

Package: CoreMicrobiomeR (via r-universe)

September 1, 2024

Type Package

Title Identification of Core Microbiome

Version 0.1.0

Imports fastmatch, vegan, SRS, edgeR, ggplot2, ggrepel, stats, plotly,
reshape2

Maintainer Mohammad Samir Farooqi <samirfarooqi8@gmail.com>

Description The Core Microbiome refers to the group of microorganisms that are consistently present in a particular environment, habitat, or host species. These microorganisms play a crucial role in the functioning and stability of that ecosystem. Identifying these microorganisms can contribute to the emerging field of personalized medicine. The 'CoreMicrobiomeR' is designed to facilitate the identification, statistical testing, and visualization of this group of microorganisms. This package offers three key functions to analyze and visualize microbial community data. This package has been developed based on the research papers published by Pereira et al.(2018) <doi:10.1186/s12864-018-4637-6> and Beule L, Karlovsky P. (2020) <doi:10.7717/peerj.9593>.

License GPL-3

Encoding UTF-8

LazyData true

Author Sorna A M [aut], Mohammad Samir Farooqi [aut, cre], Dwijesh Chandra Mishra [aut], Krishna Kumar Chaturvedi [aut], Anu Sharma [aut], Prawin Arya [aut], Sudhir Srivastava [aut], Sharanbasappa [aut], Girish Kumar Jha [aut], Kabilan S [ctb]

Depends R (>= 2.10)

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2024-04-03 20:03:02 UTC

Contents

CoreMicrobiome	2
demo_md	4
demo_otu	4
demo_tax	5
group_bar_plots	6
significance	7
stacked_bar_plots	8
visualize	9

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

CoreMicrobiome	<i>Identification of Core Microbiome</i>
----------------	--

Description

This function provides a comprehensive pipeline for processing OTU (Operational Taxonomic Unit) tables, taxonomic tables, and metadata tables. It applies various filtering methods based on user-defined parameters to select core OTUs and non-core OTUs calculates alpha and beta diversity measures. The pipeline can be customized with different normalization methods and filtering criteria. Taxa are ranked in descending according to the cumulative sum obtained. This method assigns taxa to the core if they are in the top X% of reads. Any taxa which appears before some cutoff percentage is included in the core.

Usage

```
CoreMicrobiome(otu_table, tax_table, metadata_table, filter_type, ...,
method, beta_diversity_method, top_percentage)
```

Arguments

otu_table	A dataframe of OTUs where the first row is the OTU ID and column names refer to sites/sample names.
tax_table	A dataframe of taxonomies where the first row is the OTU ID and column names refer to taxonomic classification.
metadata_table	A dataframe of sites/samples where the first row is the sites/sample names and column names refer to groups of samples.
filter_type	Filtering method type, includes "abundance_fun_filter", "occupancy_fun_filter", or "combined_filter".
...	Other parameters. These are ignored, except in filter_type = "abundance_fun_filter" which accepts min_count, prop, min_total_count parameter, and in filter_type = "occupancy_fun_filter" which accepts percent parameter, and also filter_type = "combined_filter" which accepts percent, min_count, prop, min_total_count parameters.

method Different normalization methods, includes "rrarefy", "srs", "css", "tmm", "tmmwsp", "rle", "upperquartile" or "none". The default method is tmm.

beta_diversity_method Different beta diversity methods, includes "bray", "jaccard", "mountford". The default method is bray.

top_percentage Percentage used for Core OTUs identification and the default is 10 percent.

Value

This function gives the list which consist of following results.

'final_otu_table_bef_filter' otu_table obtained after sorting according to the provided tax_table and metadata_table

'filtered_md_table' metadata_table obtained after sorting according to the provided otu_table

'final_otu_aft_filter' otu_table obtained after filtering according to the user defined filtering method

'normalized_table' normalized_otu_table obtained after normalizing according to the user defined normalization method

'alpha_diversity' Alpha diversity measures of the samples

'beta_diversity' Beta diversity measures between the samples

'core_otus' Core OTUs obtained

'non_core_otus' Non Core OTUs obtained

'core_otus_tax' Taxonomy of the obtained Core OTUs

'core_otus_count_data' Original count data of the obtained core OTUs

'core_otus_relative_abundance' Relative abundance data of the obtained core OTUs

References

Pereira, M., Wallroth, M., Jonsson, V. et al. (2018). Comparison of normalization methods for the analysis of metagenomic gene abundance data. *BMC Genomics* 19, 274. <doi:https://doi.org/10.1186/s12864-018-4637-6>

Beule L, Karlovsky P. (2020). Improved normalization of species count data in ecology by scaling with ranked subsampling (SRS): application to microbial communities. *PeerJ* 8:e9593.<doi:https://doi.org/10.7717/peerj.9593>

