

Tutorial for R MicrobiomeAnalysis package

Hua Zou

2023-01-12

Contents

1	Introduction	5
1.1	Input data	5
1.2	Methods	7
1.3	Outline of this tutorial	7
1.4	Citation	8
2	How to install MicrobiomeAnalysis	9
2.1	Installation	9
2.2	Vignette	10
2.3	Authors	10
3	Data Processing	11
3.1	Loading Packages	11
3.2	Importing Data	12
3.3	Extracting specific levels	13
3.4	Summarizing specific levels	14
3.5	Data Transformation	14
3.6	Data Imputation	15
3.7	Data Normalization	16
3.8	Data Scaling	17
3.9	Data Trimming	18
3.10	Data Filtering	19
3.11	Systematic Information	20

Chapter 1

Introduction

This tutorial describes how to perform bioinformatics data analysis of microbiota using metagenomic data. It will focus on data processing, diversity analysis, biomarker identification and association investigation from microbiota expression profile which derived from **amplicon (16s rRNA)** or **Whole genome (metagenomics)** sequencing data.

The data pre-processing module includes six main procedures:

- **Data Transformation**
- **Data Imputation**
- **Data Normalization**
- **Data Scaling**
- **Data Trimming**
- **Data Filtering**

The downstream analysis includes several modules:

- **Diversity analysis**
- **Ordination analysis**
- **Clustering analysis**
- **Differential analysis**
- **Association analysis**

1.1 Input data

when using data pre-processing functions, it requires that the inputs which are from 16s or metagenomics data should be converted into phyloseq-class object:

- **Types of data.** Users can handle 16s or metagenomics data coming from any platform on a transformed phyloseq-class object when they use `import_dada2` or `import_qiime2`.

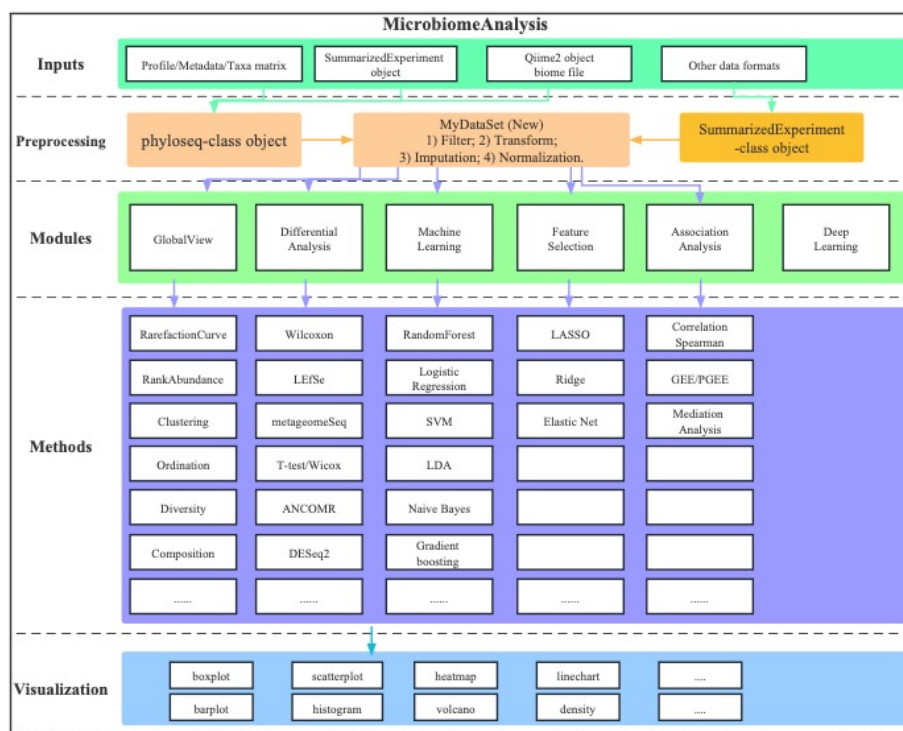


Figure 1.1: Flowchat of MicrobiomeAnalysis

- **Transformation.** The package provides multiple methods to transform individual values, such as `log10`,
- **Imputation.** Missing value imputation methods also were provided by this package.
- **Normalisation.** `MicrobiomeAnalysis` also has so many methods to normalize data sample by sample, which could remove technical effects.
- **Prefiltering.** Removing low abundant taxa also could be done by using `MicrobiomeAnalysis`.
- **Data format.** Most functions of `MicrobiomeAnalysis` are using `phyloseq`-class object as input. Therefore, users could convert their inputs into `phyloseq`-class object.

