Package ‘progenyClust’

April 12, 2016

Type Package

Title Finding the Optimal Cluster Number Using Progeny Clustering

Version 1.2

Date 2016-04-08

Author C.W. Hu

Maintainer C.W. Hu <wendyhu001@gmail.com>

Description Implementing the Progeny Clustering algorithm, the 'progenyClust' package assesses the clustering stability and identifies the optimal clustering number for a given data matrix. It uses k-means clustering as a default, provides a tailored hierarchical clustering function, and can be customized to work with other clustering algorithms and different parameter settings. The package includes a main function progenyClust(), plot and summary methods for 'progenyClust' object, a function hclust.progenyClust() for hierarchical clustering, and two example datasets (test and cell) for testing.

License AGPL-3

Imports Hmisc

Depends graphics,stats,grDevices

NeedsCompilation no

Repository CRAN

Date/Publication 2016-04-12 20:04:15

R topics documented:

`progenyClust-package` *Finding the Optimal Cluster Number Using Progeny Clustering*

Description

Implementing the Progeny Clustering algorithm based on Hu, Chenyue W., et al. "Progeny Clustering: A Method to Identify Biological Phenotypes." Scientific Reports 5 (2015), the progenyClust package assesses the clustering stability and identifies the optimal clustering number for a given data matrix. It uses kmeans clustering as default, but can be customized to work with other clustering algorithms and different parameter settings. The package includes one main function progenyClust(), plot and summary methods for "progenyClust" object, and two example dataset ("test" and "cell") for testing.

Details

Package: progenyclust
Version: 1.1
Date: 2015-11-24
License: AGPL-3
Imports: Hmisc
Depends: graphics, stats

Author(s)

C.W. Hu, Rice University
Maintainer: C.W. Hu <wendyhu001@gmail.com>

References

Hu, C.W., et al. "Progeny Clustering: A Method to Identify Biological Phenotypes." Scientific reports 5 (2015).
http://www.nature.com/articles/srep12894

`cell` *cell imaging metrics dataset*

Description

This cell dataset contains the first three principal components of the imaging metrics for 444 cells that were engineered into four patterns. The dataset therefore should include four clusters of cell samples in theory. See the references for more experimental and imaging analysis details of this data.

Usage

```
data("cell")
```

Format

A data frame with 444 observations on the following 3 variables.

`PC1` The first principal component of imaging metrics

`PC2` The second principal component of imaging metrics

`PC3` The third principal component of imaging metrics

References

Slater, John, et al. "Recapitulation and Modulation of the Cellular Architecture of a User-Chosen Cell-of-Interest Using Cell-Derived, Biomimetic Patterning." ACS nano (2015).
Hu, C.W., et al. "Progeny Clustering: A Method to Identify Biological Phenotypes." Scientific reports 5 (2015).

Examples

```
data(cell)
```

`hclust.progenyClust` *Hierarchical Clustering*

Description

hierarchical clustering function for progeny clustering

Usage

```
hclust.progenyClust(x,k,h.method='ward.D2',dist='euclidean',p=2,...)
```

Arguments

`x` a numeric matrix, data frame or `dist` object.

`k` an integer specifying the number of clusters.

`h.method` the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of `"ward.D"`, `"ward.D2"` (default), `"single"`, `"complete"`, `"average"` (= UPGMA), `"mcquitty"` (= WPGMA), `"median"` (= WPGMC) or `"centroid"` (= UPGMC).

`dist` the distance measure to be used. This must be one of `"euclidean"`, `"maximum"`, `"manhattan"`, `"canberra"`, `"binary"` or `"minkowski"`. Any unambiguous substring can be given.

`p` The power of the Minkowski distance, when `dist="minkowski"`.

`...` additional arguments in `hclust`(...).

Details

The function `hclust.progenyClust` mainly streamlines `dist`, `hclust` and `cutree` into one, and structures the output to be directly used by `progenyClust`. Most arguments and explanations were kept the same to ensure consistancy and avoid confusion. For more details, please check each individual function.

Value

`cluster` A vector of integers (from 1:k) indicating the cluster membership for each sample.

`tree` An object of class `hclust` which describes the tree produced by the clustering process.

`dist` A dissimilarity structure as produced by `dist`.

