Package 'vegtable'

January 22, 2019

Version 0.1.4

Encoding UTF-8

Title Handling Vegetation Data Sets

Depends R(>= 3.0.0), taxlist

Imports foreign, methods, plotKML, qdapRegex, sp, stringi, vegdata

Suggests devtools, vegan

LazyData true

Description Import and handling data from vegetation-plot databases, especially data stored in 'Turboveg' (<https://www.synbiosys.alterra.nl/turboveg>). Also import/export routines for exchange of data with 'Juice' (<http://www.sci.muni.cz/botany/juice>) are implemented.

License GPL (>= 2)

URL https://github.com/kamapu/vegtable

BugReports https://github.com/kamapu/vegtable/issues

Collate 'NULLing.R' 'coverconvert-class.R"vegtable-class.R"shaker-class.R' 'cross2db.R"clean.R"as.list.R"merge_taxa.R"add_releves.R' 'header.R"Extract.R"veg_relation.R' 'vegtable_stat.R"df2vegtable.R"used_synonyms.R' 'subset.R' 'coverconvert-methods.R"names.R' 'tv2coverconvert.R"tv2vegtable.R' 'crosstable.R"aggregate.R"write_juice.R"read_juice.R"vegtable2kml.R' 'layers2samples.R' 'shaker-methods.R"make_cocktail.R' 'summary.R"match_names.R' 'taxa2samples.R"count_taxa.R' 'StartMessage.R'

NeedsCompilation no

Author Miguel Alvarez [aut, cre] (<https://orcid.org/0000-0003-1500-1834>)

Maintainer Miguel Alvarez <kamapu78@gmail.com>

Repository CRAN

Date/Publication 2019-01-22 13:30:03 UTC

R topics documented:

`add_releves` *Merge relevés from data frames into vegtable objects.*

Description

Addition of plot observations into existing data sets may implicate merging data frames with `vegtable` objects.

Usage

```
## S4 method for signature 'vegtable,data.frame'
add_releves(vegtable, releves, header,
abundance, split_string, usage_ids=FALSE, layers=FALSE, layers_var,
format="crosstable", ...)
```

Arguments

vegtable	An object of class vegtable.
releves	A data frame including plot observations to be added into 'vegtable'.
header	A data frame (optional) including header information for plots.
abundance	A character value (or vector of length 2) indicating the names of abundance variable in 'vegtable'.
split_string	Character value used to split mixed abundance codes.
usage_ids	Logical value indicating whether species are as taxon usage ids (integers) or names in 'releves'.
layers	Logical value indicating whether layers are included in 'releves' or not.
layers_var	Name of the layer variable in 'vegtable'.
format	Character value indicating input format of 'releves' (whether 'crosstable' or 'databaselist').
...	Further arguments passed to function cross2db (i.e. 'na_strings').

Details

Since this function will only update slots 'samples' and 'header', consistency with slots 'layers', 'relations' and 'species' have to be checked and accordingly updated in advance.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

See Also

cross2db.

Examples

```
## No example at the moment
```

aggregate *Aggregating information into a data frame.*

Description

This function aggregates information contained in vegtable objects to a summarizing data frame.

Usage

```
## S4 method for signature 'formula'
aggregate(x, data, FUN, use_nas=TRUE, ...)
```

Arguments

x	A formula indicating the variables used for the summary.
data	Either a data frame or an object of class vegtable.
FUN	Function used to aggregate values.
use_nas	Logical value indicating whether NA's should be included in categorical variables or not.
...	Further arguments passed to the function aggregate.

Details

This function works in a similar way as crosstable.

Value

An object of class data.frame.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

See Also

aggregate.

Examples

```
## Example follows
```

as.list *Coerce an S4 object to a list.*

Description

Coercion used to explore content in S4 objects.

Usage

```
## S4 method for signature 'vegtable'
as.list(x, ...)

## S4 method for signature 'coverconvert'
as.list(x, ...)
```

Arguments

x	an object of class coverconvert or vegtable.
...	further arguments passed from or to other methods.

Details

S4 objects will be coerced to lists, where each slot in the input object becomes a member of the output list. This way allows to explore content and solve problems when validity checks fail.

Value

An object of class list.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
## Importing 'Easplist' from taxlist
library(taxlist)
data(Easplist)

## Head of slot 'taxonNames'
class(Easplist)
head(Easplist@taxonNames)

## The same after coercing to list
Easplist <- as.list(Easplist)
class(Easplist)
head(Easplist$taxonNames)
```

aspect_conv-data *Conversion of Aspect Classes to Azimuth*

Description

Conversion table required to transform values of aspect to azimuth in degrees.

Usage

```
data(aspect_conv)
```

Format

A numeric vector of values in degrees for the symbols used as names.

Author(s)

Miguel Alvarez, <kamapu78@gmail.com>.

Examples

```
library(vegtable)
data(aspect_conv)

aspect_conv[c("N", "S", "ENE", "SSW")]
```

braun_blanquet-data *Conversion of Braun-Blanquet codes to cover percentage.*

Description

Cover values conversion as coverconvert object.

Usage

