

Package ‘structSSI’

September 27, 2025

Type Package

Title Multiple Testing for Hypotheses with Hierarchical or Group Structure

Version 1.2.0

Description Performs multiple testing corrections that take specific structure of hypotheses into account, as described in Sankaran & Holmes (2014) [doi:10.18637/jss.v059.i13](https://doi.org/10.18637/jss.v059.i13).

License GPL-2

Imports methods, multtest, igraph, jsonlite, ggplot2, phyloseq, reshape2

Suggests ape, testthat

RoxygenNote 7.3.3

Depends R (>= 3.5)

biocViews Software

Encoding UTF-8

NeedsCompilation no

Author Kris Sankaran [aut, cre] (ORCID: <https://orcid.org/0000-0002-9415-1971>)

Maintainer Kris Sankaran <ksankaran@wisc.edu>

Repository CRAN

Date/Publication 2025-09-27 08:40:02 UTC

Contents

structSSI-package	2
Adaptive.GBH	3
chlamydiae	5
estimate.pi0	6
EstimatedHFDRCControl	7
GBH-class	8
gtemp	10

hFDR.adjust	11
hypothesesTree-class	13
Oracle.GBH	15
PlotHypTree	16
treePValues	17

Index	19
--------------	-----------

structSSI-package	<i>Structured Simultaneous and Selective Inference for Grouped or Hierarchically Structured Data</i>
-------------------	--

Description

Performs multiple corrections that take specific structure of hypotheses into account. The two procedures implemented are the Hierarchically FDR controlling procedure of Benjamini and Yekutieli and the Group Benjamini-Hochberg procedure of Hu, Zhou, and Zhao. The methods are applicable whenever information about hierarchical or group relationship between hypotheses can be ascertained before any data analysis.

Details

Package: structSSI
Type: Package
Version: 1.0
Date: 2012-03-14
License: GPL-2

This package implements two recently developed techniques in the field of selective and simultaneous inference (SSI). The first method is the Adaptive Groups Benjamini-Hochberg procedure of Hu, Zhou, and Zhao 2011. The second is the Hierarchical FDR Controlling Procedure of Benjamini and Yekutieli. Both methods attempt to employ apriori known information about the relationships between hypotheses in testing them and correcting for the multiple testing problem. These methods have been employed in genetics, QTL analysis, and clinical trials; more deatils about these applications can be read in the references stated below.

The Group Benjamini-Hochberg procedure is implemented in its adaptive and oracle varieties through the functions `Adaptive.GBH` and `Oracle.GBH`, respectively. The Hierarchical Procedure is implemented in the function `hFDR.adjust` and uses the class `hypothesesTree` to organize the information required for the procedure. These functions describe the procedures in more detail. Further, the references listed below present the derivations and applications of these two procedures.

Author(s)

Kris Sankaran
Maintainer: ksankaran@wisc.edu

References

- Benjamini, Y, and Yekutieli, D. Hierarchical fdr testing of trees of hypotheses. 2002.
- Hu, J.X., Zhao, H., and Zhou, H.H. False discovery rate control with groups. *Journal of the American Statistical Association*, volume 104, number 491. Pages 1215-1227. 2010.
- Sankaran, K and Holmes, S. structSSI: Simultaneous and Selective Inference for Grouped or Hierarchically Structured Data. *Journal of Statistical Software*, 59(13), 1-21. 2014. <https://jstatsoft.org/v59/i13/>
- Yekutieli, D. Hierarchical false discovery rate-controlling methodology. *Journal of the American Statistical Association*, 103(481):309-316, 2008.

