Package 'SOHPIE'

October 23, 2023

Type Package

Title Statistical Approach via Pseudo-Value Information and Estimation

Version 1.0.6

Description 'SOHPIE' (pronounced as SOFIE) is a novel pseudo-value regression approach for differential co-abundance network analysis of microbiome data, which can include additional clinical covariate in the model. The full methodological details can be found in Ahn S and Datta S (2023) <arXiv:2303.13702v1>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Depends R (>= 3.5.0)

Imports robustbase, parallel, dplyr, fdrtool, gtools, stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Language en-US

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Seungjun Ahn [cre, aut, trl] (<https://orcid.org/0000-0002-4816-8924>),
    Somnath Datta [ctb, ths]

Maintainer Seungjun Ahn <seungjun.ahn@mountsinai.org>

Repository CRAN

Date/Publication 2023-10-23 17:50:02 UTC

R topics documented:

---

asso_mat   *asso_mat*

Description

A function to estimate an association matrix. This function also includes re-estimation for leave-one-out sample.

Usage

```
asso_mat(OTUdat, group)
```

Arguments

OTUdat	An OTU table with subjects in rows and taxa in columns.
group	Indices of the subjects in a category of binary group variable.

Value

A list of an association matrix and reestimated association matrix is returned, which are estimated via SparCC.

Examples

```
# In this example, the subset of the American Gut Project data will be used.
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]

# Obtain indices of each grouping factor
```

```
# In this example, a variable indicating the status of living
# with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
newindex_grpA = which(combinedamgut$bin_dog == 0)
newindex_grpB = which(combinedamgut$bin_dog == 1)

# Now, we estimate (and re-estimate) association matrices
# for each group separately.
asso_matA = asso_mat(OTUdat=OTUtab, group=newindex_grpA)
asso_matB = asso_mat(OTUdat=OTUtab, group=newindex_grpB)
```

---

coeff	*coeff*

Description

A function to retrieve a vector of coefficient estimates of all predictor variables in the pseudo-value regression model.

Usage

```
coeff(SOHPIEres)
```

Arguments

SOHPIEres   An object called after running SOHPIE_DNA.

Value

A table that includes coefficient estimates for all variables included in the fitted model.

Examples

```
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]
# Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.
# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
 newindex_grpA = which(combinedamgut$bin_dog == 0)
 newindex_grpB = which(combinedamgut$bin_dog == 1)
```

```
SOHPIEres <- SOHPIE_DNA(OTUdat = OTUtab, clindat = phenodat,
groupA = newindex_grpA, groupB = newindex_grpB, c = 0.5)

# coeff() function will return coefficient estimates only.
coeff(SOHPIEres)
```

---

coeff_specific_var   *coeff_specific_var*

Description

A function to retrieve a vector of coefficient estimates of each taxa for one specific variable.

Usage

```
coeff_specific_var(coefftab, varname)
```

Arguments

coefftab	A table that includes coefficient estimates for a specific variable.
varname	Specify the name of the variable of interest.

Value

A vector of coefficient estimates for a single variable from the model.

Examples

```
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]
# Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.
# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
 newindex_grpA = which(combinedamgut$bin_dog == 0)
 newindex_grpB = which(combinedamgut$bin_dog == 1)

SOHPIEres <- SOHPIE_DNA(OTUdat = OTUtab, clindat = phenodat,
groupA = newindex_grpA, groupB = newindex_grpB, c = 0.5)

# coeff() function will return coefficient estimates only.
coefftab <- coeff(SOHPIEres)
```

```
# coeff_specific_var() will return coefficient estimates of
# a single variable of interest.
coeff_specific_var(coefftab = coefftab, varname = "bin_dog")
```

---

combinedamgut *A Subdata from the American Gut Project study data*

Description

A pre-processed OTU table and clinical data were obtained from the American Gut Project, available in the SpiecEasi R package.

Usage

```
data(combinedamgut)
```

Format

combinedamgut

References

McDonald D, et al. American Gut: an Open Platform for Citizen Science Microbiome Research. mSystems, 2018;3(3):e00031-18. (PubMed)

Examples

```
data(combinedamgut)
```

---

combineddietswap *A Subdata from the Diet Swap study data*

Description

A pre-processed OTU table and clinical data from the geographical epidemiology study (aka the Diet Exchange Study) is available in the microbiome R package.

