

Package ‘ecan’

July 7, 2023

Title Ecological Analysis and Visualization

Version 0.2.1

Description Support ecological analyses such as ordination and clustering.
Contains consistent and easy wrapper functions of 'stat', 'vegan', and
'labdsv' packages, and visualisation functions of ordination and clustering.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 3.5.0)

URL <https://github.com/matutosi/ecan>
<https://github.com/matutosi/ecan/tree/develop> (devel)

Imports MASS, cluster, dendextend, dplyr, ggplot2, jsonlite, labdsv,
magrittr, purrr, rlang, stringr, tibble, tidyr, vegan

Suggests ggdendro, knitr, rmarkdown, testthat (>= 3.0.0),

Config/testthat/edition 3

NeedsCompilation no

Author Toshikazu Matsumura [aut, cre]

Maintainer Toshikazu Matsumura <matutosi@gmail.com>

Repository CRAN

Date/Publication 2023-07-07 12:20:06 UTC

R topics documented:

cluster	2
d	3
df2table	4
draw_layer_construction	5
gen_example	6
ind_val	7
is_one2multi	8

ordination	9
pad2longest	11
read_biss	11
shdi	12

Index	13
--------------	-----------

cluster	<i>Helper function for clustering methods</i>
---------	-----------------------------------------------

Description

Helper function for clustering methods

Helper function for calculate distance

Add group names to hclust labels.

Add colors to dendrogram

Usage

```
cluster(x, c_method, d_method)
```

```
distance(x, d_method)
```

```
cls_add_group(cls, df, indiv, group, pad = TRUE)
```

```
cls_color(cls, df, indiv, group)
```

Arguments

x	A community data matrix. rownames: stands. colnames: species.
c_method	A string of clustering method. "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC), "centroid" (= UPGMC), or "diana".
d_method	A string of distance method. "correlation", "manhattan", "euclidean", "canberra", "clark", "bray", "kulczynski", "jaccard", "gower", "altGower", "morisita", "horn", "mountford", "raup", "binomial", "chao", "cao", "mahalanobis", "chisq", "chord", "aitchison", or "robust.aitchison".
cls	A result of cluster or dendrogram.
df	A data.frame to be added into ord scores
indiv, group	A string to specify individual and group name of column in df.
pad	A logical to specify padding strings.

Value

cluster() returns result of clustering. \$clustering_method: c_method \$distance_method: d_method
 distance() returns distance matrix.

Inputing cls return a color vector,
 inputing dend return a dend with color.

Examples

```
library(dplyr)
df <-
  tibble::tibble(
    stand = paste0("ST_", c("A", "A", "A", "B", "B", "C", "C", "C", "C")),
    species = paste0("sp_", c("a", "e", "d", "e", "b", "e", "d", "b", "a")),
    abundance = c(3, 3, 1, 9, 5, 4, 3, 3, 1))
cls <-
  df2table(df) %>%
  cluster(c_method = "average", d_method = "bray")
library(ggdendro)
# show standard cluster
ggdendro::ggdendrogram(cls)

# show cluster with group
data(dune, package = "vegan")
data(dune.env, package = "vegan")
cls <-
  cluster(dune, c_method = "average", d_method = "bray")
df <- tibble::rownames_to_column(dune.env, "stand")
cls <- cls_add_group(cls, df, indiv = "stand", group = "Use")
ggdendro::ggdendrogram(cls)
```

Description

Calculating diversity

Usage

d(x)

h(x, base = exp(1))

Arguments

x, base A numeric vector.

Value

A numeric vector.

df2table	<i>Convert data.frame and table to each other.</i>
----------	----------------------------------------------------

Description

Convert data.frame and table to each other.

Usage

```
df2table(df, st = "stand", sp = "species", ab = "abundance")
```

```
table2df(tbl, st = "stand", sp = "species", ab = "abundance")
```

```
dist2df(dist)
```

Arguments

df A data.frame.

st, sp, ab A string.

tbl A table. community matrix. rownames: stands. colnames: species.

dist A distance table.

Value

df2table() return table,
table2df() return data.frame,
dist2df() return data.frame.

Examples

```
tibble::tibble(
  st = paste0("st_", rep(1:2, times = 2)),
  sp = paste0("sp_", rep(1:2, each = 2)),
  ab = runif(4)) %>%
dplyr::bind_rows(., .) %>%
print() %>%
df2table("st", "sp", "ab")
```

`draw_layer_construction`*Draw layer construction plot*

Description

Draw layer construction plot

Add mid point and bin width of layer heights.

Compute mid point of layer heights.

Compute bin width of layer heights.

