# Package ‘RADanalysis’

April 28, 2025

**Type** Package

**Title** Normalization and Analysis of Rank Abundance Distributions

**Version** 1.0.1

**Date** 2025-04-26

**Description** Implementation of the MaxRank normalization method, which enables standardization of Rank Abundance Distributions (RADs) to a specified number of ranks. Rank abundance distributions are widely used in biology and ecology to describe species abundances, and are mathematically equivalent to complementary cumulative distribution functions (CCDFs) used in physics, linguistics, sociology, and other fields. The method is described in Saeedghalati et al. (2017) <doi:10.1371/journal.pcbi.1005362>.

**License** GPL-3

**LazyData** TRUE

**RoxygenNote** 7.3.2

**Encoding** UTF-8

**Depends** R (>= 2.10)

**Imports** sfsmisc, scales, stats, graphics

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Mohmmadkarim Saeedghalati [aut, cre],
Farnoush Farahpour [aut],
Daniel Hoffmann [aut]

**Maintainer** Mohmmadkarim Saeedghalati <arsham@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-04-28 18:40:12 UTC

# Contents

## `gut_nrads` *Result of normalization of gut_otu_table with* `max_rank = 400` *and* `average_over = 2000`.

### Description

Normalized rads created from `gut_otu_table` with `max_rank = 400` and `average_over = 2000`. `gut_otu_table` has 18 gut samples from 3 individuals, prior, under and after using Ciprofloxacin (Cp) antibiotic. Each row is the abundance of gut microbiome for one sample. The row names are as follows:

### Usage

```
data(gut_nrads)
```

### Format

Contain the result of `RADnormalization_matrix(input = gut_otu_table,max_rank = 400,average_over = 2000)`

### Details

- A1, B1 and C1 are samples from individuals A, B and C 60 days prior using Cp.
- A2a is the sample from individual A 6 days prior using Cp.
- A2b is the sample from individual A 2 days prior using Cp.
- A2c, B2 and C2 are samples from individuals A, B and C one day prior using Cp.
- A3a is the sample from individual A 3 days after the first day of Cp administration.
- A3b, B3 and C3 are the samples from individual A, B and C 5 days after the first day of Cp administration.
- A4, B4 and C4 are the samples from individual A, B and C 33 days after the first day of Cp administration.

- A5, B5 and C5 are the samples from individual A, B and C 180 days after the first day of Cp administration.

points 1 and 2 are considered pre-Cp, points 3 are considered under-Cp and point 4 and 5 are considered post-Cp.

order of rows are similar to gut_otu_table

## gut_otu_table *Abundance table of gut microbiome.*

### Description

Dethlefsen et al. 2008 (http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.0060280) have treated healthy individuals with the antibiotic Ciprofloxacin and monitored the states of the gut microbiome before the treatment, during the treatment, and some time after the treatment.

### Usage

```
data(gut_otu_table)
```

### Format

A matrix with 18 rows and 5670 columns.

### Details

Abundance table of 18 gut samples from 3 individuals, prior, under and after using Ciprofloxacin (Cp) antibiotic. Each row is the abundance of gut microbiome for one sample. The row names are as follows:

- A1, B1 and C1 are samples from individuals A, B and C 60 days prior using Cp.
- A2a is the sample from individual A 6 days prior using Cp.
- A2b is the sample from individual A 2 days prior using Cp.
- A2c, B2 and C2 are samples from individuals A, B and C one day prior using Cp.
- A3a is the sample from individual A 3 days after the first day of Cp administration.
- A3b, B3 and C3 are the samples from individual A, B and C 5 days after the first day of Cp administration.
- A4, B4 and C4 are the samples from individual A, B and C 33 days after the first day of Cp administration.
- A5, B5 and C5 are the samples from individual A, B and C 180 days after the first day of Cp administration.

points 1 and 2 are considered pre-Cp, points 3 are considered under-Cp and point 4 and 5 are considered post-Cp.

### Source

Dethlefsen, Les, et al. "The pervasive effects of an antibiotic on the human gut microbiota, as revealed by deep 16S rRNA sequencing." PLoS biol 6.11 (2008): e280.

