

# Package ‘erhcv’

December 10, 2018

**Type** Package

**Title** Equi-Rank Hierarchical Clustering Validation

**Version** 0.1.4

**License** GPL (>= 2)

**Author** Simon-Pierre Gadoury <spgadou@me.com>

**Maintainer** Simon-Pierre Gadoury <spgadou@me.com>

**Description** Assesses the statistical significance of clusters for a given dataset through bootstrapping and hypothesis testing of a given matrix of empirical Spearman's rho, based on the technique of S. Gaiser et al. (2010) <doi:10.1016/j.jmva.2010.07.008>.

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Depends** R (>= 3.5.0)

**Suggests** HAC, knitr, rmarkdown, nCopula

**Imports** stringr, stringi, utils, Rdpack, data.tree

**RdMacros** Rdpack

**Collate** 'VerifyTree.R' 'ClusterNodeSelection.R' 'EliminateCluster.R'  
'GetLeaves.R' 'GetPairs.R' 'compareTrees.R' 'hclust2tree.R'  
'tree2plot.R'

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-12-10 16:00:03 UTC

## R topics documented:

ClusterNodeSelection . . . . .	2
CompareTrees . . . . .	3
EliminateCluster . . . . .	4
GetLeaves . . . . .	4
GetPairs . . . . .	5

hclust2tree . . . . .	5
tree2plot . . . . .	6
VerifyTree . . . . .	6

<b>Index</b>	<b>8</b>
--------------	----------

---

ClusterNodeSelection    *Subcluster significance test*

---

## Description

Test the significance of a subcluster directly under a given cluster. If it is not significant, the subcluster is eliminated and its element are merged in the main cluster. Bootstrap samples of the underlying data needs to be provided, as well as the structure under consideration.

## Usage

```
ClusterNodeSelection(cluster, testPos, alpha, data, BootData)
```

## Arguments

cluster	the main cluster (of the form provided by the function hclust2tree)
testPos	the position of the subcluster to test, directly under the main cluster
alpha	the confidence level for the tests
data	the underlying data
BootData	the dataframe of bootstrap samples of Spearman rho, with columns named "(i,j)", where "i" and "j" are different leaves

## Details

The hypothesis testing is made with the matrix of Spearman's rho for a given dataset, see (Gaiser and Schmid 2010).

## Value

The main cluster, with or without the node under test, whether the hypothesis can be rejected or not.

## Author(s)

Simon-Pierre Gadoury

## References

Gaiser S, Schmid F (2010). "On testing equality of pairwise rank correlations in a multivariate random vector." *Journal of Multivariate Analysis*, **101**(10), 2598–2615.

---

CompareTrees	<i>Compare two tree structures</i>
--------------	------------------------------------

---

### Description

Provide a logical answer to the question "Are two trees equivalent?".

### Usage

```
CompareTrees(tree_test, tree_ref)
```

### Arguments

tree_test	(testing tree) a tree structure in the form of nested lists (as the output of <code>hclust2tree</code> , for example)
tree_ref	(reference tree) a tree structure in the form of nested lists (as the output of <code>hclust2tree</code> , for example)

### Details

This test is particularly useful when two trees are not ordered in the same way. As the structure are lists, one could argue that this function is a modification of the `identical` base function, overcoming the ordering.

### Value

boolean: TRUE implies that the two trees are equivalent.

### Author(s)

Simon-Pierre Gadoury

### Examples

```
## Comparison between "identical" and "compareTrees"
##
## The trees are "identical"

tree1 <- list(list(list(5, 6), list(7, 8), 3, 4), list(9, 10), 2, 1)
tree2 <- list(list(list(5, 6), list(7, 8), 3, 4), list(9, 10), 2, 1)

CompareTrees(tree1, tree2)
identical(tree1, tree2)

## The trees are "equivalent" (notice the leaves 1 and 2 interchanged)

tree1 <- list(list(list(5, 6), list(7, 8), 3, 4), list(9, 10), 2, 1)
tree2 <- list(list(list(5, 6), list(7, 8), 3, 4), list(9, 10), 1, 2)
```

```
CompareTrees(tree1, tree2)
identical(tree1, tree2)
```

---

EliminateCluster      *Subcluster elimination*

---

**Description**

Eliminate a subcluster and merge its elements to the main cluster.

