# Package 'MicrobiomeStat'

October 12, 2022

**Type** Package

**Title** Statistical Methods for Microbiome Compositional Data

**Version** 1.1

**Date** 2022-01-23

**Author** Xianyang Zhang [aut],
Jun Chen [aut, cre],
Huijuan Zhou [ctb]

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

**Description** A suite of methods for powerful and robust microbiome data analysis addressing zero-inflation, phylogenetic structure and compositional effects (Zhou et al. (2021)<arXiv:2104.00242>). The methods can be applied to the analysis of other (high-dimensional) compositional data arising from sequencing experiments.

**Depends** R (>= 3.5.0)

**Imports** ggplot2, matrixStats, Matrix, parallel, stats, utils, statmod, MASS, ggrepel, lmerTest, foreach, modeest, phyloseq

**NeedsCompilation** yes

**License** GPL-3

**Encoding** UTF-8

**Repository** CRAN

**Date/Publication** 2022-01-24 00:02:41 UTC

## R topics documented:

`linda` *Linear (Lin) Model for Differential Abundance (DA) Analysis of High-dimensional Compositional Data*

## Description

The function implements a simple, robust and highly scalable approach to tackle the compositional effects in differential abundance analysis of high-dimensional compositional data. It fits linear regression models on the centered log2-ratio transformed data, identifies a bias term due to the transformation and compositional effect, and corrects the bias using the mode of the regression coefficients. It could fit mixed-effect models for analysis of correlated data.

## Usage

```
linda(
  feature.dat,
  meta.dat,
  phyloseq.obj = NULL,
  formula,
  feature.dat.type = c('count', 'proportion'),
  prev.filter = 0,
  mean.abund.filter = 0,
  max.abund.filter = 0,
  is.winsor = TRUE,
  outlier.pct = 0.03,
  adaptive = TRUE,
  zero.handling = c('pseudo-count', 'imputation'),
  pseudo.cnt = 0.5,
  corr.cut = 0.1,
  p.adj.method = "BH",
  alpha = 0.05,
  n.cores = 1,
  verbose = TRUE
)
```

## Arguments

| | |
|---|---|
| `feature.dat` | a matrix of counts/proportions, row - features (OTUs, genes, etc) , column - samples. |
| `meta.dat` | a data frame containing the sample meta data. If there are NAs, the corresponding samples will be removed in the analysis. |
| `phyloseq.obj` | a `phyloseq` object containing the OTU count data and sample data. |
| `formula` | a character string for the formula. The formula should conform to that used by `lm` (independent data) or `lmer` (correlated data). For example: `formula = '~x1*x2+x3+(1|id)'`. At least one fixed effect is required. |

`feature.dat.type`
: the type of the feature data. It could be "count" or "proportion".

`prev.filter`
: the prevalence (percentage of non-zeros) cutoff, under which the features will be filtered. The default is 0.

`mean.abund.filter`
: the mean relative abundance cutoff, under which the features will be filtered. The default is 0.

`max.abund.filter`
: the max relative abundance cutoff, under which the features will be filtered. The default is 0.

`is.winsor`
: a logical value indicating whether winsorization should be performed to replace outliers (high values). The default is TRUE.

`outlier.pct`
: the expected percentage of outliers. These outliers will be winsorized. The default is 0.03.

`adaptive`
: a logical value indicating whether the approach to handle zeros (pseudo-count or imputation) will be determined based on the correlations between the log(sequencing depth) and the explanatory variables in `formula` when `feature.dat` is 'count'. If TRUE and the correlation p-value for any explanatory variable is smaller than or equal to `corr.cut`, the imputation approach will be used; otherwise, the pseudo-count approach will be used.

`zero.handling`
: a character string of 'pseudo-count' or 'imputation' indicating the zero handling method used when `feature.dat` is 'count'. If 'pseudo-count', `apseudo.cnt` will be added to each value in `feature.dat`. If 'imputation', then we use the imputation approach using the formula in the referenced paper. Basically, zeros are imputed with values proportional to the sequencing depth. When `feature.dat` is 'proportion', this parameter will be ignored and zeros will be imputed by half of the minimum for each feature.

`pseudo.cnt`
: a positive numeric value for the pseudo-count to be added if `zero.handling` is 'pseudo-count'. Default is 0.5.

