Package ‘dynamicTreeCut’

September 17, 2007

Version 1.08

Date 2007-09-17

Title Dynamic Tree Cut Methods for detection of clusters in hierarchical clustering dendrograms.

Author Peter Langfelder <Peter.Langfelder@gmail.com> and Bin Zhang <binzhang.ucla@gmail.com>, with contributions from Steve Horvath <SHorvath@mednet.ucla.edu>

Maintainer Peter Langfelder <Peter.Langfelder@gmail.com>

Depends R, stats

ZipData no

License GPL version 2 or newer

Description Contains methods for detection of clusters in hierarchical clustering dendrograms.

Title Dynamic Tree Cut Methods for detection of clusters in hierarchical clustering dendrograms.

URL http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/BranchCutting/

R topics documented:

cutreeDynamic ................................................................. 2
cutreeDynamicTree .......................................................... 4
cutreeHybrid ................................................................. 5
dynamicTreeCut-package ...................................................... 7
indentSpaces ................................................................. 8
printFlush ................................................................. 9

Index 10
cutreeDynamic

Adaptive branch pruning of hierarchical clustering dendrograms.

Description

This wrapper provides a common access point for two methods of adaptive branch pruning of hierarchical clustering dendrograms.

Usage

```r
cutreeDynamic(dendro, cutHeight = NULL, minClusterSize = 20, method = "hybrid", deepSplit = (ifelse(method == "dynamic", 1, FALSE)), maxCoreScatter = NULL, minGap = NULL, maxAbsCoreScatter = NULL, minAbsGap = NULL, clusterTrim = 0, labelUnlabeled = TRUE, distM = NULL, useMedoids = FALSE, maxDistToLabel = cutHeight, respectSmallClusters = TRUE, verbose = 2, indent = 0)
```

Arguments

dendro  a hierarchical clustering dendrogram such as one returned by `hclust`.
cutHeight Maximum joining heights that will be considered.
minClusterSize Minimum cluster size.
method Chooses the method to use. Recognized values are "hybrid" and "tree".
depthSplit For method "hybrid", can be either logical or integer in the range 0 to 3. For method "tree", must be logical. In both cases, provides a rough control over sensitivity to cluster splitting. The higher the value (or if `TRUE`), the more and smaller clusters will be produced. For the "hybrid" method, a finer control can be achieved via `maxCoreScatter` and `minGap` below.
maxCoreScatter Only used for method "hybrid". Maximum scatter of the core for a branch to be a cluster, given as the fraction of `cutHeight` relative to the 5th percentile of joining heights. See Details.
minGap Only used for method "hybrid". Minimum cluster gap given as the fraction of the difference between `cutHeight` and the 5th percentile of joining heights.
maxAbsCoreScatter Only used for method "hybrid". Maximum scatter of the core for a branch to be a cluster given as absolute heights. If given, overrides `maxCoreScatter`.
minAbsGap Only used for method "hybrid". Minimum cluster gap given as absolute height difference. If given, overrides `minGap`.
**cutreeDynamic**

`cutreeDynamic` is a wrapper for two related but different methods for cluster detection in hierarchical clustering dendrograms.

In order to make the shape parameters `maxCoreScatter` and `minGap` more universal, their values are interpreted relative to `cutHeight` and the 5th percentile of the merging heights (we arbitrarily chose the 5th percentile rather than the minimum for reasons of stability). Thus, the absolute maximum allowable core scatter is calculated as `maxCoreScatter * (cutHeight - refHeight) + refHeight` and the absolute minimum allowable gap as `minGap * (cutHeight - refHeight)`, where `refHeight` is the 5th percentile of the merging heights.

**Value**

A vector of numerical labels giving assignment of objects to modules. Unassigned objects are labeled 0, the largest module has label 1, next largest 2 etc.