Author(s)

C.W. Hu, Rice University

References

Hu, C.W., et al. "Progeny Clustering: A Method to Identify Biological Phenotypes." Scientific reports 5 (2015).
`http://www.nature.com/articles/srep12894`

Examples

```
# a 3-cluster 2-dimensional example dataset
data('test')

# default progeny clsutering
progenyClust(test,FUNclust=hclust.progenyClust,ncluster=2:5)->pc

# plot the scores to select the optimal cluster number
plot(pc)

# plot the clustering results with the optimal cluster number
plot(pc,test)
```

`plot.progenyClust` *Plot Progeny Clustering Results*

Description

Plot the cluster number selection results and visualizes the clustering results.

Usage

```
## S3 method for class 'progenyClust'
plot(x,data=NULL,k=NULL,errorbar=FALSE,xlab='',ylab='',...)
```

Arguments

`x` a `progenyClust` object.

`data` the full or a subset of the oringal data matrix that was used for clustering. If unspecified, the function will plot stability scores for cluster number selection; If specified, the function will plot the data in scatter plots with colors annotated by clustering memberships (Please see details below).

`k` integer specifying the cluster number for visualizing the clustering results of original data: only takes into effect when argument `data` is provided, and needs to be a cluster number that was previously investigated in `progenyClust` to generate the `progenyClust` object `x`. The default is the optimal number of clusters.

`errorbar` logical flag: specifies whether the error bars should be drawn. The error bars can only be drawn when progeny clustering is repeated multiple times, i.e. input argument "repeats" in function progenyClust is greater than 1.

`xlab` character string specifying the name of the x axis.

`ylab` character string specifying the name of the y axis.

`...` additional graphical arguments in `plot`(...).

Details

The plot function provides two types of visualization: (1) visualizing stability scores, and (2) visualizing clustering results. To visualize the stability scores that are output from `progenyClust` function, please run the plot function without specifying the input argument `data`. The resulting plot visualizes the stability score at each cluster number. This plot can provide an overview of clustering stability, and can facilitate selecting the optimal cluster number.

The plot function can also visualize the clustering results in scatter plots by specifying the input argument `data`. Since the goal is to view how the original data is clustered with certain cluster number, `data` needs to contain exactly the same number of samples as in the original data that was used to run the `progenyClust` function. If `data` contains more than two features, a table of scatter plots will be created to show clustering results within each pair of dimensions. `data` with more than 20 features/columns will not be accepted, but a subset of `data` with selected features can be used in this case. The input argument `k` specifies the cluster number at which the clustering result is shown. Note that `k` needs to be a cluster number that was previously examined by `progenyClust` when generating the `progenyClust` object `x`. If `k` is not provided, the function will use the optimal cluster number determined by the Gap criterion only if `method='gap'`, and will use the optimal number determined by the Score criterion if `method='gap'` or `method='both'` when running `progenyClust`.

Value

returns plots as described in Details.

Author(s)

C.W. Hu, Rice University

References

Hu, C.W., et al. "Progeny Clustering: A Method to Identify Biological Phenotypes." Scientific reports 5 (2015).
http://www.nature.com/articles/srep12894

Examples

```
# a 3-cluster 2-dimensional example dataset
data('test')

# default progeny clsutering
progenyClust(test,ncluster=2:5)->pc

# plot the scores to select the optimal cluster number
plot(pc)

# plot the clustering results with the optimal cluster number
plot(pc,test)
```

---

progenyClust *Progeny Clustering*

Description

Select the optimal number for clustering using Progeny Clustering.

Usage

```
progenyClust(data, FUNclust = kmeans, method = "gap", score.invert = F, ncluster = 2:10,
size = 10, iteration = 100, repeats = 1, nrandom = 10, ...)

## S3 method for class 'progenyClust'
summary(object,...)
```

Arguments

`data`	data matrix or data frame for clustering: each row correpsonds to a sample or observation, whereas each column corresponds to a feature or variable.
`FUNclust`	clustering function: accepts data as its first argument and the number for clustering as the second argument; returns a list containing a component called 'cluster' which is a vector of integers recording the clustering assignment for all samples. The default function is kmeans.

