Package 'ReorderCluster'

June 21, 2022

Type Package

Title Reordering the Dendrogram According to the Class Labels

Version 2.0

Date 2022-06-18

Author Natalia Novoselova,Frank Klawonn, Junxi Wang

Maintainer Natalia Novoselova <novos65@mail.ru>

Description Tools for performing the leaf reordering for the dendrogram that preserves the hierarchical clustering result and at the same time tries to group instances from the same class together.

License GPL (>= 3)

LinkingTo Rcpp

Depends R (>= 2.13.0),gtools,gplots,graphics,Rcpp (>= 0.12.1)

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-06-21 13:50:05 UTC

R topics documented:

`ReorderCluster-package` *optimal reordering of the hierarchical tree according to class labels*

Description

The package includes the functions to make the near optimal reordering of the leaves of dendrogram according to the class labels, preserving its initial structure. The reordering algorithm tries to group instances from the same class together. With such an optimised dendrogram it is easier to interpret the interrelation between the available class labels and clusters. Moreover, it is possible to see whether one can predict the class for unlabeled instances based on the distance that was used for clustering.

Details

Package:	ReorderCluster
Type:	Package
Version:	1.0
Date:	2014-07-21
License:	GPL (>= 3)

The basic unit of ReorderCluster package is the function `RearrangeJoseph`, which makes the initialization of all the auxiliary matrices and the sequence of the necessary functions' call to perform the leaf reordering according to class labels. The function `RearrangeJoseph` implements the preprocessing of the merging matrix of the hierarchical clustering by calling the auxiliary function `testBar` to form the node (subtree) structure using the merging matrix, the auxiliary functions `CalcMerge` to mark the nodes with identical class labels, the auxiliary functions `SubTree` to simplify the initial hierarchical tree by constructing the modified tree, where the subtrees with equal class labels are merged into the single element. The function `RearrangeJoseph` calls the function `OrderingJoseph` to execute the dynamic programming algorithm, which calculates the values of the evaluation function for each subtree of the dendrogram and thereafter finds the optimized sequence of leaves using the function `funMerge`. The evaluation function is calculated on the basis of the existing class labels.

Functions

`RearrangeJoseph`	Makes the calls to sequence of functions to perform the leaf reordering according to class labels.
`OrderingJoseph`	Calculates the evaluation function for each subtree using the dynamic programming approach.
`OrderingJosephC`	Calculates the evaluation function for each subtree using the dynamic programming approach calling the
`RearrangeData`	Provides the main scheme to perform optimal reordering of the hierarchical tree according to class labels
`funMerge`	Recovers the optimal sequence of leaves of the hierarchical tree using the backtracking scheme.
`CalcMerge`	Forms the auxiliary vectors to mark the nodes with identical class labels.
`SubTree`	Simplifies the initial hierarchical tree by reducing the number of nodes.
`testBar`	For each inner node of the dendrogram forms two vectors with left and write subtrees' elements.

colorDendClass	Makes the plot of the dendrogram, visualizing the class label information with different colors of dendro
testData1	Generates the simulated dataset with 400 genes and 100 experiments with 3 class labels.
testData2	Generates the simulated dataset with 90 genes and 90 experiments with 3 class labels.

Installing and using

To install this package, make sure you are connected to the internet and issue the following command in the R prompt:

```
install.packages("ReorderCluster")
```

To load the package in R:

```
library(ReorderCluster)
```

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Z. Bar-Joseph, D.K. Gifford, and T.S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. Bioinformatics, 17:22-29, 2001. Therese Biedl,Brona Brejova, Erik D,Demaine, Angele M.Hanmel and Tomas Vinar:Optimal Arrangement of Leaves in the Tree Representing Hierarchical Clustering of Gene Expression Data/Technical report 2001-14.

See Also

CRAN packages **cba**, **seriation** includes the function for optimal leaf reordering of the dendrogram with respect to the minimizing the sum of the distances along the (Hamiltonian) path connecting the leaves in the given order.

