Package 'IFAA'

October 18, 2020

Title IFAA: Robust Association Identification and Inference for Absolute Abundance in Microbiome Analyses

Version 1.0.0

Description IFAA is a novel approach to make inference on the association of covariates with the absolute abundance (AA) of microbiome in an ecosystem. It can be also directly applied to relative abundance (RA) data to make inference on AA because the ratio of two RA is equal ratio of their AA. This algorithm can estimate and test the associations of interest while adjusting for potential confounders. The estimates of this method have easy interpretation like a typical regression analysis. High-dimensional covariates are handled with regularization and it is implemented by parallel computing. This algorithm finds optimal reference taxa/OTU/ASV and uses permutation to control FDR as described in the papers listed in the URL below.

License GNU General Public License version 2

Encoding UTF-8

URL https://arxiv.org/abs/1909.10101v3, https://pubmed.ncbi.nlm.nih.gov/30923584/

LazyData true

RoxygenNote 7.1.1

Depends picasso (>= 1.2.0),
expm (>= 0.999-3),
foreach (>= 1.4.3),
rlecuyer (>= 0.3-3),
Matrix (>= 1.2-14),
HDCI (>= 1.0-2),
parallel,
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. This argument can also take file directory path. For example, MicrobData="C:\\...\\microbiomeData.tsv".
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. This argument can also take file directory path. For example, CovData="C:\\...\\covariatesData.tsv".
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. Default is "mcp". Other methods are under development.
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

To model the association, the following equation is used:

$$\log(\mathcal{Y}_i^k)|\mathcal{Y}_i^k > 0 = \beta^{0k} + X_i^T\beta^k + W_i^T\gamma^k + Z_i^T b_i + \epsilon_i^k, \;\; k = 1, ..., K+1,$$

where $\mathcal{Y}_i^k$ is the AA of taxa k in subject i in the entire ecosystem. X_i is the covariate matrix. W_i is the confounder matrix. Z_i is the design matrix for random effects. β^k is the regression coefficients that will be estimated and tested with the `IFAA()` function. The challenge in microbiome analysis is that $\mathcal{Y}_i^k$ can not be oberved. What is observed is its small proportion: $Y_i^k = C_i\mathcal{Y}_i^k$ where C_i is an unknown number between 0 and 1 that denote the observed proportion. The IFAA method can handle this challenge by identifying and employing reference taxa. The `IFAA()` will estimate the parameter β^k and their 95% confidence intervals. High-dimensional X_i is handled by regularization.

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. This argument can also take file directory path. For example, MicrobData="C:\\...\\microbiomeData.tsv".
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. This argument can also take file directory path. For example, CovData="C:\\...\\covariatesData.tsv".
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. Default is "mcp". Other methods are under development.
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 regression model for MZILN() can be expressed as follows:

$$\log\left(\frac{\mathcal{Y}_i^k}{\mathcal{Y}_i^{K+1}}\right)|\mathcal{Y}_i^k > 0, \mathcal{Y}_i^{K+1} > 0 = \alpha^{0k} + \mathcal{X}_i^T\alpha^k + \epsilon_i^k, \quad k = 1, ..., K,$$

where $\mathcal{Y}_i^k$ is the AA of taxa k in subject i in the entire ecosystem. $\mathcal{Y}_i^{K+1}$ is the reference taxon (specified by user). $\mathcal{X}_i$ is the covariate matrix for all covariates including confounders. α^k is the regression coefficients along with their 95% confidence intervals that will be estimated by the MZILN() function. High-dimensional X_i is handled by regularization.

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