1.2 Methods

1.2.1 Some background knowledge

We list here the main methodological or theoretical concepts you need to know to be able to efficiently apply `MicrobiomeAnalysis`:

- **phyloseq-class, SummarizedExperiment-class:** the former object contains otu table, sample data and taxa table etc which are usually used in metagenomic data analysis. The latter one comprises expression data and metadata of gene, metabolites etc.

1.2.2 Key publications

- **phyloseq object.** (McMurdie and Holmes, 2013) `phyloseq`: an R package for reproducible interactive analysis and graphics of microbiome census data.
- **Permutational Multivariate Analysis of Variance (PERMANOVA).** (Anderson, 2014) Permutational multivariate analysis of variance (PERMANOVA).
- **ALDEx2.** (Fernandes et al., 2014) Unifying the analysis of high-throughput sequencing datasets: characterizing RNA-seq, 16S rRNA gene sequencing and selective growth experiments by compositional data analysis.

1.3 Outline of this tutorial

- **Chapter 2:** How to install `MicrobiomeAnalysis`
- **Chapter 3:** Data Processing

1.4 Citation

Kindly cite by using `citation("MicrobiomeAnalysis")` if you think `MicrobiomeAnalysis` helps you. Alternative way is Zou H (2022). `MicrobiomeAnalysis`: An R package for analysis and visualization in metagenomics. R package version 1.0.3, <URL:<https://github.com/HuaZou/MicrobiomeAnalysis/>>.

Chapter 2

How to install MicrobiomeAnalysis

Before installation, you should install the two following software.

- R 4.1.2 (R Core Team, 2021) or later release [Download link](#).
- Rstudio Desktop [Download link](#).

2.1 Installation

First, installing either R package devtools or remotes:

```
if (!requireNamespace(c("devtools", "remotes"), quietly = TRUE)) {  
  install.packages(c("devtools", "remotes"))  
}  
  
devtools::install_github("HuaZou/MicrobiomeAnalysis")  
# remotes::install_github("HuaZou/MicrobiomeAnalysis")  
  
# specific release  
devtools::install_github("HuaZou/MicrobiomeAnalysis@release-v1.0.3")
```

Alternatively, you can install the latest released package from releases:

```
install.packages("MicrobiomeAnalysis*.tar.gz", repos = NULL, type = "source")
```

The MicrobiomeAnalysis package took dplyr, phyloseq, purrr, magrittr, tibble, metagenomeSeq, biomformat, SummarizedExperiment, ggplot2 and caret etc as directly import packages.

2.2 Vignette

For brief introduction to use `MicrobiomeAnalysis`, please refer to vignette and choose *html* format to do a practice.

```
utils::browseVignettes(package="MicrobiomeAnalysis")
```

2.3 Authors

- Hua Zou

Chapter 3

Data Processing

Outline of this Chapter:

- Loading Packages
- Importing Data
- Extracting specific levels
- Summarizing specific levels
- Data Transformation
- Data Imputation
- Data Normalization
- Data Scaling
- Data Trimming
- Data Filtering

3.1 Loading Packages

```
library(dplyr)
library(tibble)
library(phyloseq)
library(SummarizedExperiment)
library(MicrobiomeAnalysis)
```

3.2 Importing Data

- Converting the output of dada2 into phyloseq object

```
seq_tab <- readRDS(
  system.file("extdata", "dada2_seqtab.rds",
    package = "MicrobiomeAnalysis"))
tax_tab <- readRDS(
  system.file("extdata", "dada2_taxtab.rds",
    package = "MicrobiomeAnalysis"))
sam_tab <- read.table(
  system.file("extdata", "dada2_samdata.txt",
    package = "MicrobiomeAnalysis"),
  sep = "\t", header = TRUE, row.names = 1)

ps_dada2 <- import_dada2(
  seq_tab = seq_tab,
  tax_tab = tax_tab,
  sam_tab = sam_tab)

ps_dada2
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 232 taxa and 20 samples ]
## sample_data() Sample Data: [ 20 samples by 4 sample variables ]
## tax_table() Taxonomy Table: [ 232 taxa by 6 taxonomic ranks ]
## refseq() DNASTringSet: [ 232 reference sequences ]
```

