

Package ‘vegdata’

July 2, 2014

Version 0.6-6

Encoding UTF-8

Date Jan 2014

Title Functions to access vegetation databases (Turboveg) and prepare vegetation data especially its taxonomy for analysis

Author Florian Jansen <jansen@uni-greifswald.de>

Maintainer Florian Jansen <jansen@uni-greifswald.de>

Depends R (>= 3.0.0), foreign

Imports XML

Suggests vegan, labdsv, akima, gWidgets, googleVis, indicpecies, knitr

LazyData Yes

VignetteBuilder knitr

Description Handling of vegetation data sets from Turboveg, VegetWeb and ESVeg Exports. Taxonomic harmonization of datasets with appropriate taxonomic lists (e.g. the German taxonomical standard list ``GermanSL’’).

License GPL (>= 2)

URL <http://www.botanik.uni-greifswald.de/GermanSL.html>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-01-24 18:46:55

R topics documented:

vegdata-package	2
elbaue	3
ESveg	4
lc	5
monotypic	6
syntab	7
tax	8
taxval	10
tv.biblio	12
tv.compRefl	13
tv.coverperc	14
tv.metainfo	15
tv.obs	15
tv.site	16
tv.traits	17
tv.veg	18
tv.write	20
Index	22

vegdata-package	<i>Functions to access data from vegetation databases and evaluate taxon names (with GermanSL).</i>
-----------------	---

Description

This package provides a set of functions to load data from (at present Turboveg and VegetWeb) databases. It is also possible to semi-automatically check and adapt scientific plant names (with appropriate reference lists) and to produce a syntaxonomic (rel/abs) frequency table.

Details

Package:	vegdata
Type:	Package
License:	GPL version 2 or newer
LazyLoad:	yes

Use `tv.veg` to prepare data directly for further analyses. Set option `taxval` to TRUE, if your database is referenced with GermanSL or equivalent taxonomic reference list and you want to realize taxonomic checks and adaptations. For more details see `vignette('vegdata')`.

Author(s)

Florian Jansen

Maintainer: Florian Jansen <jansen@uni-greifswald.de>

References

Jansen, F., Dengler, J (2011) Plant names in vegetation databases - a neglected source of bias, *Journal of vegetation science*, 21(6), 1179-1186. <http://dx.doi.org/10.1111/j.1654-1103.2010.01209.x>

Jansen, Florian and Dengler, Juergen (2008) GermanSL - eine universelle taxonomische Referenzliste fuer Vegetationsdatenbanken, *Tuexenia*, 28, 239-253.

elbaue

Species Data and Altitude from floodplains of the river Elbe, Germany.

Description

The elbaue data frame has 33 sites (rows) and 53 species (columns).

Details

Data frame elbaue.env contains the following variables:

RELEVE_NR a unique number

DATE a fictitious date of the plot survey

SURF_AREA the plot area

FLOOD logical, inundated floodplain ("Altaue"); (1=recently inundated, 0= no inundation)

OLD logical; former floodplain = Altaue; Durch Deiche von der rezenten Aue getrennter Auenbereich;(1= old floodplain)]

BORDER border of floodplain = Auenrand; Grenze der Aue zu anderen Naturraeumen, haeufig vermoort (1= Auengrenzbereich)

INTENS intensity of land use = Intensitaet der Landnutzung; Drei Klassen= 1: sporadische Nutzung, 2: jaehrliche Nutzung (geringe Intensitaet), 3: jaehrliche Nutzung

MGL mean groundwater level = Mittlerer Grundwasserstand [cm]; ueber zwei Jahre aus Tageswerten gemittelter Wasserstand

SDGL standard deviation of groundwater level = Standardabweichung der Wassergang-Zeitreihe [cm]; Mass fuer die Groesse der ueber zwei Jahre gemittelten Wasserstaenden

InUnD duration of inundation period = Ueberflutungsdauer Log[Tage/Jahr]; Logarithmus der ueber zwei Jahre gemittelten Ueberflutungsdauer

InUnD_50 duration of inundation period above 50cm = Dauer von Wasserstaenden hoeher 50cm ueber Flur Log[Tage/Jahr]; Logarithmus der ueber zwei Jahre gemittelten Werte

References

Leyer, Ilona and Wesche, Carsten 2007: *Multivariate Statistik in der Oekologie*, p. 221, Springer, Berlin.

