Package 'TBEST'

February 19, 2015

Type Package

Title Tree Branches Evaluated Statistically for Tightness

Version 5.0

Date 2014-10-30

Author Guoli Sun, Alex Krasnitz

Maintainer Guoli Sun <guolisun87@gmail.com>

Description Our method introduces mathematically well-defined measures for tightness of branches in a hierarchical tree. Statistical significance of the findings is determined, for all branches of the tree, by performing permutation tests, optionally with generalized Pareto p-value estimation.

License GPL-2

Depends parallel, signal, fdrtool

NeedsCompilation no

Repository CRAN

Date/Publication 2014-12-29 17:56:43

R topics documented:

`best` *An object of class "best"*

Description

Description: This object is a list of three items. It contains a statistical assessment of the tightness of branches in a hierarchical tree.

Value

`Call`	An object of class Call, specifying the parameters used.
`data`	A matrix from which the distance matrix used for growing the tree is computed, with the rows corresponding to the items being clustered.
`indextable`	If measure of tightness is not `"slb"`, this is a matrix with the number of rows one less than the number of items being clustered. Each row corresponds to an internal node in the tree. The columns are as follows. First two columns specify the merging order of the tree, as in the `merge` component of the class `hclust`. The third column contains the node heights, as in the `height` component of `hclust`. The fourth column provides the number of leaves for each node. The corresponding column names are `"index1"`, `"index2"`, `"height"`, `"clustersize"`. The remaining columns come in pairs. If the name of the first column in a pair is `"x"`, the name of the second one is `"px"`. The first column in each pair tabulates a mesure of tightness; the second column provides the corresponding p-value. If measure of tightness is `"slb"`, this is a list with two variable, a matrix like above except without column of p-value and a p-value suggesting the significance of two-way split of input data.

Author(s)

Guoli Sun, Alex Krasnitz

See Also

`SigTree`,`plot.best`

Examples

```
## Not run:
data(leukemia)
mytable<-SigTree(data.matrix(leukemia),mystat="all",
       mymethod="ward",mymetric="euclidean",rand.fun="shuffle.column",
       distrib="Rparallel",njobs=2,Ptail=TRUE,tailmethod="ML")
class(mytable)
names(mytable)
mytable<-SigTree(data.matrix(leukemia),mystat="slb",
       mymethod="ward",mymetric="euclidean",rand.fun="shuffle.column",
       distrib="Rparallel",njobs=2,Ptail=FALSE)
class(mytable)
```

```
names(mytable)

## End(Not run)
```

LeafContent *Find names of leaves belongigng to given branches of a hierarchical tree*

Description

Description: find the names of all items comprising one or more branches of a hierarchical tree.

Usage

```
LeafContent(myinput, mynode=NA)
```

Arguments

myinput	An object of class hclust, best or partition.
mynode	An integer vector of the numbers of branches whose leaf content is desired. The hclust convention is used for numbering branches and leaves, i.e., the branch numbers can take any value between (-N) and (N-1) excluding 0, where N is the number of leaves in the tree. A negative value refers to an individual leaf whose number is minus that value. If myinput is of class partition, this argument is ignored. The function lists the leaf content for each of the branches that form the partition.

Value

A list of items, of the same length as mynode. Each item corresponds to a branch listed in myneode and is a character vector containing the names of the leaves in the branch.

Author(s)

Guoli Sun, Alex Krasnitz

Examples

```
data(leukemia)
hc<-hclust(dist(data.matrix(leukemia)),"ward")
#find the name of leaf 29
LeafContent(hc,mynode=c(-29))
#find the name of leaf 29 and leaves belonging to node 29
LeafContent(hc,mynode=c(-29,29))
## Not run:
mytable<-SigTree(data.matrix(leukemia),mystat="fldc",
        mymethod="ward",mymetric="euclidean",rand.fun="shuffle.column",
        distrib="Rparallel",njobs=2,Ptail=TRUE,tailmethod="ML")
LeafContent(mytable,mynode=c(-29,29))
```

```
mypartition<-PartitionTree(x=mytable,siglevel=0.001,statname="fldc",sigtype="raw")
LeafContent(mypartition)

## End(Not run)
```

`leukemia` *Leukemia data*

Description

This data set represents mRNA expression of 500 genes in 38 patient cases of leukemia. These 38 cases fall into 3 subtypes: AML (11), T-lineage ALL (8) and B-lineage ALL (19). The set was obtained by removing 499 genes from Golub's leukemia data, to facilitate the execution of examples for this package.

