Package ‘IFAA’

October 16, 2020

Title IFAA: Robust association identification and Inference For Absolute Abundance in microbiome analyses

Version 0.0.0.9000

Description IFAA is a novel approach to make inference on the association of covariates with the absolute abundance (AA) of microbiome in an ecosystem.

License GNU General Public License version 2

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends picasso (>= 1.2.0),
glmnet (>= 2.0-16),
expm (>= 0.999-3),
foreach (>= 1.4.3),
snow (>= 0.4-2),
doSNOW (>= 1.0.15),
rlecuyer (>= 0.3-3),
Matrix (>= 1.2-14),
HDCI (>= 1.0-2),
doParallel (>= 1.0.11),
future (>= 1.12.0)

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

IFAA *Robust association identification and inference for absolute abundance in microbiome analyses*

Description

Make inference on the association of covariates of microbiome

Usage

```
IFAA(
  MicrobData,
  CovData,
  linkIDname,
  testCov = NULL,
  ctrlCov = NULL,
  testMany = T,
  ctrlMany = F,
  nRef = 40,
  nRefMaxForEsti = 1,
  nPermu = 40,
  x1permut = T,
  refTaxa = NULL,
  reguMethod = c("mcp"),
  fwerRate = 0.25,
  paraJobs = NULL,
  bootB = 500,
  bootLassoAlpha = 0.05,
  standardize = F,
  sequentialRun = F,
  allFunc = allUserFunc(),
  refReadsThresh = 0.2,
  SDThresh = 0.05,
  SDquantilThresh = 0,
  balanceCut = 0.2,
  seed = 1
)
```

Arguments

`MicrobData` Microbiome data matrix containing microbiome abundance with each row per sample and each column per taxon/OTU/ASV. It should contain an `"id"` variable to correspond to the `"id"` variable in the covariates data: `CovData`.

`CovData` Covariates data matrix containing covariates and confounders with each row per sample and each column per variable. It should also contain an `"id"` variable to correspond to the `"id"` variable in the microbiome data: `MicrobData`.

`linkIDname` Variable name of the `"id"` variable in both `MicrobData` and `CovData`. The two data sets will be merged by this `"id"` variable.

`testCov` Covariates that are of primary interest for testing and estimating the associations. It corresponds to X_i in the equation. Default is `NULL` which means all covariates are `testCov`.

ctrlCov Potential confounders that will be adjusted in the model. It corresponds to W_i in the equation. Default is NULL which means all covariates except those in testCov are adjusted as confounders.

testMany This takes logical value TRUE or FALSE. If TRUE, the testCov will contain all the variables in CovData provided testCov is set to be NULL. The default value is TRUE which does not do anything if testCov is not NULL.

ctrlMany This takes logical value TRUE or FALSE. If TRUE, all variables except testCov are considered as control covariates provided ctrlCov is set to be NULL. The default value is TRUE which does not do anything if ctrlCov is not NULL.

nRef The number of randomly picked reference taxa used in phase 1. Default number is 40.

nRefMaxForEsti The maximum number of reference taxa used in phase 2. The default is 1.

nPermu The number of permutation used in phase 1. Default number is 40.

x1permut This takes a logical value TRUE or FALSE. If true, it will permute the variables in testCov. If false, it will use residual-permutation proposed by Freedman and Lane (1983).

refTaxa A vector of taxa or OTU or ASV names. These are reference taxa specified by the user to be used in phase 1. If the number of reference taxa is less than 'nRef', the algorithm will randomly pick extra reference taxa to make up 'nRef'. The default is NULL since the algorithm will pick reference taxa randomly.

reguMethod regularization approach used in phase 1 of the algorithm. Take value "mcp" or "lasso", default is "mcp".

fwerRate The family wise error rate for identifying taxa/OTU/ASV associated with testCov in phase 1. Default is 0.25.

paraJobs If sequentialRun is FALSE, this specifies the number of parallel jobs that will be registered to run the algorithm. Default is 8. If specified as NULL, it will automatically detect the cores to decide the number of parallel jobs.

bootB Number of bootstrap samples for obtaining confidence interval of estimates in phase 2. The default is 500.

bootLassoAlpha The significance level in phase 2. Default is 0.05.

standardize This takes a logical value TRUE or FALSE. If TRUE, all design matrix X in phase 1 and phase 2 will be standardized in the analyses. Default is FALSE.

sequentialRun This takes a logical value TRUE or FALSE. Sometimes parallel jobs can not be successfully run for unknown reasons. For example, socket related errors may pop up or some slave cores return simple error instead of numerical results. In those scenarios, setting sequentialRun = TRUE may help, but it will take more time to run. Default is FALSE.

refReadsThresh The threshold of non-zero sequencing reads for choosing the reference taxon in phase 2. The default is 0.2 which means at least 20% non-zero sequencing reads.