Examples

```
## Not run:
  elbaue <- tv.veg('elbaue')
  elbaue.env <- tv.site('elbaue')

## End(Not run)
```

ESveg

Load vegetation data from ESveg formatted data files

Description

Tabulates vegetation tables resp. header data from ESveg formatted file. Until now only very basic support, without any taxonomic emendation and layer combination.

Usage

```
ESveg.obs(db, ...)
ESveg.site(db, ...)
```

Arguments

db	Name and path to your ESveg formatted file
...	additional arguments

Value

The function returns an matrix object of class 'veg'

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

Examples

```
## Not run:
path <- system.file(package = "vegdata")
obs <- ESveg.obs(file.path(path,'tvdata', 'Data', 'elbaue.xml'))
veg <- tv.veg(obs=obs, refl='GermanSL 1.2', taxval=TRUE, convcode=FALSE, )
dimnames(veg)

## End(Not run)
```


layer	comb
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

See Also

[tv.veg](#)

monotypic	<i>Creates a list of (regional) monotypic taxa from the given taxonomic checklist</i>
-----------	---

Description

More or less internal function to check the reference lists and to create lists of monotypic taxa.

Usage

```
monotypic(refl, nr.member = 1, reflist.type = c('Turboveg', 'EDIT'), write = FALSE,
filename, tv_home, ...)
```

Arguments

refl	The name of the taxonomic reference list.
nr.member	Number of members in the next taxonomic level to be checked.
reflist.type	Type or origin of the taxonomic list: Turboveg 2.0 format or from the European Distributed Institut of Taxonomy.
write	Should the list of monotypic species be written into a CSV file for further use.
filename	Name of the file in case of write=TRUE
tv_home	Turboveg installation path, see tv.home
...	additional arguments

Value

Dataframe of monotypic taxa.

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

syntab	<i>Syntaxonomic frequency tables</i>
--------	--------------------------------------

Description

Calculate and display relative or absolute frequency tables with or without use of function `multipatt` from package `indicspecies`

Usage

```
syntab(veg, clust, type = c('rel', 'abs', 'mean.cover'), fullnames=FALSE,
mupa=NULL, dec=0, refl, ...)
## S3 method for class 'syntab'
print(x, zero.print = ".", trait, limit = 1, minstat = 0, alpha = 0.05, ...)
```

Arguments

<code>veg</code>	Vegetation dataframe
<code>clust</code>	Vector with cluster information with length equal to number of rows of <code>veg</code>
<code>type</code>	Relative or absolute frequency, mean species response values or strength of association (see function <code>multipatt</code> in package <code>indicspecies</code>).
<code>fullnames</code>	Replace rownames (LETTERCODES) with full scientific names.
<code>mupa</code>	Either logical for (not) using <code>multipatt</code> from package <code>indispecies</code> to detect significance of cluster association strength or supply output from previous use of <code>multipatt</code> .
<code>x</code>	Object from function <code>syntab</code>
<code>zero.print</code>	Replacement for zero values.
<code>trait</code>	Optional vector of trait values to be plotted behind the species.
<code>limit</code>	Minimum value to display.
<code>minstat</code>	Minimal indicator value
<code>alpha</code>	Significance threshold.
<code>dec</code>	Number of decimals in result.
<code>refl</code>	Name of Turboveg taxonomic reference list to use for <code>fullnames</code> .
<code>...</code>	additional arguments

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

See Also

package `indicspecies` from M. Caceres with function `multipatt` for indicator species analysis along multiple cluster combinations