Usage

```
data(leukemia)
```

Format

A data frame with 38 observations (rows) of 500 variables (columns).

Details

Bone marrow samples obtained from acute leukemia patients at the time of diagnosis.

Source

http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi

References

T.R. Golub, D.K. Slonim et al(1999) Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression;

Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub(2003) Consensus Clustering: A resampling-based method for class discovery and visualization of gene expression microarray data

Examples

```
data(leukemia)
dim(leukemia)
```

partition *An object of class "partition"*

Description

Description: This object is a list of four items, which jointly specify a detailed partition of a hierarchical tree into tight branches.

Value

`Call`	An object of class Call, specifying the function call which generated the list.
`best`	An object of class "best", see `best` for more info.
`sigvalue`	A two-column matrix, with one row per each internal node of the tree. The first column enumerates the nodes. The second column profides the significance estimate for the tightness of the node.
`partition`	A two-column data frame specifying the partitition. The first column is a character vector with the names of the leaves. The second column provides the number of the part to which the leaf belongs.

Author(s)

Guoli Sun, Alex Krasnitz

See Also

`PartitionTree`,`best`,`SigTree`

Examples

```
## Not run:
data(leukemia)
mytable<-SigTree(data.matrix(leukemia),mystat="all",
       mymethod="ward",mymetric="euclidean",rand.fun="shuffle.column",
       distrib="Rparallel",njobs=2,Ptail=TRUE,tailmethod="ML")
class(mytable)
mypartition<-PartitionTree(x=mytable,siglevel=0.001,statname="fldc",
       sigtype="raw")
class(mypartition)
names(mypartition)

## End(Not run)
```

PartitionTree *Find the most detailed partition of a tree into tight branches.*

Description

Description: The function finds the most detailed partition of a hierartchical tree into tight branches, given a level of significance for tightness.

Usage

```
PartitionTree(x,siglevel=0.05,statname="fldc",
        sigtype=c("raw","corrected","fdr"))
```

Arguments

x	An object of class best, such as computed by function SigTree.
siglevel	Threshold of significance for tightness of branches. Default is 0.05.
statname	A character string specifying the name of measure of tighness whose is significance is to be used for partition. The choices are "fldc"(default),"bldc","fldcc".
sigtype	A character string specifying how the significance threshold siglevel should be interpreted. If "raw", the threshold will be applied directly to the p-values tabulated for each tree node in x. With "corrected" chosen, the threshold will be applied to the p-values corrected for multiplicity: p_cor = 1 - (1 - p)^(N - 2), where N is the number of leaves in the tree. of significance. If "fdr", siglevel is interpreted as a threshold on false discovery rate.

Value

An object of class partition. See ?partition for details.

Author(s)

Guoli Sun, Alex Krasnitz

See Also

SigTree,partition,best

Examples

```
## Not run:
data(leukemia)
mytable<-SigTree(data.matrix(leukemia),mystat="all",
        mymethod="ward",mymetric="euclidean",rand.fun="shuffle.column",
        distrib="Rparallel",njobs=2,Ptail=TRUE,tailmethod="ML")
mypartition<-PartitionTree(x=mytable,siglevel=0.001,statname="fldc",
```

```
        sigtype="raw")
partition1<-mypartition$partition
sigmatrix1<-mypartition$sigvalue
fix(partition1)
fix(sigmatrix1)

## End(Not run)
```

`plot.best` *Plot a dendrogam of a hierarchical cluster with branches labeled by their numbers and significance estimates of tightness.*

Description

Description: A plot method for the class `best`.