RADanalysis *RADanalysis: A package for normalization of abundance tables to desired number of ranks using MaxRank Normalization method.*

### Description

RADanalysis package has tools for normalizing rank abundance distributions (RAD) to a desired number of ranks using MaxRank Normalization method. RADs are commonly used in biology/ecology and mathematically equivalent to complementary cumulative distributions (CCDFs) which are used in physics, linguistics and sociology and more generally in data science.

### Rank Abundance Distributions (RAD)

Rank Abundance Distributions (RADs) are a way to capture the distribution of biological species in communities, where we use the term "species" for all types of distinct biological entities, e.g. microbial species in a microbiome, viral strains in a quasi-species, the diverse variants B cells in a person, etc. A RAD can be thought of as a plot with the two axes rank (x-axis) and abundance (y-axis). For the most abundant species we draw a point at the (x,y) coordinates (1,a1) , with a1 the abundance of this most abundant species. For the second most abundant species we draw a point at (2,a2).

### MaxRank Normalization

MaxRank normalization is the method to normalize RADs. MaxRank normalization maps all rank abundance vectors to the same rank range from 1 to a common maximum rank R. First we chose the maximum rank or "MaxRank" or "R". Second generated for each sample s a pool of N_s of all individuals in s. From this pool we drew individuals at random with uniform probability and without replacement as long as the number of sampled ranks of the original RAD did not exceed R. In this way we generated a new, reduced abundance vector of R ranks, with a reduced number of individuals. Division of these reduced abundances by sum of reduced abundances transforms the reduced abundance vector to a probability distribution for the R ranks with rank probabilities summing up to 1. If R < total number of ranks in the original sample , the random drawing of individuals from the pool in general introduces a sampling error in the abundances. To control this error, one should repeat the procedure several times (typically 10-100 times) and averaged over all sampled abundance distributions.

### Source

Saeedghalati et al. 2016 "Quantitative comparison of abundance structures of genetic communities", submitted

## RADnormalization *Normalizes an abundance vector to the desired number of ranks.*

### Description

Normalizes an abundance vector to the desired number of ranks.

### Usage

```
RADnormalization(
  input,
  max_rank,
  average_over = 1,
  min_rank = 1,
  labels = FALSE,
  count_data = TRUE,
  method = "upperlimit"
)
```

### Arguments

`input` A vector which contains the abundance values (an abundance vector).

`max_rank` The desired rank to which this method normalizes the input.

`average_over` Number of times, a normalized RAD is created and averaged to produce the result.

`min_rank` The minimum rank to which this method normalizes the input.

`labels` A logical. If `TRUE` the label of each rank (ids in the input vector) will be returned.

`count_data` A logical. `TRUE` means that the input vector contains counts (integer values) otherwise contains the relative abundances. In the current version only counts are accepted.

`method` Sets the stop criterion for normalization. This should be one of "lowerlimit", "middle" or "upperlimit". Method affects the final result. lowerlimit: Samples from species pool one by one, until reaches max_rank. middle: Samples from species pool with random size until the sampled vector has desired ranks (max_rank). upperlimit: Removes from species pool one by one, until reaches max_rank.

### Value

A list of following items:

$norm_rad: Normalized RAD sum up to 1. If `labels` = `TRUE`, it would also contain the labels.

$norm_rad_count: A matrix of `average_over` rows and `max_rank` columns. Each row contains one normalized RAD. These normalized RADs are averaged and sum up to 1 in order to make norm_rad

$norm_rad_mean_sd: Standard deviation of the mean for all the ranks in `norm_rad`. This vector is created using the values in `norm_rad_count`

$inputs: A list which contains inputs used for creating normalized RADs.

### See Also

RADnormalization_matrix for normalize an entire otutable, representative_point for study the representative of groups of samples in a multi-dimensional scaling plot, representative_RAD for study the representative of group of norm rads.

