Package ‘vegdata’

May 13, 2020

Version 0.9.8

Encoding UTF-8

Title Access Vegetation Databases and Treat Taxonomy

Author Florian Jansen <florian.jansen@uni-rostock.de>

Maintainer Florian Jansen <florian.jansen@uni-rostock.de>

Depends R (>= 2.0.0), foreign

Imports utils, xml2, plyr

Suggests vegan, labdsv, akima, googleVis, indicspecies, uuid, knitr, stringr, testthat

LazyData Yes

VignetteBuilder knitr

Description Handling of vegetation data from different sources (
Turboveg <http://www.synbiosys.alterra.nl/turboveg/>;
the German national repository <http://www.vegetweb.de> and others.
Taxonomic harmonization (given appropriate taxonomic lists,
e.g. the German taxonomic standard list ``GermanSL'', <http://germansl.infinitenature.org>).

License GPL (>= 2)

URL http://germansl.infinitenature.org

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-13 12:40:02 UTC

R topics documented:

`vegdata-package` *Functions to access data from vegetation databases and evaluate taxon names.*

Description

This package provides a set of functions to load data from vegetation databases (at present Turboveg and vegetweb.de). Taxa can be (semi-)automatically be checked and adapted depending the scientific question. For this a hierachical taxonomic reference list is needed.

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 <florian.jansen@uni-rostock.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; 1= sporadic use, 2= yearly with low intensity, 3= yearly

MGL mean groundwater level in 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)
```

`isc`	*Indicate site conditions with community weighted mean values of traits or with mode of gradient classes (sum of species amplitudes).*

Description

Calculates community weighted mean trait values, like mean Ellenberg indicator values. Alternatively (method = 'mode') environmental conditions can be calculated according to the concept of sums of amplitudes of species along ecological gradients.

Usage

```
isc(veg, refl, trait.db = 'ecodbase.dbf', ivname, keyname = 'LETTERCODE',
method = c('mean', 'mode'), weight, db, ...)
showindiplot(veg, trait.db, plotid, weight, keyname = 'LETTERCODE')
```

Arguments

`veg`	Vegetation matrix with plots in rows and species in columns
`refl`	Name of Turboveg taxonomic reference list
`trait.db`	data frame with species trait values
`ivname`	Name of the trait in trait.db to be used
`keyname`	Name of the column in trait dataframe to join with colnames of veg table
`method`	mean (weighted value of single traits, or mode (maximum) of trait classes)
`weight`	additional weight, e.g niche breath of species
`db`	name of Turboveg database
`plotid`	number or id of the plot to show
`...`	additional arguments

Details

Zero trait values will be handled as NA values.

Value

Vector with the ecological classification of sites. Either mean trait values or mode of gradient classes.

Author(s)

Florian Jansen <florian.jansen@uni-rostock.de>

Examples

```
## Not run:
db <- 'elbaue'
veg <- tv.veg(db, cover.transform='sqrt', check.critical = FALSE)
site <- tv.site(db, verbose = FALSE)
# Exclude plots with very high water level fluctuation
veg <- veg[site$SDGL < 60,]
veg <- veg[,colSums(veg) > 0]
site <- site[site$SDGL < 60,]
# Load species trait value database
traits <- tv.traits(db)

# Mean indicator values of Ellenberg F values
mEIV_F <- isc(veg, trait.db = traits, ivname = 'OEK_F', method = 'mean')
plot(site$MGL, mEIV_F, xlab = 'Mean groundwater level')

# Mode (most frequent level) of Ellenberg F values
library(reshape)
traitmat <- cast(traits, LETTERCODE ~ OEK_F)
traitmat <- traitmat[,-14]
ilevel <- isc(veg, trait.db = traitmat, ivname = as.character(1:11), method = 'mode')
boxplot(site$MGL ~ ordered(ilevel, levels = levels(ilevel)[c(2,4,3,5,6:10,1)]))

