Package 'StructFDR'

April 13, 2017

Type Package

Title False Discovery Control Procedure Integrating the Prior Structure Information

Version 1.2

Date 2017-04-12

Author Jun Chen

Maintainer Jun Chen <chen.jun2@mayo.edu>

Description Perform more powerful false discovery control (FDR) for microbiome data, taking into account the prior phylogenetic relationship among bacteria species. As a general methodology, it is applicable to any type of (genomic) data with prior structure information.

Depends R (>= 3.1.0), nlme, ape, cluster, dirmult, matrixStats

Suggests MASS, knitr, rmarkdown, ggplot2, reshape

VignetteBuilder knitr

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2017-04-13 17:53:02 UTC

R topics documented:

TreeFDR-package *False Discovery Rate (FDR) Control Procedure Integrating the Prior Structure Information*

Description

The package is designed to perform more powerful false discovery control (FDR) for microbiome data, taking into account the prior phylogenetic relationship among bacteria species. As a general methodology, it is applicable to any type of (genomic) data with prior structure information, as long as a distance metric between features are defined.

Details

Package:	TreeFDR
Type:	Package
Version:	1.0
Date:	2016-10-28
License:	GPL-2

The package contains two major function `TreeFDR` and `StructFDR`, which perform tree-based or general structure-based FDR control.

Author(s)

Jun Chen

Maintainer: Jun Chen <chen.jun2@mayo.edu>

References

Jian Xiao, Hongyuan Cao and Jun Chen (2016). False discovery rate control incorporating phylogenetic tree increases detection power in microbiome-wide multiple testing. Submitted.

Examples

```
require(ape)
require(nlme)
require(cluster)
require(StructFDR)

# Generate a caelescence tree and partition into 10 clusters
set.seed(1234)
n <- 20
p <- 200
tree <- rcoal(p)
D <- cophenetic(tree)
clustering <- pam(D, k=10)$clustering
```

```
# Simulate case-control data, assuming cluster 2 is differential
X.control <- matrix(rnorm(n*p), p, n)
X.case <- matrix(rnorm(n*p), p, n)
eff.size <- rnorm(sum(clustering == 2), 0.5, 0.2)
X.case[clustering == 2, ] <- X.case[clustering == 2, ] + eff.size
X <- cbind(X.control, X.case)
Y <- gl(2, n)

# Define testing and permutation function
test.func <- function (X, Y) {
obj <- apply(X, 1, function(x) {
ttest.obj <- t.test(x ~ Y)
c(ttest.obj$p.value, sign(ttest.obj$statistic))
})
    return(list(p.value=obj[1, ], e.sign=obj[2, ]))
}

perm.func <- function (X, Y) {
return(list(X=X, Y=sample(Y)))
}

# Call TreeFDR
tree.fdr.obj <- TreeFDR(X, Y, tree, test.func, perm.func)

# Compare TreeFDR and BH
tree.fdr.obj$p.adj
tree.fdr.obj$p.adj[clustering == 2]
BH.p.adj <- p.adjust(tree.fdr.obj$p.unadj, 'fdr')
BH.p.adj[clustering == 2]

# Adjusted statistics vs clustering
par(mfrow=c(1, 2))
plot(clustering, tree.fdr.obj$z.unadj)
plot(clustering, tree.fdr.obj$z.adj)
```

`AdjStats` *Prior Structure-Adjusted Statistic*

Description

Produce the prior structure-adjusted statistic.

Usage

```
AdjStats(y, V, k, mu, fudge=0.005)
```

Arguments

y	a vector or a matrix of unadjusted z-values
V	a correlation matrix defined based on the prior structure
k	a numeric value representing the ratio of variance components
mu	a numeric value of the mean of the prior distribution
fudge	a small numeric value added to the diagonal of the correlation matrix to improve stability

Value

a vector or a matrix of adjusted z-values

Author(s)

Jun Chen

References

Jian Xiao, Hongyuan Cao and Jun Chen (2016). False discovery rate control incorporating phylogenetic tree increases detection power in microbiome-wide multiple testing. Submitted.