Examples

```
## Not run:
elbaue <- tv.veg('elbaue')
elbaue.env <- tv.site('elbaue')
clust <- vector('integer', nrow(elbaue.env))
clust[elbaue.env$MGL < -50 & elbaue.env$SDGL < 50] <- 1
clust[elbaue.env$MGL < -50 & elbaue.env$SDGL >= 50] <- 2
clust[elbaue.env$MGL >= -50 & elbaue.env$SDGL >= 50] <- 3
clust[elbaue.env$MGL >= -50 & elbaue.env$SDGL < 50] <- 4
levels(clust) <- c('dry.ld','dry.hd', 'wet.hd','wet.ld')
traits <- tv.traits()
trait <- data.frame(EIV_F = traits$OEK_F, EIV_N = traits$OEK_N)
rownames(trait) <- traits$ABBREVIAT
st <- syntab(elbaue, clust, mupa=TRUE, fullnames=TRUE)
print(st, limit=30, trait=trait)

## End(Not run)
```

tax

Query of Turboveg 2 taxonomic reference lists including concept synonymy and taxonomic hierarchy.

Description

Input is either species number (integer), shortletter (7 characters) or full (exact!) species name.

Usage

```
## Default S3 method:
tax(x, refl, verbose = FALSE, syn = TRUE, concept = NULL, strict = FALSE,
vernacular = FALSE, simplify = FALSE, quiet = FALSE, reflist.type = 'Turboveg', ...)
child(x, refl = tv.refl(), gen = 4, tree = FALSE, quiet = FALSE, syn = FALSE, ...)
parent(x, refl = tv.refl(), rank, quiet = FALSE, ...)
syn(x, refl = tv.refl(), quiet = FALSE, ...)
```


Arguments

x	Species number, lettercode or species name
refl	Taxonomic reference list
verbose	Load tax.dbf with additional taxonomic information (e.g. Secundum) instead of species.dbf
syn	Return also synonym names
concept	Name of alternative taxon view file within the reference list director
strict	Exact match or partial matching with grep
vernacular	Search in vernacular names instead of scientific names.
simplify	Will simplify species names for matching.
gen	Number of child generations to return
quiet	Hide screen messages
reflist.type	Type of taxonomic reference list to use. Until now only Turboveg lists are supported in the official package.
tree	Opens a gWidgets window with interactive taxonomic tree view. Requires package gWidgets
rank	Taxonomical level of parentship to find
...	additional attributes

Details

concept: GermanSL is a list with a single taxon view according to the standard lists of the different taxon groups (e.g. Wisskirchen and Haeupler for higher plants, see). Nevertheless a high number of synonyms is included which allows in many cases the transformation into different concepts. For illustration the concept of *Armeria maritima* from Korneck 1996 is included, which accepts e.g. *Armeria maritima ssp. bottendorfensis*. *simplify*: Before string comparison with reference list names it will eliminate diacritic marks, double consonants, "th", "y" and others. If `genus=TRUE` critical, i.e. non stable endings of genus names will be ignored, `epithet = TRUE` will eliminate specific endings for the last epithet.

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

References

Jansen, F. and Dengler, J. (2008) GermanSL - eine universelle taxonomische Referenzliste für Vegetationsdatenbanken. *Tuexenia*, 28, 239-253.

See Also

package vegdata

Examples

```
## Not run:
## GermanSL in Turboveg installation path needed
tax(27)
tax('Achillea millefolium')
tax('ACHIMILL')

## End(Not run)
## Not run:
childs(0, gen=1)
childs(94419, tree=TRUE)

## End(Not run)
```

taxval

Handling of taxonomy in vegetation data.

Description

Performs taxonomic valuation of species names according to synonymy, taxonomic level, unambiguity of biotic content etc. Necessary prerequisite is information about taxonomic status (synonymy) and hierarchy (next higher aggregat). Until now only applicable for reference list 'GermanSL' (>= vers. 1.1, see References Section), which is valid in Germany and adjacent countries.