`method` character string indicating the criterion used to pick the optimal cluster number.
’gap’: the default value, selecting the cluster number that has the biggest or smallest (when score.invert=TRUE) gap from its neighboring numbrs. The optimal cluster number is picked based on the input data only, and is not compared against any random datasets, thus is quick to compute. Note that this method does not evaluate the minimum and maximum cluster numbers.
’score’: selects the cluster number that has the highest or lowest (when score.invert=TRUE) score when comparing against scores generated from random datasets. Due to the repeats on progeny clustering on random datasets, this method is slower to compute.
’both’: uses and outputs results from both the ’gap’ and ’score’ criteria.

`score.invert` logical flag: specifies whether the score should be inverted. The default score is the ratio of true classification probabilities over false classification probilities. The inverted score is the ratio of false classification over true classification probilities, which can prevent the algorithm from generating infinite score values in cases of perfect clustering. When score.invert=TRUE, the optimla cluster number is picked based on the lowest score.

`ncluster` sequence of integers specifying candidate cluster numbers for evaluation: ncluster needs to be continuous if the method ’gap’ is chosen.

`size` integer specifying the number of progenies generated from each cluster. Default value is 10.

`iteration` integer specifying the number of times the algorithm samples progenies and evalutes similarity among progenies. Default value is 100.

`repeats` integer specifying the number of times the algorithm should be run: needs to be greater than 0. Values greater than 1 output standard deviations of the scores, which are plotted as error bars in print(...,errorbar=T,...) function. Default value is 1.

`nrandom` integer specifying the number of random datasets used to generate reference scores when using method ’score’. Default value is 10.

`object` the S3 object of class "progenyClust".

`...` additional arguments for FUNclust in progenyClust(...).

Value

progenyClust returns an object of class "progenyClust" which has a plot and summary method. It is a list with the following components:

`cluster` matrix of clustering memberships for all samples under given cluster numbers: each row corresponds to a sample; each column corresponds to a given cluster number.

`score` matrix of stability scores from clustering the input data under given cluster numbers: each column corresponds to a given cluster number; each row corresponds to a repeat, the number of which is defined by ’repeats’ in the input argument.

`random.score` matrix of stability scores from clustering random datasets under given cluster numbers: each column corresponds to a given cluster number; each row corresponds to a random dataset, the number of which is defined by ’nrandom’ in the input argument.

`random.score` matrix of stability scores from clustering random datasets under given cluster numbers: each column corresponds to a given cluster number; each row corresponds to a random dataset, the number of which is defined by ’nrandom’ in the input argument.

`mean.gap` vector of mean stability scores based on the ’gap’ criterion when the input argument ’method’ is set to be ’gap’ or ’both’.

`mean.score` vector of mean stability scores based on the ’score’ criterion when the input argument ’method’ is set to be ’score’ or ’both’.

`sd.gap` vector of standard deviations of stability scores for each given cluster number based on the ’gap’ criterion, when the input argument ’method’ is set to be ’gap’ or ’both’.

`sd.score` vector of standard deviations of stability scores for each given cluster number based on the ’score’ criterion, when the input argument ’method’ is set to be ’score’ or ’both’.

`call` the call with arguments specified.

`ncluster` the specified value of input argument ’ncluster’.

`method` the specified value of input argument ’method’.

`score.invert` the specified value of input argument ’score.invert’.

Author(s)

C.W. Hu, Rice University

References

Hu, C.W., et al. "Progeny Clustering: A Method to Identify Biological Phenotypes." Scientific reports 5 (2015).
http://www.nature.com/articles/srep12894

Examples

```
# a 3-cluster 2-dimensional example dataset
data('test')

# default progeny clsutering
progenyClust(test,ncluster=2:5)->pc

summary(pc)
plot(pc)
```

`test` *3-cluster 2-dimensional test dataset*

Description

This test dataset contains 3 clusters centered around (-1,2),(2,0) and (-1,-2) in a 2-dimensional space. Each cluster consists of 50 samples that were drawn from bivariate normal distributions with a common identity covariance matrix.

Usage

```
data("test")
```

Format

A data frame with 150 observations on the following 2 variables.

V1 numeric vector of coordinates in x axis

V2 numeric vector of coordinates in y axis

References

Hu, C.W., et al. "Progeny Clustering: A Method to Identify Biological Phenotypes." Scientific reports 5 (2015).
http://www.nature.com/articles/srep12894

Examples

```
data(test)
```

Index