---

alcohol *Alcohol data set*

Description

The data set was taken from a study of the dietary association with the gut microbiome (Wu, 2011). Here we include the alcohol intake as the primary phenotype. The OTU abundance data consists of 98 samples with 949 OTUs with prevalence > 10%. The raw counts are normalized to account for variable library sizes.

Usage

```
data(alcohol)
```

Format

The format is: chr "alcohol"

Details

A list containing the normalized OTU data (`X`, counts divided by GMPR size factors), the taxonomic lineages of the OTUs (`otu.name`), the alcohol intake phenotype (High and Low) (`Y`), the tree (`tree`).

Source

Wu, Gary D., et al. "Linking long-term dietary patterns with gut microbial enterotypes." Science 334.6052 (2011): 105-108.

Examples

```
data(alcohol)
```

`EstHyper` *Estimate the Hyper-parameter Using Generalized Least Squares*

Description

Estimate the hyper-parameter by calling the `gls` function. The errors are allowed to be correlated and/or have unequal variances.

Usage

```
EstHyper(y, D, init.val)
```

Arguments

`y`	a vector of z-values.
`D`	a distance matrix defined based on the prior structure. Diagonal have to be zeros.
`init.val`	initial values for the transformed hyper-parameter. Default is 0.

Value

a vector of estimated hyper-parameter values plus log likelihood.

Author(s)

Jun Chen

References

Jian Xiao, Hongyuan Cao and Jun Chen (2016). False discovery rate control incorporating phylogenetic tree increases detection power in microbiome-wide multiple testing. Submitted.

MicrobiomeSeqTreeFDR *False Discovery Rate (FDR) Control Integrating Prior Tree Structure for Microbiome Data Based on F-test and Residual Permutation.*

Description

The function is a wrapper of TreeFDR with a pre-defined test based on F-statistic and residual permutation.

Usage

```
MicrobiomeSeqTreeFDR (otu.tab, tree, meta.dat, grp.name, adj.name = NULL,
raw.count = FALSE, B = 100)
```

Arguments

otu.tab	a data matrix, rows are the features and columns are the samples.
tree	an object of "phylo" class. The tree of OTUs.
meta.dat	a data frame containing the variables specified by grp.name and adj.name.
grp.name	a character string indicating the variable of major interest. Can be categorical or numerical.
adj.name	a character vector indicating the variables to be adjusted. Can be categorical or numerical.
raw.count	a logical value indicating whether X are raw counts. If raw counts are supplied, internal normalization/transformation (GMPR/sqrt) will be performed. The default is FALSE. The user should be responsible for selecting the appropriate normalization/transformation methods.
B	the number of permutations. The default is 100 since the permutation test is very fast.

Value

p.adj	TreeFDR adjusted p-values.
p.unadj	raw p-values.
z.adj	moderated z-values. The scale may be different from the raw z-values.
z.unadj	raw z-values.
k, rho	the estimates of the hyperparameters. The values indicate the informativeness of the prior structure.

Author(s)

Jun Chen

References

Jun Chen, et al. (2017). A fast and effective permutation-based framework for differential abundance analysis of microbiome data. To be submitted.

See Also

TreeFDR

Examples

```
require(StructFDR)

# Generate data
data(throat.parameter)
data.obj <- SimulateData(nCases = 50, nControls = 50, nOTU = 400, nCluster = 20,
depth = 10000, p.est = throat.parameter$p.est, theta = throat.parameter$theta,
        scene = 'S2', signal.strength = 4)
meta.dat <- data.frame(group = factor(data.obj$y), sex = sample(gl(2, 50)))
beta.true <- data.obj$beta.true

# Call TreeFDR
tree.fdr.obj <- MicrobiomeSeqTreeFDR(data.obj$X, data.obj$tree, meta.dat, 'group', 'sex', B = 20)
tree.p.adj <- tree.fdr.obj$p.adj

# Empirical power and FDR
(tree.emp.pwr <- sum(tree.p.adj <= 0.05 & beta.true != 0) / sum(beta.true != 0))
(tree.emp.fdr <- sum(tree.p.adj <= 0.05 & beta.true == 0) / sum(tree.p.adj <= 0.05))
```