Usage

```
taxval(obs, refl, db, concept=NULL, syn = c('adapt','conflict','preserve'),
ag = c('conflict', 'adapt', 'preserve'),
rank, mono = c('species', 'higher', 'lower', 'preserve'), monolist = "monotypic-D",
uncertain = NULL, maxtaxlevel = 'ROOT', check.critical = TRUE, sink = TRUE, ...)
comb.species(x, sel, newname, refl)
```

Arguments

obs	data.frame of observations in TURBOVEG format, for example loaded with tv.obs
refl	Name of taxonomic reference list
db	a name of a Turboveg database directory containing tvabund.dbf, tvhabita.dbf and twin.set
concept	Character vector calling the desired taxonomical concepts dataframe(s), see details
syn	Treatment of synonyms, see details
ag	Treatment of childs and parents within the dataset, see details
rank	If ag='adapt', rank specifies the taxonomic rank to which taxa should be coarsened to. All higher taxa will be deleted.

mono	Should monotypic taxa be combined at subspecies = 'lower' or species level = 'higher'
monolist	Name of monotypic species list, must be in dbase format and in the same directory as the reference list, e.g. "monotypic-D" for the area of germany.
uncertain	List of length two, first the column name of uncertainty information, second a dataframe with uncertainty value and in column two one of 'delete', 'aggregate', 'preserve', see example.
maxtaxlevel	Maximum taxonomic levels to be used. See details.
check.critical	Check for critical names in your dataset and give warnings.'
sink	Write information about taxonomic harmonisation in a temporary file or print it on standrad output.
x	Dataframe of class 'veg'. See tv.veg
sel	Vector of species (column names) to be combined.
newname	Name of the combined taxon.
...	Other parameters passed to functions.

Details

Working with vegetation datasets, especially from different sources needs taxonomic valuation. The function tries to automate this process. Therefore the German taxonomic reference list (GermanSL, <http://geobot.botanik.uni-greifswald.de/reflist/>) contains additional taxon attributes (tax.dbf) and monotypic taxa of Germany (monotypic.dbf). Without an appropriate species list (see [tax](#)) the function will not work.

Before we replace synonyms we can choose to evaluate the data under a different taxonomical concepts than the original GermanSL. In the moment only an outline of such an approach is realised, because complete lists of differing taxonyms and there taxonomical re-evaluation have to be provided which are not available. With the inbuild test-dataset and the interpretation of the *Armeria maritima* complex you can have a shot on applying different concepts using a dataset referenced with GermanSL and using *Korneck1996.dbf*. See package vignette.

The three possible values for synonyms and child/parent taxa are: *preserve*: Leave everything untouched; *conflict*: Dissolve only in case of conflicts, e.g. if a subspecies occurs also at the species level within the same dataset. In this case the subspecies will be aggregated to the higher level. *adapt*: All respective taxa will be adapted, e.g. set to species level.

Monotypic taxa, e.g. a species which occur only with 1 subspecies in the survey area. They have to be combined, since otherwise two different (valid) taxa would denominate the same entity. If lower the higher taxon (e.g. species rank) is replaced by the lower level (subspecies rank). If neither lower nor higher monotypic species are preserved. Since the list of monotypic species strongly depends on the considered area you have to choose, which area is covered by your database and create an appropriate list of monotypic taxa. Within the package "monotypic-D.dbf" is provided as a compilation of monotypic species within the GermanSL list (see [tv.mono](#)).

Option *maxtaxlevel* determines the maximum taxonomic level within the given names, which should be used. All higher taxon observations are deleted. If you have a single field observation determined as *Asteraceae spec.* all your observations of taxa from that family will be aggregated to the family level, if you choose *ag=conflict*.

Value

Functions return the input dataframe of observations with harmonised taxon numbers.

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

References

Jansen, F. and Dengler, J. (2008) GermanSL - eine universelle taxonomische Referenzliste fl"ur Vegetationsdatenbanken. *Tuexenia*, 28, 239-253. Jansen, F. and Dengler, J. (2010) Plant names in vegetation databases - a neglected source of bias. *Journal of Vegetation Science*, 21, 1179-1186.

See Also

[tv.veg](#), [tv.obs](#)

Examples

```
## Not run:
# Turboveg installation needed
obs <- taxval(db='taxatest')
## For explanations see vignette('vegdata').

veg <- tv.veg('taxatest')
veg <- comb.species(veg, c('ARMEM-E', 'ARMEM-H'))

## End(Not run)
```

tv.biblio

Check bibliographic references from Turboveg codes

Description

Check bibliographic references from Turboveg codes

Usage

```
tv.biblio(x='all', site, quiet=FALSE, tv_home, ...)
```

Arguments

x	Turboveg reference code(s), e.g. "000001"
site	If you want to calculate the number of vegetation plots per reference, please indicate the header data, see tv.site
quiet	If you want to print the reference to the screen.
tv_home	Turbowin installation path. If not specified function tv.home tries to discover.
...	additional arguments

Value

Dataframe of (selected) biblioreferences (when assigned to an object).