Examples

```
data(leukemia)
cpp=FALSE
rownames(leukemia)=leukemia[,1]
leukemia=leukemia[,-1]
matr=leukemia[,-101]
class=leukemia[,101]

matr=as.matrix(matr)
dd=dim(matr)
```

```
label=unique(class)

Rowcolor=rainbow(length(label))
rc=matrix(0,length(class),1)
ds=matrix(0,length(class),1)

for (j in 1:length(label))
   {
index=which(class==label[j])
rc[index]=Rowcolor[j]
}

dist=dist(matr)
hc <- hclust(dist)

dend=as.dendrogram(hc)

my_palette <- colorRampPalette(c("green", "black", "red"))(n = 399)
hv <- heatmap.2(matr,Rowv=dend,scale = "none",Colv=NA,
                col=my_palette, RowSideColors = rc,trace="none",dendrogram="row")
  legend("topright",legend=label,col=Rowcolor,pch=15,cex=0.8)

  res=RearrangeJoseph(hc,as.matrix(dist),class,cpp)

hcl=res$hcl

  dend=as.dendrogram(hcl)

hv <- heatmap.2(matr,Rowv=dend,scale = "none",Colv=NA,
                col=my_palette, RowSideColors = rc,trace="none",dendrogram="row")
  legend("topright",legend=label,col=Rowcolor,pch=15,cex=0.8)
```

`CalcMerge` *Forms the binary vector to mark the nodes with identical class labels.*

Description

This is the auxiliary function of the package. It generates the binary vector to mark the nodes with identical class labels.

Usage

```
CalcMerge(hc,node,class)
```

Arguments

`hc`	An object of class hclust which describes the tree produced by the clustering process.There are such components: merge, height, order, labels,call,method,dist.method.

node	a list of lists of length n-1 testBar, where each single list stores two vectors, consisting of elements of the left and right subtrees of the corresponding node in the merging matrix hc
class	a vector of length n, which stores the class labels of the dataset objects.

Details

This is the auxiliary function of the package. It generates the binary vector to mark the nodes with identical class labels. The node or subtree with the elements, having the same class labela are marked by value 1, otherwise 0. The output vector is used as input to function SubTree, which simplifies the hierarchical tree in order to increase the efficiency of the dynamic programming algorithm.

Value

flag	a binory vector of length n-1.

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Therese Biedl,Brona Brejova, Erik D,Demaine, Angele M.Hanmel and Tomas Vinar:Optimal Arrangement of Leaves in the Tree Representing Hierarchical Clustering of Gene Expression Data/Technical report 2001-14

See Also

RearrangeJoseph, OrderingJoseph

Examples

```
data(leukemia)
rownames(leukemia)=leukemia[,1]
leukemia=leukemia[,-1]
matr=leukemia[,-101]
class=leukemia[,101]

matr=as.matrix(matr)
dist=dist(matr)
hc <- hclust(dist)
node=testBar(hc)
flag=CalcMerge(hc,node,class)
```

colorDendClass *Makes the plot of the dendrogram, visualizing the class label information with different colors of dendrogram edges.*

Description

Makes the plot of the dendrogram, visualizing the class label information with different colors of dendrogram edges.

Usage

```
colorDendClass(dend, class)
```

Arguments

dend	the clustering dendrogram of the analyzed dataset.
class	a vector of length n, which stores the class labels of the dataset objects.

Details

Makes the plot of the dendrogram, visualizing the class label information with different colors of dendrogram edges.

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Z. Bar-Joseph, D.K. Gifford, and T.S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. Bioinformatics, 17:22-29, 2001.

See Also

RearrangeJoseph, OrderingJoseph

Examples

```
data(leukemia)
rownames(leukemia)=leukemia[,1]
leukemia=leukemia[,-1]
matr=leukemia[,-101]
class=leukemia[,101]

matr=as.matrix(matr)
dist=dist(matr)
hc <- hclust(dist)
```

```
dend=as.dendrogram(hc)

label=unique(class)

Rowcolor=rainbow(length(label))
rc=matrix(0,length(class),1)

for (j in 1:length(label))
{
index=which(class==label[j])
rc[index]=Rowcolor[j]
}

colorDendClass(dend,rc[hc$order])
```

---

funMerge *Recover the optimal sequence of leaves in the hierarchical tree according to available class labels.*

Description

This is the auxiliary function of the package. It recovers the optimal sequence of leaves in the tree. Recovering is made using a backtracking scheme with the help of matrices maxI and maxJ, which store the codes of the intermediate elements, providing the best value of the evaluation function for each subtree. The sequence corresponds to the optimised leaf ordering according to evaluation function. During the recovering process for some nodes the coressponding subtrees are switched and the modified merge matrix is formed. It returns a "matrix" object, which presents the modified merge matrix with some nodes with switched subtrees. The function output is used in the function RearrangeJoseph.