Usage

```
## S3 method for class 'best'
plot(x,mystat="fldc",siglevel=0.05,sigtype=c("raw","corrected","fdr"),
        partition=NA,print.num=TRUE,print.lab=TRUE,float=0.01,col.best=c(2,3),
        cex.best=0.8,cex.leaf=0.8,font.best=NULL,main=NULL,sub=NULL,xlab=NULL,
        metric.args=list(),...)
```

Arguments

`x`	An object of class `best`, such as computed by the `SigTree` function.
`mystat`	A measure of tightness for which p-values are to be shown in the plot. Default is `"fldc"`. Other options are `"fldcc"` and `"bldc"`.
`siglevel`	A threshold level of significance for tightness of branches used when `partition=NA`. Default is 0.05. If the estimate of significance for a node is below threshold, it will be shown on the plot next to the node.
`sigtype`	A character string specifying how the significance threshold `siglevel` should be interpreted. If `"raw"`, the threshold will be applied directly to the p-values tabulated for each tree node in `x`. With `"corrected"` chosen, the threshold will be applied to the p-values corrected for multiplicity: `p_cor = 1 - (1 - p)^(N - 2)`, where `N` is the number of leaves in the tree. of significance. If `"fdr"`, `siglevel` is interpreted as a threshold on false discovery rate.
`partition`	An object of class `partition`, such as computed by the `PartitionTree` function.
`print.num`	Logical. If true, the branch numbers will be indicated.
`print.lab`	Logical. If true, the labels will be displayed at the bottom of dendrogram.
`float`	A numeric value that can change the vertical location of pvalues.
`col.best`	A character vector of length 2, indicating the colors to be used for the p-values and for the numbers of the nodes.

cex.best	A numeric value for the text size of the branch labels.
cex.leaf	A numeric value for the text size of the leaf labels.
font.best	An integer which specifies font choice of text on the plot. See ?par function parameter font for details.
main	A character string specifying the title of the plot.
sub	A character string specifying a subtitle of the plot.
xlab	A caracter string specifying the label of horizontal axis.
metric.args	Additional argument from user supplied dissimilariity(distance) function. See details and examples below for further explanation.
...	Further arguments to be passed on to the plot function.

Details

The function plots a dendrogram of the hierarchical tree as specified by the x argument, an object of class "best". When argument partition is set to an object of class "partition", and a partition does exist (see partition for description), this plot provides the significance estimates for the nodes that form the partition. Otherwise, this function puts legends on all tight nodes with significance estimates no more than siglevel. To obtain the leaves descending from a given node, refer to function LeafContent.

Value

A plot with all branch numbers and significant pvalues in the hierarchical tree.

Author(s)

Guoli Sun, Alex Krasnitz

See Also

SigTree, PartitionTree,best,partition

Examples

```
## Not run:
data(leukemia)
mytable<-SigTree(data.matrix(leukemia),mystat="all",
        mymethod="ward",mymetric="euclidean",rand.fun="shuffle.column",
        distrib="Rparallel",njobs=2,Ptail=TRUE,tailmethod="ML")
plot(x=mytable,mystat="fldc",siglevel=0.001,sigtype="raw",hang=-1)
mypartition<-PartitionTree(x=mytable,siglevel=0.001,statname="fldc",
        sigtype="raw")
plot(x=mytable,mystat="fldc",partition=mypartition)
plot(x=mytable,mystat="fldc",partition=mypartition,print.num=F)
#with user-defined functions
mydist<-function(x,y){return(dist(x)/y)}
myrand<-function(x,z){return(apply(x+z,2,sample))}
mytable<-SigTree(data.matrix(leukemia),mystat="fldc",
mymethod="ward",mymetric="mydist",rand.fun="myrand",
```

```
distrib="Rparallel",njobs=2,Ptail=TRUE,tailmethod="MOM",metric.args=list(3),
rand.args=list(2))
plot(mytable,metric.args=list(3))
plot(mytable,metric.args=list(3),cex.leaf=1.5)

## End(Not run)
```

SigTree *Perform statistical analysis of tightness for branches of a hierarchical cluster.*

Description

Description: Given data from which a hierarchical tree is grown, compute measures of tightness for each branch, sample from the null distribution of these measures in the randomized data and compute the corresponding p-values.

