

Package ‘Maaslin2’

October 17, 2020

Title Maaslin2

Version 1.2.0

Depends R (>= 3.6)

Description MaAsLin2 is comprehensive R package for efficiently determining multivariable association between clinical metadata and microbial meta'omic features. MaAsLin2 relies on general linear models to accommodate most modern epidemiological study designs, including cross-sectional and longitudinal, and offers a variety of data exploration, normalization, and transformation methods. MaAsLin2 is the next generation of MaAsLin.

License MIT + file LICENSE

LazyData false

Imports robustbase, biglm, pcaPP, edgeR, metagenomeSeq, lpsymphony, pscl, pbapply, car, dplyr, vegan, chemometrics, ggplot2, pheatmap, cplm, logging, data.table, lmerTest, hash, optparse, MASS, MuMIn, grDevices, stats, utils

Suggests knitr, testthat (>= 2.1.0)

VignetteBuilder knitr

Collate fit.R utility_scripts.R viz.R Maaslin2.R

URL <http://huttenhower.sph.harvard.edu/maaslin2>

biocViews Metagenomics, Software, Microbiome, Normalization

BugReports <https://github.com/biobakery/maaslin2/issues>

git_url <https://git.bioconductor.org/packages/Maaslin2>

git_branch RELEASE_3_11

git_last_commit 9529c31

git_last_commit_date 2020-04-27

Date/Publication 2020-10-16

Author Himel Mallick [aut],
Ali Rahnavard [aut],
Lauren McIver [aut, cre]

Maintainer Lauren McIver <lauren.j.mciver@gmail.com>

R topics documented:

Maaslin2	2
Index	4

Maaslin2

MaAsLin2 is the next generation of MaAsLin. MaAsLin is a multi-variate statistical framework that finds associations between clinical metadata and potentially high-dimensional experimental data.

Description

MaAsLin2 was developed to find associations between microbiome multi'omics features and complex metadata in population-scale epidemiological studies. The software includes multiple analysis methods, normalization, and transform options to customize analysis for your specific study.

Usage

```
Maaslin2(
  input_data,
  input_metadata,
  output,
  min_abundance = 0.0,
  min_prevalence = 0.1,
  normalization = "TSS",
  transform = "LOG",
  analysis_method = "LM",
  max_significance = 0.25,
  random_effects = NULL,
  fixed_effects = NULL,
  correction = "BH",
  standardize = TRUE,
  cores = 1,
  plot_heatmap = TRUE,
  plot_scatter = TRUE,
  heatmap_first_n = 50
)
```

Arguments

<code>input_data</code>	The tab-delimited input file of features.
<code>input_metadata</code>	The tab-delimited input file of metadata.
<code>output</code>	The output folder to write results.
<code>min_abundance</code>	The minimum abundance for each feature.
<code>min_prevalence</code>	The minimum percent of samples for which a feature is detected at minimum abundance.
<code>max_significance</code>	The q-value threshold for significance.
<code>normalization</code>	The normalization method to apply.
<code>transform</code>	The transform to apply.
<code>analysis_method</code>	The analysis method to apply.
<code>random_effects</code>	The random effects for the model, comma-delimited for multiple effects.

<code>fixed_effects</code>	The fixed effects for the model, comma-delimited for multiple effects.
<code>correction</code>	The correction method for computing the q-value.
<code>standardize</code>	Apply z-score so continuous metadata are on the same scale.
<code>plot_heatmap</code>	Generate a heatmap for the significant associations.
<code>heatmap_first_n</code>	In heatmap, plot top N features with significant associations.
<code>plot_scatter</code>	Generate scatter plots for the significant associations.
<code>cores</code>	The number of R processes to run in parallel.

Value

Data.frame containing the results from applying the model.

Author(s)

Himel Mallick<hmallick@broadinstitute.org>,
Ali Rahnavard<rah@broadinstitute.org>,
Maintainers: Lauren McIver<lauren.j.mciver@gmail.com>,

Examples

```
input_data <- system.file(
  'extdata', 'HMP2_taxonomy.tsv', package="Maaslin2")
input_metadata <- system.file(
  'extdata', 'HMP2_metadata.tsv', package="Maaslin2")
fit_data <- Maaslin2(
  input_data, input_metadata, 'demo_output', transform = "AST",
  fixed_effects = c('diagnosis', 'dysbiosisnonIBD', 'dysbiosisUC', 'dysbiosisCD', 'antibiotics', 'age'),
  random_effects = c('site', 'subject'),
  standardize = FALSE)
```

Index

Maaslin2, [2](#)