

Vegetation data access and taxonomic harmonization

Version 0.4

Florian Jansen

October 5, 2011

Abstract

An example session to show functionality and usage of R library **vegdata**.
After installation of **vegdata** you can invoke this PDF with

```
> vignette("vegdata")
```

1 Preliminary notes

Some **vegdata** functions expect an installation, or more precisely the main directory structure, of the vegetation database program Turboveg for Windows (see '<http://www.synbiosys.alterra.nl/turboveg/>' and Hennekens & Schaminée (2001)). If the package can not find a Turboveg installation it will use the directory within the package installation path. If you want to use function **taxval** for taxonomic harmonization you will need to have GermanSL or an equally structured reference list. If you do not specify any, the most recent version of GermanSL will be used and if it can not be found within the specified path, it will be downloaded from <http://geobot.botanik.uni-greifswald.de/reflist>.

Turboveg uses dBase database format for storage. The package tries to deal with the limitations of that format but it is essential, that you use "Database -> Reindex" in Turboveg every time you delete something in your Turboveg database. Otherwise it will not be deleted immediately in the dBase file, instead it is only marked for deletion, i.e. it is still there when you access this file with R and will not be recognized as deleted until you reindex your Turboveg database.

2 Provided functionality

2.1 Database access

At the moment **vegdata** provides direct access to two different vegetation database formats:

Turboveg is a desktop program, written in VisualBasic. It provides basic functions to enter, import, maintain and export vegetation data. From the 2 000 000 vegetation plot registered in <http://www.GIVD.info> approximately 1.5 million are stored in Turboveg databases format.

VegetWeb is the German national vegetation database. VegetWeb is developed as a MySQL-Server database at the Federal Agency for Nature Conservation (BfN) and can be used via a PHP framework at <http://www.floraweb.de/vegetation/vegetweb/RechercheView.php>.

2.2 Taxonomic harmonisation

One of the most important steps in using vegetation data (from different sources) for statistical analysis is to take care about the taxonomic content of the names existing in the database. 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 (Jansen & Dengler, 2010). The package offers the function `taxval` with different options for the adjustment of synonyms, monotypic taxa, taxonomic levels, members of aggregates and undetermined species.

2.3 Cover standardization

Turboveg provides different abundance codes and all kinds of user defined cover codes can easily be added. For vegetation analysis a unique species performance platform is needed which will in most cases be the percentage cover of the observed plot area. Therefore, 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 for convenience.

2.4 Layer aggregation

The most frequently used sample unit in vegetation science is a plot based vegetation relevé (Dengler *et al.*, 2011). A Braun-Blanquet relevé is a sample of names and coverage (abundance) of species in a specified area (usually between 1 and 1000 m^2) at a specific time. It contains (at least is intended to contain) a *complete* list of photo-autotrophic plants (or a defined subset) in that plot. This 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 is wanted. In Turboveg one additional column for the most widespread attribute is included by 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, flowering status etc.) or the month of survey in a multi-seasonal survey could also be of interest and can be added in Turboveg. For analysis you may want to differentiate species with different species-plot attributes (e.g. growing in different layers). Function `tv.veg` provides possibilities for species-plot attribute handling.

2.5 Vegetation matrix

Turboveg stores relevés as a dataframe of occurrences (s. below) but almost all functions and programs for vegetation analyses use plot-species cross-tables with a 0 value for non-occurrence = observed 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 letter-codes (default) or full names (with underscores instead of blanks to match the).

