several members:

- $\text{ndiff}$: The number of elements differently clustered.
- $\text{assoc}$: Cluster mapping. $\text{assoc}[i] = j$ means cluster $i$ in first clustering is associated to cluster $j$ in second clustering. If $j = 0$, $i$ is not associated to any cluster.
- $\text{diffclust}$: A vector of integers of length $\text{ndiff}$ giving the indices of elements that are differently clustered.

See Also

- compareClusterings

Examples

```r
mapClusterings(c(0,1,1,1,2,2,2,3,3,3,3,0), c(3,1,3,3,1,1,2,0,2,4,0))
```
Index

*Topic cluster
   CLAG.clust. 2
   compareClusterings. 4
   mapClusterings. 6
*Topic datasets
   BREAST. 2
   DIM128. 5
   DIM128_subset. 5
   GLOBINE. 6

BREAST. 2

CLAG.clust. 2, 6
compareClusterings. 4, 7

DIM128. 5
DIM128_subset. 5

GLOBINE. 6

mapClusterings. 4, 5, 6