

# Vegetation data access and evaluation of taxon names

## Version 0.1.1

Florian Jansen

July 17, 2010

### Abstract

An example session to show functionality and usage of R library *vegdata*.  
After installation of package *vegdata* you can run this script with  
`> vignette("vegdata")`

The functions within the package expect an installation (more precisely the directory structure) of the vegetation database program Turboveg for Windows (see '<http://www.synbiosys.alterra.nl/turboveg/>' and Hennekens & Schaminée (2001)).

## 1 Preliminary notes

Turboveg uses dBase database format for storage. The package tries to deal with the many limitations of that format but it is essential, that you use "Database -> Reindex" in Turboveg every time you make an alteration in your Turboveg database and want to see these changed in R. Otherwise, when you delete a species occurrence in TV it will not be deleted immediately in the dBase file, instead it is only marked for deletion. That means it is still there when you access this file with R and will be deleted not until you reindex (*Database -> Reindex*) your Turboveg database.

## 2 Provided functionality

### 2.1 Taxonomic evaluation

One of the most important step in using vegetation data (from different sources) is to take care about the taxonomic content of the used taxon names. That is, to make sure, that exactly one (correct and valid) name defines one biological entity. Most researchers remember to convert synonyms to valid names, but in many cases the care about e.g. monotypic subspecies or ambiguous taxonomic levels, is lacking. The package offers the function `tv.taxval()` with options for the adjustment of formas, synonyms, monotypic taxa, subspecies, members of aggregates and undetermined genera.

### 2.2 Cover standardization

Turboveg provides different abundance codes and all kinds of user defined cover codes can easily be added. For vegetation analysis an unique platform is needed which will mostly be the percentage cover of the observed area, so for every abundance code class the mean cover percentage is defined in Turboveg. Since different scales can occur in a database and the storage format of the code table in Turboveg is somewhat strange, the function `tv.coverperc()` provides automatic conversion.

## 2.3 Layer aggregation

The main sample unit in vegetation science is a so called relevé. A Braun-Blanquet relevé is a sample of names and coverage (abundance) of species in a specified area (usually between 1 and 1000 sqm). It contains (is intended to contain) a complete list of phytoautotrophic plants (or a special guild) in that area. The information can be stored in a three-column list of relevé ID, Taxon ID and performance measure (e.g. cover code).

Often additional information about the kind of occurrence of the specific plants is wanted. In Turboveg one additional column for the most widespread attribute is default: growth height classes. E.g. in a forest it is of interest, if a woody species reaches full height (tree layer) or occurs only as a small individual (herb layer). Other attributes like micro location (hummock or depression, rock or dead wood), development stage (juvenile or not, age, flowering status etc.) or the month of survey in a multi-seasonal survey could be of interest and can be added in Turboveg. For analysis you may want to differentiate species growing in different layers. Function `tv.veg()` provides options for species-plot attribute handling.

## 2.4 Vegetation matrix

Turbogev stores relevés as a list of occurrences (s. above) but almost all functions and programs for vegetation analyses are using plot / species tables with a 0 value for non-occurrence (measured absence). Function `tv.veg()` inflates the Turboveg list to matrix format with plots in rows and species in columns. Column names can be either species numbers, species lettercodes or full names (with underscores instead of blanks).

# 3 Examples

Maybe the best way to introduce into the functionalities of the package is a session with example code.

## 3.1 Preparations

We load the library as usual into our R environment.

```
> library(vegdata)
```

The package includes some example datasets and the taxonomic reference list Germansl 1.1, which can be loaded either with option `sysPath=TRUE` or after copying it to the “Data” and “Species” directories of your Turbowin installation.

If you want to learn more about the taxonomic reference list for Germany, please look at Jansen & Dengler (2008) or `'http://geobot.botanik.uni-greifswald.de/portal/reflist'`.

The package functions try to guess as many necessary things as possible. For that the path of your Turboveg installation is searched on MSWindows operation systems in the order 'O:/Turbowin', 'C:/Turbowin', 'C:/Programs/Turbowin' and 'C:/Programme/Turbowin'. Turboveg runs also on Linux systems with Wine. There it is searched in `'~/wine/drive_c/Turbowin'`. In case of trouble, please specify option `tv_home` explicitly.