Usage

```
funMerge(ind,row,col,hc,node,maxI,maxJ,cpp)
```

Arguments

ind	the value of the last row of the merge matrix, which is equal to n-1, where n is the number of data objects.
row	the leftmost element of the root tree T. It is defined as a row of the maximal element of the matrix A, holding the values of evaluation function for each subtree of the hierarchical tree.
col	the rightmost element of the root tree T. It is defined as a column of the maximal element of the matrix A, holding the values of evaluation function for each subtree of the hierarchical tree.
hc	An object of class hclust which describes the tree produced by the clustering process.There are such components: merge, height, order, labels,call,method,dist.method.

node — a list of lists of length n-1 testBar, where each single list stores two vectors, consisting of elements of the left and right subtrees of the corresponding node in the merging matrix hc

maxI — a nxn numeric matrix. Each cell (i,j) of the matrix maxI stores the code of the intermediate element i1, that provides the best value A(i,j) of the evaluation function for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element i1 is the rightmost element of the left subtree of this node(i,j).

maxJ — a nxn numeric matrix. Each cell (i,j) of the matrix maxJ stores the code of the intermediate element j1, that provides the best value A(i,j) of the evaluation function for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element j1 is the leftmost element of the right subtree of this node(i,j).

cpp — a binary variable, which allows to switch between dynamic programming algorithm, realized in R language OrderingJoseph and the same algorithm as the C++ function OrderingJosephC

Details

the function recovers the optimal sequence of leaves of the hierarchicla tree. The leftmost and rightmost elements of the root tree are used as the inputs function parameters. Than with the help of the matrices maxI, maxJ, the sequence of leaves which gives the best function value and corresponds to the optimised leaf ordering can be recovered using a backtracking scheme.

Value

hc — modified (n-1)x2 merge matrix with some nodes switched in the process of backtracking

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Therese Biedl,Brona Brejova, Erik D,Demaine, Angele M.Hanmel and Tomas Vinar:Optimal Arrangement of Leaves in the Tree Representing Hierarchical Clustering of Gene Expression Data/Technical report 2001-14

See Also

RearrangeJoseph, OrderingJoseph

Examples

```
data(leukemia)
rownames(leukemia)=leukemia[,1]
leukemia=leukemia[,-1]
```

```
matr=leukemia[,-101]
class=leukemia[,101]

matr=as.matrix(matr)
dist=dist(matr)
hc <- hclust(dist)

coef=1.5
num=dim(hc$merge)[1]
A=array(1,c(num+1,num+1))
r=array(1,c(num+1,num+1))

maxI=array(0,c(num+1,num+1))
maxJ=array(0,c(num+1,num+1))
ind=num

node=testBar(hc)
flag=CalcMerge(hc,node,class)
## change matrix hc$merge
hO<-hc
nodeO<-node
out=SubTree(ind,flag,node,hc,A,r,coef)
hc=out$hc
node=out$node
A=out$A
r=out$r

res=OrderingJosephC(ind-1, hc$merge, node, A, r, maxI, maxJ, class, coef)
A=res$A
maxI=res$maxI
maxJ=res$maxJ
r=res$r

col=which.max(apply(A[node[[ind]]$left,node[[ind]]$right,drop=FALSE],2,max))
row=which.max(apply(A[node[[ind]]$left,node[[ind]]$right,drop=FALSE],1,max))
col=node[[ind]]$right[col]
row=node[[ind]]$left[row]

hcl=funMerge(ind,row,col,hO,node,maxI,maxJ,TRUE)
```

`leukemia` *Real biological dataset to perform the analysis.*

Description

The Leukemia dataset includes the bone marrow samples obtained from acute leukemia patients at the time of diagnosis: 25 acute myeloid leukemia (AML) samples and 47 acute lymphoblastic leukemia (ALL) (24 acute 9 T-lineage acute lymphoblastic leukemia samples; and 38 B-lineage ALL samples), The most informative 100 genes were selected to proceed with reordering algorithm. The dataset can be downloaded from the website http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi.The last column stores the class labels for dataset objects.

Usage

```
data(leukemia)
```

Format

A data frame with 72 observations on the 102 variables.

Source

Molecular classification of Cancer:class discovery and class prediction by gene expression monitoring / Golub T.R., Slonim D.K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J.P., Coller H., Loh M.L., Downing J.R., Caligiuri M.A., et al. // Nature. – Vol. 286, 1999, P.531-537.

Examples

```
data(leukemia)
```

---

leukemia72 *Real biological dataset to perform the analysis.*

Description

The Leukemia dataset includes the bone marrow samples obtained from acute leukemia patients at the time of diagnosis: 25 acute myeloid leukemia (AML) samples and 47 acute lymphoblastic leukemia (ALL) (24 acute 9 T-lineage acute lymphoblastic leukemia samples; and 38 B-lineage ALL samples), The most informative 100 genes were selected to proceed with reordering algorithm. The dataset can be downloaded from the website http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi.The last column stores the class labels for dataset objects.

Usage

```
data(leukemia72)
```

Format

A data frame with 72 observations on the 102 variables.