Examples

```
## Example of using the Adaptive Benjamini-Hochberg Procedure.

set.seed(2249)
unadjp <- c(runif(40, 0, 0.001), runif(30, 0, 0.1), runif(130, 0, 1))
names(unadjp) <- paste("hyp", c(1:200))
groups <- c(sample(1:3, 200, replace = TRUE))
result <- Adaptive.GBH(unadjp, groups, method = "lsl", alpha = 0.05)

## Example of using the Hierarchical FDR controlling procedure.
library('igraph')
library('ape')

alternative.indices <- sample(1:49, 30)
unadj.p.values <- vector("numeric", length = 49)
unadj.p.values[alternative.indices] <- runif(30, 0, 0.01)
unadj.p.values[-alternative.indices] <- runif(19, 0, 1)
unadj.p.values[c(1:5)] <- runif(5, 0, 0.01)
names(unadj.p.values) <- paste("Hyp ", c(1:49))

tree <- as.igraph(rtree(25))
V(tree)$name <- names(unadj.p.values)
tree.el <- get.edgelist(tree)

hyp.tree <- hFDR.adjust(unadj.p.values, tree.el, 0.05)
plot(hyp.tree)
```

Adaptive.GBH

Adaptive Group Benjamini-Hochberg Procedure

Description

Performs the Group Benjamini-Hochberg procedure using adaptive estimates of the proportions of null p-values in given groups. The method is applicable when we know some a-priori structure about whether certain hypotheses can be grouped. Once the hypotheses are grouped and tested individually, a Benjamini-Hochberg correction is performed within each of the groups. Finally, the fact that the Benjamini-Hochberg correction controls the FDR at level $q \cdot \pi_{0_group}$ within each group

(q is the level used in the and p-value comparison and π_0 is the proportion of null hypotheses within the particular group) is used to increase the power of the procedure. The procedure is described in more detail in the paper "False Discovery Rate Control with Groups" by Hu, Zhao, and Zhou (see references below).

Usage

```
Adaptive.GBH(unadj.p, group.index, alpha = 0.05, method = "lsl", lambda = 0.5)
```

Arguments

<code>unadj.p</code>	A vector of the unadjusted p-values resulting from a multiple testing experiment.
<code>group.index</code>	A vector of the same length as the vector of unadjusted p-values, where a "G" in the jth coordinate indicates that the jth unadjusted p-values in <code>unadj.p</code> belongs to group "G". This code can be either a factor giving group names, or a numeric index.
<code>alpha</code>	The level of that we are attempting to control the FDR at.
<code>method</code>	The method for adaptively estimating the proportion of true null hypotheses within a vector of unadjusted p-values. The possible options are "storey", "lsl", and "tst". See the documentation for <code>estimatePi0</code> for more details.
<code>lambda</code>	The value of the tuning parameter for estimating the proportion of null hypotheses, in the "storey" method.

Details

Hu, J.X., Zhao, H., and Zhou, H.H. False discovery rate control with groups. *Journal of the American Statistical Association*, volume 104, number 491. Pages 1215-1227. 2010.

Sankaran, K and Holmes, S. structSSI: Simultaneous and Selective Inference for Grouped or Hierarchically Structured Data. *Journal of Statistical Software*, 59(13), 1-21. 2014. <https://jstatsoft.org/v59/i13/>

Value

An object of class GBH, which provides the adjusted p-values.

Examples

```
# These are the unadjusted p-values corresponding to the outcome of some
# multiple testing experiment. The first 500 hypotheses are null and the last
# 1500 are true alternatives.
unadjp <- c(runif(500, 0, 0.01), runif(1500, 0, 1))
names(unadjp) <- paste("Hyp:", 1:2000)

# Here there are two groups total we have randomly assigned hypotheses to
# these two groups.
group.index <- c(sample(1:2, 2000, replace = TRUE))

# Perform the GBH adjustment.
result <- Adaptive.GBH(unadjp, group.index, method = "storey")
```

```
# A summary of the GBH adjustment
summary(result)

# The estimated proportions of null hypotheses, between groups
result@pi0

# Visualizing the sorted p-values, before and after adjustment
plot(result, adjust = TRUE)
plot(result, adjust = FALSE)
```

chlamydiae*Chlamydiae Sub-Taxa of Global Patterns Data*

Description

A phyloseq class object containing data on 16s rRNA diversity within the Chlamydiae bacteria taxon, originally appearing in PNAS.

Usage

```
data(chlamydiae)
```

Format

A phyloseq object

Details

A small subtree of the GlobalPatterns dataset, originally available in the phyloseq package.

