Package ‘MarZIC’

December 3, 2023

Type Package

Title Marginal Mediation Effects with Zero-Inflated Compositional Mediator

Version 1.0.0

Description A way to estimate and test marginal mediation effects for zero-inflated compositional mediators. Estimates of Natural Indirect Effect (NIE), Natural Direct Effect (NDE) of each taxon, as well as their standard errors and confident intervals, were provided as outputs. Zeros will not be imputed during analysis. See Wu et al. (2022) <doi:10.3390/genes13061049>.

URL https://www.mdpi.com/2073-4425/13/6/1049

License GPL-2

Imports Rcpp (>= 1.0.9), NlcOptim, betareg, doParallel, foreach, pracma, mathjaxr, dirmult, stats, parallel

Suggests knitr, rmarkdown, RUnit, BiocGenerics, BiocStyle

VignetteBuilder knitr

RdMacros mathjaxr

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.2.3

Encoding UTF-8

NeedsCompilation yes

R topics documented:

`MarZIC` *Marginal Mediation Model for Zero-Inflated Compositional Mediators*

Description

MarZIC is used for calculating marginal mediation effects for zero-inflated compositional mediators. For microbiome data, the marginal outcome model for the jth taxon (or OTU, ASV) is:

$$Y = \beta_0 + \beta_1 M_j + \beta_2 1_{M_j>0} + \beta_3 X + \beta_4 X 1_{M_j>0} + \beta_5 X M_j + \epsilon$$

where $1_{()}$ is indicator function, X is the covariate of interest and M_j is the relative abundance of the jth taxon. The probability of M_j being structure zero (ie, true zeros) is:

$$\log(\frac{\Delta_j}{1-\Delta_j}) = \gamma_0 + \gamma_1 X$$

The mean of M_j in compositional structure is modeled as:

$$\log(\frac{\mu_j}{1-\mu_j}) = \alpha_0 + \alpha_1 X$$

Typically, users just need to feed the first seven inputs to the function: ‘MicrobData‘, ‘CovData‘, ‘lib_name‘, ‘y_name‘, ‘x_name‘, ‘conf_name‘ and ‘taxa_of_interest‘.

Usage

```
MarZIC(
  MicrobData,
  CovData,
  lib_name,
  y_name,
  x_name,
  conf_name = NULL,
  x4_inter = TRUE,
  x5_inter = TRUE,
  taxa_of_interest = "all",
  mediator_mix_range = 1,
  transfer_to_RA = TRUE,
  num_cores = max(detectCores() - 2, 1),
  adjust_method = "fdr",
  fdr_rate = 0.05,
  taxDropThresh = 0.8,
  taxDropCount = 4 * (length(conf_name) + 2),
  zero_prop_NIE2 = 0.1,
  zero_count_NIE2 = 4 * (length(conf_name) + 2),
  SDThresh = 0.05,
  SDx = 0.05,
  SDy = 0.05
)
```

Arguments

`MicrobData` A dataset contains microbiome data. The microbiome data could be relative abundance or absolute abundance. Subjects with missing value will be removed during analysis.

`CovData` A dataset contains outcome, library size and covariates.

`lib_name` Name of library size variable within colData.

`y_name` Name of outcome variable within colData.

`x_name` Name of covariate of interest within colData.

`conf_name` Name of confounders within colData. Defaule is NULL, meaning no confounder.

`x4_inter` Whether to include the interaction term β_4. Default is TRUE.

`x5_inter` Whether to include the interaction term β_5. Default is TRUE.

`taxa_of_interest`
A character vector for taxa names indicating taxa that should be analyzed. Default is "all", meaning all taxa should be included into analysis.

`mediator_mix_range`
Number of mixtures in mediator. Default is 1, meaning no mixture.

`transfer_to_RA` Logical variable indicating whether the microbiome data should be transferred to relative abundance. Default is TRUE. If TRUE, microbiome data will be rescaled by its row sum.

`num_cores` Number of CPU cores to be used in parallelization task.

`adjust_method` P value adjustment method. Same as p.adjust. Default is "fdr".