SimulateData *Data Simulation Function to Study the Performance of TreeFDR.*

Description

We include five scenarios ('S1-S5'). S1-S3 are phylogeny-informative/clade-consistent scenarios while S4-S5 are phylogeny-noninformative/clade-inconsistent scenarios. In 'S1', we simulate two large clusters of differentially abundant OTUs. The fold changes (effect sizes) for OTUs from the same cluster are the same. In 'S2', we weaken the assumption, and generate variable fold changes for OTUs from the same cluster. In 'S3', we simulate many small clusters (10) of differentially abundant OTUs with the same effect sizes. In 'S4', we still simulate two large clusters of differentially abundant OTUs but we allow opposite effects for OTUs from the same cluster. This violates the assumption of similar effects for closely related OTUs. In 'S5', we pick 10% random OTUs without respect to the underlying phyologeny.

Usage

```
SimulateData(nCases = 50, nControls = 50, nOTU = 400, nCluster = 20, depth = 10000,
     p.est, theta, scene, signal.strength = 4, otu.no.min = 40,
     otu.no.max = 80, zero.pct = 0, balanced = FALSE)
```

Arguments

nCases, nControls	the number of case and control samples.
nOTU	the number of OTUs simulated.
nCluster	the number of clusters to be clustered.
depth	mean library sizes/sequencing depth. The library sizes are simulated from negative binomial distribution of size=25.
p.est, theta	the parameters (proportion vector and dispersion parameter) of the Dirichlet distribution.
scene	simulation scenarios. 'S1', 'S2', 'S3', 'S4', 'S5' denote five scenarios, respectively.
signal.strength	the strength of signal (related to the mean and sd of the log fold change).
otu.no.min, otu.no.max	the minimum and maximum numbers of differentially abundant OTUs. Defaults are 40 and 80.
zero.pct	the percentage of non-differential OTUs within the cluster/clade. Applicable to 'S1' and 'S2'
balanced	a logical value indicating whether the fold changes should be multiplied to cases samples (FALSE, increase/decrease in cases, no change for controls) only or to both case and control samples (TRUE, increase in case and control samples). Balanced design will have similar power for all OTUs.

Value

y	a vector of group membership. 0 = control, 1 = case.
X	a matrix of normalized OTU counts. row: OTUs; column: samples.
beta.true	a vector of the true log fold changes for all OTUs. 0s for non-differential OTUs.
D	a matrix of the cophenetic distances among the simulated OTUs.
tree	a simulated coalescent tree of the 'phylo' class.
clustering	a vector of cluster memberships for the OTUs based on PAM.

Author(s)

Jun Chen

References

Jian Xiao, Hongyuan Cao and Jun Chen (2016). False discovery rate control incorporating phylogenetic tree increases detection power in microbiome-wide multiple testing. Submitted.

Examples

```
# Generate data set for different scenarios S1-S5
require(StructFDR)
data(throat.parameter)
# Scene 1
data.obj <- SimulateData(nCases = 50, nControls = 50, nOTU = 400, nCluster = 20,
depth = 10000, p.est = throat.parameter$p.est, theta = throat.parameter$theta,
        scene = 'S1', signal.strength = 4)
# Scene 2
data.obj <- SimulateData(nCases = 50, nControls = 50, nOTU = 400, nCluster = 20,
depth = 10000, p.est = throat.parameter$p.est, theta = throat.parameter$theta,
        scene = 'S2', signal.strength = 4)
# Scene 3
data.obj <- SimulateData(nCases = 50, nControls = 50, nOTU = 400, nCluster = 100,
depth = 10000, p.est = throat.parameter$p.est, theta = throat.parameter$theta,
        scene = 'S3', signal.strength = 4)
# Scene 4
data.obj <- SimulateData(nCases = 50, nControls = 50, nOTU = 400, nCluster = 20,
depth = 10000, p.est = throat.parameter$p.est, theta = throat.parameter$theta,
        scene = 'S4', signal.strength = 2)
# Scene 5
data.obj <- SimulateData(nCases = 50, nControls = 50, nOTU = 400, nCluster = 20,
depth = 10000, p.est = throat.parameter$p.est, theta = throat.parameter$theta,
        scene = 'S5', signal.strength = 4)
```