Source

<https://github.com/joey711/phyloseq/raw/c2605619682acb7167487f703d135d275fead748/data/GlobalPatterns.RData>

References

McMurdie P.J., Holmes S. (2013). phyloseq: A Bioconductor Package for Handling and Analysis of High-Throughput Phylogenetic Sequence Data. PLoS ONE 8(4): e61217. <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0061217>.

Caporaso, J. G., et al. (2011). Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample. PNAS, 108, 4516-4522. PMID: PMC3063599

This can be viewed/downloaded at: <https://www.pnas.org/content/108/suppl.1/4516.short>
Sankaran, K and Holmes, S. structSSI: Simultaneous and Selective Inference for Grouped or Hierarchically Structured Data. Journal of Statistical Software, 59(13), 1-21. 2014. <https://jstatsoft.org/v59/i13/>

Examples

```
if (requireNamespace("phyloseq", quietly = TRUE)) {
  data(chlamydiae)
}
```

estimate.pi0

Estimation of Proportion of Null Hypotheses among p-values

Description

This function makes three routines available for estimating the true proportion of null hypotheses from a vector of unadjusted p-values arising from a multiple testing experiment. The specific methods are the Least Slope method (`lsl`), the Two Step Test method (`tst`), and the Storey tail proportion of p-values method (`storey`). These methods are derived and explained in the references given below.

Usage

```
estimate.pi0(pvalues, method, alpha = 0.05, lambda = 0.5)
```

Arguments

<code>pvalues</code>	A vector of the unadjusted p-values resulting from a multiple testing experiment.
<code>method</code>	The technique used to estimate the proportion of true null hypotheses. Valid arguments are <code>lsl</code> , <code>tst</code> , or <code>storey</code> , which correspond to the Least Slope, Two Step Test, and Storey tail proportion of p-values methods, respectively.
<code>alpha</code>	In the Two Step Test method, the level of the Benjamini-Hochberg procedure used to estimate the proportion of true null hypotheses.
<code>lambda</code>	In the Storey tail proportion of p-values method, the cutoff used to differentiate p-values.

Details

The Least Slope method uses the insight that, if we plot the sorted unadjusted p-values, then the p-values corresponding to null hypotheses will have steep slopes, as they are uniformly distributed between 0 and 1. If we find the position where the slope of the sorted p-values increases the most, we can fix that slope and extrapolate to the x-axis, and the position where the line intersects the x-axis is the proportion of p-values estimated to be null. The formal derivation is presented in the reference below.

Storey's method uses the idea that most of the p-values above some parameter λ (usually set to 0.5) corresponds to null hypotheses. The number of hypotheses in this interval can then be used to estimate the number of hypotheses overall which are null hypotheses.

The Two Step Test method uses the idea that the result of a multiple comparisons procedure gives an estimate for how many hypotheses are null. That is, if we perform the BH procedure on 100 hypotheses, and 10 of them are rejected, then a reasonable estimate of the proportion of null hypotheses among those 100 is $\pi_0 = 0.9$.

- Benjamini, Y, Krieger, A.M., and Yekutieli, D. Adaptive linear step-up procedures that control the false discovery rate. *Biometrika*, 93(3):491, 2006.
- Benjamini, Y, and Hochberg, Y. “On the adaptive control of the false discovery rate in multiple testing with independent statistics.” *Journal of Educational and Behavioral Statistics*, 25(1):60, 2000.
- Sankaran, K and Holmes, S. structSSI: Simultaneous and Selective Inference for Grouped or Hierarchically Structured Data. *Journal of Statistical Software*, 59(13), 1-21. 2014. <https://jstatsoft.org/v59/i13/>
- Storey, J.D., Taylor, J.E., and Siegmund, D. Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(1):187-205. 2004.

Value

An estimate of the proportion of true null hypotheses from the result of the multiple testing experiment that the unadjusted p-values were extracted from.