- Converting the qiime2 output of dada2 into phyloseq object

```
otuqza_file <- system.file(
  "extdata", "table.qza",
  package = "MicrobiomeAnalysis")
taxaqa_file <- system.file(
  "extdata", "taxonomy.qza",
  package = "MicrobiomeAnalysis")
sample_file <- system.file(
  "extdata", "sample-metadata.tsv",
  package = "MicrobiomeAnalysis")
treeqza_file <- system.file(
  "extdata", "tree.qza",
  package = "MicrobiomeAnalysis")
ps_qiime2 <- import_qiime2(
  otu_qza = otuqza_file, taxa_qza = taxaqa_file,
  sam_tab = sample_file, tree_qza = treeqza_file
)
```

```
ps_qiime2
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 770 taxa and 34 samples ]
## sample_data() Sample Data: [ 34 samples by 9 sample variables ]
## tax_table() Taxonomy Table: [ 770 taxa by 7 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 770 tips and 768 internal nodes ]
```

- Converting inputs into SummarizedExperiment object

```
data("Zeybel_2022_protein")
assay <- SummarizedExperiment::assay(Zeybel_2022_protein) %>%
  data.frame()
rowData <- SummarizedExperiment::rowData(Zeybel_2022_protein) %>%
  data.frame()
colData <- SummarizedExperiment::colData(Zeybel_2022_protein) %>%
  data.frame()
metadata <- list(lab="hua", type="protein")

assay <- assay[1:10, 1:10]

se_protein <- import_SE(
  object = assay,
  rowdata = rowData,
  coldata = colData,
  metadata = metadata)

se_protein
```

```
## class: SummarizedExperiment
## dim: 10 10
## metadata(2): lab type
## assays(1): ''
## rownames(10): IL8 VEGFA ... uPA IL6
## rowData names(3): ProteinID LOD prop
## colnames(10): P101001 P101003 ... P101013 P101016
## colData names(47): PatientID Gender ... Right_leg_fat_free_mass Right_leg_total_body_water
```

3.3 Extracting specific levels

- Extracting “Genus” levels phyloseq object

```
ps_genus <- aggregate_taxa(x = ps_dada2,
                           level = "Genus")
ps_genus
```

```
## phyloseq-class experiment-level object
```

```
## otu_table()   OTU Table:           [ 66 taxa and 20 samples ]
## sample_data() Sample Data:        [ 20 samples by 4 sample variables ]
## tax_table()   Taxonomy Table:      [ 66 taxa by 7 taxonomic ranks ]
```

3.4 Summarizing specific levels

- Phyloseq object contains from Kingdom to the the specific taxonomic levels (Phylum)

```
ps_summarize_genus <- summarize_taxa(
  ps = ps_dada2,
  level = "Genus")
ps_summarize_genus

## phyloseq-class experiment-level object
## otu_table()   OTU Table:           [ 66 taxa and 20 samples ]
## sample_data() Sample Data:        [ 20 samples by 4 sample variables ]
## tax_table()   Taxonomy Table:      [ 66 taxa by 1 taxonomic ranks ]
```

3.5 Data Transformation

5 methods to transform individual values (by **individual value**).

- “log10”, the transformation is $\log_{10}(\text{object})$, and if the data contains zeros the transformation is $\log_{10}(1 + \text{object})$.
- “log10p”, the transformation is $\log_{10}(1 + \text{object})$.
- “SquareRoot”, the transformation is Square Root.
- “CubicRoot”, the transformation is Cubic Root.
- “logit”, the transformation is Zero-inflated Logit Transformation (Does not work well for microbiome data).

Here is for phyloseq-class.

```
data("Zeybel_2022_gut")
ps_transform <- transform_abundances(
  object = Zeybel_2022_gut,
  level = "Phylum",
  transform = "log10p")

head(ps_transform@otu_table@.Data[, 1:5], 3)

##               P101003    P101007    P101010    P101012    P101018
## p__Actinobacteria -1.651096 -2.0986943 -2.5452009 -1.6773074 -1.8931057
## p__Bacteroidetes  -0.330115 -0.1437857 -0.3379679 -0.1574846 -0.1295813
## p__Chloroflexi     0.000000  0.0000000  0.0000000  0.0000000  0.0000000
```

- Here is for SummarizedExperiment-class.

```
data("Zeybel_2022_protein")
se_transform <- transform_abundances(
  object = Zeybel_2022_protein,
  transform = "SquareRoot")

head(assay(se_transform)[, 1:5], 3)

##          P101001 P101003 P101004 P101007 P101009
## IL8    2.246691 2.063092 2.170020 2.400327 2.168238
## VEGFA  3.344589 3.302819 3.246115 3.321395 3.258289
## CD8A   3.398494 3.185073 3.132900 3.262700 3.075131
```

3.6 Data Imputation

11 methods to impute missing value (NAs or Zeros)