Examples

```
#To run input data
core_1 <- CoreMicrobiome(
  otu_table = demo_otu,
  tax_table = demo_tax,
  metadata_table = demo_md,
  filter_type = "occupancy_fun_filter", #Or "abundance_fun_filter", Or "combined_filter"
  percent = 0.5,
  method = "css", # Or "srs", "rrarefy", "tmm", "tmmwsp", "rle", "upperquartile", "none"
  beta_diversity_method = "jaccard",
  top_percentage = 10 # Adjust the percentage as needed for core/non-core OTUs
)
```

```
#To view the core otus obtained
core_1[["core_otus"]]
#To view the taxonomy of the obtained core otus
core_1[["core_otus_tax"]]
```

demo_md

Arabidopsis thaliana - Metadata dataset

Description

This dataset was given by Lundberg et al., (2012). The metadata table contains additional information about each sample (root) included in the study. It typically includes details about the experimental conditions, environmental factors, sample genotype, location, and other relevant contextual information. Metadata is crucial for linking the microbial community data to specific experimental variables and understanding how the root microbiome might vary in response to different factors. The original dataset contains 1049 samples(rows) and factors like soil type, genotype, treatment, developmental stage and replication information in columns for each particular sample.

Usage

demo_md

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 103 rows and 6 columns.

Details

Here only the portion of the dataset is taken for running the functions. The dataset contains 103 rows and 6 columns.

Source

[doi:10.1038/nature11237](https://doi.org/10.1038/nature11237)

demo_otu

Arabidopsis thaliana - OTU dataset

Description

This dataset was given by Lundberg et al., (2012). The OTU table is a central part of the data set. It represents the abundance or presence/absence of different microbial taxa (operational taxonomic units) in the root samples of *Arabidopsis thaliana*. Each column in the OTU table corresponds to a specific sample (root) from the study, and each row represents a different OTU, which could be a species or a group of closely related organisms. The table contains numerical values representing the count of each OTU in the corresponding samples. The original dataset contains 18783 rows of OTUs and 1439 samples(columns).

Usage

demo_otu

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 188 rows and 1440 columns.

Details

Here only the portion of the dataset is taken for running the functions.

The dataset contains 188 rows and 1440 columns.

Source

[doi:10.1038/nature11237](https://doi.org/10.1038/nature11237)

demo_tax

Arabidopsis thaliana - Taxonomy dataset

Description

This dataset was given by Lundberg et al., (2012). The taxonomy table provides information about the taxonomic identity of the OTUs listed in the OTU table. Each row in the taxonomy table corresponds to an OTU from the OTU table, and the columns provide details about the taxonomic classification of that OTU, such as kingdom, phylum, class, order, family, genus, and species. This information allows researchers to identify the microbial species or groups that are present in the root samples. The original dataset contains 777 rows of OTUs and Phylum, Class, Order, Family in columns corresponding to particular OTU.

Usage

demo_tax

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 188 rows and 5 columns.

Details

Here only the portion of the dataset is taken for running the functions. The dataset contains 188 rows and 5 columns.

Source

[doi:10.1038/nature11237](https://doi.org/10.1038/nature11237)

group_bar_plots	<i>Grouped Bar Plots Based on Sample Size</i>
-----------------	---

Description

The `grouped_bar_plots` function is designed for generating grouped bar plots to visualize data. It takes a OTU table before filtering and OTU table after filtering as input containing data for multiple samples and creates a series of grouped bar plots, each representing a specific group of samples.

Usage

```
group_bar_plots(otu_table_bef_filtering, otu_table_aft_filtering,
  num_samples_per_plot)
```

Arguments

`otu_table_bef_filtering`
A data frame of OTUs before filtering where the first row is the OTU ID and column names refer to sites/sample names

`otu_table_aft_filtering`
A data frame of OTUs after filtering where the first row is the OTU ID and column names refer to sites/sample names

`num_samples_per_plot`
The number of samples to be displayed in each grouped bar plot.

Value

A list of interactive grouped bar plots, showing the change in sample size before and after filtering OTU table

Examples

```
#To run input data
core_1 <- CoreMicrobiome(
  otu_table = demo_otu,
  tax_table = demo_tax,
  metadata_table = demo_md,
  filter_type = "occupancy_fun_filter", #Or "abundance_fun_filter", Or "combined_filter"
  percent = 0.5,
  method = "css", # Or "srs", "rrarefy", "tmm", "tmmwsp", "rle", "upperquartile", "none"
  beta_diversity_method = "jaccard",
  top_percentage = 10 # Adjust the percentage as needed for core/non-core OTUs
)
#To run grouped bar plot function
plot_group_bar <- group_bar_plots(core_1$final_otu_table_bef_filter,
  core_1$final_otu_table_aft_filter, 10)
#To view the grouped bar plot
plot_group_bar[[1]]
```

 significance

Testing the Significance of the Identified Core Microbiome

Description

This function performs a two-sample variance test to assess the statistical significance of differences in abundance between core OTUs and non-core OTUs. It takes two data frames as input, representing the abundance of core OTUs and non-core OTUs, and returns the results of the variance test. It tells whether the identified core represents the particular environment or habitat.

Usage

```
significance(core_ids, non_core_ids)
```

Arguments

core_ids	A Dataframe of core OTUs where the first row is the OTU ID and column names refer to sites/sample names
non_core_ids	A Dataframe of non_core OTUs where the first row is the OTU ID and column names refer to sites/sample names

Value

This function gives the list which consist of following results.