Examples

```
true.p.1 <- runif(20, 0, 0.01)
null.p.1 <- runif(980, 0, 1)
p.val.1 <- c(true.p.1, null.p.1)

estimate.pi0(p.val.1, "lsl")
estimate.pi0(p.val.1, "storey", lambda = 0.2)
estimate.pi0(p.val.1, "tst")
```

EstimatedHFDRControl *Estimate FDR control variations for HFDR procedure*

Description

This function estimates two types of HFDR control appropriate for trees of hypotheses. If the BH procedure is applied at level alpha within each of the tree families, this is defined as

$$FDR.est := \alpha * [\#discoveries + \#families\ tested] / [\#discoveries + 1]$$

where a discovery is defined as an adjusted p value below alpha within the entire tree or at the tips for tree and tips FDR, respectively.

Usage

```
EstimatedHFDRControl(hyp.tree)
```

Arguments

`hyp.tree` An object of class `HypothesesTree`, such as the result to a call of `hFDR.adjust`.

Details

Yekutieli, D. Hierarchical false discovery rate-controlling methodology. *Journal of the American Statistical Association*, 103(481):309-316, 2008. Benjamini, Y, and Yekutieli, D. Hierarchical fdr testing of trees of hypotheses. 2002. Sankaran, K and Holmes, S. structSSI: Simultaneous and Selective Inference for Grouped or Hierarchically Structured Data. *Journal of Statistical Software*, 59(13), 1-21. 2014. <https://jstatsoft.org/v59/i13/>

Value

A list with the following elements,

<code>tree</code>	The estimated full-tree FDR.
<code>tip</code>	The estimated outer-nodes FDR.
<code>n.families.tested</code>	The number of families of hypotheses tested by the HFDR procedure.
<code>n.tree.discoveries</code>	The number of discoveries over the whole tree.
<code>n.tip.discoveries</code>	The number of discoveries among the tree tips.

Examples

```
library("igraph")
library("ape")

alternative.indices <- sample(1:49, 30)
unadj.p.values <- vector("numeric", length = 49)
unadj.p.values[alternative.indices] <- runif(30, 0, 0.01)
unadj.p.values[-alternative.indices] <- runif(19, 0, 1)
unadj.p.values[c(1:5)] <- runif(5, 0, 0.01)
names(unadj.p.values) <- paste("Hyp ", c(1:49))

tree <- as.igraph(rtree(25))
V(tree)$name <- names(unadj.p.values)
tree.el <- get.edgelist(tree)

hyp.tree <- hFDR.adjust(unadj.p.values, tree.el, 0.05)

EstimatedHFDRControl(hyp.tree)
```


Description

This defines a class GBH for managing outputs from the Group Benjamini-Hochberg procedure. This object makes it easy to print, summarize, and plot the results of the testing procedure.

Prints the entire table of adjusted p-values and their associated FDR adjusted significance levels, together with the estimated proportions of null hypotheses, within each group.

Shows results from multiple testing via GBH. Also supplies the estimated proportion of null hypothesis within each group and a table of counts of adjusted significance across groups.

Show results of testing hypothesis, sorted according to GBH adjusted significance, shape coded according to group membership, and color coded according to pre and post GBH p-value adjustment.

Usage

```
## S4 method for signature 'GBH'
initialize(.Object, ...)

## S4 method for signature 'GBH'
show(object)

## S4 method for signature 'GBH'
summary(object)

## S4 method for signature 'GBH,ANY'
plot(x, title = "GBH Adjustment", ...)
```

Arguments

<code>.Object</code>	Dummy to initialize S4 class
<code>...</code>	Any other arguments are accepted, but they will be ignored.
<code>object</code>	A GBH object whose hypotheses we want to summarize.
<code>x</code>	A GBH object whose p-values to plot.
<code>title</code>	The name added to the top of the plot. Defaults to 'GBH Adjustment'.

Slots

`p.vals` Object of class 'data.frame'. Each row corresponds to an individual hypothesis. The different columns correspond to, * `hypothesisIndex`: The index of the current hypothesis in the `unadjp` vector * `hypothesisName`: The name of the current hypothesis, from the names of the `unadjp` vector * `unadjp`: The unadjusted p-values input from `unadjp` * `adjp`: The adjusted p-values, after the GBH adjustment. * `group`: The group to which the original hypothesis belonged * `significance`: A code for the significance of each hypothesis

`pi0` Object of class 'numeric'. The proportion of null hypotheses within each group. This is either known a priori or estimated adaptively from the unadjusted p-values.