- “LOD”: specific Limit Of Detection which provides by user.
- “half_min”: half minimal values across samples except zero.
- “median”: median values across samples except zero.
- “mean”: mean values across samples except zero.
- “min”: minimal values across samples except zero.
- “knn”: k-nearest neighbors samples.
- “rf”: nonparametric missing value imputation using Random Forest.
- “global_mean”: a normal distribution with a mean that is down-shifted from the sample mean and a standard deviation that is a fraction of the standard deviation of the sample distribution.
- “svd”: missing values imputation based Singular value decomposition.
- “QRILC”: missing values imputation based quantile regression. (default: “none”).

phyloseq-class as inputs

```
data("Zeybel_2022_gut")
ps_impute <- impute_abundance(
  object = Zeybel_2022_gut,
  level = "Phylum",
  group = "LiverFatClass",
  ZerosAsNA = TRUE,
  RemoveNA = TRUE,
  cutoff = 20,
  method = "knn")
```

```
head(ps_impute@otu_table@.Data[, 1:5], 3)
```

	P101003	P101007	P101010	P101012	P101018
p__Actinobacteria	0.0223308	0.0079672	0.0028497	0.0210229	0.0127907
p__Bacteroidetes	0.4676113	0.7181486	0.4592320	0.6958497	0.7420252
p__Firmicutes	0.4818712	0.2715033	0.4577191	0.2712132	0.1701021

- Inputs is from SummarizedExperiment-class.

```
data("Zeybel_2022_protein")
se_impute <- impute_abundance(
  object = Zeybel_2022_protein,
  group = "LiverFatClass",
  ZerosAsNA = TRUE,
  RemoveNA = TRUE,
  cutoff = 20,
  method = "knn")
head(assay(se_impute)[, 1:5], 3)
```

	P101001	P101003	P101004	P101007	P101009
IL8	5.047325	4.25600	4.70867	5.76131	4.70094
VEGFA	11.186140	10.90848	10.53712	11.03153	10.61631
CD8A	11.549635	10.14454	9.81491	10.64507	9.45627

3.7 Data Normalization

Normalizing the OTU_table in phyloseq-class object sample by sample to reduce the effects of systematic differences such as library size (**by sample**).

- “rarefy”: random subsampling counts to the smallest library size in the data set.
- “TSS”: total sum scaling, also referred to as “relative abundance”, the abundances were normalized by dividing the corresponding sample library size.
- “TMM”: trimmed mean of m-values. First, a sample is chosen as reference. The scaling factor is then derived using a weighted trimmed mean over the differences of the log-transformed gene-count fold-change between the sample and the reference.
- “RLE”, relative log expression, RLE uses a pseudo-reference calculated using the geometric mean of the gene-specific abundances over all samples. The scaling factors are then calculated as the median of the gene counts ratios between the samples and the reference.

- “CSS”: cumulative sum scaling, calculates scaling factors as the cumulative sum of gene abundances up to a data-derived threshold.
- “CLR”: centered log-ratio normalization.
- “CPM”: pre-sample normalization of the sum of the values to 1e+06.

```
data("caporaso")
ps_norm <- normalize(
  object = caporaso,
  method = "TSS")

head(ps_norm@otu_table@.Data[, 1:5], 3)
```

```
##                               L1S140          L1S208 L1S8 L1S281 L3S242
## New.CleanUp.ReferenceOTU647      0 0.0000000000      0      0      0
## 14030                            0 0.0000000000      0      0      0
## New.CleanUp.ReferenceOTU858      0 0.0001013993      0      0      0
```

3.8 Data Scaling

Data scaling adjusts each variable/feature by a scaling factor computed based on the dispersion of the variable (**by variable/feature**).

- “mean_center”: values minus mean statistic.
- “zscore”: mean-centered and divided by the standard deviation of each variable.
- “pareto”: mean-centered and divided by the square root of the standard deviation of each variable.
- “range”: mean-centered and divided by the range of each variable. (default: “none”).

phyloseq-class as inputs

```
data("enterotypes_arumugam")
ps_scale <- scale_variables(
  object = enterotypes_arumugam,
  level = "Phylum",
  method = "range")

head(ps_scale@otu_table@.Data[, 1:5], 3)
```

```
##                AM-AD-1      AM-AD-2  AM-F10-T1  AM-F10-T2  DA-AD-1
## Acidobacteria  0.00000000  0.00000000  0.0000000  0.00000000  0.0000000
## Actinobacteria 0.03379937 -0.05187261  0.0180014  0.882710081 -0.1132282
## Bacteroidetes -0.21758868 -0.21769150 -0.0870175  0.004106405  0.4083952
```