```
data(braun_blanquet)
```

Format

An object of class coverconvert.

Details

Object of class coverconvert contains conversion tables usually from a categorical variable (a cover scale) to a numerical one (equivalent percentage cover value). cover values are stored as range for each level in the scale (minimum and maximum cover value).

See Also

coverconvert, transform.

Examples

```
library(vegtable)
data(braun_blanquet)

## Quick displays
names(braun_blanquet)
summary(braun_blanquet)
summary(braun_blanquet$b_bbds)
```

---

clean *Clean orphaned records in vegtable object.*

Description

Delete entries in slots header and species orphaned by manipulation of slots.

Usage

```
## S4 method for signature 'vegtable'
clean(object, times=2, ...)
```

Arguments

object	A vegtable object.
times	Numeric value indicating how many times should be the cleaning be repeated.
...	Further arguments passed from or to other methods.

Details

Orphaned records generated by modifications in some slots may cause a loss on the validity of vegtable objects. This function should be applied to optimise the allocated size of a vegtable object, as well. Since running cleaning only once does not assure the deletion of all orphaned entries, it is recommended to run it at least twice. This repetition of cleaning is controlled by the argument 'times'.

Value

A clean vegtable object.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(Kenya_veg)

## Direct manipulation of slot header generates an invalid object
Kenya_veg@header <- Kenya_veg@header[1:50,]
summary(Kenya_veg)

## Now apply cleaning
Kenya_veg <- clean(Kenya_veg)
summary(Kenya_veg)
```

`count_taxa` *Count taxa within a taxlist object*

Description

Counting number of taxa within `taxlist` objects or character vectors containing taxon names.

Usage

```
## S4 method for signature 'vegtable'
count_taxa(object, level, include_lower=FALSE, ...)

## S4 method for signature 'formula'
count_taxa(object, data, include_lower=FALSE, ...)
```

Arguments

`object`	An object of class `vegtable` or a formula.
`data`	An object of class `vegtable`.
`level`	Character value indicating the taxonomic rank of counted taxa.
`include_lower`	Logical value, whether lower taxonomic ranks should be included at the requested level.
`...`	further arguments passed among methods.

Details

This function provides a quick calculation of taxa in `vegtable` objects, considering only records in slot samples. Such records can be also merged from lower ranks.

For the formula method, units without any requested taxa will not appear in the output data frame. If no taxa at all is occurring at the requested level in any unit, an error message will be retrieved.

Value

An data frame with the number of taxa from requested level at requested units for the formula method, or just an integer value.

Author(s)

Miguel Alvarez, <kamapu78@gmail.com>.

Examples

```
library(vegtable)

## Different alternatives
count_taxa(Kenya_veg)
head(count_taxa(~ releveID, Kenya_veg))
head(count_taxa(species ~ releveID, Kenya_veg))
head(count_taxa(species ~ releveID, Kenya_veg, TRUE))
head(count_taxa(family ~ releveID, Kenya_veg, TRUE))
```

---

coverconvert-class *Cover conversion tables.*

Description

Cover conversion tables for vegtable objects.

Details

This class implements conversions from different cover scales in percentage cover. For transformations to percentage cover, the function transform should be than used.

Slots

value List containing the levels of each scale.

conversion List with the respective start and end cut levels for the scale levels.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

See Also

tv2coverconvert, braun_blanquet.

Examples

```
library(vegtable)
showClass("coverconvert")

## Add a custom scale
Scale <- new("coverconvert")
Scale$my_scale <- list(
value=factor(c("low","medium","high"), levels=c("low","medium","high")),
conversion=c(0,50,75,100))
summary(Scale)
```

`crosstable` *Generating cross tables from database lists.*

Description

This function is generating cross tables, which are the most common format used by statistical packages analysing vegetation data (e.g. vegan).

Usage

```
## S4 method for signature 'formula,data.frame'
crosstable(formula, data, FUN, na_to_zero=FALSE,
use_nas=TRUE, ...)

## S4 method for signature 'formula,vegtable'
crosstable(formula, data, FUN, na_to_zero=FALSE,
use_nas=TRUE, ...)

cross2db(object, layers=FALSE, na_strings)
```

Arguments

`formula`	A formula indicating the variables used in the cross table.
`data`	Either a data frame or an object of class `vegtable`.
`FUN`	Function used to aggregate values.
`na_to_zero`	A logical value indicating whether zeros should be inserted into empty cells or not.
`use_nas`	Logical value indicating whether NAs should be considered as levels for categorical variables or not.
`...`	Further arguments passed to the function `aggregate`.
`object`	A data frame including a cross table.
`layers`	Logical value, whether the cross table includes a layer column or not.
`na_strings`	Character vector indicating no records in the cross table.

Details

Most applications and displays of vegetation data use preferentially the cross table format. For convenience, the formula has the form `'abundance ~ plot + species + ...'`. Additional variables used for rows (`...`) can be for instance the layers.