`StructFDR` *False Discovery Rate (FDR) Control Integrating a General Prior Structure*

Description

The procedure is based on an empirical Bayes hierarchical model, where a structure-based prior distribution is designed to utilize the prior structure information. A moderated statistic based on posterior mean is used for permutation-based FDR control. By borrowing information from neighboring features defined based a distance metric, it is able to improve the statistical power of detecting associated features while controlling the FDR at desired levels.

Usage

```
StructFDR(X, Y, D, test.func, perm.func, eff.sign = TRUE, B = 20, q.cutoff = 0.5,
        alpha = 1, adaptive = c('Fisher', 'Overlap'), alt.FDR = c('BH', 'Permutation'),
           ...)
```

Arguments

`X`	a data matrix, rows are the features and columns are the samples.
`Y`	a vector of the phenotypic values, where association tests are being assessed

D	a matrix of pairwise distances between features. It determines how the features are related and the neighborhood for information borrowing.
test.func	a function that performs the actual tests. It takes X, Y and ... as the inputs, and returns a list with two slots p.value and e.sign, which are vectors of p-values and signs of the effects.
perm.func	a function that performs the permutation tests. It takes X, Y and ... as the inputs, and returns a list with two slots X and Y, which contain the permuted data.
eff.sign	a logical value indicating whether the direction of the effects should be considered. If it is true (default), negative and positive effects provide conflicting information.
B	the number of permutations. The default is 20. If computation time is not a big concern, B=100 is suggested to achieve excellent reproducibility between different runs.
q.cutoff	the quantile cutoff to determine the feature sets to estimate the number of false positives under the null. This cutoff is to protect the signal part of the distributions. The default is 0.5.
alpha	the exponent applied to the distance matrix. Large values have more smoothing effects for closely related features. The default is 1.
adaptive	the proposed procedure is most powerful when the signal structure conforms the prior structure. When this assumption is seriously violated, it loses power. We provide two heuristic adaptive approaches to compensate the power loss in such situations. 'Fisher' approach compares the number of hits by our method to the alternative FDR approach at an FDR of 20% and uses the alternative FDR approach if the number of hits is significantly less based on Fisher's exact test; 'Overlap' method selects the alternative approach when it fails to identify half of the hits from the alternative approach at an FDR of 20%. The default is 'Fisher'.
alt.FDR	the alternative FDR control used when the proposed approach is powerless. The default is 'BH' procedure and another option is the permutation-based FDR control ('Permutation')
...	further arguments such as covariates to be passed to test.func

Value

p.adj	StructFDR adjusted p-values.
p.unadj	raw p-values.
z.adj	moderated z-values. The scale may be different from the raw z-values.
z.unadj	raw z-values.
k, rho	the estimates of the hyperparameters. The values indicate the informativeness of the prior structure.

Author(s)

Jun Chen

References

Jian Xiao, Hongyuan Cao and Jun Chen (2016). False discovery rate control incorporating phylogenetic tree increases detection power in microbiome-wide multiple testing. Submitted.