Inputs is from SummarizedExperiment-class.

```
data("Zeybel_2022_protein")

se_impute <- impute_abundance(
  object = Zeybel_2022_protein,
  group = "LiverFatClass",
  ZerosAsNA = TRUE,
  RemoveNA = TRUE,
  cutoff = 20,
  method = "knn")

se_scale <- scale_variables(
  se_impute,
  method = "zscore")

head(assay(se_scale)[, 1:5], 3)

##          P101001    P101003    P101004    P101007    P101009
## IL8    -0.6666419 -0.7058454 -0.5725945 -0.4119527 -0.5532427
## VEGFA   1.3465093  1.5156366  1.3859023  1.2384224  1.4615621
## CD8A    1.4657131  1.2605319  1.1432227  1.1174021  1.0664467
```

3.9 Data Trimming

Trimming samples or features whose prevalence is less than threshold

- “both”, prevalence of features and samples more than cutoff.
- “feature”, prevalence of features more than cutoff.
- “feature_group”, prevalence of features more than cutoff by groups.
- “sample”, prevalence of samples more than cutoff.

phyloseq-class as inputs

```
data("Zeybel_2022_gut")
ps_trim <- trim_prevalence(
  Zeybel_2022_gut,
  group = "LiverFatClass",
  level = "Phylum",
  cutoff = 0.1,
  trim = "feature_group")

ps_trim

## phyloseq-class experiment-level object
## otu_table()   OTU Table:             [ 5 taxa and 42 samples ]
```

```
## sample_data() Sample Data:      [ 42 samples by 46 sample variables ]
## tax_table()   Taxonomy Table:   [ 5 taxa by 3 taxonomic ranks ]
```

Inputs is from SummarizedExperiment-class.

```
data("Zeybel_2022_protein")
se_trim <- trim_prevalence(
  Zeybel_2022_protein,
  cutoff = 0.99,
  trim = "both")
se_trim
```

```
## class: SummarizedExperiment
## dim: 66 54
## metadata(0):
## assays(1): ''
## rownames(66): IL8 VEGFA ... TNFB CSF_1
## rowData names(3): ProteinID LOD prop
## colnames(54): P101001 P101003 ... P101095 P101096
## colData names(47): PatientID Gender ... Right_leg_fat_free_mass Right_leg_total_body_water
```

3.10 Data Filtering

Filtering feature who is low relative abundance or unclassified (Ref: (Thingholm et al., 2019))

- Feature is more than Mean relative abundance across all samples;
- Feature is more than Minimum relative abundance at least one sample.

phyloseq-class as inputs

```
data("Zeybel_2022_gut")
Zeybel_2022_gut_counts <- phyloseq::transform_sample_counts(
  Zeybel_2022_gut, function(x) {round(x * 107)})
```

```
# absolute abundance
ps_filter_absolute <- filter_abundance(
  object = Zeybel_2022_gut_counts,
  level = "Genus",
  cutoff_mean = 100,
  cutoff_one = 1000,
  unclass = FALSE)
```

```
ps_filter_absolute
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:      [ 94 taxa and 42 samples ]
```

```
## sample_data() Sample Data:      [ 42 samples by 46 sample variables ]
## tax_table()   Taxonomy Table:   [ 94 taxa by 7 taxonomic ranks ]
```

```
# relative abundance
ps_filter_relative <- filter_abundance(
  object = Zeybel_2022_gut,
  level = "Genus",
  cutoff_mean = 1e-04,
  cutoff_one = 1e-03,
  unclass = TRUE)

ps_filter_relative
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:        [ 67 taxa and 42 samples ]
## sample_data() Sample Data:      [ 42 samples by 46 sample variables ]
## tax_table()   Taxonomy Table:   [ 67 taxa by 7 taxonomic ranks ]
```

Inputs is from SummarizedExperiment-class.

```
data("Zeybel_2022_protein")
se_filter <- filter_abundance(
  object = Zeybel_2022_protein,
  cutoff_mean = 5,
  cutoff_one = 8)

se_filter
```

```
## class: SummarizedExperiment
## dim: 40 54
## metadata(0):
## assays(1): ''
## rownames(40): 4E_BP1 AXIN1 ... uPA VEGFA
## rowData names(3): ProteinID LOD prop
## colnames(54): P101001 P101003 ... P101095 P101096
## colData names(47): PatientID Gender ... Right_leg_fat_free_mass Right_leg_total_body
```