For objects of class `vegtable`, the formula can also include variables from the species list (for example `'AcceptedName'`, `'AuthorName'`) or even taxon traits.

Value

An object of class `data.frame`.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)

## load Kenya_veg and subset to reference 2331 (Bronner 1990)
data(Kenya_veg)
Kenya_veg <- subset(Kenya_veg, REFERENCE == 2331, slot="header")

## transform cover to percentage
Kenya_veg <- transform(Kenya_veg, to="cover_perc", rule="middle")

## cross table of the first 5 plots
Cross <- crosstable(cover_perc ~ releveID + AcceptedName + AuthorName,
        Kenya_veg[1:5,], mean, na_to_zero=TRUE)
head(Cross)
```

---

df2vegtable *Convert a data frame into a vegtable object.*

Description

Conversion of a data frame containing a crosstable of abundance or cover of species in single plots.

Usage

```
df2vegtable(x, species, layer, ...)
```

Arguments

`x`	A data frame formatted for a taxlist object.
`species`	Numeric or integer indicating the position of the column with species names.
`layer`	Numeric or integer indicating the position of the column with layers.
`...`	Further arguments passed from or to other methods.

Details

This function coerces a data frame containing a vegetation cross table into a vegtable object. The input data frame 'x' may include information on the layers or not.

Value

A vegtable object.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
## Creating data set 'dune_veg'
library(vegtable)
library(vegan)

## Load data from vegan
data(dune)
data(dune.env)

## Conversion to vegtable
dune_veg <- data.frame(species=colnames(dune), t(dune), stringsAsFactors=FALSE,
        check.names=FALSE)
dune_veg <- df2vegtable(dune_veg, species=1)

summary(dune_veg)

## Adding environmental variables
dune.env$ releveID <- as.integer(rownames(dune.env))
header(dune_veg) <- dune.env

summary(dune_veg)
```

---

dune_veg-data *Dutch Dune Meadows.*

Description

Data set from the package vegan, converted to a vegtable object.

Usage

```
data(dune_veg)
```

Format

An object of class vegtable.

Source

Original data were imported from dune.

References

Jongman RHG, ter Braak CJF, van Tongeren OFR (1987). *Data analysis in community and landscape ecology.* Pudoc, Wageningen, NL.

Examples

```
library(vegtable)
data(dune_veg)
summary(dune_veg)
```

Extract *Select or replace elements in objects.*

Description

Methods for quick access to slot 'header' of vegtable objects or for access to single cover scales in coverconvert objects. Also replacement methods are implemented.

Usage

```
## S4 method for signature 'vegtable'
x$name

## S4 method for signature 'vegtable'
x[i, j, ..., drop=FALSE]

## S4 method for signature 'coverconvert'
x$name
```

Arguments

x	Object of class vegtable.
...	Further arguments passed to or from other methods.
name	A name to access.
i,j	Indices for access.
drop	A logical value passed to Extract.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(Kenya_veg)

## Range of latitude values in database
range(Kenya_veg$LATITUDE)

## Summary of countries
summary(Kenya_veg$COUNTRY)
summary(droplevels(Kenya_veg$COUNTRY))

## First 5 samples
summary(Kenya_veg[1:5,])
```

---

header *Retrieve or replace slot header in 'vegtable' objects.*

Description

Retrieve or replace the content of slot 'header' in vegtable objects.

Usage

```
## S4 method for signature 'vegtable'
header(x, ...)

header(x) <- value
```

Arguments

x	Object of class vegtable.
value	Data frame to be set as slot header.
...	Further arguments passed to or from other methods.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(Kenya_veg)

head(header(Kenya_veg))
```

`Kenya_veg-data` *Vegetation-Plots from Kenya.*

Description

A subset of SWEA-Dataveg including five references providing plots collected in Kenya.

Usage

```
data(Kenya_veg)
```

Format

An object of class `vegtable`.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>), Michael Curran (<currmi01@gmail.com>).

Source

http://www.givd.info/ID/AF-00-006.

References

Bronner G (1990). *Vegetation and land use in the Mathews Range area, Samburu-District, Kenya*. J. Cramer, Berlin.

Bussmann RW (1994). *The forests of Mount Kenya – vegetation, ecology, destruction and management of a tropical mountain forest ecosystem*. Universität Bayreuth.

Bussmann RW (2002). Islands in the desert – forest vegetation of Kenya's smaller mountains and highland areas. *Journal of East African Natural History* 91: 27–79.

Fujiwara K, Furukawa T, Kiboi SK, Mathenge S, Mutiso P, Hayashi H, Meguro S (2014). Forest types and biodiversity around the Great Rift Valley in Kenya. *Contributii Botanice* 49: 143–178.

Schmitt K (1991). *The vegetation of the Aberdare National Park Kenya*. Universitätsverlag Wagner, Innsbruck.

Examples

```
library(vegtable)
data(Kenya_veg)
summary(Kenya_veg)
```

layers2samples *Add information from slot 'layers' into slot 'samples'.*

Description

Slot layers may include additional information that should be moved to samples in order to use it by subset, aggregate or crosstable methods.