`fdr_rate` FDR cutoff for significance. Default is 0.05.

`taxDropThresh` The threshold of dropping taxon due to high zero percentage. Default is 0.9, meaning taxon will be dropped for analysis if zero percentage is higher than 90%.

`taxDropCount` The threshold of dropping taxon due to not enough non-zero observation counts. Default is 4 * (length(conf_name)+2), meaning taxon will be dropped if non-zero observation is less than four times of number of covariates plus 1.

`zero_prop_NIE2` The threshold of zero percentage for calculating NIE2. Default is 0.1, meaning NIE2 will be calculated for taxon with zero percentage greater than 10%.

`zero_count_NIE2`
The threshold of zero counts for calculating NIE2. Default is 4 * (length(conf_name)+2), meaning NIE2 will be calculated for taxon with zero counts greater than four times of number of covariates plus 1.

`SDThresh` The threshold of dropping taxon due to low coefficient of variation (CV) to avoid constant taxon. Default is 0.05, meaning any taxon has CV less than 0.05 will be dropped.

`SDx` The threshold of stopping analysis due to low CV of covariate of interest. Default is 0.05, meaning when CV of covariate of interest is less than 0.05, the analysis will be stopped.

`SDy` The threshold of stopping analysis due to low CV of outcome. Default is 0.05, meaning when CV of outcome. is less than 0.05, the analysis will be stopped.

Value

A ‘list‘ of ‘4‘ datasets containing the results for ‘NIE1‘, ‘NIE2‘, ‘NDE‘, and ‘NIE‘. Each dataset has row representing each taxon, 6 columns for ‘Estimates‘, ‘Standard Error‘, ‘Lower bound for 95 ‘Adjusted p value‘, ‘Significance indicator‘.

References

Wu et al.(2022) MarZIC: A Marginal Mediation Model for Zero-Inflated Compositional Mediators with Applications to Microbiome Data. Genes 2022, 13, 1049.

Examples

```
{
library(MarZIC)

## A make up example with 2 taxon and 20 subjects.
set.seed(1)
nSub <- 20
nTaxa <- 2
## generate covariate of interest X
X <- rbinom(nSub, 1, 0.5)
## generate confounders
conf1<-rnorm(nSub)
conf2<-rbinom(nSub,1,0.5)
## generate mean of each taxon. All taxon are having the same mean for simplicity.
mu <- exp(-5 + X + 0.1 * conf1 + 0.1 * conf2) /
 (1 + exp(-5 + X + 0.1 * conf1 + 0.1 * conf2))
phi <- 10

## generate true RA
M_taxon<-t(sapply(mu,function(x) dirmult::rdirichlet(n=1,rep(x*phi,nTaxa))))

P_zero <- exp(-3 + 0.3 * X + 0.1 * conf1 + 0.1 * conf2) /
 (1 + exp(-3 + 0.3 * X + 0.1 * conf1 + 0.1 * conf2))

non_zero_ind <- t(sapply(P_zero,function(x) 1-rbinom(nTaxa,1,rep(x,nTaxa))))

True_RA<-t(apply(M_taxon*non_zero_ind,1,function(x) x/sum(x)))

## generate outcome Y based on true RA
Y <- 1 + 100 * True_RA[,1] + 5 * (True_RA[,1] > 0) + X + conf1 + conf2 + rnorm(nSub)

## library size was set to 10,000 for all subjects for simplicity.
libsize <- 10000

## generate observed RA
observed_AA <- floor(M_taxon*libsize*non_zero_ind)

observed_RA <- t(apply(observed_AA,1,function(x) x/sum(x)))
colnames(observed_RA)<-paste0("rawCount",seq_len(nTaxa))
CovData <- cbind(Y = Y, X = X, libsize = libsize, conf1 = conf1, conf2 = conf2)


## run the analysis
res <- MarZIC(
  MicrobData = observed_RA,
  CovData = CovData,
  lib_name = "libsize",
  y_name = "Y",
  x_name = "X",
  conf_name = c("conf1","conf2"),
  taxa_of_interest = NULL,
  num_cores = 1,
  mediator_mix_range = 1
)
}
```

Index