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

See Also

[tv.site](#)

tv.compRef1

Compare different taxonomical reference lists.

Description

The function checks for different taxon numbers and, or taxon names in two TURBOVEG reference lists.

Usage

```
tv.compRef1(refl1, refl2, tv_home, check.nr=FALSE,
simplify = TRUE, verbose=FALSE, Sink=TRUE,
filter.1, filter.2, new = FALSE, file="compRef1.txt", ...)
```

Arguments

refl1	First reference list to compare.
refl2	Second reference list to compare.
tv_home	TURBOVEG installation path. If not specified, guessed by codetv.home
check.nr	Check equality of species numbers.
simplify	normalize taxon names with function taxname.simplify
verbose	Print species names on screen.
Sink	Write text file with differences.
filter.1	Character vector of filter keywords for refl1 to omit taxa from the comparison.
filter.2	Character vector of filter keywords for refl2 to omit taxa from the comparison.
new	Write new combined TURBOVEG reference list.
file	Name of the sink file.
...	Additional arguments.

Author(s)

Florian Jansen

References

Jansen, F. and Dengler, J. (2010) Plant names in vegetation databases - a neglected source of bias. *Journal of Vegetation Science*, 21, 1179-1186.

See Also

[tax](#)

 tv.coverperc

Cover code translation

Description

Translate cover code into percentage cover values for Turboveg database observations.

Usage

```
tv.coverperc(db, obs, RelScale, tv_home, tvscale, quiet = FALSE, ...)
```

Arguments

db	the name of the Turboveg database
obs	dataframe of observations, containing Cover Codes, coded in tvscale.dbf of Turboveg installation
RelScale	dataframe of CoverScale codes per releve, if empty it is read from the database
tv_home	Path to Turboveg installation.
tvscale	Cover scale.
quiet	Suppress messages.
...	Further options.

Value

obs	data.frame of observations with additional column COVER_PERC
-----	--

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

Examples

```
## For examples see in vignette('vegdata').
```

tv.metainfo	<i>Show metainfo of vegetation database or ecodbase</i>
-------------	---

Description

Showing "metainfo.txt" when specified and saved in Turboveg database directory. When db = 'eco' and refl specified, metainfo of species attribute table is displayed.

Usage

```
tv.metainfo(db, refl='GermanSL 1.2', tv_home, filename = 'metainfo.txt', ...)
```

Arguments

db	Turboveg database name
refl	Turboveg taxonomic reference list, declaration only necessary for ecodbase info
tv_home	Turboveg installation path
filename	Name of metainfo file residing in database directory
...	additional arguments

Details

Since Turboveg provides no formalised method to store information about database fields, I suggest to save a simple text file, named for example "metainfo.txt" into the directory of your Turboveg database.

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

tv.obs	<i>Dataframe of plot-species observations directly from Turboveg</i>
--------	--

Description

Dataframe of plot-species observations directly from Turboveg.

Usage

```
tv.obs(db, tv_home, ...)
```

Arguments

db	Name of your Turboveg database. This is the directory name containing tv-abund.dbf, tvhabita.dbf and tvwin.set. Please include pathnames below but not above Turbowin/Data.
tv_home	Turbowin installation path. If not specified function tv.home tries to discover.
...	additional arguments

Value

Data.frame of species occurrences in Turboveg format, that is every occurrence is a row with relev'e number, species number, layer, cover code and optional additional species-plot information.