Source

Molecular classification of Cancer:class discovery and class prediction by gene expression monitoring / Golub T.R., Slonim D.K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J.P., Coller H., Loh M.L., Downing J.R., Caligiuri M.A., et al. // Nature. – Vol. 286, 1999, P.531-537.

Examples

```
data(leukemia72)
```

`OrderingJoseph` *Makes the calculation of the evaluation function for each subtree of the hierarchical tree using the dynamic programming approach*

Description

The function realizes the dynamic programming approach in order to find the optimal reordering of the leaves of the hierarchical tree. The optimal reordering is made according to the available class labels. As an output the four auxiliary matrices A, R, maxI, maxJ are returned to the `RearrangeJoseph` function. The cells of the A matrix store the values of the evaluation function for each subtree of the hierarchical tree.

Usage

```
OrderingJoseph(ind, hc, node, A, r, maxI, maxJ, class, coef)
```

Arguments

`ind`	the value of the last row of the merge matrix, which is equal to n-1, where n is the number of data objects.
`hc`	An object of class hclust which describes the tree produced by the clustering process.There are such components: merge, height, order, labels,call,method,dist.method.
`node`	a list of lists of length n-1 `testBar`, where each single list stores two vectors, consisting of elements of the left and right subtrees of the corresponding node in the merging matrix `hcmatr`
`A`	a numerical nxn matrix, which stores the values of the evaluation function fir the subtrees. Each cell (i,j) of the matrix A stores the best ordering (according to the evaluation function value) of the subtree starting with element i and ending with element j.
`r`	a numerical nxn matrix, each cell stores the number of the elements with the same class label, starting from the leftmost or the rightmost element of the subtree.
`maxI`	a nxn numeric matrix. Each cell (i,j) of the matrix maxI stores the code of the intermediate element i1, that provides the best value A(i,j) of the evaluation function for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element i1 is the rightmost element of the left subtree of this node(i,j).
`maxJ`	a nxn numeric matrix. Each cell (i,j) of the matrix maxJ stores the code of the intermediate element j1, that provides the best value A(i,j) of the evaluation function for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element j1 is the leftmost element of the right subtree of this node(i,j).
`class`	a numerical or character vector, containing the class labels of the dataset objects.
`coef`	a parameter used in the expression for the calculation of the evaluation function. It reinforces the influence of longer sequences. Its values are in the interval]1,2].

Details

The function performs the main part of the reordering process, forming the matrix A of values of the evaluation function for each subtree of the initially formed hierarchical tree. At the beginning all cells of the matrix A and r are equal 1, than the best ordering is calculated for the nodes, consisting of two elements. At each following iteration the best ordering of the more complex node is estimated on the basis of the best selected orderings received for its left and right subtrees. The process stops when the root node is reached. At each stage of this bottom up recursive process of computing the optimised function value for each inner node we updata the marices maxI,maxJ in order to store the intermediate leaves that provide the optimized value for each pair of (i,j) elements of this node. To evaluate the complex node not only the function values for its left and right subtrees are taken into account but also the coincidence of the class labels of the rightmost element of the left subtree and the leftmost element of the write subtree. When the class labels are equal the resulting value for the node is not simply the sum of the values of two its subtrees but is recalculated using the values of the auxiliary matrix r. Matrix r with dimensions nxn stores for each subtree the number of elements with the same class label and are not interrupted by elements of other class labels starting from leftmost or the rightmost element of the subtree.

Value

`A` a numerical nxn matrix, which stores the values of the evaluation function fir the subtrees. Each cell (i,j) of the matrix A stores the best ordering (according to the evaluation function value) of the subtree starting with element i and ending with element j.

`maxI` a nxn numeric matrix. Each cell (i,j) of the matrix maxI stores the code of the intermediate element i1, that provides the best value A(i,j) of the evaluation function for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element i1 is the rightmost element of the left subtree of this node(i,j).

`maxJ` a nxn numeric matrix. Each cell (i,j) of the matrix maxJ stores the code of the intermediate element j1, that provides the best value A(i,j) of the evaluation function for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element j1 is the leftmost element of the right subtree of this node(i,j).

`r` a numerical nxn matrix, each cell stores the number of the elements with the same class label, starting from the leftmost or the rightmost element of the subtree.