Usage

```
draw_layer_construction(  
  df,  
  stand = "stand",  
  height = "height",  
  cover = "cover",  
  group = "",  
  ...  
)  
  
add_mid_p_bin_w(df, height = "height")  
  
mid_point(x)  
  
bin_width(x)
```

Arguments

<code>df</code>	A dataframe including columns: stand, layer height and cover. Optional column: stand group.
<code>stand, height, cover, group</code>	A string to specify stand, height, cover, group column.
<code>...</code>	Extra arguments for <code>geom_bar()</code> .
<code>x</code>	A numeric vector.

Value

`draw_layer_construction()` returns gg object, `add_mid_p_bin_w()` returns dataframe including `mid_point` and `bin_width` columns. `mid_point()` and `bin_width()` return a numeric vector.

Examples

```

library(dplyr)
n <- 10
height_max <- 20
ly_list <- c("B", "S", "K")
st_list <- LETTERS[1]
sp_list <- letters[1:9]
st_group <- NULL
sp_group <- rep(letters[24:26], 3)
cover_list <- 2^(0:4)
df <- gen_example(n = n, use_layer = TRUE,
                 height_max = height_max, ly_list = ly_list,
                 st_list = st_list, sp_list = sp_list,
                 st_group = st_group, sp_group = sp_group,
                 cover_list = cover_list)

# select stand and summarise by sp_group
df %>%
  dplyr::group_by(height, sp_group) %>%
  dplyr::summarise(cover = sum(cover), .groups = "drop") %>%
  draw_layer_construction(group = "sp_group", colour = "white")

```

gen_example

Generate vegetation example

Description

Stand, species, and cover are basic. Layer, height, st_group, are sp_group optional.

Usage

```

gen_example(
  n = 300,
  use_layer = TRUE,
  height_max = 20,
  ly_list = "",
  st_list = LETTERS[1:9],
  sp_list = letters[1:9],
  st_group = NULL,
  sp_group = NULL,
  cover_list = 2^(0:6)
)

```

Arguments

n A numeric to generate no of occurrences.
use_layer A logical. If FALSE, height_max and ly_list will be omitted.

height_max A numeric. The highest layer of samples.
 ly_list, st_list, sp_list, st_group, sp_group
 A string vector. st_group and sp_group are optional (default is NULL). Length
 of st_list and sp_list should be the same as st_group and sp_group, respectively.
 cover_list A numeric vector.

Value

A dataframe with columns: stand, layer, species, cover, st_group and sp_group.

Examples

```
n <- 300
height_max <- 20
ly_list <- c("B1", "B2", "S1", "S2", "K")
st_list <- LETTERS[1:9]
sp_list <- letters[1:9]
st_group <- rep(LETTERS[24:26], 3)
sp_group <- rep(letters[24:26], 3)
cover_list <- 2^(0:6)
gen_example(n = n, use_layer = TRUE,
            height_max = height_max, ly_list = ly_list,
            st_list = st_list, sp_list = sp_list,
            st_group = st_group, sp_group = sp_group,
            cover_list = cover_list)
```

ind_val

Helper function for Indicator Species Analysis

Description

Calculating diversity indices such as species richness (s), Shannon's H' (h), Simpson' D (d), Simpson's inverse D (i).

Usage

```
ind_val(
  df,
  stand = NULL,
  species = NULL,
  abundance = NULL,
  group = NULL,
  row_data = FALSE
)
```

Arguments

df	A data.frame, which has three cols: stand, species, abundance. Community matrix should be converted using table2df().
stand, species, abundance	A text to specify each column. If NULL, 1st, 2nd, 3rd column will be used.
group	A text to specify group column.
row_data	A logical. TRUE: return row result data of labdsv::indval().

Value

A data.frame.

Examples

```
library(dplyr)
library(tibble)
data(dune, package = "vegan")
data(dune.env, package = "vegan")
df <-
  dune %>%
  table2df(st = "stand", sp = "species", ab = "cover") %>%
  dplyr::left_join(tibble::rownames_to_column(dune.env, "stand"))
ind_val(df, abundance = "cover", group = "Moisture")
```

is_one2multi

Check cols one-to-one, or one-to-multi in data.frame

Description

Check cols one-to-one, or one-to-multi in data.frame

Usage

```
is_one2multi(df, col_1, col_2)

is_one2one(df, col_1, col_2)

is_multi2multi(df, col_1, col_2)

cols_one2multi(df, col, include_self = TRUE)

select_one2multi(df, col, include_self = TRUE)

unique_length(df, col_1, col_2)
```


Arguments

df A data.frame
 col, col_1, col_2 A string to specify a colname.
 include_self A logical. If TRUE, return value including input col.

Value

is_one2multi(), is_one2one(), is_multi2multi() return a logical. cols_one2multi() returns strings of colnames that has one2multi relation to input col. unique_length() returns a list.