3 Preparations

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

We load the library as usual into our R environment.

```
> library(vegdata)
```

Many functions use the directory structure of Turbowin. The first time such a function is called, function `tv.home` tries to find your Turboveg installation path. Depending on whether you have Turboveg installed on your computer or not, it will give you a message about the Turboveg installation path or the path to the Turboveg directory structure of package vegdata.

```
> tv.home()
```

```
#####
Turboveg root directory is set to /home/florian/.wine/drive_c/Turbowin
If you want to change this use: options(tv_home="<path_to_your_Turbowin_root>")
#####
```

If you want to change this use:

```
> options(tv_home = "path_to_your_Turboveg_root_directory")
```

4 Service functions

```
> tv.db()
```

```
[1] "" "elbaue" "taxatest"
```

will give you a list of possible database names (directories within the Turboveg Data directory).

```
> tv.refl()
```

```
[1] "GermanSL 1.2"
```

GermanSL 1.2 is the default Taxonomic reference list in the actual vegdata package. Whenever you use a Turboveg database name in a function, the Reference list will be read from the database configuration file "tvwin.set", if possible.

Package vegdata contains several service functions to query the taxonomic information contained in the reference list.

```
> tax("Achillea millefolium")
```

Taxonomic evaluation list (species.dbf) of version GermanSL 1.2 not available.
I will try to download the reference now.

	SPECIES_NR	LETTERCODE		ABBREVIAT	
18	27	ACHI#MI		Achillea millefolium agg.	
20	31	ACHIMIL		Achillea millefolium	
21	32	ACHIM-M	Achillea millefolium subsp.	millefolium	
22	33	ACHIM-S	Achillea millefolium subsp.	sudetica	
8678	20096	ACHICOL	Achillea millefolium subsp.	collina	
8679	20097	ACHIPAN	Achillea millefolium subsp.	pannonica	
8680	20098	ACHIPAN	Achillea millefolium var.	lanata	
13219	26082	ACHIMIL	Achillea millefolium var.	firma	
26247	90019	ACHI*AB	Achillea millefolium agg. x	nobilis	
26248	90020	ACHIM*P	Achillea millefolium x	pannonica	
			NATIVE NAME	SYNONYM	VALID_NR
18		Artengruppe	Wiesen-Schafgarbe	FALSE	27
20		Gewöhnliche	Wiesen-Schafgarbe	FALSE	31
21		Gewöhnliche	Wiesen-Schafgarbe i.e.S.	FALSE	32
22		Sudeten-	Wiesenschafgarbe	FALSE	33
8678			<NA>	TRUE	29
8679			<NA>	TRUE	34
8680			<NA>	TRUE	34
13219			<NA>	TRUE	31
26247			<NA>	TRUE	90028
26248			<NA>	FALSE	90020

"GermanSL 1.2" is not included in vegdata to keep the R package small. Instead the reference list will be automatically downloaded into the tv_home directory, if it is not installed but needed. If you want to use a different list, specify refl=<Name of your list>. Function tax will use the given character string to look for all (partially) matching species names within the reference list

```
> tax("Achillea millefolium", strict = TRUE, verbose = TRUE)
```

SPECIES_NR	LETTERCODE	ABBREVIAT	AUTHOR	SYNONYM	VALID_NR	VALID_NAME
20	31	ACHIMIL	Achillea millefolium	L.	FALSE	31 Achillea millefolium
		NATIVENAME	GRUPPE	RANG	AGG	AGG_NAME
20	Gew\u0094hnliche	Wiesen-Schafgarbe	S	SPE	27	Achillea millefolium agg.
		NACHWEIS				SECUNDUM HYBRID BEGRUEND EDITSTATUS
20	BfN(Wisskirchen u. Haeupler 1998)	BfN(Wisskirchen u. Haeupler 1998)	<NA>	<NA>		BfN

Additional to the Turboveg standard fields comprehensive information for every taxon is stored in an extra file (tax.dbf). If you set option strict=TRUE, only the species with exact match to the given character string will be returned.

```
> tax("Elytrigia repens")$ABBREVIAT
```

```
[1] "Elytrigia repens subsp. arenosa" "Elytrigia repens"
[3] "Elytrigia repens var. caesia"   "Elytrigia repens var. littoralis"
[5] "Elytrigia repens var. repens"
```

```
> syn("Elytrigia repens")
```

Name swarm of Elytrigia repens :