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Therese Biedl,Brona Brejova, Erik D,Demaine, Angele M.Hanmel and Tomas Vinar:Optimal Arrangement of Leaves in the Tree Representing Hierarchical Clustering of Gene Expression Data/Technical report 2001-14

See Also

RearrangeJoseph, funMerge

Examples

```
data(leukemia)
rownames(leukemia)=leukemia[,1]
leukemia=leukemia[,-1]
matr=leukemia[,-101]
class=leukemia[,101]

matr=as.matrix(matr)
dist=dist(matr)
hc <- hclust(dist)

coef=1.5
num=dim(hc$merge)[1]
A=array(1,c(num+1,num+1))
r=array(1,c(num+1,num+1))

maxI=array(0,c(num+1,num+1))
maxJ=array(0,c(num+1,num+1))
ind=num

node=testBar(hc)
flag=CalcMerge(hc,node,class)
## change matrix hc$merge
hO<-hc
nodeO<-node
out=SubTree(ind,flag,node,hc,A,r,coef)
hc=out$hc
node=out$node
A=out$A
r=out$r

res=OrderingJoseph(ind, hc, node, A, r, maxI, maxJ, class, coef)
```

OrderingJosephC — *Makes the calculation of the evaluation function for each subtree of the hierarchical tree using the dynamic programming approach (C++ version)*

Description

The function realizes the dynamic programming approach in order to find the optimal reordering of the leaves of the hierarchical tree. The optimal reordering is made according to the available class labels. As an output the four auxiliary matrices A, R, maxI, maxJ are returned to the RearrangeJoseph function. The cells of the A matrix store the values of the evaluation function for each subtree of the hierarchical tree.

Usage

```
OrderingJosephC(ind, hc, node, A, r, maxI, maxJ, nclass, coef)
```

Arguments

ind	the value of the last row of the merge matrix, which is equal to n-1, where n is the number of data objects.
hc	An object of class hclust which describes the tree produced by the clustering process.There are such components: merge, height, order, labels,call,method,dist.method.
node	a list of lists of length n-1 testBar, where each single list stores two vectors, consisting of elements of the left and right subtrees of the corresponding node in the merging matrix hcmatr
A	a numerical nxn matrix, which stores the values of the evaluation function fir the subtrees. Each cell (i,j) of the matrix A stores the best ordering (according to the evaluation function value) of the subtree starting with element i and ending with element j.
r	a numerical nxn matrix, each cell stores the number of the elements with the same class label, starting from the leftmost or the rightmost element of the subtree.
maxI	a nxn numeric matrix. Each cell (i,j) of the matrix maxI stores the code of the intermediate element i1, that provides the best value A(i,j) of the evaluation function for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element i1 is the rightmost element of the left subtree of this node(i,j).
maxJ	a nxn numeric matrix. Each cell (i,j) of the matrix maxJ stores the code of the intermediate element j1, that provides the best value A(i,j) of the evaluation function for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element j1 is the leftmost element of the right subtree of this node(i,j).
nclass	a numerical or character vector, containing the class labels of the dataset objects.
coef	a parameter used in the expression for the calculation of the evaluation function. It reinforces the influence of longer sequences. Its values are in the interval]1,2].

Details

The function performs the main part of the reordering process, forming the matrix A of values of the evaluation function for each subtree of the initially formed hierarchical tree. At the beginning all cells of the matrix A and R are equal 1, than the best ordering is calculated for the nodes, consisting of two elements. At each following iteration the best ordering of the more complex node is estimated on the basis of the best selected orderings received for its left and right subtrees. The process stops when the root node is reached. At each stage of this bottom up recursive process of computing the optimised function value for each inner node we updata the marices maxI,maxJ in order to store the intermediate leaves that provide the optimized value for each pair of (i,j) elements of this node. To evaluate the complex node not only the function values for its left and right subtrees are taken into account but also the coincidence of the class labels of the rightmost element of the left

subtree and the leftmost element of the write subtree. When the class labels are equal the resulting value for the node is not simply the sum of the values of two its subtrees but is recalculated using the values of the auxiliary matrix r. Matrix r with dimensions nxn stores for each subtree the number of elements with the same class label and are not interrupted by elements of other class labels starting from leftmost or the rightmost element of the subtree.

Value

A	a numerical nxn matrix, which stores the values of the evaluation function fir the subtrees. Each cell (i,j) of the matrix A stores the best ordering (according to the evaluation function value) of the subtree starting with element i and ending with element j.
maxI	a nxn numeric matrix. Each cell (i,j) of the matrix maxI stores the code of the intermediate element i1, that provides the best value A(i,j) of the evaluation function for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element i1 is the rightmost element of the left subtree of this node(i,j).
maxJ	a nxn numeric matrix. Each cell (i,j) of the matrix maxJ stores the code of the intermediate element j1, that provides the best value A(i,j) of the evaluation function for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element j1 is the leftmost element of the right subtree of this node(i,j).
r	a numerical nxn matrix, each cell stores the number of the elements with the same class label, starting from the leftmost or the rightmost element of the subtree.