`corr.cut`
: a numerical value between 0 and 1, indicating the significance level used for determining the zero-handling approach when `adaptive` is TRUE. Default is 0.1.

`p.adj.method`
: a character string indicating the p-value adjustment approach for addressing multiple testing. See R function `p.adjust`. Default is 'BH'.

`alpha`
: a numerical value between 0 and 1 indicating the significance level for declaring differential features. Default is 0.05.

`n.cores`
: a positive integer. If `n.cores > 1` and formula is in a form of mixed-effect model, `n.cores` parallels will be conducted. Default is 1.

`verbose`
: a logical value indicating whether the trace information should be printed out.

## Value

A list with the elements

variables — a vector of variable names of all fixed effects in formula. For example: formula = '~x1*x2+x3+(1|id)'. Suppose x1 and x2 are numerical, and x3 is a categorical variable of three levels: a, b and c. Then the elements of variables would be ('x1', 'x2', 'x3b', 'x3c', 'x1:x2').

bias — a numeric vector; each element corresponds to one variable in variables; the estimated bias of the regression coefficients due to the compositional effect.

output — a list of data frames with columns 'baseMean', 'log2FoldChange', 'lfcSE', 'stat', 'pvalue', 'padj', 'reject', 'df'; names(output) is equal to variables; the rows of the data frame corresponds to features. Note: if there are features being excluded due to filtering, the number of the rows of the output data frame will be not equal to the number of the rows of feature.dat. Features are identified by the row names. If the row names of feature.dat are NULL, then 1 : nrow(feature.dat) is set as the row names of feature.dat.

- baseMean: 2 to the power of the intercept coefficients (normalized by one million)
- log2FoldChange: bias-corrected coefficients
- lfcSE: standard errors of the coefficients
- stat: log2FoldChange / lfcSE
- pvalue: 2 * pt(-abs(stat), df)
- padj: p.adjust(pvalue, method = p.adj.method)
- reject: padj <= alpha
- df: degrees of freedom. The number of samples minus the number of explanatory variables (intercept included) for fixed-effect models; estimates from R package lmerTest with Satterthwaite method of approximation for mixed-effect models.

feature.dat.use — the actual feature table used in the differential analysis after filtering, winsorization and zero handling.

meta.dat.use — the meta data used in the abundance analysis (only variables in formula are stored; samples that have NAs are removed; numerical variables are scaled).

## Author(s)

Huijuan Zhou, Jun Chen, Xianyang Zhang

## References

Huijuan Zhou, Kejun He, Jun Chen, and Xianyang Zhang. LinDA: Linear Models for Differential Abundance Analysis of Microbiome Compositional Data. https://arxiv.org/abs/2104.00242

## Examples

```
data(smokers)

ind <- smokers$meta$AIRWAYSITE == 'Throat'
otu.tab <- as.data.frame(smokers$otu[, ind])
```

```
depth <- colSums(otu.tab)
meta <- cbind.data.frame(Smoke = factor(smokers$meta$SMOKER[ind]),
                         Sex = factor(smokers$meta$SEX[ind]),
                         Site = factor(smokers$meta$SIDEOFBODY[ind]),
                         SubjectID = factor(smokers$meta$HOST_SUBJECT_ID[ind]))

# Differential abundance analysis using the left throat data
ind1 <- meta$Site == 'Left' & depth >= 1000
linda.obj  <- linda(otu.tab[, ind1], meta[ind1, ], formula = '~Smoke+Sex',
           feature.dat.type = 'count',
           prev.filter = 0.1, is.winsor = TRUE, outlier.pct = 0.03,
           p.adj.method = "BH", alpha = 0.1)




linda.plot(linda.obj, c('Smokey', 'Sexmale'),
           titles = c('Smoke: n v.s. y', 'Sex: female v.s. male'),
           alpha = 0.1, lfc.cut = 1, legend = TRUE, directory = NULL,
            width = 11, height = 8)

rownames(linda.obj $output[[1]])[which(linda.obj $output[[1]]$reject)]


# Differential abundance analysis pooling both the left and right throat data
# Mixed effects model is used

ind  <- depth >= 1000
linda.obj <- linda(otu.tab[, ind], meta[ind, ], formula = '~Smoke+Sex+(1|SubjectID)',
           feature.dat.type = 'count',
           prev.filter = 0.1, is.winsor = TRUE, outlier.pct = 0.03,
           p.adj.method = "BH", alpha = 0.1)



# For proportion data
otu.tab.p <- t(t(otu.tab) / colSums(otu.tab))
ind1 <- meta$Site == 'Left' & depth >= 1000
lind.obj  <- linda(otu.tab[, ind1], meta[ind1, ], formula = '~Smoke+Sex',
           feature.dat.type = 'proportion',
           prev.filter = 0.1, is.winsor = TRUE, outlier.pct = 0.03,
           p.adj.method = "BH", alpha = 0.1)
```