See Also

TreeFDR

Examples

```
require(ape)
require(nlme)
require(cluster)
require(StructFDR)

# Generate a caelescence tree and partition into 10 clusters
set.seed(1234)
n <- 20
p <- 200
tree <- rcoal(p)
# Pairwise distance matrix.
D <- cophenetic(tree)
clustering <- pam(D, k=10)$clustering

# Simulate case-control data, assuming cluster 2 is differential
X.control <- matrix(rnorm(n*p), p, n)
X.case <- matrix(rnorm(n*p), p, n)
eff.size <- rnorm(sum(clustering == 2), 0.5, 0.2)
X.case[clustering == 2, ] <- X.case[clustering == 2, ] + eff.size
X <- cbind(X.control, X.case)
Y <- gl(2, n)

# Define testing and permutation function
test.func <- function (X, Y) {
obj <- apply(X, 1, function(x) {
ttest.obj <- t.test(x ~ Y)
c(ttest.obj$p.value, sign(ttest.obj$statistic))
})
    return(list(p.value=obj[1, ], e.sign=obj[2, ]))
}

perm.func <- function (X, Y) {
return(list(X=X, Y=sample(Y)))
}

# Call StructFDR
tree.fdr.obj <- StructFDR(X, Y, D, test.func, perm.func)

# Compare StructFDR and BH
tree.fdr.obj$p.adj
tree.fdr.obj$p.adj[clustering == 2]
BH.p.adj <- p.adjust(tree.fdr.obj$p.unadj, 'fdr')
```

```
BH.p.adj[clustering == 2]

# Adjusted statistics vs clustering
par(mfrow=c(1, 2))
plot(clustering, tree.fdr.obj$z.unadj)
plot(clustering, tree.fdr.obj$z.adj)
```

`throat.parameter` *Dirichlet parameters used for generating the artificial data sets.*

Description

These parameters were derived based on a microbiome data set for studying the effect of smoking on the upper respiratory tract microbiome. The left throat microbiome was used to estimate the relative proportions of 778 most abundant OTUs. The dispersion parameter was selected to achieve medium level of zeroinflation (around 50% zeros).

Usage

```
data(throat.parameter)
```

Format

The format is: chr "throat.parameter"

Details

A list containing the proportion vector (`throat.p.est`) and the dispersion (`throat.theta`).

Source

Charlson ES, Chen J, Custers-Allen R, Bittinger K, Li H, et al. (2010) Disordered Microbial Communities in the Upper Respiratory Tract of Cigarette Smokers. PLoS ONE 5(12): e15216.

Examples

```
data(throat.parameter)
```

TreeFDR *False Discovery Rate (FDR) Control Integrating Prior Tree Structure for Microbiome Data*

Description

The procedure is based on an empirical Bayes hierarchical model, where a structure-based prior distribution is designed to utilize the phylogenetic tree. A moderated statistic based on posterior mean is used for permutation-based FDR control. By borrowing information from neighboring bacterial species, it is able to improve the statistical power of detecting associated bacterial species while controlling the FDR at desired levels.

Usage

```
TreeFDR(X, Y, tree, test.func, perm.func, eff.sign = TRUE, B = 20, q.cutoff = 0.5,
      alpha = 1, adaptive = c('Fisher', 'Overlap'), alt.FDR = c('BH', 'Permutation'),
        ...)
```

Arguments

X	a data matrix, rows are the features and columns are the samples.
Y	a vector of the phenotypic values, where association tests are being assessed
tree	an object of "phylo" class
test.func	a function that performs the actual tests. It takes X, Y and ... as the inputs, and returns a list with two slots p.value and e.sign, which are vectors of p-values and signs of the effects.
perm.func	a function that performs the permutation tests. It takes X, Y and ... as the inputs, and returns a list with two slots X and Y, which contain the permuted data.
eff.sign	a logical value indicating whether the direction of the effects should be considered. If it is true (default), negative and positive effects provide conflicting information.
B	the number of permutations. The default is 20. If computation time is not a big concern, B=100 is suggested to achieve excellent reproducibility between different runs.
q.cutoff	the quantile cutoff to determine the feature sets to estimate the number of false positives under the null. This cutoff is to protect the signal part of the distributions. The default is 0.5.
alpha	the exponent applied to the distance matrix. Large values have more smoothing effects for closely related species. The default is 1. If the underlying structure assumption is weak, consider decrease the value to 0.5.
adaptive	the proposed procedure is most powerful when the signal is shared in a clade. When this assumption is seriously violated, it loses power. We provide two heuristic adaptive approaches to compensate the power loss in such situations. 'Fisher' approach compares the number of hits by our method to the alternative