	SPECIES_NR	ABBREVIAT	SYNONYM	SECUNDUM
4076	6541	Agropyron repens subsp. caesium	TRUE BfN(Wisskirchen u. Haeupler 1998)	
4079	6544	Elymus repens subsp. repens s. l.	TRUE BfN(Wisskirchen u. Haeupler 1998)	
4789	10260	Elymus repens subsp. caesium	TRUE BfN(Wisskirchen u. Haeupler 1998)	
8712	20143	Agropyron caesium	TRUE BfN(Wisskirchen u. Haeupler 1998)	
8730	20167	Agropyron repens subsp. repens	TRUE BfN(Wisskirchen u. Haeupler 1998)	
9888	21639	Elytrigia repens	TRUE BfN(Wisskirchen u. Haeupler 1998)	
12063	24393	Triticum repens	TRUE BfN(Wisskirchen u. Haeupler 1998)	
13913	27778	Elymus repens	FALSE BfN(Wisskirchen u. Haeupler 1998)	
14005	27914	Agropyron repens	TRUE BfN(Wisskirchen u. Haeupler 1998)	
		EDITSTATUS		
4076		BfN		
4079		Korrektur		
4789		BfN		
8712		BfN		
8730		BfN		
9888		BfN		
12063		BfN		
13913		BfN		
14005		BfN		

```
> childs(27, quiet = TRUE)$ABBREVIAT
```

```
[1] "Achillea collina"           "Achillea millefolium"
[3] "Achillea pannonica"        "Achillea roseoalba"
[5] "Achillea setacea"          "Achillea pratensis"
[7] "Achillea lanulosa"          "Achillea collina x millefolium"
[9] "Achillea collina x pannonica" "Achillea collina x pratensis"
[11] "Achillea collina x roseoalba" "Achillea collina x setacea"
[13] "Achillea millefolium x pannonica" "Achillea pratensis x roseoalba"
[15] "Achillea millefolium subsp. millefolium" "Achillea millefolium subsp. sudetica"
```

```
> parents("ACHIMIL")
```

Parents of *Achillea millefolium* (31):

	SPECIES_NR	ABBREVIAT	SECUNDUM RANG	GENERATION
18	27	<i>Achillea millefolium</i> agg.	BfN(Wisskirchen u. Haeupler 1998)	AGG 1
20641	60728	<i>Achillea</i> species	BfN(Wisskirchen u. Haeupler 1998)	GAT 2
20471	60463	Asteraceae species	Wisskirchen u. Haeupler 1998	FAM 3
20445	60415	Asterales species	Wisskirchen u. Haeupler 1998	ORD 4
20324	60079	Asteridae species	Wisskirchen u. Haeupler 1998	UKL 5
20318	60071	Magnoliopsida species	Wisskirchen u. Haeupler 1998	KLA 6
20309	60049	Magnoliophytina species	Wisskirchen u. Haeupler 1998	UAB 7
20283	60000	Spermatophyta species	Wisskirchen u. Haeupler 1998	ABT 8
29376	94419	"Gef\u0084ápfanze" species	[ad-hoc-Taxon fÃ¼r GermanSL]	AG2 9
10	0	"Gr\u0081nliches etwas"	[ad-hoc-Taxon fÃ¼r GermanSL]	ROOT 10
	EDITSTATUS			
18	BfN			
20641	BfN			
20471	Korrektur			
20445	BfN			
20324	BfN			
20318	Korrektur			
20309	Korrektur			
20283	Korrektur			
29376	Ergaenzung			
10	Ergaenzung			

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

5 Taxonomic harmonisation

Care about the taxonomic content of the datasets is crucial for every analysis. Some of these steps can be automated with an appropriate taxonomic reference. For background and details see (Jansen & Dengler, 2010).

```
> db <- "taxatest"
```

Defines the vegetation database name according to the name of the Turboveg database directory name

```
> tv.metainfo(db)
```