## End(Not run)
```

`lc` *Templates for pseudo-species according to Turboveg layer informations.*

Description

Differentiates species according to layer (tree, shrub, juvenile etc.) or other species-plot informations from Turboveg. The data.frames lc.0 and lc.1 are templates for layer aggregation/differentiation.

Details

Column `layer` point to the Turboveg layer specification (see Turboveg Help) and column `comb` defines the aggregation.

`lc.0` = Use every layer differentiation from 0 to 9 in Turboveg database as pseudo-species.

layer	comb
0	0
1	1
2	2
3	3
4	4
5	5
6	6

7	7
8	8
9	9

`lc.1` = Default layer combination in `tv.veg`. Differenciates tree and shrub layers, all other layers are combined.

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

`lc.all` = Do not use any layer differentiation.

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 <`florian.jansen@uni-rostock.de`>

See Also

tv.veg

monotypic *Creates a list of (regional) monotypic taxa from the given taxonomic checklist*

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 <florian.jansen@uni-rostock.de>

syntab *Syntaxonomic frequency tables*

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'), mupa, dec=0, refl, ...)
## S3 method for class 'syntab'
print(x, zero.print = ".", trait, limit = 1, minstat = 0, alpha = 0.05, ...)
```

Arguments

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

Author(s)

Florian Jansen <florian.jansen@uni-rostock.de>

See Also

Package indicspecies 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()
m <- match(rownames(st$syntab), traits$LETTERCODE, nomatch = 0)
trait <- traits[m, c("OEK_F","OEK_N")]
rownames(trait) <- traits$LETTERCODE[m]
st <- syntab(elbaue, clust, mupa=TRUE)
print(st, limit=30, trait=trait)
# Manipulation of the syntaxonomic table
sttable <- st$syntab
sttable <- sttable[sttable$p.value < 0.05 & !is.na(sttable$p.value), !names(sttable)
taxa <- tax(rownames(sttable))
rownames(sttable) <- taxa[match(rownames(sttable), taxa$LETTERCODE, nomatch = 0),'TaxonName']
write.csv(sttable, 'sttable.csv')

## End(Not run)
```

tax *Query of Turboveg 2 taxonomic reference lists including (if available) concept synonomy 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, detailed = FALSE, syn = TRUE, concept = NULL, strict = FALSE,
simplify = FALSE, quiet = FALSE, reflist.type = 'Turboveg', ...)
child(x, refl = tv.refl(), gen = 4, quiet = FALSE, syn = FALSE, ...)
parent(x, refl = tv.refl(), rank, quiet = FALSE, ...)
syn(x, refl = tv.refl(), quiet = FALSE, ...)
taxname.abbr(x, hybrid = c('remove', 'retain'),
species = FALSE, cf = FALSE, ...)
taxname.simplify(x, genus=TRUE, epithet=TRUE,  hybrid = c("remove", "retain"),
      rank = c("remove", "retain"), concept.status = c("remove", "retain"), ...)
parse.taxa(x, epis)
taxname.removeAuthors(x)
```

Arguments

`x`	Species number, lettercode or species name(s)
`refl`	Taxonomic reference list
`detailed`	In old Turboveg versions detailed taxonomic information could only be given in an extra file which was called tax.dbf in GermanSL. Compatibility mode.
`syn`	Return also synonym names
`concept`	Name of the file with an alternative taxon view stored in the reference list directory, see details.
`strict`	Exact match or partial matching with `grep`
`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.

`rank`	Taxonomical level of taxa to find
`hybrid`	remove hybrid markers for comparisons
`species`	use spec., sp., or species for genus level taxa
`cf`	remove 'in doubt' marker
`genus`	simplify genus name part
`epithet`	simplify epithet(s)
`concept.status`	remove s. str. s. l.
`epis`	vector of rank marker strings, if missing: 'subsp.', 'var.', 'v.'
`...`	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 huge 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*.

taxname.simplify: Before string comparison with reference list names it will eliminate diacritic marks, double consonants, "th", "y" versus "i(i)" and other frequent differences in writing style. If genus = TRUE (non stable) endings of genus names will be ignored, epithet = TRUE will eliminate endings for the epithet part.

taxname.abbr: standardisation of rank names

parse.taxa: parse genus and epitheta from name strings.

taxname.removeAuthors Remove name authors from full scientific name strings.