Examples

```
df <- tibble::tibble(
  x = rep(letters[1:6], each = 1),
  x_grp = rep(letters[1:3], each = 2),
  y = rep(LETTERS[1:3], each = 2),
  y_grp = rep(LETTERS[1:3], each = 2),
  z = rep(LETTERS[1:3], each = 2),
  z_grp = rep(LETTERS[1:3], times = 2))

unique_length(df, "x", "x_grp")

is_one2one(df, "x", "x_grp")
is_one2one(df, "y", "y_grp")
is_one2one(df, "z", "z_grp")
```

 ordination

Helper function for ordination methods

Description

Helper function for ordination methods

Usage

```
ordination(tbl, o_method, d_method = NULL, ...)

ord_plot(ord, score = "st_scores", x = 1, y = 2)

ord_add_group(ord, score = "st_scores", df, indiv, group)

ord_extract_score(ord, score = "st_scores", row_name = NULL)
```

Arguments

<code>tbl</code>	A community data matrix. rownames: stands. colnames: species.
<code>o_method</code>	A string of ordination method. "pca", "ca", "dca", "pcoa", or "nmds". "fspa" is removed, because package dave was archived.
<code>d_method</code>	A string of distance method. "correlation", "manhattan", "euclidean", "canberra", "clark", "bray", "kulczynski", "jaccard", "gower", "altGower", "morisita", "horn", "mountford", "raup", "binomial", "chao", "cao", "mahalanobis", "chisq", "chord", "aitchison", or "robust.aitchison".
<code>...</code>	Other parameters for PCA.
<code>ord</code>	A result of ordination().
<code>score</code>	A string to specify score for plot. "st_scores" means stands and "sp_scores" species.
<code>x, y</code>	A column number for x and y axis.
<code>df</code>	A data.frame to be added into ord scores
<code>indiv, group, row_name</code>	A string to specify indiv, group, row_name column in df.

Value

ordination() returns result of ordination. `$st_scores`: scores for stand. `$sp_scores`: scores for species. `$eig_val`: eigen value for stand. `$results_raw`: results of original ordination function. `$ordination_method`: `o_method`. `$distance_method`: `d_method`. `ord_plot()` returns ggplot2 object. `ord_extract_score()` extracts stand or species scores from ordination result. `ord_add_group()` adds group data.frame into ordination scores.

Examples

```
library(ggplot2)
library(vegan)
data(dune)
data(dune.env)

df <-
  table2df(dune) %>%
  dplyr::left_join(tibble::rownames_to_column(dune.env, "stand"))
sp_dammy <-
  tibble::tibble("species" = colnames(dune),
                 "dammy_1" = stringr::str_sub(colnames(dune), 1, 1),
                 "dammy_6" = stringr::str_sub(colnames(dune), 6, 6))
df <- dplyr::left_join(df, sp_dammy)

ord_dca <- ordination(dune, o_method = "dca")
ord_pca <-
  df %>%
  df2table() %>%
  ordination(o_method = "pca")
```

```
ord_dca_st <-
  ord_extract_score(ord_dca, score = "st_scores")
ord_pca_sp <-
  ord_add_group(ord_pca,
    score = "sp_scores", df, indiv = "species", group = "dammy_1")
```

pad2longest *Pad a string to the longest width of the strings.*

Description

Pad a string to the longest width of the strings.

Usage

```
pad2longest(string, side = "right", pad = " ")
```

Arguments

string Strings.
 side Side on which padding character is added (left, right or both).
 pad Single padding character (default is spaces).

Value

Strings.

Examples

```
x <- c("a", "ab", "abc")
pad2longest(x, side = "right", pad = " ")
```

read_biss *Read data from BiSS (Biodiversity Investigation Support System) to data frame.*

Description

BiSS data is formatted as JSON.

Usage

```
read_biss(txt, join = TRUE)
```


Index

`add_mid_p_bin_w`
 `(draw_layer_construction)`, 5

`bin_width` `(draw_layer_construction)`, 5

`cls_add_group` `(cluster)`, 2
`cls_color` `(cluster)`, 2
`cluster`, 2
`cols_one2multi` `(is_one2multi)`, 8

`d`, 3
`df2table`, 4
`dist2df` `(df2table)`, 4
`distance` `(cluster)`, 2
`draw_layer_construction`, 5

`gen_example`, 6

`h` `(d)`, 3

`ind_val`, 7
`is_multi2multi` `(is_one2multi)`, 8
`is_one2multi`, 8
`is_one2one` `(is_one2multi)`, 8

`mid_point` `(draw_layer_construction)`, 5

`ord_add_group` `(ordination)`, 9
`ord_extract_score` `(ordination)`, 9
`ord_plot` `(ordination)`, 9
`ordination`, 9

`pad2longest`, 11

`read_biss`, 11

`select_one2multi` `(is_one2multi)`, 8
`shdi`, 12

`table2df` `(df2table)`, 4

`unique_length` `(is_one2multi)`, 8