`adaptive` Object of class 'logical'. An indicator of whether the proportion `pi0` was estimated adaptively from the data or known a priori.

`alpha` Object of class 'numeric'. The level at which the FDR is controlled, during the GBH procedure.

Examples

```
# These are the unadjusted p-values corresponding to the outcome of some
# multiple testing experiment. The first 500 hypotheses are null and the last
# 1500 are true alternatives.

unadjp <- c(runif(500, 0, 0.01), runif(1500, 0, 1))
names(unadjp) <- paste('Hyp: ', 1:2000)

# These are the unadjusted p-values corresponding to the outcome of some
# multiple testing experiment. The first 500 hypotheses are null and the last
# 1500 are true alternatives.
unadjp <- c(runif(500, 0, 0.01), runif(1500, 0, 1))
names(unadjp) <- paste('Hyp: ', 1:2000)

# Here there are two groups total we have randomly assigned hypotheses to these
# two groups.
group.index <- c(sample(1:2, 2000, replace = TRUE))

# Perform the GBH adjustment.
result <- Adaptive.GBH(unadjp, group.index, method = 'storey')

# A summary of the GBH adjustment
summary(result)
```

gtemp

Global mean land-ocean temperature deviations

Description

Global mean land-ocean temperature deviations, measured by the Goddard Institute for Space Studies and supplied in package `astsa`, used in teaching Shumway and Stoffer's Time Series Analysis textbook.

Usage

```
data(gtemp)
```

Format

A data frame with 130 observations on the following 2 variables.

`year` a numeric vector

`temp` a numeric vector

Source

<https://data.giss.nasa.gov/gistemp/>

References

<https://www.stat.pitt.edu/stoffer/tsa3/>

Sankaran, K and Holmes, S. structSSI: Simultaneous and Selective Inference for Grouped or Hierarchically Structured Data. Journal of Statistical Software, 59(13), 1-21. 2014. <https://jstatsoft.org/v59/i13/>

Examples

```
data(gtemp)
if (interactive()) {
  plot(gtemp, type = 'l')
}
```

hFDR.adjust

FDR Controlling Procedure for Hierarchically Structured Hypotheses

Description

This function implements the Hierarchical FDR controlling procedure developed Benjamini and Yekutieli. The procedure incorporates structural information about the hierarchical relationships between hypotheses in order to increase power and interpretability of a testing procedure while controlling the False Discovery Rate at a prespecified level. It is applicable whenever there is a natural hierarchical structure between the hypotheses being tested before the data analysis begins. For example, the method has been used before in Clinical Studies, where nodes deeper in the tree correspond to secondary or tertiary endpoints. It has also been used in QTL analysis, where we first make statements about regions of chromosomes being associated with specific brain activity and then focus on more and more detailed subsets of chromosomes.

Usage

```
hFDR.adjust(unadjp, tree.el, alpha = 0.05)
```

Arguments

unadjp	A vector of raw p-values resulting from an experiment. The names of this vector should be contained in the edgelist parameterizing the hierarchical structure between hypothesis, inputted as tree.el.
tree.el	The edgelist parameterizing the hierarchical structure between hypotheses. The edges must be stored so that each edge is a row of a two column matrix, where the first column gives the parent and the second gives the child.
alpha	The level of FDR control within families of the tree. Note that this is NOT necessarily the level of FDR control within the entire tree. Refer to the paper by Yekutieli and Benjamini for bounds on various FDR criterion.

Details

The FDR controlling procedure is described in more detail in the paper by Yekutieli and Benjamini 2009. The idea is to control for multiple testing error within families of hypotheses, and only test a descendant family of hypotheses if the associated parent hypotheses was deemed significant in the higher level. The families of hypotheses are taken to be the children of any particular node, and error is controlled within these families using the Benjamini-Hochberg procedure. Different bounds can be proven for the FDR when considered along whole tree, within a single level, and tips. In particular, the whole tree FDR is typically controlled at three times the FDR control within individual families, and this result holds for arbitrary hypotheses tests and configurations of trees.