Usage

```
## S4 method for signature 'vegtable,character,character'
layers2samples(object, layer,
variable, ...)
## S4 method for signature 'vegtable,character,missing'
layers2samples(object, layer,
variable, ...)
```

Arguments

object	An object of class vegtable.
layer	Character value indicating a target layer.
variable	Character vector with the names of variables to be inserted in slot 'samples'.
...	Further arguments to be passed among methods.

Details

If names of variables are not provided, all variables from the respective layer table will be inserted in slot 'samples'.

Value

An object of class vegtable with variables added to samples.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
## No example available for this function.
```

make_cocktail *Produce a Cocktail classification.*

Description

Classification of vegtable objects according to **Cocktail** algorithms.

Usage

```
## S4 method for signature 'shaker,vegtable'
make_cocktail(shaker, vegtable, which, cover,
syntax="Syntax", FUN=sum, ...)
```

Arguments

shaker	An object of class shaker containing the respective cocktail definitions.
vegtable	An object of class vegtable containing the vegetation observations to be classified.
which	Integer or character indicating the definition to be applied for classification.
cover	Name of the cover variable in vegtable.
syntax	Character value indicating the name of the retrieved variable including the final classification of plots.
FUN	Function used for merging multiple occurrence of species in a single plot.
...	Further arguments passed from or to other methods.

Details

Cocktail algorithms are logical functions selecting plots according to either occurrence of species groups and cover values of single species. A group will be declared as occurring in a plot when at least a half of its members is present in the plot.

This function inserts single columns with logical values indicating whether a plot is classified in the vegetation unit or not. An additional column (name provided in argument 'syntax') compile all vegetation units, indicating with a '+' symbol those plots classified in more than one vegetation unit. When only a part of the formulas will be used, it should be specified by the argument 'which'.

Value

A data frame corresponding to the slot 'header' of input object 'vegtable', including the results of Cocktail classification for the respective plots.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

References

Alvarez M (2017). Classification of aquatic and semi-aquatic vegetation in two East African sites: Cocktail definitions and syntaxonomy. *Phytocoenologia*.

Bruelheide H (2000). A new measure of fidelity and its application to defining species groups. *Journal of Vegetation Science* 11: 167–178.

Kočí M, Chytrý M, Tichý L (2003). Formalized reproduction of an expert-based phytosociological classification: a case study of subalpine tall-forb vegetation. *Journal of Vegetation Science* 14: 601–610.

See Also

shaker, vegtable, Wetlands.

Examples

```
library(vegtable)

## Example from Alvarez (2017)
data(Wetlands)

Wetlands_veg@header <- make_cocktail(Wetlands, Wetlands_veg, cover="percen")
summary(as.factor(Wetlands_veg@header$Syntax))

## Same but only for two vegetation units
data(Wetlands)
Wetlands_veg@header <- make_cocktail(Wetlands, Wetlands_veg,
which=c("HY1","HY2"), cover="percen")
summary(as.factor(Wetlands_veg@header$Syntax))
```

---

match_names *Search matchings between character and taxlist objects.*

Description

Names provided in a character vector will be compared with names stored in slot `'taxonNames'` of an object of class taxlist by using the function stringsim.

Usage

```
## S4 method for signature 'character,vegtable'
match_names(x, object, ...)
```

Arguments

x	A character vector with names to be compared.
object	An object of class taxlist to be compared with.
...	Further arguments passed to match_names.

Details

This method is applied to the slot 'species' in the input vegtable object.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

See Also

match_names, stringsim.

Examples

```
## No example at the moment
```

---

merge_taxa *Merge concepts.*

Description

Merge taxon concepts form into single ones or insert accepted names to slot samples.

Usage

```
## S4 method for signature 'vegtable,numeric,missing'
merge_taxa(object, concepts, level, ...)

## S4 method for signature 'vegtable,missing,character'
merge_taxa(object, concepts, level, ...)

## S4 method for signature 'vegtable'
taxa2samples(object, merge_to, ...)
```

Arguments

object	Object of class vegtable.
concepts	Numeric (integer) vector including taxon concepts to be merged.
level,merge_to	Character value indicating the level to which the taxa have to be merged.
...	Further arguments passed to merge_taxa ('taxlist' method).

Details

This method is applied to a function defined in the package taxlist and only modify the slot 'species' in the input 'object'.

The use of 'taxa2samples' with 'merge_to' argument will produce a similar result as using 'merge_taxa' with 'level' argument, but 'taxa2samples' will replace the records in slot samples by the respective accepted names without any modification in slot species.

Value

An object of class vegtable.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(Kenya_veg)

## Merge Olea capensis into one
summary(subset(Kenya_veg@species, grepl("Olea capensis", TaxonName),
slot="names"), "all")
Kenya_veg <- merge_taxa(Kenya_veg, c(52041,50432,50235))

## Check Olea capensis again
summary(subset(Kenya_veg@species, grepl("Olea capensis", TaxonName),
slot="names"), "all")

## Effect of taxa2samples by counting taxa
count_taxa(Kenya_veg, level="genus")

Kenya_veg <- taxa2samples(Kenya_veg, merge_to="genus")
count_taxa(Kenya_veg, level="genus")
```

---

names *Retrieve names of vegtable and coverconvert objects.*

Description

Quick access to column names in slot header and names of conversion codes.