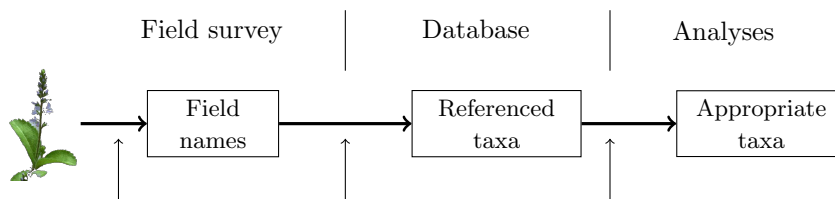
Metainformation, i.e. information about the kind of available information should always be given for every database. Since Turboveg does not ask and provide such information, write a simple text file called metainfo.txt and save it within the database folder. Turboveg does not provide any metadata handling. Database **taxatest** is an artificial dataset to show functionalities and necessary steps for taxonomic harmonization.

Let's have a look at the Turboveg data structure.

```
> obs.tax <- tv.obs(db)
```

```
reading observations ...
```

```
> species <- tax("all", syn = TRUE)
> obs.tax$Name <- species$ABBREVIAT[match(obs.tax$SPECIES_NR, species$SPECIES_NR)]
> head(obs.tax[, 1:4])
```



1. Field interpretation

- document your source(s) of taxonomic interpretation (Flora)
- specify determination certainty
- collect herbarium specimen

2. Database entry

- document field records / original literature
- reference as conservative as possible to a taxonomic reference list with all relevant taxa (synonyms, field aggregates, horticultural plants, ...)
- document your interpretations

3. Preparation for analyses

- convert synonyms
- summarize monotypic taxa
- clean up nested taxa
- clean up taxonomic ranks
- ...

Three steps of taxonomic interpretation

- need of appropriate tools (software, reference lists)
- standards
- threefold attention

Figure 1: Steps of taxonomic interpretation

RELEVE_NR	SPECIES_NR	COVER_CODE	LAYER	
1	2	27	2b	0
2	2	4685	4	1
3	2	4685	1	2
4	2	4685	1	6
5	1	31	3	6
6	1	20096	+	6

This condensed format shows only presences of species observations. Every species observation is stored in one row and the membership to a specific vegetation plot is given in column **RELEVE_NR**.

5.1 Explaining function `taxval`

We are using the taxonomic reference list GermanSL (see Jansen & Dengler 2008, Tuexenia) which contains not only information about synonymy of species names, but also about the taxonomic hierarchy. This enables several semi-automatic enhancements of the taxonomic information stored in your vegetation database. If

your database is not referenced to GermanSL (and can not be converted), you have to dismiss function `taxval` (option `tax=FALSE` in `tv. veg`) and do the taxonomic harmonization by hand (function `comb.species`).

```
> obs.taxval <- taxval(obs.tax, mono = "lower")
```

Original number of names: 25

5 Synonyms found in dataset, adapted

SPECIES_NR	ABBREVIAT	Freq.1	VALID_NR	VALID_NAME
20010	Cardamine pratensis subsp. pratensis	1	15133	Cardamine pratensis
20096	Achillea millefolium subsp. collina	1	29	Achillea collina
20583	Armeria maritima subsp. bottendorffensis	1	20585	Armeria maritima subsp. halleri
25203	Abies alpestris	2	4269	Picea abies
27309	Armeria bottendorffensis	1	20585	Armeria maritima subsp. halleri

Freq.2

0

0

0

0

0

1 monotypic taxa found in dataset, set to lower rank.

AGG_NR	AGG_NAME	AGG_RANG	MEMBER_NR	MEMB_NAME	MEMB_RANG
66142	Acoraceae species	FAM	61329	Acorus species	GAT

1 monotypic taxa found in dataset, set to lower rank.

AGG_NR	AGG_NAME	AGG_RANG	MEMBER_NR	MEMB_NAME	MEMB_RANG
61329	Acorus species	GAT	69	Acorus calamus	SPE

No taxa higher than ROOT found.