### Examples

```
data("gut_otu_table")
rads <- gut_otu_table
original_rad <- sort(rads[1,],decreasing = TRUE)
#removing zeros
original_rad <- original_rad[original_rad > 0]
plot(original_rad,ylim = c(1,max(original_rad)),log = "xy", xlab = "Rank",ylab = "Abundance",
     main = "RAD of first sample",pch = 19,type = "b",cex = 0.5)
print(paste("number of ranks present in the original rad is:",length(original_rad)))
norm_rad <- RADnormalization(input = rads[1,],max_rank = 500,average_over = 50)
points(x = norm_rad$norm_rad * sum(norm_rad$norm_rad_count[1,]) ,pch = 19,cex = 1, type = "l",
       col = "blue",lwd = 4)
points(x = norm_rad$norm_rad_count[1,],pch = 19,cex = 1, type = "l",col = "red",lwd = 3)
points(x = norm_rad$norm_rad_count[2,],pch = 19,cex = 1, type = "l",col = "green",lwd = 3)
legend("bottomleft",legend = c("Original RAD","possible norm rad","possible norm rad",
                               paste("nrad averaged over 50 realizations, times",
                               sum(norm_rad$norm_rad_count[1,]))),
       col = c("black","red","green","blue"),lwd = 2,bty = "n")
```

## RADnormalization_matrix

*Normalizes an abundance table to the desired number of ranks*

### Description

Normalizes an abundance table to the desired number of ranks

### Usage

```
RADnormalization_matrix(
  input,
  max_rank,
  average_over = 1,
  min_rank = 1,
  labels = FALSE,
  count_data = TRUE,
  sample_in_row = TRUE,
  method = "upperlimit",
  verbose = TRUE
)
```

### Arguments

`input` A vector or matrix which contains the abundance values (an abundance table).

`max_rank` The desired rank to which this method normalizes the input.

`average_over` Number of times, a normalized RAD is created and averaged to produce the result.

`min_rank` The minimum rank to which this method normalizes the input.

`labels` A logical. If `TRUE` the label of each rank (ids in the input vector) will be returned.

`count_data` A logical. `TRUE` means that the input vector contains counts (integer values) otherwise contains the relative abundances. In the current version only counts are accepted.

`sample_in_row` A logical. `TRUE` means that the abundance vector of samples are represented in rows otherwise in columns.

`method` Sets the stop criterion for normalization. This should be one of "lowerlimit", "middle" or "upperlimit". Method affects the final result. lowerlimit: Samples from species pool one by one, until reaches max_rank. middle: Samples from species pool with random size until the sampled vector has desired ranks (max_rank). upperlimit: Removes from species pool one by one, until reaches max_rank.

`verbose` A logical. If `TRUE`, prints the progress in percent in console.

### Value

A list of following items:

$norm_matrix A matrix which contains normalized RADs sum up to 1. If `labels = TRUE`, it would also contain the labels.

$inputs A list which contains inputs used for creating normalized RADs. It does not contain `input` because it could be very big.

### See Also

`RADnormalization` for normalize an abundance vector. This function return more details compared to `RADnormalization_matrix`, `representative_point` for study the representative of groups of samples in a multi-dimensional scaling plot, `representative_RAD` for study the representative of group of norm rads.

### Examples

```
data("gut_otu_table")
rads <- gut_otu_table
#plot original rads
line_cols <- c("green","red","blue")
sample_classes <- c(1,1,1,1,2,2,3,3,1,1,2,3,3,1,1,2,3,3)
plot(1,xlim = c(1,2000),ylim = c(1,20000),col = "white",log  = "xy",
     xlab = "Rank",ylab = "Abundance",main = "Original RADs from antibiotic data set")
for(i in 1:nrow(rads)){
    temp <- sort(rads[i,],decreasing = TRUE)
    temp <- temp[temp>0]
    lines(x = temp,lwd = 2,col = line_cols[sample_classes[i]])
}
legend("bottomleft",bty = "n",legend = c("pre Cp","under Cp","post Cp"),col = line_cols,lwd = 3)


nrads <- RADnormalization_matrix(input = rads,max_rank = 400,average_over = 20,sample_in_row = TRUE)
nrads <- nrads$norm_matrix

plot(1,xlim = c(1,400),ylim = c(4e-5,1),col = "white",log  = "xy",
     xlab = "Rank",ylab = "Abundance",
     main = "NRADs from antibiotic data set with R = 400 \n with average_over = 20")
for(i in 1:nrow(nrads)){
    lines(x = nrads[i,],lwd = 2,col = line_cols[sample_classes[i]])
}
legend("bottomleft",bty = "n",legend = c("pre Cp","under Cp","post Cp"),col = line_cols,lwd = 3)
```

## representative_point *Computes representative point based on the coordinates of points which are in the same group.*

### Description

Computes representative point based on the coordinates of points which are in the same group.

