Package ‘BOUTH’

November 5, 2018

Type Package

Title Bottom-up Tree Hypothesis Tests

Version 0.1.0

Author Yunxiao Li, Yi-Juan Hu and Glen A. Satten

Maintainer Yunxiao Li <yunxiao.li@emory.edu>

Description Testing hypotheses that have a branching tree dependence structure in a bottom-up manner, with false discovery rate control

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0.9000

Depends R (>= 2.1.0)

Imports dplyr(>= 0.7.0), RColorBrewer(>= 1.1-2)

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`bouth` *Bottom-up Tree Hypothesis Tests*

Description

This function implements the (one-stage and two-stage) bottom-up approach to testing hypotheses that have a branching tree dependence structure, with false discovery rate control. Our motivating example comes from testing the association between a trait of interest and groups of microbes that have been organized into operational taxonomic units (OTUs) or amplicon sequence variants (ASVs). Given p-values from association tests for each individual OTU or ASV, we would like to know if we can declare that a certain species, genus, or higher taxonomic grouping can be considered to be associated with the trait. If a large proportion of species from that genus influence the trait, we should conclude the genus influences the trait. Conversely if only a few of the species from a genus are non-null, then a better description of the microbes that influence occurrence of the trait is a list of associated species. Finding taxa that can be said to influence a trait in this sense is the first goal of our approach. The second goal is to locate the highest taxa in the tree for which we can conclude many taxa below, but not any ancestors above, influence risk; we refer to such taxa as driver taxa.

Usage

```
bouth(anno.table, pvalue.leaves, na.symbol = "unknown", far = 0.1,
  tau = 0.3, is.weighted = TRUE)
```

Arguments

`anno.table` an `n` by `l` data.frame that specifies the annotation (i.e., grouping) of `n` leaf nodes at `l` levels. The `l` levels are ordered with the highest level (the root node) first and the lowest level (the leaf nodes) last, so that each row represents a path from the root node to a leaf node.

`pvalue.leaves` a vector of p-values at leaf nodes, the names of which should match the values in the last column of `anno.table`.

`na.symbol` a character string which is to be inpretreted as `NA` values. The default is 'unknown'.

`far` nominal false assignment rate (FAR), the error rate in analogy with the false discovery rate. If `far` takes one value, it corresponds to the one-stage bottom-up test with overal FAR control at the specified level. If `far` takes a vector of two values, it corresponds to the two-stage bottom-up test with the test of level-1 nodes controlling FAR at the level of the first value of `far` and the test of the remaining levels controlling FAR at the level of the second value of `far`. The default is 0.1.

`tau` a pre-specified constant to prevent nodes with large p-values from being detected if a large number (say, `m`) of null hypotheses can be easily rejected because of very low p-values, in which case `q x m` nodes with large p-values can be said to be detected, while still controlling the overall error rate at level `q`. The default is 0.3.

`is.weighted` a logical value indicating whether the weighted or unweighted test is performed. The default is 'TRUE'.

Value

A list consisting of

`results.by.level`

a data frame that gives the number of detected nodes at each level along with information on the test for that level.

results.by.node
: a data frame that gives detailed results at each node (i.e., leaf and inner nodes), including the (derived) p-value, the indicator of a detection or not, and the indicator of a driver node or not.

tree
: A tree structure used in the function graphlan.

References

Li, Y., Satten, GA., Hu Y.-J. "A bottom-up approach to testing hypotheses that have a branching tree dependence structure, with false discovery rate control" XXX(2018)

Examples

```
data(IBD)

## one-stage weighted bottom-up test on the IBD data
test.1 = bouth(anno.table = IBD$tax.table, pvalue.leaves = IBD$pvalue.otus,
na.symbol = "unknown", far = 0.1, is.weighted = TRUE)

## extract all detected nodes
test.1$results.by.node[test.1$results.by.node$is.detected, ]

## extract all detected driver nodes
test.1$results.by.node[test.1$results.by.node$is.driver,]


## two-stage (weighted) bottom-up test on the IBD data
test.2 = bouth(anno.table = IBD$tax.table, pvalue.leaves = IBD$pvalue.otus,
na.symbol = "unknown", far = c(0.05, 0.05))
```

---

graphlan *Visualizing the results of* bouth *with GraPhlAn*

Description

Create a tree figure for visualizing the results of bouth

Usage

```
graphlan(bouth.out, show.leaf = FALSE, output.dir = getwd(),
  graphlan.dir = NULL, clade.separation = NULL, branch.thickness = NULL,
  branch.bracket.depth = NULL, branch.bracket.width = NULL,
  clade.marker.size = NULL)
```

Arguments

bouth.out	an output object from the bouth function.
show.leaf	a logical value indicating whether or not the leaf nodes are shown in the output figure.
output.dir	the directory for storing the output figure. The default is the current working directory.
graphlan.dir	the directory where the GraPhlAn package is located.

Note

This function is dependent on the python package GraPhlAn. Parameters `clade.separation`, `branch.thickness`, `branch.bracket.depth`, `branch.bracket.width`, and `clade.marker.size` can be set the same as in the python package. Details can be found at (https://bitbucket.org/nsegata/graphlan/overview).

References

Asnicar, F, et al. "Compact graphical representation of phylogenetic data and metadata with GraPhlAn." PeerJ 3 (2015): e1029.

Examples

```
data(IBD)

## performing the one-stage, weighted bottom-up test on the IBD data
test.1 = bouth(anno.table = IBD$tax.table, pvalue.leaves = IBD$pvalue.otus,
na.symbol = "unknown", far = 0.1, is.weighted = TRUE)


## suppose the GraPhlAn package is located at graphlan_directory/
graphlan(bouth.out = test.1, graph.dir = graphlan_directory)
```

`IBD` *An example dataset*

Description

A real data example from a study on human gut microbiome and inflammatory bowel disease (IBD). The preprocessing was performed on the online platform QIITA (https://qiita.ucsd.edu).

Usage

```
IBD
```

Format

tax.table A 2360 by 8 data frame that gives taxomonic assignment to 2360 operational taxonomic units (OTUs) from 8 levels, which from the top to bottom are kingdom, phylum, class, order, family, genus, species, and OTUs. The value ‘unknown’ indicates missing assignment at some levels.

pvalue.otus A vector of p-values for testing the differential abundance of each of the 2360 OTUs between the ulcerative colitis (UC) and control groups.

References

Halfvarson, J, et al. "Dynamics of the human gut microbiome in inflammatory bowel disease." Nature microbiology 2.5 (2017): 17004.

Examples

```
data(IBD)
```

Index