**Author(s)**

Peter Langfelder, [Peter.Langfelder@gmail.com](mailto:Peter.Langfelder@gmail.com)
References


See Also

hclust, cutreeHybrid, cutreeDynamicTree.

cutreeDynamicTree  Dynamic dendrogram pruning based on dendrogram only

Description

Detect clusters in a hierarchical dendrogram using a variable cut height approach. Uses only the information in the dendrogram itself is used (which may give incorrect assignment for outlying objects).

Usage

cutreeDynamicTree(dendro, maxTreeHeight = 1, deepSplit = TRUE, minModuleSize = 50)

Arguments

dendro Hierarchical clustering dendrogram such produced by hclust.

maxTreeHeight Maximum joining height of objects to be considered part of clusters.

deepSplit If TRUE, method will favor sensitivity and produce more smaller clusters. When FALSE, there will be fewer bigger clusters.

minModuleSize Minimum module size. Branches containing fewer than minModuleSize objects will be left unlabeled.

Details

A variable height branch pruning technique for dendrograms produced by hierarchical clustering. Initially, branches are cut off at the height maxTreeHeight; the resulting clusters are then examined for substructure and if subclusters are detected, they are assigned separate labels. Subclusters are detected by structure and are required to have a minimum of minModuleSize objects on them to be assigned a separate label. A rough degree of control over what it means to be a subcluster is implemented by the parameter deepSplit.

Value

A vector of numerical labels giving assignment of objects to modules. Unassigned objects are labeled 0, the largest module has label 1, next largest 2 etc.
**cutreeHybrid**

**Author(s)**

Bin Zhang, (binzhang.ucla@gmail.com), with contributions by Peter Langfelder, (Peter.Langfelder@gmail.com).

**References**

http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/BranchCutting

**See Also**

cutreeHybrid

---

**cutreeHybrid**

*Hybrid adaptive tree cut for hierarchical clustering dendrograms.*

**Description**

Detect clusters in a dendrogram produced by the function `hclust`.

**Usage**

```r
cutreeHybrid(dendro, cutHeight = NULL, minClusterSize = 20,
  deepSplit = 1,
  maxCoreScatter = NULL, minGap = NULL,
  maxAbsCoreScatter = NULL, minAbsGap = NULL, clusterTrim = 0,
  labelUnlabeled = TRUE, distM = NULL,
  useMedoids = FALSE, maxDistToLabel = cutHeight,
  respectSmallClusters = TRUE, verbose = 2, indent = 0)
```

**Arguments**

- `dendro`: a hierarchical clustering dendrogram such as one returned by `hclust`.
- `cutHeight`: Maximum joining heights that will be considered.
- `minClusterSize`: Minimum cluster size.
- `deepSplit`: Either logical or integer in the range 0 to 3. Provides a rough control over sensitivity to cluster splitting. The higher the value, the more and smaller clusters will be produced. A finer control can be achieved via `maxCoreScatter` and `minGap` below.
- `maxCoreScatter`: Maximum scatter of the core for a branch to be a cluster, given as the fraction of `cutHeight` relative to the 5th percentile of joining heights. See Details.
- `minGap`: Minimum cluster gap given as the fraction of the difference between `cutHeight` and the 5th percentile of joining heights.
- `maxAbsCoreScatter`: Maximum scatter of the core for a branch to be a cluster given as absolute heights. If given, overrides `maxCoreScatter`.
- `minAbsGap`: Minimum cluster gap given as absolute heights.
\textbf{Details}

The function detects clusters in a hierarchical dendrogram based on the shape of branches on the dendrogram. For details on the method, see \url{http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/BranchCutting}.

In order to make the shape parameters \texttt{maxCoreScatter} and \texttt{minGap} more universal, their values are interpreted relative to \texttt{cutHeight} and the 5th percentile of the merging heights (we arbitrarily chose the 5th percentile rather than the minimum for reasons of stability). Thus, the absolute maximum allowable core scatter is calculated as \texttt{maxCoreScatter} * (\texttt{cutHeight} - \texttt{refHeight}) + \texttt{refHeight} and the absolute minimum allowable gap as \texttt{minGap} * (\texttt{cutHeight} - \texttt{refHeight}), where \texttt{refHeight} is the 5th percentile of the merging heights.