Using the default function settings you only have to specify the Turboveg database name. That is the name also occurring in Turbowin dialogues and can be found below the directory “Data”. If you use subdirectories you have to include those (e.g. `'testdata/taxatest'`). For general information about database structure see Turboveg Help).

```
> db1 <- "taxatest"
```

**Main functions** The package contains two main functions:

`tv.veg()` is a wrapper for a number of other functions to support the database load from Turboveg, taxonomic amendment, cover unification, layer combination and matrix generation.

`tv.site()` will load the site (header) data and does some basic corrections caused by Turboveg's database format.

Before you start to analyse a foreign dataset first check if there is a metainfo about the dataset available. Turboveg does not provide any metadata handling. So we recommend a simple text file named "metainfo.txt" and stored in the (Turboveg) database directory which can be loaded by:

```
> tv.metainfo(db1, sysPath = TRUE)
```

### 3.2 Site data

```
> site <- tv.site(db1, sysPath = TRUE)
```

The following columns contain no data and are omitted

```
[1] REFERENCE TABLE_NR NR_IN_TAB PROJECT AUTHOR SYNTAXON UTM
[8] ALTITUDE EXPOSITION MOSS_IDENT LICH_IDENT
```

The following numeric columns contain only 0 values and are omitted

```
[1] COV_TOTAL COV_TREES COV_SHRUBS COV_HERBS COV_MOSSES COV_LICHEN COV_ALGAE
[8] COV_LITTER COV_WATER COV_ROCK TREE_HIGH TREE_LOW SHRUB_HIGH SHRUB_LOW
[15] HERB_HIGH HERB_LOW HERB_MAX CRYPT_HIGH
```

The following numeric fields contain 0 values:

```
[1] INCLINATIO
```

Please check if these are really measured as 0 values or if they are not measured and wrongly assigned because of Dbase restrictions.

If so, use something like:

```
site$Column_name[site$Column_name==0] <- NA
summary(site[,c('INCLINATIO')])
```

The function `tv.site()` is quite straightforward. After loading the file `tvhabita.dbf` of the specified database, warnings are given for plots without specified relevé area or date and the fields are checked if they are empty (a lot of predefined header fields in Turboveg are often unused) or contain probably mistakable 0 values in numerical fields, due to DBase deficiencies (DBase can not handle NA = not available values reliably). It is stated in the output, if you have to check and possibly correct 0 values.

### 3.3 Vegetation data

Now we care about the species occurrence data.

Simple loading of species observation data from Turboveg is done by function `tv.obs()`

```
> obs <- tv.obs(db1, sysPath = TRUE)
```

```
reading observations ...
```

```
> head(obs)
```

	RELEVE_NR	SPECIES_NR	COVER_CODE	LAYER	DET_CERT	SEASON	MICROREL	FLOWER
1	2	27	2b	0	0	0	Schlenke	0
2	2	4685	4	1	0	0	Schlenke	0
3	2	4685	1	2	1	0	Schlenke	0
4	2	4685	1	6	0	0	<NA>	10
5	1	31	3	6	0	0	<NA>	0
6	1	20096	+	6	0	0	Schlenke	1

Data is stored in Turboveg as a flat table of occurrence values, that is one species-plot occurrence per row. Field *RELEVE\_NR* contains the plot number, *SPECIES\_NR* the taxon codes, *COVER\_CODE* the performance code and all other columns show species-plot attributes like growth height classes.

### 3.3.1 Names and entities

If you want to know the species name for a species number or letter code or vice versa you can use:

```
> tax("ACERNEG", sysPath = TRUE)
```

	SPECIES_NR	LETTERCODE	SHORTNAME	ABBREVIAT	NATIVENAME	AUTHOR	SYNONYM	VALID_NR
6	8	ACERNEG	Acer negundo	Acer negundo	Eschen-Ahorn	21	FALSE	8

```
> tax(27, tax = TRUE, sysPath = TRUE)
```

	SPECIES_NR	LETTERCODE	ABBREVIAT	SYNONYM	VALID_NR		
3529	27	<NA>	Achillea millefolium agg.	FALSE	27		
		VALID_NAME	GRUPPE	RANG	AGG	AGG_NAME	
3529	Achillea millefolium agg.	S	AGG	60728	Achillea spec.		
		NACHWEIS			SECUNDUM	HYBRID	
3529	BfN (Wisskirchen & Haeupler 1998)	BfN (Wisskirchen & Haeupler 1998)				0	
	BEGRUENDUN	IN_QUELLE_	AUTONYM	ELTER_1	ELTER_2	ELTER_3	EDITSTATUS
3529	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	BfN

As stated in the beginning the care about the taxonomic integrity of your database should stay be the beginning of your vegetation analyses. For Turboveg databases with taxonomic reference list GermanSL (versions 0.9, 1.0 or 1.1) this can be done semi-automatic.