Author(s)

Florian Jansen <`florian.jansen@uni-rostock.de`>

References

Jansen, F. and Dengler, J. (2008) GermanSL - eine universelle taxonomische Referenzliste f\"ur Vegetationsdatenbanken. Tuexenia, 28, 239-253.

See Also

package vegdata

Examples

```
taxname.abbr('Achillea millefolium s. millefolium')
taxname.simplify('Vicia sylvatica') == taxname.simplify('Vicia silvaticum')


## Not run:
## GermanSL in Turboveg installation path needed
```

```
tax(27)
tax('Achillea millefolium')
tax('ACHIMILL')

## End(Not run)
## Not run:
child(0, gen=1)
child(27)

## End(Not run)
```

taxval *Handling of taxonomy in vegetation data.*

Description

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

Usage

```
taxval(obs, refl, db,
ag = c('conflict', 'adapt', 'preserve'),
rank, mono = c('species', 'higher', 'lower', 'preserve'), monolist = "monotypic-D",
maxtaxlevel = 'AGG', taxlevels, check.critical = TRUE, interactive = FALSE, ...)
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`

`ag` Treatment of children 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 in this taxonomic tree will be deleted, see maxtaxlevel.

`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.

`maxtaxlevel` Maximum taxonomic levels to be used. See details.

`taxlevels` dataframe of ordered taxonomic rank levels

`check.critical`	Check for critical names in your dataset and give warnings.'
`interactive`	Do you want to adapt the list of changes.
`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://germansl.infinitenature.org) 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.

Possible values for adapting the taxonomic hierarchy within the dataset (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`: Dissolve all nested taxa to e.g. species level for option ag. For this option also option `rank`, specifying the rank to which the taxa shall be adapted, must be given.

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.csv"` 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 obervations of taxa from that family will be aggregated to the family level, if you choose ag=conflict.

`Interactive` If you want to manually adapt the taxonomic harmonization `interactive=TRUE` will create a table with all original names and NewTaxonID's according to the chosen rules. The table will be saved as `taxvalDecisionTable.csv` in your actual working directory. You can manipulate the column NewTaxonID. If you run `taxval` again (e.g. through function `tv.veg`) and a file with this name exist in your working directory, it will be used.

Value

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

Author(s)

Florian Jansen <florian.jansen@uni-rostock.de>

References

Jansen, F. and Dengler, J. (2008) GermanSL - eine universelle taxonomische Referenzliste f\"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', db, dict = tv.dict(db), quiet=FALSE, tv_home, ...)
```

Arguments

`x`	Turboveg reference code(s), e.g. "000001"
`db`	Database name. Needed to select appropriate TV Dictionary folder.
`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.
`dict`	Name of Turboveg Dictionary (term lists for header data) if not the default one.
`...`	additional arguments

Value

Dataframe of (selected) bibliographic references (when assigned to an object).

Author(s)

Florian Jansen <florian.jansen@uni-rostock.de>

See Also

tv.site

---

tv.compRefl *Compare different taxonomical reference lists.*

Description

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

Usage

```
tv.compRefl(refl1, refl2, tv_home, check.nr=FALSE,
simplify = TRUE, verbose=FALSE, Sink=TRUE,
new = FALSE, file="compRefl.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.
new	Write new combined TURBOVEG reference list.
file	Name of the sink file.
...	Additional arguments.

Author(s)

Florian Jansen <florian.jansen@uni-rostock.de>

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 <florian.jansen@uni-rostock.de>

Examples

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

`tv.metadata` *Show metainfo of vegetation database or ecodbase*

Description

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

Usage

```
tv.metadata(db, refl, tv_home, filename = 'metadata.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 "metadata.txt" into the directory of your Turboveg database.