Usage

```
data(combineddietswap)
```

Format

'combineddietswap'

References

O'Keefe, SJ, et al. Fat, fibre and cancer risk in african americans and rural africans. Nat Commun. 2015;6:6342. (PubMed)

Examples

```
data(combineddietswap)
```

DCtaxa_tab	*DCtaxa_tab*

Description

A function to obtain a list consisting of taxa that are significantly differentially connected (DC) between two biological or clinical conditions. These DC taxa are resulted from the pseudo-value regression method with additional covariates. In addition, a user can extract the names of DC taxa only.

Usage

```
DCtaxa_tab(qvaltab, groupvar, alpha)
```

Arguments

qvaltab	A table with adjusted p-values (or q-value in this package).
groupvar	Specify the name of binary indicator variable.
alpha	A level of significance (e.g. 0.05).

Value

q-values and names of significantly DC taxa (e.g. taxa name) based on SOHPIE_DNA function.

Examples

```
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]
# Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.
# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
 newindex_grpA = which(combinedamgut$bin_dog == 0)
 newindex_grpB = which(combinedamgut$bin_dog == 1)
```

```
SOHPIEres <- SOHPIE_DNA(OTUdat = OTUtab, clindat = phenodat,
groupA = newindex_grpA, groupB = newindex_grpB, c = 0.5)

# Create an object to keep the table with q-values using qval() function.
qvaltab <- qval(SOHPIEres)

# Please do NOT forget to provide the name of variable in DCtaxa_tab(groupvar = ).
DCtaxa_tab <- DCtaxa_tab(qvaltab = qvaltab, groupvar = "bin_dog", alpha = 0.3)
DCtaxa_tab
```

---

pseudoreg   *pseudoreg*

Description

A function to regress pseudo-values across set of covariates.

Usage

```
pseudoreg(pseudoval, clindat, c)
```

Arguments

pseudoval	Jackknife pseudo-values calculated.
clindat	A metadata/phenotypic data consisting of the clinical and demographic variables that the user wants to include in the regression. (e.g., binary group indicator for intervention vs. control, continuous age, ...)
c	The choice of trimming proportion for the least trimmed estimator of robust regression. A value has to be between 0.5 and 1 as specified in ltsReg() function in robustbase package.

Value

A pseudo-value regression is fitted. Please use pseudoreg.summary() to output p-values, q-values, and coefficient estimates.

Examples

```
# In this example, the subset of the American Gut Project data will be used.
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]

#Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
```

```
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.

# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living
# with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
newindex_grpA = which(combinedamgut$bin_dog == 0)
newindex_grpB = which(combinedamgut$bin_dog == 1)

# Now, we estimate (and re-estimate) association matrices
# for each group separately.
asso_matA = asso_mat(OTUdat=OTUtab, group=newindex_grpA)
asso_matB = asso_mat(OTUdat=OTUtab, group=newindex_grpB)

# Calculate the network centrality.
thetahat_grpA = thetahats(asso_matA$assomat)
thetahat_grpB = thetahats(asso_matB$assomat)

# Obtain network centrality for the re-estimated association matrices.
thetahat_drop_grpA = sapply(asso_matA$reest.assomat, thetahats)
thetahat_drop_grpB = sapply(asso_matB$reest.assomat, thetahats)

# Sample sizes for each group.
n_A <- length(newindex_grpA)
n_B <- length(newindex_grpB)

# Now calculate jackknife pseudo-values for each group.
thetatilde_grpA = thetatildefun(thetahat_grpA, thetahat_drop_grpA, n_A)
thetatilde_grpB = thetatildefun(thetahat_grpB, thetahat_drop_grpB, n_B)

thetatilde = rbind(thetatilde_grpA, thetatilde_grpB)

# Map the column names (taxa names)
colnames(thetatilde) = colnames(OTUtab)

# Fit a pseudo-value regression using jackknife pseudovalues
# and phenotypic data. A reminder that the phenotypic data should
# contain a set of predictor variables to be fitted.
fitmod = pseudoreg(pseudoval=thetatilde, clindat=phenodat, c=0.5)
```

---

pseudoreg.summary   *pseudoreg.summary()*

Description

A function to output summary results (p-values, q-values, and coefficient estimates) from the fitted pseudo-value regression.

Usage

```
pseudoreg.summary(pseudo.reg.res, taxanames)
```

Arguments

pseudo.reg.res   A fitted pseudo-value regression using pseudoreg()

taxanames   Names of taxa from the OTU table.