Usage

```
## S4 method for signature 'coverconvert'
names(x)

## S4 method for signature 'vegtable'
names(x)

## S4 method for signature 'vegtable'
dimnames(x)
```

Arguments

x	An object of class vegtable or coverconvert.

Details

These methods provide a quick display of the contents in coverconvert and vegtable objects.

Value

Either a vector or a list (in the case of 'dimnames') with the names of variables.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(Kenya_veg)

## All possibilities shown in 'Usage'
names(Kenya_veg@coverconvert)
names(Kenya_veg)
dimnames(Kenya_veg)
```

---

read_juice *Importing 'Juice' tables.*

Description

This function imports vegetation tables exported from 'Juice' (http://www.sci.muni.cz/botany/juice).

Usage

```
read_juice(file, encoding="LATIN-1", sep=";", na="", ...)
```

Arguments

file	Character value indicating the name of the file exported from 'Juice'.
encoding	Argument passed to readLines.
sep	Separator used to split rows into columns.
na	Character used as not available values.
...	Further arguments passed to readLines.

Details

For a properly import, you may strictly follow the export steps in 'Juice':

- Menu `File -> Export -> Table -> to Spreadsheet Format File`
- Check the option `Export covers in %`

In the `header` (see **Value**), the first column (`juice_nr`) corresponds to the plot number assigned by 'Juice' at import, while the column `db_nr` is the number originally assigned to the plot (e.g. 'Turboveg' ID).

Value

A list with two elements:

cross_table A data frame of species by plot.

header A data frame with header data.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)

## Installed 'Juice' version of 'Wetlands_veg'
Veg <- file.path(path.package("vegtable"), "juice", "Wetlands_juice.txt")
Veg <- read_juice(Veg)
names(Veg)
```

---

set_pseudo,set_group,set_formula

Set slots in shaker objects.

Description

Building `shaker` objects for Cocktail classifications.

Usage

```
## S4 method for signature 'shaker,taxlist,character'
set_pseudo(shaker, companion, pseudo,
pseudo_id, authority=FALSE, enc_cont="latin1", enc_gr="utf8", ...)

## S4 method for signature 'shaker,vegtable,character'
set_pseudo(shaker, companion, pseudo, ...)

## S4 method for signature 'shaker,taxlist,character'
```

```
set_group(shaker, companion, group,
group_id, authority=FALSE, enc_cont="latin1", enc_gr="utf8", ...)

## S4 method for signature 'shaker,vegtable,character'
set_group(shaker, companion, group, ...)

## S4 method for signature 'shaker,taxlist,character'
set_formula(shaker, companion, formula,
formula_id, authority=FALSE, enc_cont="latin1", enc_gr="utf8", ...)

## S4 method for signature 'shaker,vegtable,character'
set_formula(shaker, companion, formula,
...)
```

Arguments

`shaker`	Object of class `shaker` to be modified.
`companion`	Either a `taxlist` or a `vegtable` object.
`pseudo,group`	Character vector with names of taxa included in a pseudo-species or a species group.
`formula`	Character vector including a formula as definition of a vegetation unit.
`pseudo_id,group_id,formula_id`	Character value as name of the pseudo-species, species group or defined vegetation unit.
`authority`	Logical value indicating whether author names should be included in the taxon name or not.
`enc_cont,enc_gr`	Encodings used for special characters.
`...`	Further arguments passes from or to other methods.

Details

These functions are implemented for constructing or complementing `shaker` objects. Note that construction of those objects will always require a `'companion'` object, which is either an object of class `taxlist` or `vegtable`.

Value

A `shaker` object.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

See Also

`shaker`, `make_cocktail`.

Examples

```
library(vegtable)
data(Wetlands)

## Construct the 'shaker' object anew
Wetlands <- new("shaker")

## Set a pseudo-species
Wetlands <- set_pseudo(Wetlands, Wetlands_veg, c("Cyperus latifolius",
"Cyperus exaltatus"))

## Set a species group
Wetlands <- set_group(Wetlands, Wetlands_veg, group_id="Cyperus papyrus",
group=c(
                "Cyperus papyrus",
                "Cyclosorus interruptus",
                "Lepistemon owariense"))

## Set a fromula
Wetlands <- set_formula(Wetlands, Wetlands_veg, formula_id="HE1",
formula="groups:'Cyperus papyrus' | species:'Cyperus papyrus > 50'")

## Summaries
summary(Wetlands)
summary(Wetlands, Wetlands_veg)
```

---

shaker-class *Class containing Cocktail algorithms.*

Description

Objects used for collecting Cocktail definitions.