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Therese Biedl,Brona Brejova, Erik D,Demaine, Angele M.Hanmel and Tomas Vinar:Optimal Arrangement of Leaves in the Tree Representing Hierarchical Clustering of Gene Expression Data/Technical report 2001-14

See Also

RearrangeJoseph, funMerge

Examples

```
data(leukemia)
rownames(leukemia)=leukemia[,1]
leukemia=leukemia[,-1]
matr=leukemia[,-101]
nclass=leukemia[,101]
```

```
matr=as.matrix(matr)
dis=dist(matr)
hc <- hclust(dis)

coef=1.5
num=dim(hc$merge)[1]
A=array(1,c(num+1,num+1))
r=array(1,c(num+1,num+1))

maxI=array(0,c(num+1,num+1))
maxJ=array(0,c(num+1,num+1))
ind=num

node=testBar(hc)
flag=CalcMerge(hc,node,nclass)
## change matrix hc$merge
hO<-hc
nodeO<-node
out=SubTree(ind,flag,node,hc,A,r,coef)
hc=out$hc
node=out$node
A=out$A
r=out$r

res=OrderingJosephC(ind-1, hc$merge, node, A, r, maxI, maxJ, nclass, coef)
```

`RearrangeData` *Sample function to perform optimal reordering of the hierarchical tree according to class labels*

Description

Provides the main scheme to perform optimal reordering of the hierarchical tree according to class labels for any dataset

Usage

```
RearrangeData(matr,class,clustering.method="complete",cDend=FALSE,dirpath="",cpp=TRUE)
```

Arguments

`matr`	a dataset, a matrix of feature values for several cases.
`class`	a character vector or numeric vector, containing the class labels of the dataset objects
`clustering.method`	a method for performing the hierarchical clustering. The method names corresponds to the methods, that are used for the `hclust` function. The default value is "complete".

cDend	A binary variable with values TRUE/FALSE enables or desable the visualiztion of the color dendrogram. Default is FALSE.
dirpath	A path to the directory, where the heat map is saved. When the path is empry (default) the heat map is visualized on the display.
cpp	a binary variable, which allows to switch between dynamic programming algorithm, realized in R language OrderingJoseph and the same algorithm as the C++ function OrderingJosephC

Details

The function visualizes the heat map for the results of the hiererchical clustering of the input dataset, than the call to the function RearrangeJoseph enables the reordering of the dendrogram according to the supplied class labels. At the end the resultant dendrogram and the heat map are visualized.

Value

order	a numeric vector of the reordered leaves of the debdrogram.
A	a numerical nxn matrix, which stores the values of the evaluation function for the subtrees. Each cell (i,j) of the matrix A stores the best ordering (according to the evaluation function value) of the subtree starting with element i and ending with element j.
class	a vector of labels of the dataset objects, reordered according to the vector order.

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Z. Bar-Joseph, D.K. Gifford, and T.S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. Bioinformatics, 17:22-29, 2001.

See Also

RearrangeJoseph, OrderingJoseph

Examples

```
data(leukemia)
rownames(leukemia)=leukemia[,1]
leukemia=leukemia[,-1]
matr=leukemia[,-101]
class=leukemia[,101]
matr=as.matrix(matr)
res=RearrangeData(matr,class)
```

RearrangeJoseph — *Makes the initialization of auxiliary matrices and calls to sequence of functions to perform the reordering of the elements (leaves) of the hierarchical tree according to class labels of the data objects.*

Description

The function makes the preprocessing of the hierarchical merging matrix in order to simplify the hierarchical tree. For each node of the hierarchical tree forms two vectors, consisting of elements of its left and write subtrees. The function initialize the four auxiliary matrices, which are necessary for the main dynamic programming algorithm `OrderingJoseph` for reordering the leaves of the hierarchical tree according to available class labels. The function calls the main function codeOrderingJoseph and than make use of the A, maxI and maxJ matrices to recive the optimal sequence of the leaves in the tree by calling the function `funMerge`. It outputs the vector of row indices of the merge matrix, which coressponds to the nodes with switched subtrees. The merge matrix with switched subtrees present the optimal reordering of the hierarchical tree.