Value

A pseudo-value regression is fitted. Please use pseudoreg.summary() to output p-values, q-values, and coefficient estimates.

Examples

```
# In this example, the subset of the American Gut Project data will be used.
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]

#Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.

# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living
# with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
newindex_grpA = which(combinedamgut$bin_dog == 0)
newindex_grpB = which(combinedamgut$bin_dog == 1)

# Now, we estimate (and re-estimate) association matrices
# for each group separately.
asso_matA = asso_mat(OTUdat=OTUtab, group=newindex_grpA)
asso_matB = asso_mat(OTUdat=OTUtab, group=newindex_grpB)

# Calculate the network centrality.
thetahat_grpA = thetahats(asso_matA$assomat)
thetahat_grpB = thetahats(asso_matB$assomat)

# Obtain network centrality for the re-estimated association matrices.
thetahat_drop_grpA = sapply(asso_matA$reest.assomat, thetahats)
thetahat_drop_grpB = sapply(asso_matB$reest.assomat, thetahats)

# Sample sizes for each group.
n_A <- length(newindex_grpA)
n_B <- length(newindex_grpB)
```

```
# Now calculate jackknife pseudo-values for each group.
thetatilde_grpA = thetatildefun(thetahat_grpA, thetahat_drop_grpA, n_A)
thetatilde_grpB = thetatildefun(thetahat_grpB, thetahat_drop_grpB, n_B)

thetatilde = rbind(thetatilde_grpA, thetatilde_grpB)

# Map the column names (taxa names)
colnames(thetatilde) = colnames(OTUtab)

# Fit a pseudo-value regression using jackknife pseudovalues
# and phenotypic data. A reminder that the phenotypic data should
# contain a set of predictor variables to be fitted.
fitmod = pseudoreg(pseudoval=thetatilde, clindat=phenodat, c=0.5)

# Extract summary results from the fitted model from fitmod object above.
summary.result = pseudoreg.summary(pseudo.reg.res=fitmod, taxanames=colnames(OTUtab))
```

---

pval   *pval*

Description

A function to retrieve a vector of p-values of each taxa for all variables that are included in the pseudo-value regression model.

Usage

```
pval(SOHPIEres)
```

Arguments

SOHPIEres   An object called after running SOHPIE_DNA.

Value

A table that includes p-values for all predictor variables considered in the regresson.

Examples

```
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]
# Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.
# Obtain indices of each grouping factor
```

```
# In this example, a variable indicating the status of living with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
 newindex_grpA = which(combinedamgut$bin_dog == 0)
 newindex_grpB = which(combinedamgut$bin_dog == 1)

SOHPIEres <- SOHPIE_DNA(OTUdat = OTUtab, clindat = phenodat,
groupA = newindex_grpA, groupB = newindex_grpB, c = 0.5)

# Create an object to keep the table with p-values using qval() function.
pvaltab <- pval(SOHPIEres)
```

---

pval_specific_var   *pval_specific_var*

Description

A function to retrieve a vector of p-values of each taxa for one specific variable. In other words, this will be useful for quickly accessing the taxa-specific p-values for main binary group variable (or other specific variable/covariate).

Usage

```
pval_specific_var(pvaltab, varname)
```

Arguments

pvaltab   A table that includes p-values for a specific variable.

varname   Specify the name of the variable of interest.

Value

A vector of p-values for a single variable from the model.

Examples

```
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]
# Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.
# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
 newindex_grpA = which(combinedamgut$bin_dog == 0)
```

```
  newindex_grpB = which(combinedamgut$bin_dog == 1)

SOHPIEres <- SOHPIE_DNA(OTUdat = OTUtab, clindat = phenodat,
groupA = newindex_grpA, groupB = newindex_grpB, c = 0.5)

# Create an object to keep the table with p-values using pval() function.
pvaltab <- pval(SOHPIEres)

# Retrieve a vector of p-values for a single variable of interest.
pval_specific_var(pvaltab = pvaltab, varname = "bin_dog")
```

---

qval   *qval*

Description

A function to retrieve a vector of q-values of each taxa for all variables that are included in the pseudo-value regression model.

Usage

```
qval(SOHPIEres)
```

Arguments

SOHPIEres   An object called after running SOHPIE_DNA.

Value

A table that includes q-values for all predictor variables considered in the regresson.