To run the taxonomic adjustments of the example dataset use `tv.taxval()`

```
> obs <- tv.taxval(db1, obs, sysPath = TRUE)
```

Original number of taxa: 20

4 Synonyms found in dataset, adapted

	SPECIES_NR	ABBREVIAT	Freq_Member	VALID_NR
5230	27309	Armeria bottendorffensis	1	20585
7609	20096	Achillea millefolium subsp. collina	1	29
7763	25203	Abies alpestris	2	4269
9853	20583	Armeria maritima subsp. bottendorffensis	1	20585

  

	VALID_NAME	Freq_Agg
5230	Armeria maritima subsp. halleri	0
7609	Achillea collina	0
7763	Picea abies	0
9853	Armeria maritima subsp. halleri	0

1 Variants, forms, subspecies etc. found also at higher level in dataset, combined at higher level

	SPECIES_NR	ABBREVIAT	Freq_Member	AGG	AGG_NAME
1830	33	Achillea millefolium subsp. sudetica	1	31	Achillea millefolium

  

	Freq_Agg
1830	1

3 members of occurring aggregates in dataset, aggregated:

	SPECIES_NR	ABBREVIAT	Freq_Member	AGG	AGG_NAME
1828	31	Achillea millefolium	2	27	Achillea millefolium agg.
12032	2923	Hieracium pilosella	1	12273	Hieracium subg. Pilosella
12188	29	Achillea collina	1	27	Achillea millefolium agg.

  

	Freq_Agg
1828	1
12032	1
12188	1

1 Monotypic taxa found in dataset, species converted to lower rank.

	AGG_NR	AGG_NAME	AGG_RANG	MEMBER_NR	MEMB_NAME	MEMB_RANG
966	66142	Acoraceae spec.	FAM	61329	Acorus spec.	GAT

1 Monotypic taxa found in dataset, species converted to lower rank.

	AGG_NR	AGG_NAME	AGG_RANG	MEMBER_NR	MEMB_NAME	MEMB_RANG
851	61329	Acorus spec.	GAT	69	Acorus calamus	SPE

Undetermined genera and above preserved!

Number of taxa after validation: 13

Warning: Critical Pseudonym(s) in dataset, please check

	to_check	checknr	check against	SPECIES_NR	VALID_NR	SYNONYM
4876	Galium mollugo	2555	<NA>	27395	2549	TRUE

  

NACHWEIS

4876 BfN (Wisskirchen & Haeupler 1998)

Warning: Critical species in dataset, please check

	to_check	checknr	check against	SPECIES_NR	VALID_NR	SYNONYM
12097	Dactylis glomerata	1843	<NA>	26585	1842	TRUE
12052	Galium mollugo	2555	<NA>	26777	2548	TRUE

  

NACHWEIS

12097 BfN (Wisskirchen & Haeupler 1998)

12052 BfN (Wisskirchen & Haeupler 1998)

Have a look at `?tv.taxval` or `args(tv.taxval)` to change standard options.

Taxonomic evaluation of vegetation data sets can only be performed with checklists containing appropriate taxonomic information (see `tax.dbf` and `monotypic-D.dbf` for GermanSL 1.1, Jansen & Dengler (2008)).

If your database is not referenced with GermanSL you can not use `tv.taxval()` and you have to execute `tv.veg()` with option `tax=FALSE` or convert your database to GermanSL (Export to XML in Turboveg and re-import choosing the new GermanSL) assuming you have a central european database.

German SL is based upon the **taxon views** (Berendsohn (1995) of available checklists for Germany but contains more than 16,000 synonyms which can be used to switch between different taxon views.

To deal with a different taxonomic concept than the one used in GermanSL, you can use the option **concept**. For this a file is necessary indicating the new synonymy status, valid names and new aggregation. Within the package only a small example list (**korneck1996.dbf**) for the taxonomic view of *Armeria maritima* from (Korneck *et al.*, 1996) is implemented. Please compare the following examples.

```
> tv.taxval("taxatest", sysPath = TRUE)
> tv.taxval("taxatest", concept = "korneck1996", sysPath = TRUE)
```