Usage

```
RearrangeJoseph(hc, dis,class,cpp)
```

Arguments

hc	An object of class hclust which describes the tree produced by the clustering process.There are such components: merge, height, order, labels,call,method,dist.method.
dis	an n by n distance matrix, each (i,j) element corresponds to the Euclidean distance between ith and jth data object.
class	a vector of length n, which stores the class labels of the dataset objects.
cpp	a binary variable, which allows to switch between dynamic programming algorithm, realized in R language `OrderingJoseph` and the same algorithm as the C++ function `OrderingJosephC`

Details

This function initializes four auxiliary matrices A, r , maxI and maxJ. Matrix A of dimension nxn stores the results of the calculation of the evaluation function values for each subtree of the hierarchical tree. Each cell (i,j) of the matrix A stores the best ordering (according to the evaluation function value) of the subtree starting with element i and ending with element j. The algorithms use the dynamic programming concept and works, analyzing the hierarchical tree from the bottom to the top (root node). At the beginning all cells of the matrix A and r are equal 1, than the best ordering is calculated for the nodes, consisting of two elements. At each following iteration the best ordering of the more complex node is estimated on the basis of the best selected orderings received for its left and right subtrees. The process stops when the root node is reached. At each stage of this bottom up recursive process of computing the optimised function value for each inner node we updata the marices maxI,maxJ in order to store the intermediate leaves that provide the optimized value for each pair of (i,j) elements of this node. To evaluate the complex node not only the function

values for its left and right subtrees are taken into account but also the coincidence of the class labels of the rightmost element of the left subtree and the leftmost element of the write subtree. When the class labels are equal the resulting value for the node is not simply the sum of the values of two its subtrees but is recalculated using the values of the auxiliary matrix r. Matrix r with dimensions nxn stores for each subtree the number of elements with the same class label starting from leftmost or the rightmost element of the subtree.

Each cell (i,j) of the matrix maxI stores the code of the intermediate element i1, that provides the best evaluation value A(i,j) for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element i1 is the rightmost element of the left subtree of this node(i,j). Each cell (i,j) of the matrix maxJ stores the code of the intermediate element j1, that provides the best evaluation value A(i,j) for the subtree or node(i,j), which has element i as the leftmost and element j as the rightmost. Element j1 is the leftmost element of the right subtree of this node(i,j). After performing the reordering procedure `OrderingJoseph` using the dynamic programming concept the recovering of the sequence of leaves which gives the best evaluation function value is performed `funMerge`. At first we find out the leftmost and rightmost elements of the root node or the whole tree T. After that the sequence of leaves is recovered using a backtracking scheme with the help of matrices maxI and maxJ, which store the codes of the intermediate elements, providing the best value of the evaluation function for each subtree. The sequence corresponds to the optimized leaf ordering according to evaluation function. As a result the optimized merge matrix is formed, with some nodes, having their subtrees switched.

Value

`hcl`	A modified merging matrix of dimension nx2 which describes the tree produced optimal reordering of the tree leaves.
`A`	An auxiliary matrix A of dimension nxn, which stores the results of the calculation of the evaluation function values for each subtree of the hierarchical tree.
`max`	An maximal element of the matrix A.

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Therese Biedl,Brona Brejova, Erik D,Demaine, Angele M.Hanmel and Tomas Vinar:Optimal Arrangement of Leaves in the Tree Representing Hierarchical Clustering of Gene Expression Data/Technical report 2001-14

See Also

`OrderingJoseph`, `OrderingJosephC`, `funMerge`

Examples

```
data(leukemia)
rownames(leukemia)=leukemia[,1]
leukemia=leukemia[,-1]
matr=leukemia[,-101]
class=leukemia[,101]

matr=as.matrix(matr)
dist=dist(matr)
hc <- hclust(dist)

res<-RearrangeJoseph(hc,as.matrix(dist),class,TRUE)
```

SubTree — *Simplifies the initial hierarchical tree by reducing the number of nodes. Constructs the new merging matrix with some inner nodes substituted by one element from the coressponding subtree.*

Description

This is the auxiliary function of the package. It creates the four-column matrix for reordering, that contains the results of merging the subtrees of the dendrogram with identical class labels into single elements.The result is the simplifies tree, which allows to spare the computational time and to increase the efficiency of the reordering algorithm `OrderingJoseph`.

Usage

```
SubTree(ind,flag,node,hc,A,r,coef)
```

Arguments

`ind`	the value of the last row of the merge matrix, which is equal to n-1, where n is the number of data objects.
`flag`	a binory vector of length n-1.
`node`	a list of lists of length n-1 `testBar`, where each single list stores two vectors, consisting of elements of the left and right subtrees of the corresponding node in the merging matrix `hc`
`hc`	An object of class hclust which describes the tree produced by the clustering process.There are such components: merge, height, order, labels,call,method,dist.method.
`A`	a numerical nxn matrix, which stores the values of the evaluation function fir the subtrees. Each cell (i,j) of the matrix A stores the best ordering (according to the evaluation function value) of the subtree starting with element i and ending with element j.
`r`	a numerical nxn matrix, each cell stores the number of the elements with the same class label, starting from the leftmost or the rightmost element of the subtree.

coef — a parameter used in the expression for the calculation of the evaluation function. It reinforces the influence of longer sequences. Its values are in the interval]1,2].