8 child taxa found in dataset, adapted

SPECIES_NR	ABBREVIAT	Freq.1	AGG	AGG_NAME	Freq.2
29	Achillea collina	1	27	Achillea millefolium agg.	1
31	Achillea millefolium	1	27	Achillea millefolium agg.	1
33	Achillea millefolium subsp. sudetica	1	31	Achillea millefolium	1
27	Achillea millefolium agg.	1	60728	Achillea species	1
2923	Hieracium pilosella	1	12273	Hieracium subg. Pilosella	1
15133	Cardamine pratensis	2	1105	Cardamine pratensis agg.	1
20945	Cardamine dentata	1	1105	Cardamine pratensis agg.	1
1105	Cardamine pratensis agg.	1	60845	Cardamine species	1

3 child taxa found in dataset, adapted

SPECIES_NR	ABBREVIAT	Freq.1	AGG	AGG_NAME	Freq.2
27	Achillea millefolium agg.	1	60728	Achillea species	1
31	Achillea millefolium	1	27	Achillea millefolium agg.	1
1105	Cardamine pratensis agg.	1	60845	Cardamine species	1

1 child taxa found in dataset, adapted

SPECIES_NR	ABBREVIAT	Freq.1	AGG	AGG_NAME	Freq.2
27	Achillea millefolium agg.	1	60728	Achillea species	1

Number of taxa after validation: 13

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

to_check	check_No	check against	SPECIES_NR	SECUNDUM
Galium mollugo	2555	Galium mollugo auct.	27395	BfN(Wisskirchen u. Haeupler 1998)

Warning: Critical species in dataset, please check

	to_check	check_No	check against	SPECIES_NR	SECUNDUM
Dactylis glomerata	1843	Dactylis glomerata s. l.	26585	BfN(Wisskirchen u. Haeupler 1998)	
Galium mollugo	2555	Galium mollugo s. l.	26777	BfN(Wisskirchen u. Haeupler 1998)	

The database contains 25 different names in the beginning.

Synonyms 5 of the species names are synonyms and are therefore transferred to legal taxon names, respectively numbers (see option `syn='adapt'`). If you want to preserve synonyms, choose option `syn='conflict'` or `'preserve'`.

Monotypic species within the area Monotypic taxa are valid taxa which are the only child of their next higher taxonomic rank within the survey area. By default they will be converted by `taxval` to the higher rank. For instance *Poa trivialis* is in Germany only represented by *Poa trivialis subsp. trivialis*. Both taxa are valid, but for most analysis only one name for these identical entities must be used. By default a list of monotypic taxa within the GermanSL (whole Germany) is considered (see `tv.mono('GermanSL 1.2')`). The default is to set all monotypic species to the higher rank (because many monotypic subspecies can occur in vegetation databases).

If necessary, the procedure has to be repeated through the taxonomic

Trimming the hierarchy If your database contains the taxon *Asteraceae spec.*, the `taxval` code to follow would aggregate occurrences of all your Asteraceae to the family level. To prevent this you can delete all observations above a certain taxonomic level. The default is not to trim the hierarchy (`ROOT = "Greenish Something"` is the toplevel).

Solving the nestedness If your database contains *Achillea millefolium* but also *Achillea millefolium agg.* for most analysis it will be necessary to coarsen the first (option `ag='conflict'`) because *A. millefolium agg.* will probably include further occurrences of *Achillea millefolium*.

The procedure has to be repeated until all occurring taxonomical levels are considered.

Especially with aggregates and their members the coarsening to the higher level can be a sad fate. If you have 100 occurrences of *Achillea millefolium* but a single one with *A. mill. agg.* you might want to clean your observational dataframe beforehand or do the aggregation afterwards manually with `tv.veg(db, ag='preserve')` and a manual correction with function `comb.species` (see below).