- ‘statistic’ Calculated F test statistic
- ‘parameter’ The numerator degrees of freedom (num df), and the denominator degrees of freedom (denom df)
- ‘p-value’ Probability value
- ‘alternative’ The alternative hypothesis for this test is that the true ratio of variances is not equal to 1. This suggests that the variances of the two data sets are different
- ‘conf.int’ 95 percent confidence interval limit for the ratio of variances
- ‘estimate’ Ratio of variances between core_data and non_core_data calculated
- ‘method’ The test performed is an F test, which compares the variances of the two data sets
- ‘data.name’ The data used for the test are core_ids and non_core_ids

Examples

```
#To run input data
core_1 <- CoreMicrobiome(
  otu_table = demo_otu,
  tax_table = demo_tax,
  metadata_table = demo_md,
  filter_type = "occupancy_fun_filter", #Or "abundance_fun_filter", Or "combined_filter"
  percent = 0.5,
  method = "css", # Or "srs", "rrarefy", "tmm", "tmmwsp", "rle", "upperquartile", "none"
```

```

beta_diversity_method = "jaccard",
top_percentage = 10 # Adjust the percentage as needed for core/non-core OTUs
)
#To run significance test
f_test <- significance(core_1[["core_otus"]] , core_1[["non_core_otus"]] )

#To view the significance test result
f_test

```

stacked_bar_plots *Stacked Bar Plots Based on Relative Abundance Data*

Description

This function generates stacked bar plots for visualizing the relative abundance data of different operational taxonomic units (OTUs) in various samples.

Usage

```
stacked_bar_plots(data, num_samples_per_plot)
```

Arguments

data A data frame containing the relative abundance data for the OTUs. The first column should contain the OTU IDs, and the subsequent columns should represent samples.

num_samples_per_plot The number of samples to be displayed in each stacked bar plot.

Value

A list of interactive stacked bar plots, one for each group of samples, showing the relative abundance of OTUs in the samples.

Examples

```

#To run input data
core_1 <- CoreMicrobiome(
  otu_table = demo_otu,
  tax_table = demo_tax,
  metadata_table = demo_md,
  filter_type = "occupancy_fun_filter", #Or "abundance_fun_filter", Or "combined_filter"
  percent = 0.5,
  method = "css", # Or "srs", "rrarefy", "tmm", "tmmwsp", "rle", "upperquartile", "none"
  beta_diversity_method = "jaccard",
  top_percentage = 10 # Adjust the percentage as needed for core/non-core OTUs
)
#To run the stacked bar plots function
stacked_plots <- stacked_bar_plots(core_1$core_otus_relative_abundance, 10)

```



```
#To view the stacked bar plot
stacked_plots[[1]]
```

visualize

Visualizing the effect of minimum count on the core size

Description

The visualize function generates interactive line plots that allow users to explore the impact of different min_count values on the number of core OTUs. Users can interact with the plots to examine the relationship between filtering criteria and core OTU identification visually.

Usage

```
visualize(filtered_otu, min_count_val, max_count_val, count_val_interval,
prop, min_total_count, method, top_percentage)
```

Arguments

filtered_otu	A dataframe of OTUs obtained before filtering which is retrieved from CoreMicrobiome function where the first row is the OTU ID and column names refer to sites/sample names
min_count_val	A numeric value of Minimum count for each OTU to be present in each to be included after the filtering
max_count_val	A numeric value of Maximum count for each OTU to be present in each to be included after the filtering
count_val_interval	Count value interval for each OTU to be present in each to be included after the filtering
prop	Minimum proportion of samples in which an OTU must be present
min_total_count	Minimum total count for each OTU to be included after the filtering
method	Different normalization methods, includes "rrarefy", "srs", "css", "tmm", or "none"
top_percentage	Percentage used for obtaining the Core OTUs

Value

This function gives a line plot which shows change in number of core OTUs with minimum count

Examples

```
#To run input data
core_1 <- CoreMicrobiome(
  otu_table = demo_otu,
  tax_table = demo_tax,
  metadata_table = demo_md,
  filter_type = "occupancy_fun_filter", #Or "abundance_fun_filter", Or "combined_filter"
  percent = 0.5,
  method = "css", # Or "srs", "rrarefy", "tmm", "tmmwsp", "rle", "upperquartile", "none"
  beta_diversity_method = "jaccard",
  top_percentage = 10 # Adjust the percentage as needed for core/non-core OTUs
)
#To view the line plot
visualize(filtered_otu = core_1[["final_otu_table_bef_filter"]],
  min_count_val = 5,
  max_count_val = 25,
  count_val_interval = 5,
  prop = 0.1,
  min_total_count = 10,
  method = "srs",
  top_percentage = 10)
```

Index

* datasets

demo_md, [4](#)

demo_otu, [4](#)

demo_tax, [5](#)

CoreMicrobiome, [2](#)

demo_md, [4](#)

demo_otu, [4](#)

demo_tax, [5](#)

group_bar_plots, [6](#)

significance, [7](#)

stacked_bar_plots, [8](#)

visualize, [9](#)