Details

This function modifies the merging matrix of hc, by merging the subtrees, which have the same elements' class labels. As a result the n-1 by 2 matrix is formed with the columns coressponding to nodes of the simplified hierarchical tree. Using this merging matrix allows to rise the time efficiency of the reordering algorithm (see OrderingJoseph and RearrangeJoseph).

Value

hc — An object of class hclust which include the modified merging matrix hc$merge, with the subtrees with identical class label merged into a single element (the leftmost element of the corresponding subtree)

node — the modified input structure node (a list of lists of length n-1), which corresponds to the modified hc

A — the modified nxn matrix A, which corresponds to the modified hc.

r — a modified nxn matrix r, which corresponds to the modified hc.

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Z. Bar-Joseph, D.K. Gifford, and T.S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. Bioinformatics, 17:22-29, 2001.

See Also

RearrangeJoseph, OrderingJoseph

Examples

```
data(leukemia)
rownames(leukemia)=leukemia[,1]
leukemia=leukemia[,-1]
matr=leukemia[,-101]
class=leukemia[,101]

matr=as.matrix(matr)
dist=dist(matr)
hc <- hclust(dist)

coef=1.5
num=dim(hc$merge)[1]
A=array(1,c(num+1,num+1))
```

```
r=array(1,c(num+1,num+1))

ind=num
node=testBar(hc)
flag=CalcMerge(hc,node,class)
out=SubTree(ind,flag,node,hc,A,r,coef)
```

---

testBar *For each node (subtree) of the hierarchical tree forms two vectors, consisting of elements of its left and write subtrees.*

Description

This is the auxiliary function of the package. It makes the lists of two vector, consisting of elements of left and write subtrees for each node of the hierarchical tree. These lists are used in reordering algorithm OrderingJoseph, so that it makes easy to define the first(leftmost) and last(rightmost) element of each subtree.

Usage

```
testBar(hc)
```

Arguments

hc	An object of class hclust which describes the tree produced by the clustering process.There are such components: merge, height, order, labels,call,method,dist.method.

Details

For each node or subtree of the hierarchical clustering result the function forms the list of two vector, consisting of elements of its left and write subtrees. These lists are used in reordering algorithm OrderingJoseph, so that it makes easy to define the first(leftmost) and last(rightmost) element of each subtree, and also to determine the intermediate elements in the process of searching the best ordering for each particular subtree.

Value

node	a list of lists of length n-1, where each single list stores two vectors, consisting of elements of the left and right subtrees of the corresponding node in the merging matrix hc

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Z. Bar-Joseph, D.K. Gifford, and T.S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. Bioinformatics, 17:22-29, 2001.

See Also

CalcMerge, SubTree

Examples

```
data(leukemia)
rownames(leukemia)=leukemia[,1]
leukemia=leukemia[,-1]
matr=leukemia[,-101]
class=leukemia[,101]

matr=as.matrix(matr)
dist=dist(matr)
hc <- hclust(dist)
node=testBar(hc)
```

---

testData1 *Simulates the dataset for analysis.*

Description

Generates the simulated dataset with 400 genes and 100 experiments with 3 class labels.

Usage

```
testData1()
```

Details

Generates the simulated dataset with 400 genes and 100 experiments with 3 class labels. The last column stores the class labels for dataset objects.

Value

matr	a dataset, a matrix of feature values for several cases. The additional last column stores the values of class labels.

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Z. Bar-Joseph, D.K. Gifford, and T.S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. Bioinformatics, 17:22-29, 2001.

See Also

RearrangeJoseph, OrderingJoseph

Examples

```
matr=testData1()
```

`testData2` *Simulates the dataset for analysis.*

Description

Generates the simulated dataset with 90 genes and 90 experiments with 3 class labels.

Usage

```
testData2()
```

Details

Generates the simulated dataset with 90 genes and 90 experiments with 3 class labels. The last column stores the class labels for dataset objects.

Value

`matr`	a dataset, a matrix of feature values for several cases. The additional last column stores the values of class labels.

Author(s)

Natalia Novoselova,Frank Klawonn,Junxi wang

Maintainer: Natalia Novoselova <novos65@mail.ru>

References

Z. Bar-Joseph, D.K. Gifford, and T.S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. Bioinformatics, 17:22-29, 2001.

See Also

RearrangeJoseph, OrderingJoseph

Examples

```
matr=testData2()
```

Index