### Usage

```
representative_point(
  input,
  ids = NULL,
  coord_names = c(1, 2),
  standard_error_mean = TRUE,
  plot = FALSE,
  ...
)
```

### Arguments

`input` A matrix which contains the coordinates of samples. Usually this is the result of ordination of normalized RADs using multi-dimensional scaling (`cmdscale`). In the input matrix each row contains vector of coordinates of one sample.

`ids` Vector of row numbers of the desired group, from which a representative point is going to be represented

`coord_names` A vector which contains the coordintes number that should be used to create representative point. Default is `c(1,2)`.

`standard_error_mean` A logical. If `TRUE`, uses the standard error of the mean and plot it with representative points. It works only if `plot = TRUE`.

| | |
|---|---|
| plot | A logical. If TRUE, shows the representative points on the previous plot. |
| ... | other graphical parameters to use for plotting. This function uses internally the functions points and arrows to plot. |

## Value

A list of following parameters:

$mean: Contains the average of points. A vector with the length of coordinates used for computing the average. These coordinates are preset in coord_names.

$sd: A vector with a length similar to mean which contains the standard deviation for each coordinate.

$mean_standard_error: A vector with a length similar to mean which contain the standard deviation of the mean for each coordinate. This vector is the result of sd / sqrt(n), when n is the number of members of the group (length of sample_ids).

If plot = TRUE, representative points would be added to the previous plot.

If standard_error_mean = TRUE, the standard error of the mean would be added to the representative points.

## See Also

RADnormalization for normalize an abundance vector. This function return more details compared to RADnormalization_matrix, RADnormalization_matrix for normalize an entire otutable, representative_RAD for study the representative of group of norm rads.

## Examples

```
line_cols <- c("green","red","blue")
sample_classes <- c(1,1,1,1,2,2,3,3,1,1,2,3,3,1,1,2,3,3)
maxrank <- 400
data("gut_nrads")
nrads <- gut_nrads
nrads <- nrads$norm_matrix

#distance matrix using manhattan distance
d <- dist(x = nrads,method = "manhattan")
#ordination using classical multi-dimensional scaling
mds <- cmdscale(d = d,k = 5,eig = TRUE)

#plot the points
plot(mds$points,xlab = "First coordinate",ylab = "Second coordinate",pch = 19,cex =1,
    col = line_cols[sample_classes],
    main = "MDS plot with representative points \n of each group and error bars")

#add the representative points wit erorr bar to the previous plot
a <- representative_point(input = mds$points,ids = which(sample_classes == 1),
    col = scales::alpha(line_cols[1],0.5),
    plot = TRUE,standard_error_mean = TRUE,pch = 19, cex = 4)
a <- representative_point(input = mds$points,ids = which(sample_classes == 2),
    col = scales::alpha(line_cols[2],0.5),
    plot = TRUE,standard_error_mean = TRUE,pch = 19, cex = 4)
a <- representative_point(input = mds$points,ids = which(sample_classes == 3),
    col = scales::alpha(line_cols[3],0.5),
    plot = TRUE,standard_error_mean = TRUE,pch = 19, cex = 4)

legend("bottomleft",bty = "n",legend = c("pre Cp","under Cp","post Cp"),
    col = line_cols,pch = 19)
```

## representative_RAD *Computes representative normalized RAD of a group of normalized RADs.*

### Description

Computes representative normalized RAD of a group of normalized RADs.