Yekutieli, D. Hierarchical false discovery rate-controlling methodology. *Journal of the American Statistical Association*, 103(481):309-316, 2008.

Benjamini, Y, and Yekutieli, D. Hierarchical fdr testing of trees of hypotheses. 2002.

Sankaran, K and Holmes, S. structSSI: Simultaneous and Selective Inference for Grouped or Hierarchically Structured Data. *Journal of Statistical Software*, 59(13), 1-21. 2014. <https://jstatsoft.org/v59/i13/>

Value

An object of class `hypothesesTree` with the hypothesis tree structure, unadjusted and adjusted p-values, significance annotations, and the FDR control level used.

Examples

```
library("igraph")
library("ape")
library("structSSI")

alternative.indices <- sample(1:49, 30)
unadj.p.values <- vector("numeric", length = 49)
unadj.p.values[alternative.indices] <- runif(30, 0, 0.01)
unadj.p.values[-alternative.indices] <- runif(19, 0, 1)
unadj.p.values[c(1:5)] <- runif(5, 0, 0.01)
names(unadj.p.values) <- paste("Hyp ", c(1:49))

tree <- as.igraph(rtree(25))
V(tree)$name <- names(unadj.p.values)
tree.el <- get.edgelist(tree)

hyp.tree <- hFDR.adjust(unadj.p.values, tree.el, 0.05)

# We can visualize the difference between the unadjusted and the adjusted
# trees.
if (interactive()) {
  plot(hyp.tree, adjust = FALSE)
  plot(hyp.tree, adjust = TRUE)
}
```

hypothesesTree-class *Manage Hierarchical Test Results*

Description

This defines a class of hypothesis trees. It should allow us to manipulate the hypotheses, the results of tests associated with them, and their hierarchical relation. We set a class to make this manipulation easier and to prevent the bugs that arise from complexity. Note that the tree is defined by its adjacency matrix.

Check that the hypotheses tree is correctly initialized. It ensures that the number of unadjusted p-values, hypotheses names, and nodes in the hypotheses tree all agree. It also checks that the hypotheses tree is in fact a tree.

This prints the unadjusted and adjusted p-values of the hypotheses tree associated with the HFDR procedure.

This prints the most significant adjusted p-values, along with estimates of the FDR across the tree and at tips.

Creates an interactive plot to understand the tests along the tree that were rejected, with their adjusted and unadjusted p-values.

Usage

```
## S4 method for signature 'hypothesesTree'
initialize(.Object, ...)

## S4 method for signature 'hypothesesTree'
show(object)

## S4 method for signature 'hypothesesTree'
summary(object)

## S4 method for signature 'hypothesesTree,ANY'
plot(
  x,
  ...,
  adjust = TRUE,
  return_script = FALSE,
  width = 900,
  height = 500,
  base_font_size = 12,
  output_file_name = paste("hyp_tree", gsub("[^\\d]+", "", Sys.time(), perl = TRUE),
    ".html", sep = "")
)
```

Arguments

.Object Dummy to initialize S4 class

...	Any other arguments are accepted, but they will be ignored.
object	A hypothesesTree object whose hypotheses we want to summarize.
x	A hypothesesTree object whose hypotheses we want to plot.
adjust	Show the adjusted p-values?
return_script	Return the d3 code used to generate the plot.
width	The width of the printed tree, in pixels.
height	The height of the printed tree, in pixels.
base_font_size	The size of the plot's labels.
output_file_name	The name of the file to which the script is saved.

Slots

tree Object of class "matrix". The edgelist for the hypotheses tree. * hypothesisIndex: The index of the current hypothesis in the unadjp vector * hypothesisName: The name of the current hypothesis, from the names of the unadjp vector * unadjp: The unadjusted p-values input from unadjp * adjp: The adjusted p-values, after the GBH adjustment. * group: The group to which the original hypothesis belonged * significance: A code for the significance of each hypothesis

p.vals Object of class 'data.frame'. Each row corresponds to an individual hypothesis. The first column stores the p-values before GBH adjustment, while the second gives the hFDR adjusted p-values. The hypotheses are sorted in order of significance according to these GBH adjusted p-values. The group column gives the group membership of each hypothesis, and adj.significance codes the significance of each hypothesis, according to the GBH adjusted p-values.

alpha Object of class "numeric". The level at which the FDR is controlled among children of each parent node.