---

`linda.plot` *Plot LinDA Results*

---

## Description

The function produces the effect size plot of the differential features and volcano plot based on the output from `linda`.

## Usage

```
linda.plot(
  linda.obj,
  variables.plot,
  titles = NULL,
  alpha = 0.05,
  lfc.cut = 1,
  legend = FALSE,
  directory = NULL,
  width = 11,
  height = 8
)
```

## Arguments

| | |
|---|---|
| linda.obj | return from function linda. |
| variables.plot | vector; variables whose results are to be plotted. For example, suppose the return value variables is equal to ('x1', 'x2', 'x3b', 'x3c', 'x1:x2'), then one could set variables.plot = c('x3b', 'x1:x2'). |
| titles | vector; titles of the effect size plot and volcano plot for each variable in variables.plot. Default is NULL. If NULL, the titles will be set as variables.plot. |
| alpha | a numerical value between 0 and 1; cutoff for padj. |
| lfc.cut | a positive numerical value; cutoff for log2FoldChange. |
| legend | TRUE or FALSE; whether to show the legends of the effect size plot and volcano plot. |
| directory | character; the directory to save the figures, e.g., getwd(). Default is NULL. If NULL, figures will not be saved. |
| width | the width of the graphics region in inches. See R function pdf. |
| height | the height of the graphics region in inches. See R function pdf. |

## Value

A list of ggplot2 objects.

| | |
|---|---|
| plot.lfc | a list of effect size plots. Each plot corresponds to one variable in variables.plot. |
| plot.volcano | a list of volcano plots. Each plot corresponds to one variable in variables.plot. |

## Author(s)

Huijuan Zhou, Jun Chen, Xianyang Zhang

## References

Huijuan Zhou, Kejun He, Jun Chen, and Xianyang Zhang. LinDA: Linear Models for Differential Abundance Analysis of Microbiome Compositional Data. https://arxiv.org/abs/2104.00242

## Examples

```
data(smokers)
ind <- smokers$meta$AIRWAYSITE == 'Throat' & smokers$meta$SIDEOFBODY == 'Left'
otu.tab <- as.data.frame(smokers$otu[, ind])
depth <- colSums(otu.tab)
meta <- cbind.data.frame(Smoke = factor(smokers$meta$SMOKER[ind]),
                         Sex = factor(smokers$meta$SEX[ind]))

ind  <- depth >= 1000
linda.obj <- linda(otu.tab[, ind], meta[ind, ], formula = '~Smoke+Sex',
           feature.dat.type = 'count',
           prev.filter = 0.1, is.winsor = TRUE, outlier.pct = 0.03,
           p.adj.method = "BH", alpha = 0.1)

linda.plot(linda.obj, c('Smokey', 'Sexmale'),
           titles = c('Smoke: n v.s. y', 'Sex: female v.s. male'),
           alpha = 0.1, lfc.cut = 1, legend = TRUE, directory = NULL,
            width = 11, height = 8)
```

---

# smokers *Microbiome data from the human upper respiratory tract (left and right throat)*

---

## Description

A dataset containing "otu", "tax", meta", "genus", family"

## Usage

```
data(smokers)
```

## Format

A list with elements

**otu** otu table, 2156 taxa by 290 samples

**tax** taxonomy table, 2156 taxa by 7 taxonomic ranks

**meta** meta table, 290 samples by 53 sample variables

**genus** 304 by 290

**family** 113 by 290

## Source

https://qiita.ucsd.edu/ study ID:524 Reference: 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.

# Index