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

See Also

[tv.veg](#)

Examples

```
## Not run:
# Turboveg installation needed
obs <- tv.obs('taxatest')
head(obs)

## End(Not run)
```

tv.site

Load site data from Turboveg Database

Description

Loading Turboveg header data and do basic data evaluation. Empty columns are eliminated and warnings about possibly wrong '0' values are performed

Usage

```
tv.site(db, tv_home, quiet=FALSE, common.only = TRUE,...)
```


Arguments

db	Name of your Turboveg database. Directory name containing tvabund.dbf, tvhabita.dbf and tvwin.set.
tv_home	Turbowin installation path. Optional, if Turbowin is either on "C:/turbowin" or "C:/Programme/Turbowin".
quiet	Suppress messages
common.only	Import only header data which occur in all databases with the same name.
...	Additional options like dec for type.convert

Details

Please specify pathnames below but not above Turbowin/Data. Can be a single database or a character vector of multiple databases. In the latter case you have to assure, that all databases use the same taxonomic reference list.

You can use the example in the final output line to make a summary statistic for attributes with potentially misleading '0' values. Just delete the '\' at beginning and end.

Value

data.frame of site variables.

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

tv.traits	<i>Load species traits from Turboveg reference list</i>
-----------	---

Description

Loading Turboveg ecodbase or any other specified dBase file in this directory and do basic data evaluation. Empty columns are eliminated. meanTraits will calculate mean trait values like mean Ellenberg indicator values (see vignette).

Usage

```
tv.traits(db, trait.db = 'ecodbase.dbf', refl, quiet = FALSE, ...)
meanTraits(trait, veg, refl, trait.db = 'ecodbase.dbf', join = 'LETTERCODE',
zero.is.NA = TRUE, ...)
```

Arguments

db	Path name to the Turboveg database directory
trait.db	Name of species trait DBase file, default is 'ecodbase'
refl	Name of the taxonomic reference list, if veg is not loaded with tv.veg
quiet	Hide messages
trait	Name of the column to use from the trait table
veg	vegetation matrix
join	Column name of the trait table corresponding to the colnames of the vegetation matrix
zero.is.NA	Should zero trait values handled as NA values
...	additional arguments for tv.traits

Details

You can use the final output line to make a summary statistic for attributes with potentially misleading '0' values. Just delete the \" at beginning and end.

Value

data.frame of ecological traits, see `metainfo(refl, eco=TRUE)`.

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

Examples

```
## Not run:
veg <- tv.veg('elbaue', cover.transform='pa')
# mEIV <- meanTraits('OEK_F', veg, 'ecodbase.dbf')
site <- tv.site('elbaue')
# plot(site$MGL, mEIV)

## End(Not run)
```

tv.veg

Tabulates vegetation tables from Turboveg database

Description

Tabulates vegetation tables from Turboveg resp. VegetWeb database, including taxonomic emendation and layer combination. Using various default parameters for the included functions. It is a wrapper for `tv.obs`, `taxval`, `tv.coverperc`.

Usage

```
tv.veg(db, tv_home, taxval=TRUE, convcode=TRUE, lc = c("layer", "mean", "max", "sum", "first"),
pseudo, values='COVER_PERC', spcnames=c('short', 'long', 'numbers'), dec = 0,
cover.transform = c('no', 'pa', 'sqrt'), obs, refl, RelScale, ...)
```

Arguments

db	Name of your Turboveg database. Directory name containing tvabund.dbf, tvhabita.dbf and tvwin.set. Please specify pathnames below (if you sorted your databases in subfolders) but not above Turbowin/Data.
tv_home	Turbowin installation path.
taxval	Should taxonomic valuation (see taxval) be performed?
convcode	Should cover code be converted to percentage values?
lc	Layer combination type. Possible values: layer (default), sum, mean or max, see details
pseudo	List used for layer combinations, see details
values	Name of the variable which should be used for the vegetations matrix.
spcnames	Should species numbers be replaced by shortletters or real names?
dec	Number of decimals for cover values in the resulting vegetation matrix.
cover.transform	If you want to transform the abundance values within your samples you can choose 'pa' for presence-absence or 'squareroot' for the dec rounded square root.
obs	Observations, optional
refl	Taxonomic reference list, optional
RelScale	Vector with Cover Scale code per Releve.
...	additional arguments for included functions

Details

layer means, the different layers are combined assuming their independence (a species occurring in two layers with a cover of 50% will result in an overall cover of 75%. sum will sum up cover values of all layers)

With option pseudo you can decide, which layers should be combined. Give a list with a combination data.frame (see [lc](#) and second the name of the column for combination. The default is `pseudo = list(lc.1, c('LAYER'))`, where `lc.1` is a data.frame `data(lc.1)`, which will combine all tree layers, all shrub layers and all layers below shrubs. An alternative would be `data(lc.all)`, combining all layers. With option `pseudo=NULL` there will be no layer aggregation.