Details

These objects work as **expert systems** for recognition of defined vegetation units among plots of a `vegtable` object. A `'shaker'` object will be always dependent on a `vegtable` object, which is called `'companion'`. Since modifications in the `companion` may affect the functionality of the `'shaker'` object, it will be recommended to create the last during a session by a source script instead of recycling them from old R images.

Slots

pseudos List containing IDs of taxa that will be merged into pseudo-species.

groups List containing IDs of taxa belonging to the same Cocktail group.

dominants data frame including lists of species used as dominant species in Cocktail algorithms, as well as operators and cover values used in the formulas.

formulas List with formulas that will be used as definitions for vegetation units.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

See Also

set_pseudo, set_group, set_formula, make_cocktail.

Examples

```
library(vegtable)
showClass("shaker")
```

---

subset *Subset functions for vegtable objects.*

Description

Produce subsets of vegtable objects.

Usage

```
## S4 method for signature 'vegtable'
subset(x, subset, slot="header", keep_children=FALSE,
keep_parents=FALSE, ...)
```

Arguments

x	A vegtable object for subset.
subset	Logical vector or operation for subset.
slot	Slot to be applied for subset.
keep_children	Argument passed to subset.
keep_parents	Argument passed to subset.
...	Further arguments passed from or to other methods.

Details

This function generate subsets of vegtable objects through logical operations. Such operations can be applied either to the plots, or the relations, which are the main slots in that class.

This method can be referred to the slot species the same way as subset, then the rest of the data will include only references to the subset of species list.

Value

A S4 object of class vegtable.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(dune_veg)
summary(dune_veg)

## Select plots used as pastures
Pastures <- subset(dune_veg, Use == "Pasture", slot="header")
summary(Pastures)
```

---

summary *Summary method for vegtable objects.*

Description

Display summaries for vegtable objects.

Usage

```
## S4 method for signature 'vegtable'
summary(object, units="Kb", ...)

## S4 method for signature 'coverconvert'
summary(object, ...)

## S4 method for signature 'shaker'
summary(object, companion, authority=FALSE, ...)
```

Arguments

object	Object to be summarized.
units	Units used for object size (passed to format).
companion	Companion object (either a taxlist or a vegtable object.
authority	Logical value indicating whether authors should be displayed or not.
...	further arguments to be passed to or from other methods.

Details

Those methods are implemented for objects of the classes vegtable, coverconvert and shaker.

The method for class 'vegtable' retrieves the metadata, the size of the object, its validity and additional statistics on the content of input object.

For objects of class shaker, the function 'summary' will either retrieve general statistics when 'companion' is missing, or a more detailed display when accompained by a taxlist or vegtable object.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)

## Summary for 'vegtable' objects
data(Wetlands)
summary(Wetlands_veg)

## Summary for 'coverconvert' objects
data(braun_blanquet)
summary(braun_blanquet)

## Summary for 'shaker' objects (alone and with companion)
summary(Wetlands)
summary(Wetlands, Wetlands_veg)
```

---

transform *Convert cover scales to percent cover.*

Description

Convert values of a categorical cover scale to percentage values.

Usage

```
## S4 method for signature 'character,coverconvert'
transform(x, conversion, from=NULL,
rule="top", zeroto=0.1, ...)

## S4 method for signature 'factor,coverconvert'
transform(x, conversion, ...)

## S4 method for signature 'vegtable,missing'
transform(x, to, replace=FALSE, rule="top",
zeroto=0.1, ...)
```

Arguments

x	Either a factor or character vector, or a vegtable object.
conversion	An object of class coverconvert.
from	Scale name of values in 'x' as character value.
to	Name of the column in slot 'samples' for writing converted values.
replace	Logical value indicating whether existing cover values should be replaced or not.

rule	Rule applied for the conversion (see details).
zeroto	Value used to replace levels with bottom at 0% cover.
...	Further arguments passed from or to other methods.

Details

This function requires as input a coverconvert object which contains the conversion tables.

In the case of vegtable objects, the conversion is already embedded in the slot 'coverconvert'.

Three rules are implemented for transformation, either 'top' (values transformed to the top of the range), 'middle' (transformation at the midpoint), and 'bottom' (conversion at the lowest value of the range). In the later case, transformation ranges starting at 0% of cover can be set to a different value by the argument 'zeroto'.

When 'replace=FALSE', existing values of cover in the vegtable object will be maintained. Since there is not a standard naming of cover values, in the transformation the name of cover variable should be indicated in the argument 'to'.