Author(s)

Florian Jansen <florian.jansen@uni-rostock.de>

`tv.obs` *Dataframe of plot-species observations directly from Turboveg*

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 tvabund.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 <florian.jansen@uni-rostock.de>

See Also

`tv.veg`

Examples

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

## End(Not run)
```

tv.readXML *Read Turboveg XML*

Description

Reads Turboveg XML formatted files species-plot observations and site information to Turboveg database.

Usage

```
tv.readXML(file)
```

Arguments

file	Path name of the Turboveg XML file

Value

S3 list with elements tvwin, tvadmin, site, and obs.

Author(s)

Florian Jansen <florian.jansen@uni-rostock.de>

See Also

tv.veg, tv.site

---

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, drop=TRUE, common.only = FALSE, verbose = TRUE,
replace.names, ...)
```

Arguments

`db`	Name of your Turboveg database(s). 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".
`drop`	Drop variables without values.
`common.only`	Import only header data with the same name in all databases.
`verbose`	print warnings and hints
`replace.names`	replace variable names. Useful if using multiple source databases. Data frame with names to be replaced in first and replacing names in second column.
`...`	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 <florian.jansen@uni-rostock.de>

---

tv.traits *Load species traits from Turboveg reference list*

Description

Loading Turboveg ecodbase or any other specified dBase file in this directory and do basic data evaluation. Empty columns are eliminated.

Usage

```
tv.traits(db, trait.db = 'ecodbase.dbf', refl, ...)
```

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
...	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 <florian.jansen@uni-rostock.de>

See Also

isc

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, taxval=TRUE,tv_home,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, site, 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 abundancce values within your samples you can choose 'pa' for presence-absence or 'sqrt' for the `dec` rounded square root.

`obs` Observations, optional

`site` plot header data, see `tv.site`

`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 there independence (a species occurring in two layers with a cover of 50% will result in a 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 <florian.jansen@uni-rostock.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(lc.0,'LAYER'), genus = 'delete')

## End(Not run)
```

---

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

Description

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

Usage

```
tv.write(x, site, name, tvadmin, remarks, dict = "", cover = c("code", "perc"),
drop = FALSE, obl = TRUE, overwrite = FALSE, ...)
```

Arguments

x	Either observations data.frame with RELEVE_NR, TaxonUsageID and COVER_CODE columns or vegetation matrix of class "veg".
site	Header data for plots.
name	Name of the new database.
tvadmin	Dataframe with plot UUID's and Turboveg columns from TvAdmin.dbf. A new file with new unique identifiers will be created if omitted.
remarks	Remarks in Turboveg format if the comments for individual plots exceed 254 characters. See remarks.dbf in Turboveg databases. An empty file will be created if omitted.
dict	Turboveg dictionary name
cover	Use of covercodes or (mean) cover percentages, see Details.
drop	Drop columns which are empty or contain only NA values.
obl	Add obligatory fields defined in the TV dictionary but not present in the site data table.
overwrite	Logical. Should an existing database be overwritten.
...	Additional arguments.

Details

By default Covercode is written to Turboveg. This is only meaningful, if correct CoverScales are given in the site dataframe. Unique plot ID's are stored in TvAdmin.dbf. If you want to preserve already given UUID's you have to prepare an adequate data.frame. Look into existing TvAdmin.dbf files for necessary columns.

Value

Five files will be created in tv_home/Data/<name> directory. tvabund.dbf with occurrence information n long format, tvhabita.dbf with plot information, remarks.dbf with comments longer then 255 characters, TvAdmin.dbf with plot UUID's and tvwin.dbf with information about taxonomic reference list, and dictionary used.

Author(s)

Florian Jansen <florian.jansen@uni-rostock.de>

See Also

tv.veg

Index