3.11 Systematic Information

```
devtools::session_info()
```

```
## - Session info -----
## setting  value
## version  R version 4.1.2 (2021-11-01)
## os       macOS Monterey 12.2.1
## system   x86_64, darwin17.0
## ui       RStudio
```

```
## language (EN)
## collate en_US.UTF-8
## ctype en_US.UTF-8
## tz Asia/Shanghai
## date 2023-01-12
## rstudio 2022.07.2+576 Spotted Wakerobin (desktop)
## pandoc 2.19.2 @ /Applications/RStudio.app/Contents/MacOS/quarto/bin/tools/ (via rmarkdown)
##
## - Packages -----
## package * version date (UTC) lib source
## ade4 1.7-18 2021-09-16 [1] CRAN (R 4.1.0)
## ANCOMBC 1.4.0 2021-10-26 [1] Bioconductor
## annotate 1.72.0 2021-10-26 [1] Bioconductor
## AnnotationDbi 1.56.2 2021-11-09 [1] Bioconductor
## ape 5.6-2 2022-03-02 [1] CRAN (R 4.1.2)
## assertthat 0.2.1 2019-03-21 [1] CRAN (R 4.1.0)
## Biobase * 2.54.0 2021-10-26 [1] Bioconductor
## BiocGenerics * 0.40.0 2021-10-26 [1] Bioconductor
## BiocParallel 1.28.3 2021-12-09 [1] Bioconductor
## biomformat 1.22.0 2021-10-26 [1] Bioconductor
## Biostrings 2.62.0 2021-10-26 [1] Bioconductor
## bit 4.0.4 2020-08-04 [1] CRAN (R 4.1.0)
## bit64 4.0.5 2020-08-30 [1] CRAN (R 4.1.0)
## bitops 1.0-7 2021-04-24 [1] CRAN (R 4.1.0)
## blob 1.2.2 2021-07-23 [1] CRAN (R 4.1.0)
## bookdown 0.29 2022-09-12 [1] CRAN (R 4.1.2)
## brio 1.1.3 2021-11-30 [1] CRAN (R 4.1.0)
## bslib 0.4.0 2022-07-16 [1] CRAN (R 4.1.2)
## cachem 1.0.6 2021-08-19 [1] CRAN (R 4.1.0)
## callr 3.7.0 2021-04-20 [1] CRAN (R 4.1.0)
## caTools 1.18.2 2021-03-28 [1] CRAN (R 4.1.0)
## cli 3.4.1 2022-09-23 [1] CRAN (R 4.1.2)
## cluster 2.1.2 2021-04-17 [1] CRAN (R 4.1.2)
## codetools 0.2-18 2020-11-04 [1] CRAN (R 4.1.2)
## colorspace 2.0-3 2022-02-21 [1] CRAN (R 4.1.2)
## crayon 1.5.0 2022-02-14 [1] CRAN (R 4.1.2)
## data.table 1.14.6 2022-11-16 [1] CRAN (R 4.1.2)
## DBI 1.1.2 2021-12-20 [1] CRAN (R 4.1.0)
## DelayedArray 0.20.0 2021-10-26 [1] Bioconductor
## desc 1.4.1 2022-03-06 [1] CRAN (R 4.1.2)
## DESeq2 1.34.0 2021-10-26 [1] Bioconductor
## devtools 2.4.3 2021-11-30 [1] CRAN (R 4.1.0)
## digest 0.6.30 2022-10-18 [1] CRAN (R 4.1.2)
## dplyr * 1.0.10 2022-09-01 [1] CRAN (R 4.1.2)
## ellipsis 0.3.2 2021-04-29 [1] CRAN (R 4.1.0)
## evaluate 0.17 2022-10-07 [1] CRAN (R 4.1.2)
```