Value

Either a vector or a vegtable object.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(Kenya_veg)

## Check the available scales
summary(Kenya_veg@coverconvert)

## Conversion by default 'top' rule
Kenya_veg <- transform(Kenya_veg, to="percent")
summary(as.factor(Kenya_veg@samples$percent))

## Conversion by 'middle' rule
Kenya_veg <- transform(Kenya_veg, to="percent", rule="middle", replace=TRUE)
summary(as.factor(Kenya_veg@samples$percent))

## Conversion by 'bottom' rule
Kenya_veg <- transform(Kenya_veg, to="percent", rule="bottom", replace=TRUE)
summary(as.factor(Kenya_veg@samples$percent))
```

tv2coverconvert *Importing conversion tables from 'Turboveg' databases.*

Description

This function reads the content of cover conversion tables stored in 'Turboveg' and attempts to reformat them in a more comprehensive structure.

Usage

```
tv2coverconvert(file, as.is=TRUE)
```

Arguments

file	A connection to a DBF file containing conversion table in 'Turboveg'.
as.is	A logical value passed to read.dbf

Details

This function is used by tv2vegtable to import the respective conversion table from 'Turboveg' databases. Note that conversion tables in 'Turboveg' have only stored the middle point for each cover class in a scale, thus it will be recommended to rebuild the 'coverconvert' slot or use braun_blanquet.

Value

A coverconvert object.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

See Also

tv2vegtable, read.dbf.

Examples

```
library(vegtable)

## Installed 'Turboveg' version of "Fujiwara et al. (2014)"
TV_Home <- file.path(path.package("vegtable"), "tv_data", "popup", "Swea")
Table <- tv2coverconvert(file.path(TV_Home, "tvscale.dbf"))

## First scale have to be deleted from conversion table
Table@value <- Table@value[-1]
Table@conversion <- Table@conversion[-1]
summary(Table)
```

```
## Compare the 'Turboveg' version with a vegtable version
data(braun_blanquet)
summary(Table$br_bl)
summary(braun_blanquet$br_bl)
```

`tv2vegtable` *Import vegetation data from a 'Turboveg' databases.*

Description

`tv2vegtable` imports vegetation data sets from 'Turboveg' data bases.

Usage

```
tv2vegtable(db, tv_home=tv.home(), skip_empty_relations=TRUE, skip_scale,
clean=TRUE)
```

Arguments

`db`	Name of 'Turboveg' data base as character value.
`tv_home`	'Turboveg' installation path as character value.
`skip_empty_relations`	Logical value indicating whether empty relations may be excluded from imported database or not.
`skip_scale`	Character value indicating scales to be excluded in slot `'coverconvert'`.
`clean`	Logical value indicating whether output object should be cleaned or not.

Details

Import function for 'Turboveg' databases into an object of class `vegtable`. Most of the contents of 'Turboveg' databases are included in DBF files and therefore imported by the function `read.dbf`. The automatic setting of database path will be done by the function `tv.home` but it can be customised by the argument `'tv_home'`.

The species list will be imported by using the function `tv2taxlist` and therefore formatted as a `taxlist` object. Similarly, conversion tables will be handled as `coverconvert` objects.

Empty columns in the header will be deleted in the imported object.

Value

A `vegtable` object.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

See Also

tv2taxlist, tv2coverconvert, tv.home.

Examples

```
library(vegtable)

## Installed 'Turboveg' version of 'Fujiwara et al. (2014)'
TV_Home <- file.path(path.package("vegtable"), "tv_data")
Veg <- tv2vegtable("Fujiwara_2014", TV_Home)
summary(Veg)
```

---

used_synonyms *Retrieve synonyms used in the data set.*

Description

Plots records are rather linked to plant names than plant taxon concepts. This function provides a quick report about synonyms used in a data set (a vegtable object) and their respective accepted name.

Usage

```
## S4 method for signature 'vegtable'
used_synonyms(x, ...)
```

Arguments

x	A vegtable object.
...	Further arguments to be passed from or to another methods.

Details

This function will only retrieve synonyms that are used in plot records.

Value

A data frame with following columns:

SynonymsID Usage ID of synonyms.

Synonym The synonym itself.

SynonymAuthor Author of synonym.

TaxonConceptID ID of the taxon concept.

AcceptedNameID Usage ID of the accepted name.

AcceptedName The respective accepted name.

AcceptedNameAuthor The author of the accepted name.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

See Also

accepted_name.

Examples

```
library(vegtable)
data(Kenya_veg)

## Synonyms used in the Kenya_veg
Synonyms <- used_synonyms(Kenya_veg)
head(Synonyms)
```

---

vegtable-class *Class vegtable.*

Description

Class holding vegetation-plot data sets. Designed to content all information stored in 'Turboveg' databases in just one object.

Details

This class was designed to include information of relevés, header data and species in just one object. Objects can be created by calls of the form `new("vegtable", ...)`.

Slots

description A named character vector containing metadata.

samples A data frame with samples list.

header A data frame with plots data.

species Species list as a taxlist object.

layers A list including strata within samples as data frames.

relations A list including popup lists as data frames.

coverconvert A scale conversion object of class coverconvert.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

See Also

tv2vegtable.

Examples

```
library(vegtable)
showClass("vegtable")
```

---

vegtable2kml *Mapping of plot observations.*

Description

This function is a wrapper of kml producing and displaying KML files.

Usage

```
## S4 method for signature 'data.frame'
vegtable2kml(obj, file, coords=~ Longitude + Latitude,
                srs=CRS("+proj=longlat +datum=WGS84"))

