

# Package ‘MicrobiomeStat’

November 19, 2021

**Type** Package

**Title** Statistical Methods for Microbiome Compositional Data

**Version** 1.0

**Date** 2021-11-10

**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](https://arxiv.org/abs/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

**NeedsCompilation** yes

**License** GPL-3

**Encoding** UTF-8

**Repository** CRAN

**Date/Publication** 2021-11-19 07:00:02 UTC

## R topics documented:

linda . . . . .	2
linda.plot . . . . .	5
smokers . . . . .	7
<b>Index</b>	<b>8</b>

---

 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 log<sub>2</sub>-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,
  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.
formula	a character string for the formula. The formula should conform to that used by <code>lm</code> (independent data) or <code>lmer</code> (correlated data). For example: <code>formula = '~x1*x2+x3+(1 id)'</code> . At least one fixed effect is required.
feature.dat.type	the type of the feature data. It could be "count" or "proportion".

<code>prev.filter</code>	the prevalence (percentage of non-zeros) cutoff, under which the features will be filtered. The default is 0.
<code>mean.abund.filter</code>	the mean relative abundance cutoff, under which the features will be filtered. The default is 0.
<code>max.abund.filter</code>	the max relative abundance cutoff, under which the features will be filtered. The default is 0.
<code>is.winsor</code>	a logical value indicating whether winsorization should be performed to replace outliers (high values). The default is TRUE.
<code>outlier.pct</code>	the expected percentage of outliers. These outliers will be winsorized. The default is 0.03.
<code>adaptive</code>	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 <code>formula</code> when <code>feature.dat</code> is 'count'. If TRUE and the correlation p-value for any explanatory variable is smaller than or equal to <code>corr.cut</code> , the imputation approach will be used; otherwise, the pseudo-count approach will be used.
<code>zero.handling</code>	a character string of 'pseudo-count' or 'imputation' indicating the zero handling method used when <code>feature.dat</code> is 'count'. If 'pseudo-count', <code>apseudo.cnt</code> will be added to each value in <code>feature.dat</code> . 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 <code>feature.dat</code> is 'proportion', this parameter will be ignored and zeros will be imputed by half of the minimum for each feature.
<code>pseudo.cnt</code>	a positive numeric value for the pseudo-count to be added if <code>zero.handling</code> is 'pseudo-count'. Default is 0.5.
<code>corr.cut</code>	a numerical value between 0 and 1, indicating the significance level used for determining the zero-handling approach when <code>adaptive</code> is TRUE. Default is 0.1.
<code>p.adj.method</code>	a character string indicating the p-value adjustment approach for addressing multiple testing. See R function <code>p.adjust</code> . Default is 'BH'.
<code>alpha</code>	a numerical value between 0 and 1 indicating the significance level for declaring differential features. Default is 0.05.
<code>n.cores</code>	a positive integer. If <code>n.cores</code> > 1 and <code>formula</code> is in a form of mixed-effect model, <code>n.cores</code> parallels will be conducted. Default is 1.
<code>verbose</code>	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 <code>variables</code> ; the estimated bias of the regression coefficients due to the compositional effect.
output	a list of data frames with columns <code>'baseMean'</code> , <code>'log2FoldChange'</code> , <code>'lfcSE'</code> , <code>'stat'</code> , <code>'pvalue'</code> , <code>'padj'</code> , <code>'reject'</code> , <code>'df'</code> ; <code>names(output)</code> is equal to <code>variables</code> ; 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 <code>feature.dat</code> . Features are identified by the row names. If the row names of <code>feature.dat</code> are <code>NULL</code> , then <code>1 : nrow(feature.dat)</code> is set as the row names of <code>feature.dat</code> . <ul style="list-style-type: none"> <li>• <code>baseMean</code>: 2 to the power of the intercept coefficients (normalized by one million)</li> <li>• <code>log2FoldChange</code>: bias-corrected coefficients</li> <li>• <code>lfcSE</code>: standard errors of the coefficients</li> <li>• <code>stat</code>: <code>log2FoldChange / lfcSE</code></li> <li>• <code>pvalue</code>: <code>2 * pt(-abs(stat), df)</code></li> <li>• <code>padj</code>: <code>p.adjust(pvalue, method = p.adj.method)</code></li> <li>• <code>reject</code>: <code>padj &lt;= alpha</code></li> <li>• <code>df</code>: degrees of freedom. The number of samples minus the number of explanatory variables (intercept included) for fixed-effect models; estimates from R package <code>lmerTest</code> with Satterthwaite method of approximation for mixed-effect models.</li> </ul>
<code>feature.dat.use</code>	the actual feature table used in the differential analysis after filtering, winsorization and zero handling.
<code>meta.dat.use</code>	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.

**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$SIDE OF BODY[ind]),
                        SubjectID = factor(smokers$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
linda.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

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

### Value

A list of `ggplot2` objects.

<code>plot.lfc</code>	a list of effect size plots. Each plot corresponds to one variable in <code>variables.plot</code> .
<code>plot.volcano</code>	a list of volcano plots. Each plot corresponds to one variable in <code>variables.plot</code> .

### 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.

## 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

\* **datasets**  
smokers, [7](#)

[linda](#), [2](#)  
[linda.plot](#), [5](#)

[smokers](#), [7](#)