Usage

```
SigTree(myinput,mystat=c("all","fldc","bldc","fldcc","slb"),
        mymethod="complete",mymetric="euclidean",rand.fun=NA,
        by.block=NA,distrib=c("vanilla","Rparallel"),Ptail=TRUE,
        tailmethod=c("ML","MOM"),njobs=1,seed=NA,
        Nperm=ifelse(Ptail,1000,1000*nrow(myinput)),
        metric.args=list(),rand.args=list())
```

Arguments

myinput	A matrix with rows corresponding to items to be clustered.
mystat	A character string specifying the measures of tightness to be computed and evaluated for significance of finding. See Details for the definitions of these measures. If "all" is chosen, all the first three measures, "fldc", "bldc" and "fldcc", and the corresponding p-values are computed. Otherwise, only the specified measure and its p-value are computed.
mymethod	A character string specifying the linkage method for hierarchical clustering, to be used by the hclust function. See hclust argument method for method options.
mymetric	A character string specifying the definition of dissimilarity (distance) among the data items. The options, in addition to those for the argument method of the dist functiton, are "pearson","kendall", and "spearman". If one of the latter three is chosen, the distances are computed as as.dist(1 - cor(myinput)), with the corresponding option for the method argument of the cor function.It can also be a character string specifying a user supplied dissimilarity (distance) function for myinput. See details and examples below for further explanation.

rand.fun	A character string specifying the permutation method to be applied to myinput. If NA(default), no permutation is performed. "shuffle.column" performs a random permutation independently within each column. With "shuffle.block", a random permutation is performed independently within each block of columns, as specified by the by.block argument, and independently from the other blocks. It can also be a character string specifying a user supplied randomization function for myinput. See details and examples below for further explanation.
by.block	A vector of the same length as the column dimension of myinput, to specify the blocking of columns of myinput. It is used in conjunction with rand.fun = "shuffle.block", and is ignored otherwise.
distrib	One of "vanilla", "Rparallel" to specify the distributed computing option for the cluster assignment step. For "vanilla" (default) no distributed computing is performed. For "Rparallel" the parallel package of R core is used for multi-core processing.
Ptail	Logical. If Ptail is TRUE(default), the Generalized Pareto Distribution is used to approximate the tail of the null distribution for each of the chosen measures. Otherwise, empirical p-values are computed directly from the corresponding samples.
tailmethod	A character string only needed to be specified if the Ptail is set to TRUE. For "ML" the parameters of the Generalized Pareto Distribution are estimated by likelihood maximization; for "MOM" they are estimated by the method of moments.
njobs	A single integer specifying the number of worker jobs to create in case of distributed computation if distrib = "Rparallel"; ignored otherwise.
seed	An optional single integer value, to be used to set the random number generator seed (see details).
Nperm	A single integer specifying the size of a sample from the null distribution. See details for the default sample size.
metric.args	Additional arguments for user-supplied dissimilarity (distance) function. See details and examples below for further explanation.
rand.args	Additional arguments for user-supplied randomization function. See details and examples below for further explanation.

Details

When rand.fun is set to the name of a user supplied randomization function, the first argument of that function should be set to myinput. See examples below.

The measures of tightness are defined as follows. Denote a node in the tree by a, its sibling node by b, and their parent node by p. Let their respective geights be ha,hb,hp. Finally, let Sx mean that the measure S is computed for the node x. Then the definitions are

fldc:

```
Sa = (hp-ha)/hp
```

fldcc:

```
Sa = (hp-(ha-hb)/2)/ha
```

bldc:

`Sp = (2*hp-ha-hb)/(2*hp)`

slb:

`Sp = 2*hp-ha-hb`

The first three measures test tightnss of all internal nodes at the same time, while `slb` only tests two-way split of input data. The `seed` argument is optional. Setting the seed ensures reproducibility of sampling from the null distribution.