## S4 method for signature 'vegtable'
vegtable2kml(obj, file, coords=~ LONGITUDE + LATITUDE,
                srs=CRS("+proj=longlat +datum=WGS84"))
```

Arguments

obj	Input object containing coordinate values.
file	Character value with the name of output file (including file extension).
coords	Either a character vector or a formula indicating the names of coordinate values.
srs	Spatial reference system as proj4string.

Details

Georeferenced plots can be quickly displayed in Google Earth using this function.

Value

A KML file, which will be automatically opened in **Google Earth**.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(Kenya_veg)

## Plots containing Podocarpus observations
Kenya_veg@species <- subset(Kenya_veg@species, grepl("Podocarpus", TaxonName),
slot="names")

Kenya_veg <- subset(Kenya_veg, TaxonUsageID %in%
Kenya_veg@species@taxonNames$TaxonUsageID, slot="samples")

## Not run: vegtable2kml(Kenya_veg, "Podocarpus.kml")
```

---

vegtable_stat *General statistics from vegtable objects.*

Description

This function calculates general statistics of local 'Turboveg' databases as required by GIVD (Global Index of Vegetation-Plot Databases, https://www.givd.info).

Usage

```
vegtable_stat(vegtable)
```

Arguments

vegtable	An object of class vegtable.

Details

This function is based on a script delivered by GIVD for summarising statistics required in the descriptions of databases (see meta data in the page of the Global Index for Vegetation-Plot Databases).

Author(s)

GIVD. Adapted by Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(Kenya_veg)

## Statistics for GIVD
vegtable_stat(Kenya_veg)
```

`veg_relation` *Retrieve or replace relations in vegtable objects.*

Description

Tables providing information about levels of categorical variables in the header of a 'Turboveg' database are called `'popups'` in 'Turboveg', but `'relations'` in `vegtable`. Such variables will be converted into factors in the slot `'header'` according to the levels and their sorting in the respective relation.

Usage

```
## S4 method for signature 'vegtable,character'
veg_relation(vegtable, relation,
match_header=FALSE, ...)

veg_relation(vegtable, relation) <- value
```

Arguments

`vegtable`	An object of class `vegtable`.
`relation`	A character value indicating the veg_relation to be retrieved or replaced.
`match_header`	A logical vector, whether only levels occurring in slot `'header'` should be considered or all.
`value`	A data frame containing the new veg_relation.
`...`	Further arguments to be passed among methods.

Value

This function retrieves and object of class `'data.frame'`. In the replacement method, an object of class `vegtable` including `'value'` in the slot `'relations'`.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(Kenya_veg)

## overview of references
veg_relation(Kenya_veg, "REFERENCE")
```

Wetlands-data *Vegetation-plots from Tanzania.*

Description

A subset of SWEA-Dataveg with plots sampled in Tanzania.

Usage

```
data(Wetlands)
```

Format

An object of class `shaker` (`'Wetlands'`) and the respective companion as `vegtable` object (`'Wetlands_veg'`).

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>), Michael Curran (<currmi01@gmail.com>).

Source

http://www.givd.info/ID/AF-00-006.

References

Alvarez M (2017). Classification of aquatic and semi-aquatic vegetation in two East African sites: Cocktail definitions and syntaxonomy. *Phytocoenologia*.

Examples

```
library(vegtable)
data(Wetlands)

summary(Wetlands)
summary(Wetlands_veg)
```

write_juice *Exporting Tables for 'Juice'.*

Description

This function produce txt files as inport formats for 'Juice' (http://www.sci.muni.cz/botany/juice/).

Usage

```
## S4 method for signature 'vegtable,character,formula'
write_juice(data, file, formula,
header=c("COUNTRY","REFERENCE"), coords=c("LONGITUDE","LATITUDE"), FUN, ...)
```

Arguments

data	An object of class vegtable.
file	Character value indicating the name of output files (without file extension).
formula	A formula passed to crosstable.
header	Variables of header to be exported.
coords	Names of coordinate variables in header of 'data'.
FUN	Funtion passed to crosstable.
...	Further arguments passed to the function crosstable.

Details

This function produces two output files to be imported into a 'Juice' file: A vegetation table produced by crosstable and a header table. Both tables share the file name plus a suffix (table for the vegetation table and header for the header).

For the import in 'Juice', you may start with the table following in the menu 'File -> Import -> Table -> from Spreadsheet File (e.g. EXCEL Table)' and then follow the wizard. You further import the header table following in the menu 'File -> Import -> Header Data -> From Comma Delimited File'. Notice that the vegetation is a semi-colon delimited file, while the header is a comma delimited file.

Author(s)

Miguel Alvarez (<kamapu78@gmail.com>).

Examples

```
library(vegtable)
data(Kenya_veg)

## Only first 20 observations
Kenya_veg <- Kenya_veg[1:20,]
## Not run:
write_juice(Kenya_veg, "SWEA", FUN=mean)

## End(Not run)
```

Index