### 3.3.2 Cover values

Cover is coded in Turboveg as an alphanumeric code. Different codes can be combined by using the mean cover percentage per cover class. Function `tv.coverperc()` will do this job according to the definitions in *Turboveg/Popup/tvscale.dbf*.

```
> obs <- tv.coverperc(db1, obs, sysPath = TRUE)
```

```
Cover code used: Braun/Blanquet (old)
code      r      +      1      2      3      4      5
perc      1      2      3      13     38     68     88

Cover code used: Braun/Blanquet (new)
code      r      +      1      2m     2a     2b     3      4      5
perc      1      2      3      4      8      18     38     68     88
```

```
> head(obs)
```

	RELEV_Nr	SPECIES_Nr	COVER_CODE	LAYER	DET_CERT	SEASON	MICROREL	FLOWER	COVERSCALE
1	2	27	2b	0	0	0	Schlenke	0	02
2	2	4685	4	1	0	0	Schlenke	0	02
3	2	4685	1	2	1	0	Schlenke	0	02
4	2	4685	1	6	0	0	<NA>	10	02
5	1	27	3	6	0	0	<NA>	0	01
6	1	27	+	6	0	0	Schlenke	1	01
	COVER_PERC								
1	18								
2	68								
3	3								
4	3								
5	38								
6	2								

### 3.3.3 Pseudo-species, layer combinations and vegetation matrix

`tv.veg()` is a wrapper for the above mentioned functions and produces a vegetation matrix with relevés as rows and species as columns. Additionally care about species-plot attribute differentiation and combination, the inflation of a vegetation matrix and the handling of species codes is provided.

If we have more than one occurrence of the same species in a plot, e.g. because tree species growing as young stands and adult specimens were differentiated according to growth height classes we have to create either pseudo-species which differentiate the occurrences in the resulting vegetation matrix or to combine species occurrences from different layers. For the latter we can use different calculations i.e. mean, max, min or first value. If we assume an independent occurrence of a species in different layers, a tree with a cover of 50% in tree layer and 50% in herb layer can be accounted with an overall cover of 75%. This is done with option `lc = 'layer'`, the default.

If you want to differentiate species according to layer or other species-plot attributes you can specify which attributes should be used for differentiation, and how pseudo-species should be labelled (e.g. `speciesname.layercode`). Two example data frames for layer differentiation are included in the package. `lc.0` uses all Turboveg layers (0 to 9) for pseudo-species differentiation. `lc.1` combines tree layers and shrub layers to a maximum of three pseudo-species per taxon.

```
> lc.1
```

	LAYER	COMB
1	0	0
2	1	Tree
3	2	Tree
4	3	Tree
5	4	Shrub
6	5	Shrub
7	6	0
8	7	0
9	8	0
10	9	0

```
> veg <- tv.veg(db1, lc = "sum", comb = list(lc.1, c("LAYER")), dec = 1,
+   quiet = TRUE, sysPath = TRUE)
```

```
> veg[, 1:7]
```

	ACERPSE	ACERPSE.Shrub	ACHI#MI	ACHISPE	ACOUCL	ADONAES	AGRTS;P
1	3	13	43	3	0	3	3
2	0	0	18	0	0	0	0
3	0	0	0	0	3	0	0

If you want to use only presence/absence information in your analyses you can do:

```
> veg[veg > 0] <- 1
```

### 3.4 Additional functions

`frequstab()` produces a relative or absolute frequency table of a vegetation table classification with the possibility to filter according to threshold values. To exemplify the function we use the second dataset implemented the package. It is the demonstration dataset from (Leyer & Wesche, 2007), a selection of grassland relevés from the floodplains of the river Elbe.

```
> db2 <- "elbaue"
> veg <- tv.veg(db2, quiet = TRUE, sysPath = TRUE)
> site <- tv.site(db2, sysPath = TRUE)
```

```

> tv.metainfo(db2)

> cluster <- rep(NA, nrow(site))
> cluster[site$MGL < -100] <- "dry"
> cluster[site$MGL < -25 & site$MGL >= -100] <- "wet"
> cluster[site$MGL >= -25] <- "very.wet"

> freqtab(veg, cluster, limit = 30, sort = c("dry", "wet", "very.wet"),
+       relfr = TRUE, sysPath = TRUE)