\textbf{Value}

A list containing the following elements:

\begin{itemize}
  \item \texttt{labels} \hspace{1cm} Numerical labels of clusters, with 0 meaning unassigned, label 1 labeling the largest cluster etc.
  \item \texttt{cores} \hspace{1cm} Numerical labels indicating cores of found clusters.
  \item \texttt{smallLabels} \hspace{1cm} Numerical labels for branches that failed to be recognized clusters only because of insufficient number of objects.
  \item \texttt{trimmed} \hspace{1cm} Numerical labels indicating objects that have been trimmed from clusters.
  \item \texttt{branches} \hspace{1cm} A list detailing the detected branch structure.
\end{itemize}
Author(s)

Peter Langfelder, ⟨Peter.Langfelder@gmail.com⟩

References


See Also

hclust, as.dist

dynamicTreeCut-package

Methods for detection of clusters in hierarchical clustering dendrograms.

Description

Contains methods for detection of clusters in hierarchical clustering dendrograms.

Details

Package: dynamicTreeCut
Version: 1.08
Date: 2007-09-17
Depends: R, stats
ZipData: no
License: GPL version 2 or newer
URL: http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/BranchCutting/
Packaged: Mon Sep 17 12:32:38 2007; plangfelder
Built: R 2.4.1; 2007-09-17 12:48:47; unix

Index:

cutreeDynamic  Adaptive branch pruning of hierarchical clustering dendrograms.
cutreeDynamicTree Dynamic dendrogram pruning based on dendrogram only
cutreeHybrid Hybrid adaptive tree cut for hierarchical clustering dendrograms.
indentSpaces Spaces for indented output.
printFlush Print arguments and flush the console.
treecut-package Methods for detection of clusters in hierarchical clustering dendrograms.
Author(s)

Peter Langfelder <Peter.Langfelder@gmail.com> and Bin Zhang <binzhang.ucla@gmail.com>, with contributions from Steve Horvath <SHorvath@mednet.ucla.edu>

Maintainer: Peter Langfelder <Peter.Langfelder@gmail.com>

---

**indentSpaces**

Spaces for indented output.

---

Description

Returns a character string containing two times indent spaces.

Usage

```r
indentSpaces(indent = 0)
```

Arguments

- `indent` Desired level of indentation. The number of returned spaces will be twice this argument.

Value

A character string containing spaces, of length twice indent.

Author(s)

Peter Langfelder, ⟨Peter.Langfelder@gmail.com⟩

Examples

```r
spaces = indentSpaces(0);
print(paste(spaces, "This output is not indented..."));
spaces = indentSpaces(1);
print(paste(spaces, "...while this one is."))
```
printFlush

Print arguments and flush the console.

Description

Passes all its arguments unchaged to the standard `print` function; after the execution of print it flushes the console, if possible.

Usage

```r
printFlush(...)```

Arguments

... Arguments to be passed to the standard `print` function.

Details

Passes all its arguments unchaged to the standard `print` function; after the execution of print it flushes the console, if possible.

Value

Returns the value of the `print` function.

Author(s)

Peter Langfelder, ⟨Peter.Langfelder@gmail.com⟩

See Also

`print`
Index

*Topic cluster
  cutreeDynamic, 1
  cutreeHybrid, 5
*Topic misc
  cutreeDynamicTree, 4
*Topic package
  dynamicTreeCut-package, 7
*Topic print
  indentSpaces, 8
  printFlush, 8

as.dist, 7

cutreeDynamic, 1
  cutreeDynamicTree, 3, 4
  cutreeHybrid, 3, 5, 5

dynamicTreeCut
  (dynamicTreeCut-package), 7
  dynamicTreeCut-package, 7

hclust, 3, 4, 7

indentSpaces, 8

print, 9
  printFlush, 8