I confess that it is a strange and complete artificial example. Starting with 25 names in the beginning only 13 taxa survived the valuation. All others had to be converted.

```
> obs.taxval$Taxon <- species$ABBREVIAT[match(obs.taxval$SPECIES_NR, species$SPECIES_NR)]
> obs.taxval[order(obs.taxval$Name), c("Name", "Taxon")]
```

	Name	Taxon
10	Abies alpestris	Picea abies
15	Abies alpestris	Picea abies
8	Acer pseudoplatanus	Acer pseudoplatanus
9	Acer pseudoplatanus	Acer pseudoplatanus
5	Achillea millefolium	Achillea species
1	Achillea millefolium agg.	Achillea species
6	Achillea millefolium subsp. collina	Achillea species
11	Achillea millefolium subsp. sudetica	Achillea species
7	Achillea species	Achillea species
16	Acoraceae species	Acorus calamus
19	Adonis aestivalis	Adonis aestivalis
20	Agrostis stolonifera var. palustris	Agrostis stolonifera var. palustris
22	Armeria bottendorffensis	Armeria maritima subsp. halleri
12	Armeria maritima subsp. bottendorffensis	Armeria maritima subsp. halleri
13	Armeria maritima subsp. elongata	Armeria maritima subsp. elongata

14	Armeria maritima subsp. halleri	Armeria maritima subsp. halleri
25	Cardamine dentata	Cardamine species
27	Cardamine pratensis	Cardamine species
26	Cardamine pratensis agg.	Cardamine species
28	Cardamine pratensis subsp. pratensis	Cardamine species
29	Cardamine species	Cardamine species
18	Dactylis glomerata	Dactylis glomerata
17	Galium mollugo	Galium mollugo
21	Hieracium pilosella	Hieracium subg. Pilosella
23	Hieracium subg. Pilosella	Hieracium subg. Pilosella
24	Picea abies	Picea abies
2	Quercus robur	Quercus robur
3	Quercus robur	Quercus robur
4	Quercus robur	Quercus robur

Critical Pseudonyms Taxon misapplication is maybe the greatest danger in using survey data. Known misapplications of names (.auct) are embedded in the GermanSL. Please pay attention, if these might also be relevant for your dataset.

Completely independent from the questions of correct taxonomic naming of a specific specimen, the boundary of a taxon interpretation can differ much Jansen & Dengler (see 2010). This should be adequately solved during data entry. Nevertheless these warnings gives you a last chance to rethink the correctness of your taxon assignments.

5.2 Coarsening to a specific taxonomic level

If you want only species in your analyses and no other taxonomic level use `taxval(obs, ag='adapt', rank='SPE')`. All hierarchical levels below the species level (including the above specified monotypic sub-species) are set to species level in this case.

```
> tmp <- taxval(obs.tax, maxtaxlevel = "FAM", ag = "adapt", rank = "FAM")
```

Original number of names: 25

5 Synonyms found in dataset, adapted

SPECIES_NR	ABBREVIAT	Freq.1	VALID_NR	VALID_NAME
20010	Cardamine pratensis subsp. pratensis	1	15133	Cardamine pratensis
20096	Achillea millefolium subsp. collina	1	29	Achillea collina
20583	Armeria maritima subsp. bottendorfensis	1	20585	Armeria maritima subsp. halleri
25203	Abies alpestris	2	4269	Picea abies
27309	Armeria bottendorfensis	1	20585	Armeria maritima subsp. halleri

Freq.2

0
0
0
0
0

No taxa higher than FAM found.