SDThresh The threshold of standard deviations of sequencing reads for choosing the reference taxon in phase 2. The default is 0.5 which means the standard deviation of sequencing reads should be at least 0.5.

balanceCut The threshold of non-zero sequencing reads in each group of a binary variable for choosing the reference taxon in phase 2. The default number is 0.2 which means at least 20% sequencing reads are non-zero in each group.

seed Random seed for reproducibility. Default is 1.

Details

The `IFAA()` uses a novel approach to make inference on the association of covariates with the absolute abundance (AA) of microbiome in an ecosystem.

Value

A list containing the estimation results.

- `analysisResults$estByCovList`: A list containing estimating results for all the variables in `testCov`. See details.
- `covariatesData`: A dataset containing covariates and confounders used in the analyses.

References

Li et al.(2020) IFAA: Robust association identification and Inference For Absolute Abundance in microbiome analyses. arXiv:1909.10101v3

Zhang CH (2010) Nearly unbiased variable selection under minimax concave penalty. Annals of Statistics. 38(2):894-942.

Freedman and Lane (1983) A nonstochastic interpretation of reported significance levels. Journal of Business & Economic Statistics. 1(4):292-298.

Examples

```
data(dataM)
dim(dataM)
dataM[1:5, 1:8]
data(dataC)
dim(dataC)
dataC[1:5, ]
results <- IFAA(MicrobData = dataM,
                CovData = dataC,
                linkIDname = "id",
                testCov = c("v1", "v2"),
                ctrlCov = c("v3"), nRef = 4,
                nPermu = 4,
                fwerRate = 0.25,
                bootB = 5)
```

MZILN *Conditional regression for microbiome analysis based on multivariate zero-inflated logistic normal model*

Description

Make inference on the associations of microbiome with covariates given a user-specified reference taxon/OTU/ASV.

Usage

```
MZILN(
  MicrobData,
  CovData,
  linkIDname,
  allCov = NULL,
  refTaxa,
  reguMethod = c("mcp"),
  paraJobs = NULL,
  bootB = 500,
  bootLassoAlpha = 0.05,
  standardize = F,
  sequentialRun = T,
  allFunc = allUserFunc(),
  seed = 1
)
```

Arguments

`MicrobData` Microbiome data matrix containing microbiome abundance with each row per sample and each column per taxon/OTU/ASV. It should contain an `"id"` variable to correspond to the `"id"` variable in the covariates data: `CovData`.

`CovData` Covariates data matrix containing covariates and confounders with each row per sample and each column per variable. It should also contain an `"id"` variable to correspond to the `"id"` variable in the microbiome data: `MicrobData`.

`linkIDname` Variable name of the `"id"` variable in both `MicrobData` and `CovData`. The two data sets will be merged by this `"id"` variable.

`allCov` All covariates of interest (including confounders) for estimating and testing their associations with microbiome. Default is all covariates in covData are of interest.

`refTaxa` Reference taxa specified by the user and will be used as the reference taxa.

`reguMethod` regularization approach used in phase 1 of the algorithm. Take value `"mcp"` or `"lasso"`, default is `"mcp"`.

`paraJobs` If `sequentialRun` is `FALSE`, this specifies the number of parallel jobs that will be registered to run the algorithm. Default is `8`. If specified as `NULL`, it will automatically detect the cores to decide the number of parallel jobs.

`bootB` Number of bootstrap samples for obtaining confidence interval of estimates in phase 2. The default is `500`.

`bootLassoAlpha` The significance level in phase 2. Default is `0.05`.

`standardize` This takes a logical value `TRUE` or `FALSE`. If `TRUE`, all design matrix X in phase 1 and phase 2 will be standardized in the analyses. Default is `FALSE`.

`sequentialRun` This takes a logical value `TRUE` or `FALSE`. Sometimes parallel jobs can not be successfully run for unknown reasons. For example, socket related errors may pop up or some slave cores return simple error instead of numerical results. In those scenarios, setting `sequentialRun = TRUE` may help, but it will take more time to run. Default is `TRUE`.

`seed` Random seed for reproducibility. Default is `1`.

Details

The `MZILN()` function can implement the Multivariate Zero-Inflated Logistic Normal model. It estimate and test the association given a user-specified reference taxon/OTU/ASV, whereas the `IFAA()` does not require any user-specified reference taxa.

Value

A list containing the estimation results.

- `analysisResults$estByRefTaxaList`: A list containing estimating results for all reference taxa and all the variables in 'allCov'. See details.
- `covariatesData`: A dataset containing all covariates used in the analyses.

References

Li et al.(2018) Conditional Regression Based on a Multivariate Zero-Inflated Logistic-Normal Model for Microbiome Relative Abundance Data. Statistics in Biosciences 10(3): 587-608

Zhang CH (2010) Nearly unbiased variable selection under minimax concave penalty. Annals of Statistics. 38(2):894-942.

Examples

```
data(dataM)
dim(dataM)
dataM[1:5, 1:8]
data(dataC)
dim(dataC)
dataC[1:5, ]
results <- MZILN(MicrobData = dataM,
                CovData = dataC,
                linkIDname = "id",
                allCov=c("v1","v2","v3"),
                refTaxa=c("rawCount11"))
```

Index