**Usage**

```
EliminateCluster(cluster, pos)
```

**Arguments**

cluster      the cluster under consideration (of the form provided by the function hclust2tree)  
pos          the position of the subcluster, directly under the main cluster, to eliminate

**Value**

The main cluster, without the subcluster that was eliminated.

**Author(s)**

Simon-Pierre Gadoury

---

GetLeaves      *Leaves extractions*

---

**Description**

Extract the leaves (nodes in the bottom of the hierarchy) under (not necessarily directly under) a cluster.

**Usage**

```
GetLeaves(cluster)
```

**Arguments**

cluster      a cluster (of the form provided by the function hclust2tree)

**Value**

Vector of leaves.

**Author(s)**

Simon-Pierre Gadoury

---

GetPairs	<i>Obtain pairs of leaves</i>
----------	-------------------------------

---

**Description**

Find all the pairs of leaves between a specific subcluster and all the other subclusters under the main cluster. Pairs between the subclusters of the specific subcluster are also given.

**Usage**

```
GetPairs(cluster, pos)
```

**Arguments**

cluster	main cluster (of the form provided by the function <code>hclust2tree</code> )
pos	position of the subcluster directly under the main cluster. This subcluster is the one every other subcluster will be paired with.

**Value**

Matrix of pairs of leaves.

**Author(s)**

Simon-Pierre Gadoury

---

hclust2tree	<i>hclust object transformation</i>
-------------	-------------------------------------

---

**Description**

Transforms a `hclust` object into a tree (cluster) used in other functions of this package.

**Usage**

```
hclust2tree(clustering)
```

**Arguments**

clustering	<code>hclust</code> object
------------	----------------------------

**Author(s)**

Simon-Pierre Gadoury

tree2plot *Plot of a tree structure*

---

**Description**

Provide the plot or the data.tree object for a given tree structure.

**Usage**

```
tree2plot(tree, structure = FALSE)
```

**Arguments**

tree            the tree under consideration (of the form provided by the function hclust2tree)  
structure       logical. Should the data.tree structure be returned rather than the plot?

**Value**

a plot or data.tree object

**Author(s)**

Simon-Pierre Gadoury

---

VerifyTree *Verify tree structure*

---

**Description**

Given a matrix of data, where the rows are observations and the columns are variables, it verifies the statistical significance of hierarchical nodes provided by hclust, through the use of the empirical matrix of Spearman's rho.

**Usage**

```
VerifyTree(data, alpha = 0.95, nboot = 500, distance.method = "maximum",  
          hclust.method = "complete")
```

**Arguments**

data            data used for the clustering  
alpha           the confidence level for the tests  
nboot           the number of bootstrap samples to use  
distance.method            method for the distance matrix  
hclust.method    method for the clustering

## Details

The hypothesis testing, as well as the clustering, is made with the matrix of Spearman's rho for a given dataset, see (Gaisser and Schmid 2010).

## Value

A list, containing the bootstrap samples and the initial tree structure, modified, according to the results of the tests

## Author(s)

Simon-Pierre Gadoury

## References

Gaisser S, Schmid F (2010). "On testing equality of pairwise rank correlations in a multivariate random vector." *Journal of Multivariate Analysis*, **101**(10), 2598–2615.

## Examples

```
require(HAC)
str <- hac(type = 1, tree = list(list(list("X4", "X5", 6),
                                     "X6", 3), "X1", list("X2", "X3", 10), 1))

set.seed(2018)
U.. <- rHAC(1000, str)
U.. <- U..[,c(4, 5, 6, 1, 2, 3)]

## Tree via hclust
spear <- cor(U.., method = "sp")
clust <- hclust(dist(spear, method = "maximum"),
               method = "complete")
tree1 <- hclust2tree(clust)

## Tree after verification
tree2 <- VerifyTree(U.., alpha = 0.95,
                   distance.method = "maximum",
                   hclust.method = "complete")$Tree

## Comparison
par(mfrow = c(1, 3))
tree2plot(tree1)
tree2plot(tree2)
plot(str)
par(mfrow = c(1, 3))
```

# Index

ClusterNodeSelection, [2](#)

CompareTrees, [3](#)

EliminateCluster, [4](#)

GetLeaves, [4](#)

GetPairs, [5](#)

hclust2tree, [5](#)

identical, [3](#)

tree2plot, [6](#)

VerifyTree, [6](#)