```

```

Number of clusters: 3

dry wet very.wet
Achillea millefolium      38  0      0
Agrostis capillaris      54 14      0
Agrostis stolonifera     15 29     33
Alopecurus geniculatus    0 21     50
Alopecurus pratensis     77 64      0
Cardamine pratensis       8 64      0
Carex acuta               0 71     33
Carex praecox            69  7      0
Carex vesicaria           0 43      0
Carex vulpina            15 36      0
Elymus repens            77 21      0
Erophila verna           31  0      0
Galium palustre          15 71     50
Galium verum agg.        46  7      0
Glyceria fluitans         0 14     33
Glyceria maxima          0 29     83
Holcus lanatus            8 43      0
Juncus effusus            0 50      0
Phalaris arundinacea     15 79     50
Poa palustris            38 64      0
Poa pratensis agg.      62 43      0
Poa trivialis s. trivialis 15 57      0
Potentilla anserina       0 36      0
Ranunculus flammula       0 43      0
Ranunculus repens        31 93     17
Rorippa amphibia         0  7     50
Rumex thyrsiflorus       69  0      0
Silene flos-cuculi        0 36      0
Sium latifolium          0 43     17
Stellaria palustris       8 64      0
Taraxacum Sec. Alpina, Hamata et Ruderalia 62 29      0
Trifolium repens          0 43      0
Vicia tetrasperma        31  7      0

```

Use `help(package='vegdata')` for a complete list of available functions and data sets in `vegdata`.

At <http://geobot.botanik.uni-greifswald.de/download/> a development version of the package `vegdata` is available with additional functionalities (but less stability).

The package `vegdata` serves only as a helper for further analysis of vegetation data which can already be done by powerful R packages like `vegan`. But with the functions shown above we are now ready to execute all kinds of analyses in the wide area of vegetation analyses.



### 3.5 Vegetation analyses

For instance we can do a “Nonmetric Multidimensional Scaling with Stable Solution from Random Starts Axis Scaling and Species Scores” which is a wrapper for Kruskal’s Non-metric Multidimensional Scaling (Cox & Cox, 1994, 2001) from Jari Oksanen (Oksanen *et al.*, 2008).

```
> library(vegan)
> veg.nmds <- metaMDS(veg, distance = "bray", trymax = 5, autotransform = FALSE,
+   noshare = 1, expand = TRUE, trace = 2)
> plot(veg.nmds)
```

To show the result in comparison with environmental measurements in a nice graphic we do some magic.

```
> library(labdsv)
> library(akima)
> color = function(x) rev(topo.colors(x))
> nmds.plot <- function(nmds, site, var1, var2, disp, ...) {
+   lplot <- nrow(nmds$points)
+   lspc <- nrow(nmds$species)
+   stems <- colSums(veg)
+   filled.contour(interp(nmds$points[, 1], nmds$points[, 2], site[,
+     var1]), ylim = c(-1, 1.1), xlim = c(-1.4, 1.4), color.palette = color,
+     xlab = var1, ylab = var2, main = paste("NMDS of vegetation"),
+     key.title = title(main = var1, cex.main = 0.8, line = 1, xpd = NA),
+     plot.axes = {
+       axis(1)
+       axis(2)
+       points(veg.nmds$points[, 1], veg.nmds$points[, 2], xlab = "",
+         ylab = "", cex = 0.5, col = 2, pch = "+")
+       points(veg.nmds$species[, 1], veg.nmds$species[, 2], xlab = "",
+         ylab = "", cex = 0.2, pch = 19)
+       ordisurf(veg.nmds, site[, var2], col = "black", choices = c(1,
+         2), add = TRUE)
+       orditorp(veg.nmds, display = disp, pch = " ")
+       legend("topright", paste("GAM of ", var2), col = "black",
+         lty = 1)
+     }, ...)
+ }

> print(nmds.plot(veg.nmds, site, disp = "species", var1 = "MGL", var2 = "SDGL"))
```



- for vegetation datasoftware package for input, processing, and presentation of phytosociological data. *Journal of Vegetation Science*, 12, 589–591.
- Jansen, F. & Dengler, J. (2008). Germansl - eine universelle taxonomische referenzliste für vegetationsdatenbanken. *Tuexenia*, 28, 239–253.
- Korneck, D., Schnittler, M. & Vollmer, I. (1996). Rote Liste der Farn- und Blütenpflanzen (Pteridophyta et Spermatophyta) Deutschlands. *Schriftenreihe für Vegetationskunde*, 28, 21–187.
- Leyer, I. & Wesche, K. (2007). *Multivariate Statistik in der Ökologie*. Springer, Berlin.
- Oksanen, J., Kindt, R., Legendre, P., O'Hara, B., Simpson, G.L. & Stevens, M.H.H. (2008). *vegan: Community Ecology Package*.