Examples

```
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]
# Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.
# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
 newindex_grpA = which(combinedamgut$bin_dog == 0)
 newindex_grpB = which(combinedamgut$bin_dog == 1)

SOHPIEres <- SOHPIE_DNA(OTUdat = OTUtab, clindat = phenodat,
```

```
groupA = newindex_grpA, groupB = newindex_grpB, c = 0.5)

# Create an object to keep the table with q-values using qval() function.
qvaltab <- qval(SOHPIEres)
```

---

qval_specific_var	*qval_specific_var*

Description

A function to retrieve a vector of q-values of each taxa for one specific variable. In other words, this will be useful for quickly accessing the taxa-specific q-values for main binary group variable (or other specific variable/covariate).

Usage

```
qval_specific_var(qvaltab, varname)
```

Arguments

qvaltab	A table that includes q-values for a specific variable.
varname	Specify the name of the variable of interest.

Value

A vector of q-values for a single variable from the model.

Examples

```
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]
# Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.
# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
 newindex_grpA = which(combinedamgut$bin_dog == 0)
 newindex_grpB = which(combinedamgut$bin_dog == 1)

SOHPIEres <- SOHPIE_DNA(OTUdat = OTUtab, clindat = phenodat,
groupA = newindex_grpA, groupB = newindex_grpB, c = 0.5)

# Create an object to keep the table with q-values using qval() function.
```

```
qvaltab <- qval(SOHPIEres)

# Retrieve a vector of q-values for a single variable of interest.
qval_specific_var(qvaltab = qvaltab, varname = "bin_dog")
```

SOHPIE_DNA   *SOHPIE_DNA*

Description

A pseudo-value regression approach for differential co-abundance network analysis that adjusts for additional covariates.

Usage

```
SOHPIE_DNA(OTUdat, clindat, groupA, groupB, c)
```

Arguments

OTUdat	An OTU table with subjects in rows and taxa in columns.
clindat	A subdata consisting of the clinical and demographic variables that the user wants to include in the regression. (e.g., binary group indicator for intervention vs. control, continuous age, ...)
groupA	Indices of the subjects in the first category (e.g., not living with a dog; see example below with American Gut Project sample data) of binary group variable.
groupB	Indices of the subjects in the second category (e.g., living with a dog; see example below with American Gut Project sample data) of binary group variable.
c	The choice of trimming proportion for the least trimmed estimator of robust regression. A value has to be between 0.5 and 1 as specified in ltsReg() function in robustbase package.

Value

A list containing three data frame objects returned from this SOHPIE_DNA main function. A user will see beta coefficients, p-values, and adjusted p-values (q-values) for each predictor variables that are included in the regression model.

References

Ahn S, Datta S. Differential Co-Abundance Network Analyses for Microbiome Data Adjusted for Clinical Covariates Using Jackknife Pseudo-Values. ArXiv [Preprint]. 2023 Mar 23:arXiv:2303.13702v1. PMID: 36994149; PMCID: PMC10055480.

Examples

```
# In this example, the subset of the American Gut Project data will be used.
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]
 #Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.
# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living
# with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
newindex_grpA = which(combinedamgut$bin_dog == 0)
newindex_grpB = which(combinedamgut$bin_dog == 1)

SOHPIEres <- SOHPIE_DNA(OTUdat = OTUtab, clindat = phenodat,
groupA = newindex_grpA, groupB = newindex_grpB, c = 0.5)
```

---

sparcc	*sparcc wrapper*

Description

SpiecEasi R package, says in his package that this is "a reimplementation of SparCC algorithm (Friedman et Alm, PLoS Comp Bio, 2012)." Installation of SpiecEasi can sometimes generate errors, so I have included Dr. Huaying Fang's sparcc wrapper as one of the functions in this package for the estimation of co-abundance networks. His code was acquired from CCLasso (Fang et al, Bioinformatics, 2015), provided in GitHub (https://github.com/huayingfang/CCLasso).

Usage

```
sparcc(x, imax = 20, kmax = 10, alpha = 0.1, Vmin = 1e-04)
```

Arguments

x	count data matrix (OTU table)
imax	Number of iterations in the outer loop
kmax	max iteration steps for SparCC
alpha	threshold for strong correlation
Vmin	absolute value of correlations below this threshold are considered zero by the inner SparCC loop.

Value

This will estimate an association matrix (network) for observed OTU table.

---

stderrs   *stderrs*

Description

A function to retrieve a vector of standard error (stderr) of coefficient estimates (betahats) all predictor variables in the pseudo-value regression model.

