

Package: MicrobiomeStat (via r-universe)

August 30, 2024

Type Package

Title Statistical Methods for Microbiome Compositional Data

Version 1.2

Date 2024-03-13

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. (2022)<[doi:10.1186/s13059-022-02655-5](https://doi.org/10.1186/s13059-022-02655-5)>). 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, parallel, stats, utils, Matrix, statmod, MASS, ggrepel, lmerTest, foreach, modeest

NeedsCompilation yes

License GPL-3

Encoding UTF-8

Repository CRAN

Date/Publication 2024-04-01 22:30:02 UTC

Contents

linda	2
linda.plot	6
linda.wald.test	8
smokers	9

Index	11
--------------	-----------

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,
  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 formula 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 formula 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	numeric vector; each element corresponds to one variable in variables; the estimated bias of the regression coefficients due to the compositional effect.
output	<p>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 taxa. Note: if there are taxa being excluded due to prev.cut, the number of the rows of the output data frame will be not equal to the number of the rows of otu.tab. Taxa are identified by the rownames. If the rownames of otu.tab are NULL, then 1 : nrow(otu.tab) is set as the rownames of otu.tab.</p> <p>baseMean: 2 to the power of the intercept coefficients (normalized by one million)</p> <p>log2FoldChange: bias-corrected coefficients</p> <p>lfcSE: standard errors of the coefficients</p> <p>stat: log2FoldChange / lfcSE</p> <p>pvalue: 2 * pt(-abs(stat), df)</p> <p>padj: p.adjust(pvalue, method = p.adjust.method)</p> <p>reject: padj <= alpha</p> <p>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.</p>
covariance	a list of data frames; the data frame records the covariances between a regression coefficient with other coefficients; names(covariance) is equal to variables; the rows of the data frame corresponds to taxa. If the length of variables is equal to 1, then the covariance is NULL.
otu.tab.use	the OTU table used in the abundance analysis (the otu.tab after the preprocessing: samples that have NAs in the variables in formula or have less than lib.cut read counts are removed; taxa with prevalence less than prev.cut are removed and data is winsorized if !is.null(winsor.quan); and zeros are treated, i.e., imputed or pseudo-count added).
meta.use	the meta data used in the abundance analysis (only variables in formula are stored; samples that have NAs or have less than lib.cut read counts are removed; numerical variables are scaled).
wald	<p>a list for use in Wald test. If the fitting model is a linear model, then it includes</p> <p>beta: a matrix of the biased regression coefficients including intercept and all fixed effects; the columns correspond to taxa</p> <p>sig: the standard errors; the elements corresponding to taxa</p> <p>X: the design matrix of the fitting model</p> <p>bias: the estimated biases of the regression coefficients including intercept and all fixed effects</p> <p>If the fitting model is a linear mixed-effect model, then it includes</p> <p>beta: a matrix of the biased regression coefficients including intercept and all fixed effects; the columns correspond to taxa</p> <p>beta.cov: a list of covariance matrices of beta; the elements corresponding to taxa</p>

rand.cov: a list with covariance matrices of variance parameters of random effects; the elements corresponding to taxa; see more details in the paper of 'lmerTest'

Joc.beta.cov.rand: a list of a list of Jacobian matrices of beta.cov with respect to the variance parameters; the elements corresponding to taxa

bias: the estimated biases of the regression coefficients including intercept and all fixed effects

Author(s)

Huijuan Zhou, Jun Chen, Xianyang Zhang

References

Zhou, H., He, K., Chen, J., Zhang, X. (2022). LinDA: linear models for differential abundance analysis of microbiome compositional data. *Genome biology*, 23(1), 95.

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> .

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

Zhou, H., He, K., Chen, J., Zhang, X. (2022). LinDA: linear models for differential abundance analysis of microbiome compositional data. *Genome biology*, 23(1), 95.

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)
```

linda.wald.test	<i>Wald test for bias-corrected regression coefficients</i>
-----------------	---

Description

The function implements Wald test for bias-corrected regression coefficients generated from the `linda` function. One can utilize the function to perform ANOVA-type analyses. For example, if you have a categorical variable with three levels, you can test whether all levels have the same effect.

Usage

```
linda.wald.test(
  linda.obj,
  L,
  model = c("LM", "LMM"),
  alpha = 0.05,
  p.adj.method = "BH"
)
```

Arguments

<code>linda.obj</code>	return from the <code>linda</code> function.
<code>L</code>	A matrix for testing $Lb = 0$, where b includes the intercept and all fixed effects from running <code>linda</code> . Thus the number of columns of <code>L</code> must be equal to <code>length(variables)+1</code> , where <code>variables</code> is from <code>linda.obj</code> , which does not include the intercept.
<code>model</code>	'LM' or 'LMM' indicating the model fitted in <code>{linda}</code> is linear model or linear mixed-effect model.
<code>alpha</code>	significance level for testing $Lb = 0$.
<code>p.adj.method</code>	P-value adjustment approach. See R function <code>p.adjust</code> . The default is 'BH'.

Value

A data frame with columns

<code>Fstat</code>	Wald statistics for each taxon
<code>df1</code>	The numerator degrees of freedom
<code>df2</code>	The denominator degrees of freedom
<code>pvalue</code>	$1 - pf(Fstat, df1, df2)$
<code>padj</code>	<code>p.adjust(pvalue, method = p.adj.method)</code>
<code>reject</code>	<code>padj <= alpha</code>

Author(s)

Huijuan Zhou <huijuanzhou2019@gmail.com> Jun Chen <Chen.Jun2@mayo.edu> Xianyang Zhang <zhangxiany@stat.tamu.edu>

References

Zhou, H., He, K., Chen, J., Zhang, X. (2022). LinDA: linear models for differential abundance analysis of microbiome compositional data. *Genome biology*, 23(1), 95.

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]))

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)
# L matrix (2x3) is designed to test the second (Smoke) and the third (Sex) coefficient to be 0.
# For a categorical variable > two levels, similar L can be designed to do ANOVA-type test.
L <- matrix(c(0, 1, 0, 0, 0, 1), nrow = 2, byrow = TRUE)
result <- linda.wald.test(linda.obj, L, 'LMM', alpha = 0.1)
```

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, [9](#)

[linda](#), [2](#)

[linda.plot](#), [6](#)

[linda.wald.test](#), [8](#)

[smokers](#), [9](#)