##	fansi	1.0.2	2022-01-14	[1]	CRAN (R 4.1.2)
##	fastmap	1.1.0	2021-01-25	[1]	CRAN (R 4.1.0)
##	foreach	1.5.2	2022-02-02	[1]	CRAN (R 4.1.2)
##	fs	1.5.2	2021-12-08	[1]	CRAN (R 4.1.0)
##	genefilter	1.76.0	2021-10-26	[1]	Bioconductor
##	geneplotter	1.72.0	2021-10-26	[1]	Bioconductor
##	generics	0.1.2	2022-01-31	[1]	CRAN (R 4.1.2)
##	GenomeInfoDb	* 1.30.1	2022-01-30	[1]	Bioconductor
##	GenomeInfoDbData	1.2.7	2022-03-09	[1]	Bioconductor
##	GenomicRanges	* 1.46.1	2021-11-18	[1]	Bioconductor
##	ggplot2	3.4.0	2022-11-04	[1]	CRAN (R 4.1.2)
##	glmnet	4.1-3	2021-11-02	[1]	CRAN (R 4.1.0)
##	glue	1.6.2	2022-02-24	[1]	CRAN (R 4.1.2)
##	gplots	3.1.1	2020-11-28	[1]	CRAN (R 4.1.0)
##	gtable	0.3.0	2019-03-25	[1]	CRAN (R 4.1.0)
##	gtools	3.9.2	2021-06-06	[1]	CRAN (R 4.1.0)
##	highr	0.9	2021-04-16	[1]	CRAN (R 4.1.0)
##	htmltools	0.5.3	2022-07-18	[1]	CRAN (R 4.1.2)
##	httr	1.4.2	2020-07-20	[1]	CRAN (R 4.1.0)
##	igraph	1.2.11	2022-01-04	[1]	CRAN (R 4.1.2)
##	impute	1.68.0	2021-10-26	[1]	Bioconductor
##	IRanges	* 2.28.0	2021-10-26	[1]	Bioconductor
##	iterators	1.0.14	2022-02-05	[1]	CRAN (R 4.1.2)
##	jquerylite	0.1.4	2021-04-26	[1]	CRAN (R 4.1.0)
##	jsonlite	1.8.3	2022-10-21	[1]	CRAN (R 4.1.2)
##	KEGGREST	1.34.0	2021-10-26	[1]	Bioconductor
##	KernSmooth	2.23-20	2021-05-03	[1]	CRAN (R 4.1.2)
##	knitr	1.40	2022-08-24	[1]	CRAN (R 4.1.2)
##	lattice	0.20-45	2021-09-22	[1]	CRAN (R 4.1.2)
##	lifecycle	1.0.3	2022-10-07	[1]	CRAN (R 4.1.2)
##	limma	3.50.1	2022-02-17	[1]	Bioconductor
##	locfit	1.5-9.5	2022-03-03	[1]	CRAN (R 4.1.2)
##	magrittr	2.0.3	2022-03-30	[1]	CRAN (R 4.1.2)
##	MASS	7.3-55	2022-01-13	[1]	CRAN (R 4.1.2)
##	Matrix	1.4-0	2021-12-08	[1]	CRAN (R 4.1.0)
##	MatrixGenerics	* 1.6.0	2021-10-26	[1]	Bioconductor
##	matrixStats	* 0.61.0	2021-09-17	[1]	CRAN (R 4.1.0)
##	memoise	2.0.1	2021-11-26	[1]	CRAN (R 4.1.0)
##	metagenomeSeq	1.36.0	2021-10-26	[1]	Bioconductor
##	mgcv	1.8-39	2022-02-24	[1]	CRAN (R 4.1.2)
##	microbiome	1.16.0	2021-10-26	[1]	Bioconductor
##	MicrobiomeAnalysis	* 1.0.3	2023-01-12	[1]	Bioconductor
##	multtest	2.50.0	2021-10-26	[1]	Bioconductor
##	munsell	0.5.0	2018-06-12	[1]	CRAN (R 4.1.0)
##	nlme	3.1-155	2022-01-13	[1]	CRAN (R 4.1.2)
##	nloptr	2.0.0	2022-01-26	[1]	CRAN (R 4.1.2)