Examples

```
library('igraph')
library('ape')

alternative.indices <- sample(1:49, 30)
unadj.p.values <- vector("numeric", length = 49)
unadj.p.values[alternative.indices] <- runif(30, 0, 0.01)
unadj.p.values[-alternative.indices] <- runif(19, 0, 1)
unadj.p.values[c(1:5)] <- runif(5, 0, 0.01)
names(unadj.p.values) <- paste("Hyp ", c(1:49))

tree <- as.igraph(rtree(25))
V(tree)$name <- names(unadj.p.values)
tree.el <- get.edgelist(tree)

hyp.tree <- hFDR.adjust(unadj.p.values, tree.el, 0.05)
if (interactive()) {
  plot(hyp.tree)
}
```

Oracle.GBH

Oracle Group Benjamini-Hochberg Correction

Description

Performs the Group Benjamini-Hochberg procedure when the true proportion of null hypotheses is known within each group. The procedure is applicable whenever group structure about the relationship between different hypotheses is known before testing begins. The idea is to control the FDR within each group and to use the proportion of null hypotheses present within each group to make the testing procedure within that group either more or less conservative – this is in fact the idea behind all adaptive multiple testing procedures.

The Oracle GBH method can also be used when we would like to use the Adaptive GBH procedure but with estimates of proportions of true null hypotheses within groups that are not made directly available through the Adaptive.GBH function – in this case those estimates can be input as the argument `pi.groups` in the this function `Oracle.GBH`.

Usage

```
Oracle.GBH(unadj.p, group.index, pi.groups, alpha = 0.05)
```

Arguments

<code>unadj.p</code>	A vector of the unadjusted p-values resulting from a multiple testing experiment.
<code>group.index</code>	A vector of the same length as the vector of unadjusted p-values, where a "G" in the jth coordinate indicates that the jth unadjusted p-values in <code>unadj.p</code> belongs to group "G". This code can be either a factor giving group names, or a numeric index.
<code>pi.groups</code>	A vector of the known proportions of true null hypotheses within each of the groups. This vector should be named so that each element of the <code>group.index</code> vector correspond to one of the names of the <code>pi.groups</code> vector.
<code>alpha</code>	The level that we are attempting to control the FDR at.

Details

Hu, J.X., Zhao, H., and Zhou, H.H. False discovery rate control with groups. *Journal of the American Statistical Association*, volume 104, number 491. Pages 1215-1227. 2010.

Sankaran, K and Holmes, S. structSSI: Simultaneous and Selective Inference for Grouped or Hierarchically Structured Data. *Journal of Statistical Software*, 59(13), 1-21. 2014. <https://jstatsoft.org/v59/i13/>

Value

An object of class GBH, which provides the adjusted p-values.

Examples

```
# A very simple example, with only 5 hypotheses.
unadjp <- c(0.002, 0.1, 0.001, 0.01, 0.4)
names(unadjp) <- paste("hyp", 1:5)
groups <- c(1, 2, 1, 2, 2)

# Say we know group 1 has pi_0,1 = 0.3 and pi_0,2 = 0.9
pi.groups <- c("1" = 0.3, "2" = 0.9)
Oracle.GBH(unadjp, groups, pi.groups)

# An example where we use an external pi0 estimation routine

unadjp.2 <- c(runif(500, 0, 0.01), runif(1500, 0, 1))
names(unadjp.2) <- paste("hyp", 1:2000)
groups.2 <- c(sample(1:2, 2000, replace = TRUE))
pi.groups <- c("1" = NA, "2" = NA)
for(i in 1:2){
  pi.groups[i] <- estimate.pi0(unadjp.2[which(groups.2 == i)], method = "storey")
}

result <- Oracle.GBH(unadjp.2, groups.2, pi.groups, 0.05)
result@pi0
result@p.vals
```

PlotHypTree

Interactively Visualize a HypothesisTree

Description

Generates a D3.js-based visualization of a HypothesisTree object. Nodes are colored by p-value, and hovering over a node displays its name and either the raw or adjusted p-value, depending on the adjust argument.