Value

If `rand.fun` is set to NA, the function returns a matrix whose rows correspond to the internal nodes of the tree and whose columns contain the tree structure as in the `merge` component of the class `hclust`; the `height` component of `hclust`;and columns tabulating the values of the measures of tightness specified by the `mystat` argument. If `rand.fun` is set to a specific randomization method, an object of class `best` is returned. See `?best` for details.

Note

If `mymetric` or `rand.fun` is a customized function, make sure you have read and write permission for your working directory.

Author(s)

Guoli Sun, Alex Krasnitz

References

Theo A. Knijnenburg, Lodewyk F. A. Wessels et al (2009) Fewer permutations, more accurate P-values

See Also

`best`,`plot.best`

Examples

```
####Each column is a gene expression profile for a case of leukemia.
####Each case belongs to one of three subtypes.
data(leukemia)
#output only statistic table
mytable<-SigTree(data.matrix(leukemia),mystat="all",
        mymethod="ward",mymetric="euclidean")
class(mytable)
## Not run:
#use multicore processing to detect significant sub-clusters
mytable<-SigTree(data.matrix(leukemia),mystat="all",
mymethod="ward",mymetric="euclidean",rand.fun="shuffle.column",
distrib="Rparallel",njobs=2,Ptail=TRUE,tailmethod="ML")
class(mytable)
####Each row after the 1st describes an item belonging to one of four subtypes.
```

```
####Each column corresponds to a genomic location in one of 22 human chromosomes.
####The 1st row contains the chromosome numbers.
data(T10)
#Perform randomization within each chromosome
chrom<-as.numeric(T10[1,])
mydata<-T10[-1,]
mytable<-SigTree(data.matrix(mydata),mystat="fldc",
mymethod="ward",mymetric="euclidean",rand.fun="shuffle.block",
by.block=chrom,distrib="Rparallel",njobs=2,Ptail=TRUE,tailmethod="ML")
#Compute dissimilarity using a user-supplied distance function,
#and perform randomization using a user-supplied randomization function,
#with additional arguments.
#Both user-supplied functions are only useful as illustration.
mydist<-function(x,y){return(dist(x)/y)}
myrand<-function(x,z){return(apply(x+z,2,sample))}
mytable<-SigTree(data.matrix(leukemia),mystat="fldc",
mymethod="ward",mymetric="mydist",rand.fun="myrand",
distrib="Rparallel",njobs=2,Ptail=TRUE,tailmethod="MOM",metric.args=list(3),
rand.args=list(2))

## End(Not run)
```

T10 *Breast tumor single cells data*

Description

This data set summarizes DNA copy number variation in 100 individual cancer cells harvested from a breast tumor. The cells belong to four subtypes, differing by ploidy.There are 47 Diploid+Pseudo-diploid, 24 Hypo-diploid, 4 Aneuploid B and 25 Aneuploid A cells. Their copy number profiles are summarized in terms of 354 amplification and deletion "cores", are computed by the CORE package.

Usage

```
data(T10)
```

Format

A data frame with 101 rows and 354 columns. Each column corresponds to a core. The first row is integer and contains the chromosome number for each core. The remaining rows are numeric, with values between 0 and 1, and each represents a DNA copy number profile of a cell.

Details

Please remove the first row before computing the distance matrix.

Source

Alexander Krasnitz, Guoli Sun, Peter Andrews, and Michael Wigler(2013) Target inference from collections of genomic intervals

References

Alexander Krasnitz, Guoli Sun, Peter Andrews, and Michael Wigler(2013) Target inference from collections of genomic intervals

Examples

```
data(T10)
dim(T10)
```

Index