##	permute	0.9-7	2022-01-27	[1]	CRAN (R 4.1.2)
##	phyloseq	* 1.38.0	2021-10-26	[1]	Bioconductor
##	pillar	1.7.0	2022-02-01	[1]	CRAN (R 4.1.2)
##	pkgbuild	1.3.1	2021-12-20	[1]	CRAN (R 4.1.0)
##	pkgconfig	2.0.3	2019-09-22	[1]	CRAN (R 4.1.0)
##	pkgload	1.2.4	2021-11-30	[1]	CRAN (R 4.1.0)
##	plyr	1.8.6	2020-03-03	[1]	CRAN (R 4.1.0)
##	png	0.1-7	2013-12-03	[1]	CRAN (R 4.1.0)
##	prettyunits	1.1.1	2020-01-24	[1]	CRAN (R 4.1.0)
##	processx	3.5.2	2021-04-30	[1]	CRAN (R 4.1.0)
##	ps	1.6.0	2021-02-28	[1]	CRAN (R 4.1.0)
##	purrr	0.3.4	2020-04-17	[1]	CRAN (R 4.1.0)
##	R6	2.5.1	2021-08-19	[1]	CRAN (R 4.1.0)
##	rbibutils	2.2.7	2021-12-07	[1]	CRAN (R 4.1.0)
##	RColorBrewer	1.1-2	2014-12-07	[1]	CRAN (R 4.1.0)
##	Rcpp	1.0.9	2022-07-08	[1]	CRAN (R 4.1.2)
##	RCurl	1.98-1.6	2022-02-08	[1]	CRAN (R 4.1.2)
##	Rdpack	2.2	2022-03-19	[1]	CRAN (R 4.1.2)
##	remotes	2.4.2	2021-11-30	[1]	CRAN (R 4.1.0)
##	reshape2	1.4.4	2020-04-09	[1]	CRAN (R 4.1.0)
##	rhdf5	2.38.1	2022-03-10	[1]	Bioconductor
##	rhdf5filters	1.6.0	2021-10-26	[1]	Bioconductor
##	Rhdf5lib	1.16.0	2021-10-26	[1]	Bioconductor
##	rlang	1.0.6	2022-09-24	[1]	CRAN (R 4.1.2)
##	rmarkdown	2.17	2022-10-07	[1]	CRAN (R 4.1.2)
##	rprojroot	2.0.2	2020-11-15	[1]	CRAN (R 4.1.0)
##	RSQLite	2.2.10	2022-02-17	[1]	CRAN (R 4.1.2)
##	rstudioapi	0.13	2020-11-12	[1]	CRAN (R 4.1.0)
##	Rtsne	0.15	2018-11-10	[1]	CRAN (R 4.1.0)
##	S4Vectors	* 0.32.3	2021-11-21	[1]	Bioconductor
##	sass	0.4.2	2022-07-16	[1]	CRAN (R 4.1.2)
##	scales	1.2.1	2022-08-20	[1]	CRAN (R 4.1.2)
##	sessioninfo	1.2.2	2021-12-06	[1]	CRAN (R 4.1.0)
##	shape	1.4.6	2021-05-19	[1]	CRAN (R 4.1.0)
##	stringi	1.7.8	2022-07-11	[1]	CRAN (R 4.1.2)
##	stringr	1.4.1	2022-08-20	[1]	CRAN (R 4.1.2)
##	SummarizedExperiment	* 1.24.0	2021-10-26	[1]	Bioconductor
##	survival	3.4-0	2022-08-09	[1]	CRAN (R 4.1.2)
##	testthat	3.1.2	2022-01-20	[1]	CRAN (R 4.1.2)
##	tibble	* 3.1.8	2022-07-22	[1]	CRAN (R 4.1.2)
##	tidyr	1.2.0	2022-02-01	[1]	CRAN (R 4.1.2)
##	tidyselect	1.1.2	2022-02-21	[1]	CRAN (R 4.1.2)
##	tinytex	0.42	2022-09-27	[1]	CRAN (R 4.1.2)
##	usethis	2.1.5	2021-12-09	[1]	CRAN (R 4.1.0)
##	utf8	1.2.2	2021-07-24	[1]	CRAN (R 4.1.0)
##	vctrs	0.5.1	2022-11-16	[1]	CRAN (R 4.1.2)

```
##      vegan                2.5-7      2020-11-28 [1] CRAN (R 4.1.0)
##      withr                2.5.0      2022-03-03 [1] CRAN (R 4.1.2)
##      Wrench               1.12.0     2021-10-26 [1] Bioconductor
##      xfun                 0.34       2022-10-18 [1] CRAN (R 4.1.2)
##      XML                  3.99-0.9   2022-02-24 [1] CRAN (R 4.1.2)
##      xtable               1.8-4      2019-04-21 [1] CRAN (R 4.1.0)
##      XVector              0.34.0     2021-10-26 [1] Bioconductor
##      yaml                 2.3.6      2022-10-18 [1] CRAN (R 4.1.2)
##      zlibbioc             1.40.0     2021-10-26 [1] Bioconductor
##
##      [1] /Library/Frameworks/R.framework/Versions/4.1/Resources/library
##
## -----
```


Bibliography

- Anderson, M. J. (2014). Permutational multivariate analysis of variance (permanova). *Wiley statsref: statistics reference online*, pages 1–15.
- Fernandes, A. D., Reid, J. N., Macklaim, J. M., McMurrough, T. A., Edgell, D. R., and Gloor, G. B. (2014). Unifying the analysis of high-throughput sequencing datasets: characterizing rna-seq, 16s rrna gene sequencing and selective growth experiments by compositional data analysis. *Microbiome*, 2(1):1–13.
- McMurdie, P. J. and Holmes, S. (2013). phyloseq: an r package for reproducible interactive analysis and graphics of microbiome census data. *PloS one*, 8(4):e61217.
- R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Thingholm, L. B., Rühlemann, M. C., Koch, M., Fuqua, B., Laucke, G., Boehm, R., Bang, C., Franzosa, E. A., Hübenthal, M., Rahnavard, A., et al. (2019). Obese individuals with and without type 2 diabetes show different gut microbial functional capacity and composition. *Cell host & microbe*, 26(2):252–264.