Usage

```
PlotHypTree(
  hyp.tree,
  adjust = TRUE,
  return_script = FALSE,
  width = 900,
  height = 500,
  base_font_size = 12,
  output_file_name = NULL
)
```

Arguments

`hyp.tree` An object of class HypothesisTree.

adjust	Logical; if TRUE, use adjusted p-values for coloring. If FALSE, use raw p-values.
return_script	Logical; if TRUE, return the HTML/JS script as a character string instead of writing to a file and opening in a browser.
width	Width of the plot in pixels. Default is 900.
height	Height of the plot in pixels. Default is 500.
base_font_size	Base font size for node labels. Default is 12.
output_file_name	Optional file name for the HTML output. If NULL, a name is generated automatically.

Details

The visualization displays the hierarchical structure of hypotheses. Node color reflects the p-value, and hovering over a node shows its name and p-value.

Value

If `return_script` is TRUE, returns a character string containing the HTML/JS code. Otherwise, writes the visualization to an HTML file and opens it in the default browser (if interactive).

treePValues	<i>Create tree of p-values for phyloseq data</i>
-------------	--

Description

This helper function is used to aggregate abundances of individual microbes to higher levels in the tree and test whether those aggregated abundances are significantly different between environments, using the data structures from the phyloseq package framework.

Usage

```
treePValues(tree, abundances, environments)
```

Arguments

tree	An edgelist for a tree containing the phylogenetic relationships between different microbes.
abundances	A phyloseq class OTU table specifying the abundances of different microbes across environments.
environments	A phyloseq class Sample Data object associating the different environments with variables of interest.

Value

A vector containing the p-values of the linear model predicting the abundances of microbes aggregated to different levels in the taxonomy from environmental variables.

Examples

```
library("igraph")

# Example with random data
if(require("ape")) {
  rand.tree <- as.igraph(rtree(50))
  V(rand.tree)$name <- paste("hyp", seq_along(V(rand.tree)))
  rand.tree <- get.edgelist(rand.tree)
  X <- matrix(rnorm(50 * 4), 50, 4)
  rownames(X) <- paste("hyp", 1:50)
  colnames(X) <- 1:4
  X[, 1:2] <- X[, 1:2] + 1
  groups <- factor(c("A", "A", "B", "B"))
  treePValues(rand.tree, X, groups)
}

# Example using phyloseq
if(require("phyloseq") & require("ape")) {
  data(chlamydiae)
  abundances <- otu_table(chlamydiae)
  environments <- sample_data(chlamydiae)$SampleType
  ch.tree <- get.edgelist(as.igraph(phy_tree(chlamydiae)))
  ch.pval <- treePValues(ch.tree, abundances, environments)
}
```

Index

- * **datasets**
 - gtemp, [10](#)
- * **htest**
 - structSSI-package, [2](#)
- * **package**
 - structSSI-package, [2](#)
- Adaptive.GBH, [3](#)
- chlamydiae, [5](#)
- estimate.pi0, [6](#)
- EstimatedHFDRControl, [7](#)
- GBH (GBH-class), [8](#)
- GBH, ANY (GBH-class), [8](#)
- GBH-class, [8](#)
- gtemp, [10](#)
- hFDR.adjust, [11](#)
- hypothesesTree-class, [13](#)
- initialize, GBH-method (GBH-class), [8](#)
- initialize, hypothesesTree-method
(hypothesesTree-class), [13](#)
- Oracle.GBH, [15](#)
- plot, GBH, ANY-method (GBH-class), [8](#)
- plot, hypothesesTree, ANY-method
(hypothesesTree-class), [13](#)
- PlotHypTree, [16](#)
- show, GBH-method (GBH-class), [8](#)
- show, hypothesesTree-method
(hypothesesTree-class), [13](#)
- structSSI (structSSI-package), [2](#)
- structSSI-package, [2](#)
- summary, GBH-method (GBH-class), [8](#)
- summary, hypothesesTree-method
(hypothesesTree-class), [13](#)
- treePValues, [17](#)