Usage

```
stderrs(SOHPIEres)
```

Arguments

SOHPIEres   An object called after running SOHPIE_DNA.

Value

A table that includes standard error of betahats for all predictors regressed in the fitted model.

Examples

```
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]
# Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.
# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
 newindex_grpA = which(combinedamgut$bin_dog == 0)
 newindex_grpB = which(combinedamgut$bin_dog == 1)

SOHPIEres <- SOHPIE_DNA(OTUdat = OTUtab, clindat = phenodat,
groupA = newindex_grpA, groupB = newindex_grpB, c = 0.5)

# stderrs() function will return standard error of betahats only.
stderrs(SOHPIEres)
```

stderrs_specific_var   *stderrs_specific_var*

Description

A function to retrieve a vector of standard error of coefficient estimates (betahats) of each taxa for one specific variable.

Usage

```
stderrs_specific_var(stderrstab, varname)
```

Arguments

stderrstab	A table that includes standard error of betahat for a specific variable.
varname	Specify the name of the variable of interest.

Value

A vector of standard error of betahats for a single variable from the model.

Examples

```
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]
# Clinical/demographic covariates (phenotypic data):
# Note: All of these covariates will be included in the regression, so
# please make sure that phenodat includes the variables that will be analyzed only.
phenodat = combinedamgut[, 1:7] # first column is ID, so not using it.
# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
 newindex_grpA = which(combinedamgut$bin_dog == 0)
 newindex_grpB = which(combinedamgut$bin_dog == 1)

SOHPIEres <- SOHPIE_DNA(OTUdat = OTUtab, clindat = phenodat,
groupA = newindex_grpA, groupB = newindex_grpB, c = 0.5)

# stderrs() function will return standard error of betahats only.
stderrstab <- stderrs(SOHPIEres)

# stderrs_specific_var() will return standard error of coefficient estimates of
# a single variable of interest.
stderrs_specific_var(stderrstab = stderrstab, varname = "bin_dog")
```

---

thetahats   *thetahats*

Description

A function to compute the network centrality (i.e. total connectivity) of each microbial taxa from the association matrix.

Usage

```
thetahats(asso.matinput)
```

Arguments

asso.matinput   An input is an association matrix that is estimated from the user-provided OTU data.

Value

A vector containing network centrality of each taxa.

---

thetatildefun   *thetatildefun*

Description

A function to calculate jackknife pseudo-values

Usage

```
thetatildefun(thetahatinput, thetahatdropinput, sizegroup)
```

Arguments

thetahatinput   A network centrality calculated from association matrix for whole sample.

thetahatdropinput

Network centralities calculated from re-estimated association matrices for leave-one-out samples.

sizegroup   Sample size for group.

Value

A jackknife pseudo-value will be returned.

Examples

```
# In this example, the subset of the American Gut Project data will be used.
data(combinedamgut) # A complete data containing columns with taxa and clinical covariates.

# Note: The line below will use a toy example with the first 30 out of 138 taxa.
OTUtab = combinedamgut[ , 8:37]

# Obtain indices of each grouping factor
# In this example, a variable indicating the status of living
# with a dog was chosen (i.e. bin_dog).
# Accordingly, Groups A and B imply living without and with a dog, respectively.
newindex_grpA = which(combinedamgut$bin_dog == 0)
newindex_grpB = which(combinedamgut$bin_dog == 1)

# Now, we estimate (and re-estimate) association matrices
# for each group separately.
asso_matA = asso_mat(OTUdat=OTUtab, group=newindex_grpA)
asso_matB = asso_mat(OTUdat=OTUtab, group=newindex_grpB)

# Calculate the network centrality.
thetahat_grpA = thetahats(asso_matA$assomat)
thetahat_grpB = thetahats(asso_matB$assomat)

# Obtain network centrality for the re-estimated association matrices.
thetahat_drop_grpA = sapply(asso_matA$reest.assomat, thetahats)
thetahat_drop_grpB = sapply(asso_matB$reest.assomat, thetahats)

# Sample sizes for each group.
n_A <- length(newindex_grpA)
n_B <- length(newindex_grpB)

# Now calculate jackknife pseudo-values for each group.
thetatilde_grpA = thetatildefun(thetahat_grpA, thetahat_drop_grpA, n_A)
thetatilde_grpB = thetatildefun(thetahat_grpB, thetahat_drop_grpB, n_B)
```

Index