Value

Function returns an object of class matrix with (combined) cover values.

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

See Also

[taxval](#), [tv.coverperc](#), [tv.mono](#), [tv.obs](#), [tv.site](#)

Examples

```
## Not run:
vignette("vegdata")
# If you have Turboveg installed on your computer try for a beginning
# tv.veg('databasename', tax=FALSE).
args(tv.veg)
help('taxval')

veg <- tv.veg('taxatest')
names(veg)
tv.veg('taxatest', uncertain=list('DET_CERT', data.frame(0:2,c('pres','agg','agg'))),
pseudo=list(1c.0,'LAYER'), genus = 'delete')

## End(Not run)
```

tv.write	<i>Write species-plot observations and site information to Turboveg database.</i>
----------	---

Description

Write species-plot observations and site information to Turboveg database.

Usage

```
tv.write(x, site, name, cover = c("code", "perc"), overwrite = FALSE, ...)
```

Arguments

x	Either observations data.frame with RELEVANCE_NR, TaxonUsageID and COVER_CODE columns or vegetation matrix of class "veg".
site	Header data for plots.
name	Name of the database.
cover	Use of covercodes or (mean) cover percentages, see Details.
overwrite	Logical. Should an existing database be overwritten.
...	Additional arguments.

Details

By default covercode are written to Turboveg. This is only meaningful, if correct coverscales are stored in the site dataframe. In case of e.g. VegetWeb data it is better to choose percentage cover (and Cover code 00) for Turboveg).

Value

A dataframe of species occurrences is written in dbase format into the Data directory of the specified Turboveg installation. Accordingly the header data information (tvhabita.dbf) is written to this directory and an empty file remarks.dbf is copied from the sample dataset.

Author(s)

Florian Jansen <jansen@uni-greifswald.de>

See Also

[tv.veg](#)

Index

- *Topic **datasets**
 - elbaue, 3
- *Topic **data**
 - lc, 5
- *Topic **misc, survey**
 - tv.biblio, 12
 - tv.obs, 15
 - tv.site, 16
 - tv.write, 20
- *Topic **misc,manip,survey**
 - ESveg, 4
 - tv.veg, 18
- *Topic **misc,manip**
 - taxval, 10
- *Topic **misc**
 - monotypic, 6
 - syntab, 7
 - tax, 8
 - tv.coverperc, 14
 - tv.metainfo, 15
 - tv.traits, 17
- *Topic **package**
 - vegdata-package, 2
- *Topic **utilities**
 - tv.compRefl, 13

- child(tax), 8
- childs(tax), 8
- comb.species(taxval), 10

- elbaue, 3
- ESveg, 4

- frequstab(syntab), 7

- grep, 9

- lc, 5, 19

- meanTraits(tv.traits), 17
- monotypic, 6

- parent(tax), 8
- parents(tax), 8
- print.syntab(syntab), 7

- syn(tax), 8
- syntab, 7

- tax, 8, 11, 14
- taxval, 2, 10, 19, 20
- tv.biblio, 12
- tv.compRefl, 13
- tv.coverperc, 14, 20
- tv.db(tv.veg), 18
- tv.eco(tv.traits), 17
- tv.home, 6, 12, 16
- tv.metainfo, 15
- tv.mono, 11, 20
- tv.mono(monotypic), 6
- tv.obs, 10, 12, 15, 20
- tv.site, 12, 13, 16, 20
- tv.taxval(taxval), 10
- tv.traits, 17
- tv.veg, 2, 5, 6, 11, 12, 16, 18, 21
- tv.write, 20

- vegdata(vegdata-package), 2
- vegdata-package, 2