FDR approach at an FDR of 20% and uses the alternative FDR approach if the number of hits is significantly less based on Fisher's exact test; 'Overlap' method selects the alternative approach when it fails to identify half of the hits from the alternative approach at an FDR of 20%. The default is 'Fisher'.

`alt.FDR` the alternative FDR control used when the proposed approach is powerless. The default is 'BH' procedure and another option is the permutation-based FDR control ('Permutation')

`...` further arguments such as covariates to be passed to `test.func`

Value

`p.adj` TreeFDR adjusted p-values.

`p.unadj` raw p-values.

`z.adj` moderated z-values. The scale may be different from the raw z-values.

`z.unadj` raw z-values.

`k, rho` the estimates of the hyperparameters. The values indicate the informativeness of the prior structure.

Author(s)

Jun Chen

References

Jian Xiao, Hongyuan Cao and Jun Chen (2016). False discovery rate control incorporating phylogenetic tree increases detection power in microbiome-wide multiple testing. Submitted.

See Also

`StructFDR`

Examples

```
require(ape)
require(nlme)
require(cluster)
require(StructFDR)

# Generate a caelescence tree and partition into 10 clusters
set.seed(1234)
n <- 20
p <- 200
tree <- rcoal(p)
D <- cophenetic(tree)
clustering <- pam(D, k=10)$clustering

# Simulate case-control data, assuming cluster 2 is differential
X.control <- matrix(rnorm(n*p), p, n)
X.case <- matrix(rnorm(n*p), p, n)
```

```
eff.size <- rnorm(sum(clustering == 2), 0.5, 0.2)
X.case[clustering == 2, ] <- X.case[clustering == 2, ] + eff.size
X <- cbind(X.control, X.case)
Y <- gl(2, n)

# Define testing and permutation function
test.func <- function (X, Y) {
obj <- apply(X, 1, function(x) {
ttest.obj <- t.test(x ~ Y)
c(ttest.obj$p.value, sign(ttest.obj$statistic))
})
    return(list(p.value=obj[1, ], e.sign=obj[2, ]))
}

perm.func <- function (X, Y) {
return(list(X=X, Y=sample(Y)))
}

# Call TreeFDR
tree.fdr.obj <- TreeFDR(X, Y, tree, test.func, perm.func)

# Compare TreeFDR and BH
tree.fdr.obj$p.adj
tree.fdr.obj$p.adj[clustering == 2]
BH.p.adj <- p.adjust(tree.fdr.obj$p.unadj, 'fdr')
BH.p.adj[clustering == 2]

# Adjusted statistics vs clustering
par(mfrow=c(1, 2))
plot(clustering, tree.fdr.obj$z.unadj)
plot(clustering, tree.fdr.obj$z.adj)
```

`Ztransform` *Transform P-values to Z-values*

Description

Transform the p-values to z-values. Both two-sided and one-sided transformations are implemented.

Usage

```
Ztransform(p.value, e.sign, eff.sign = TRUE, tol = 1e-15)
```

Arguments

`p.value`	a vector of p-values
`e.sign`	a vector of signs of the effects, taken on value -1 and 1. (In effects when 'eff.sign = TRUE')

`eff.sign`	a logical value indicating whether the direction/sign of the effect should be considered
`tol`	a numeric value at which the p-value (both ends) will truncate.

Value

a vector of z-values

Author(s)

Jun Chen

References

Jian Xiao, Hongyuan Cao and Jun Chen (2016). False discovery rate control incorporating phylogenetic tree increases detection power in microbiome-wide multiple testing. Submitted.

Index