### Usage

```
representative_RAD(
  norm_rad,
  sample_ids = NULL,
  plot = FALSE,
  min_rank = 1,
  confidence = 0.95,
  with_conf = TRUE,
  ...
)
```

### Arguments

| | |
|---|---|
| `norm_rad` | A matrix which contains the normalized RADs (samples in rows). |
| `sample_ids` | Vector of row numbers of the desired group, from which a representative RAD is going to be produced. |
| `plot` | A logical. If `TRUE`, plots the repRAD. The plot would be added to the previous plot. |
| `min_rank` | The minimum rank to be considered for making repRADs. |
| `confidence` | Confidence interval of plotted repRAD. Default is 0.9. |
| `with_conf` | A logical. If `TRUE`, plots the confidence interval in addition to repRAD. Only works when `plot` is `TRUE`. |
| `...` | Other graphical parameters to use for plotting. This function uses internally the functions `lines` and `polygon` to plot. |

## Value

A list of following parameters:

$average: Contains a vector of length equal to the columns of norm_rad. This in the representative normalized RAD which is the average of normalized RAD of the group.

$population_sd: A vector of length equal to the columns of norm_rad which contains the standard deviation for each rank.

$standard_error: A vector of length equal to the columns of norm_rad which contains the standard deviation of the mean for each rank. This vector is the result of population_sd / sqrt(n), when n is the number of members of the group (length of sample_ids).

If plot = TRUE, plot of the repRAD is produced and would be added to the previous plot.

If with_conf = TRUE, confidence interval would be added to the repRAD plot.

## See Also

RADnormalization for normalize an abundance vector. This function return more details compared to RADnormalization_matrix, RADnormalization_matrix for normalize an entire otutable, representative_point for study the representative of groups of samples in a multi-dimensional scaling plot,

## Examples

```
line_cols <- c("green","red","blue")
sample_classes <- c(1,1,1,1,2,2,3,3,1,1,2,3,3,1,1,2,3,3)
maxrank <- 400
data("gut_nrads")
nrads <- gut_nrads
nrads <- nrads$norm_matrix

#plot nrads
plot(1e10,xlim = c(1,maxrank),ylim = c(2e-5,1),log="xy",
     xlab = "rank",ylab = "abundance",cex.lab = 1.5,axes = FALSE)
sfsmisc::eaxis(side = 1,at = c(1,10,100,1000,10000))
sfsmisc::eaxis(side = 2,at = c(1e-4,1e-3,1e-2,1e-1,1),las = 0)
for(i in 1:nrow(nrads)){
    points(nrads[i,],type = 'l',col = line_cols[sample_classes[i]],lwd = 0.8)
}
#plot confidence intervals of representative nrads
a <- representative_RAD(norm_rad = nrads,sample_ids = which(sample_classes == 1),
                        plot = TRUE,confidence = 0.9,with_conf = TRUE,
                        col = scales::alpha(line_cols[1],0.5),border = NA)
a <- representative_RAD(norm_rad = nrads,sample_ids = which(sample_classes == 2),
                        plot = TRUE,confidence = 0.9,with_conf = TRUE,
                        col = scales::alpha(line_cols[2],0.5),border = NA)
a <- representative_RAD(norm_rad = nrads,sample_ids = which(sample_classes == 3),
                        plot = TRUE,confidence = 0.9,with_conf = TRUE,
                        col = scales::alpha(line_cols[3],0.5),border = NA)
#plot representative nrads
a <- representative_RAD(norm_rad = nrads,sample_ids = which(sample_classes == 1),
                        plot = TRUE,with_conf = FALSE,
                        col = scales::alpha(line_cols[1],0.8),lwd = 4)
```

```
a <- representative_RAD(norm_rad = nrads,sample_ids = which(sample_classes == 2),
                      plot = TRUE,with_conf = FALSE,
                      col = scales::alpha(line_cols[2],0.8),lwd = 4)
a <- representative_RAD(norm_rad = nrads,sample_ids = which(sample_classes == 3),
                      plot = TRUE,with_conf = FALSE,
                      col = scales::alpha(line_cols[3],0.8),lwd = 4)
legend("bottomleft",bty = "n",legend = c("pre Cp","under Cp","post Cp"),
col = line_cols,lwd = 3)
```

# Index