Number of taxa after validation: 10

```
> tmp$Taxon <- species$ABBREVIAT[match(tmp$SPECIES_NR, species$SPECIES_NR)]
> tmp[order(tmp$Taxon), 9:10]
```

	Name	Taxon
8	Acer pseudoplatanus	Aceraceae species
9	Acer pseudoplatanus	Aceraceae species
16	Acoraceae species	Acoraceae species
1	Achillea millefolium agg.	Asteraceae species
5	Achillea millefolium	Asteraceae species
6	Achillea millefolium subsp. collina	Asteraceae species
7	Achillea species	Asteraceae species
11	Achillea millefolium subsp. sudetica	Asteraceae species
21	Hieracium pilosella	Asteraceae species
23	Hieracium subg. Pilosella	Asteraceae species
25	Cardamine dentata	Brassicaceae species
26	Cardamine pratensis agg.	Brassicaceae species
27	Cardamine pratensis	Brassicaceae species
28	Cardamine pratensis subsp. pratensis	Brassicaceae species
29	Cardamine species	Brassicaceae species
2	Quercus robur	Fagaceae species
3	Quercus robur	Fagaceae species
4	Quercus robur	Fagaceae species
10	Abies alpestris	Pinaceae species
15	Abies alpestris	Pinaceae species
24	Picea abies	Pinaceae species
12	Armeria maritima subsp. bottendorffensis	Plumbaginaceae species
13	Armeria maritima subsp. elongata	Plumbaginaceae species
14	Armeria maritima subsp. halleri	Plumbaginaceae species
22	Armeria bottendorffensis	Plumbaginaceae species
18	Dactylis glomerata	Poaceae species
20	Agrostis stolonifera var. palustris	Poaceae species
19	Adonis aestivalis	Ranunculaceae species
17	Galium mollugo	Rubiaceae species

Check ?taxval and args(taxval) for more options than the default.

5.3 Implementing another taxon view

If you wish to use another taxonomic concept than the default, you can use a conversion table to change synonymy etc. to catch your needs.

6 Vegetation matrices

At the moment there exist no formal class for vegetation data in R. But most functions in **vegan**, **ade4** or other packages expect vegetation data to be stored in a matrix with species in columns and plots in rows. Therefore, we need to inflate the *Turboveg* format (where zero occurrences are missing) to such a matrix.

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

6.1 Performance measures

At least in Europe most vegetation plots have some information about the cover of a species within the survey area, often given in some kind of alphanumeric code. Different codes systems can be combined by using the mean cover percentage per cover code class. Function **tv.coverperc** will do this job according to the definitions in *Turboveg/Popup/tvscale.dbf*.

```
> obs <- tv.obs(db)
```

```
reading observations ...
```

```
> obs <- tv.coverperc(db, obs)
```

```
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	COVER_PERC
1	2	27	2b	0	0	0	Schlenke	0	02	18
2	2	4685	4	1	0	0	Schlenke	0	02	68
3	2	4685	1	2	1	0	Schlenke	0	02	3
4	2	4685	1	6	0	0	<NA>	10	02	3
5	1	31	3	6	0	0	<NA>	0	01	38
6	1	20096	+	6	0	0	Schlenke	1	01	2

If option `convcode = TRUE` (the default) the covercodes used in the Turboveg database (see file `tv-scale.dbf`) are converted to (mean) percentage values according to the entries in the Turboveg Popup list "TVScale". For visual control the translated values will be printed on the screen.

A few simple possibilities for percentage cover transformations are included in function `tv.veg`, e.g. to use only presence-absence information you can choose option `cover.transform = 'pa'`.

6.2 Pseudospecies

How to account for different vegetation layers or other kinds of species differentiation?

The next step is the separation of pseudo-species. Pseudo-species are all kind of taxa split according to species-plot information beyond the performance measure which will be used within the matrix. At this point you have to decide which information should be preserved and which should be aggregated. For instance layer separation must be defined at this step. The default is to differentiate tree, shrub and herb layers but to combine finer layer specifications within them.

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 e.g. to sum up all cover percentages of different layers (`lc='sum'`) or the maximum value (`lc='max'`), mean value (`lc='mean'`). If we assume an independent occurrence of a species in different vertical layers, we can do the calculations with option `lc = 'layer'` (the default). This results in a probability sum: A species covering 50% in tree layer 1 and 50% in herb layer will get a combined cover of 75% because both layers will overlap 50% ($1 - 0.5 \cdot 0.5$).

If you want to specify pseudo-species by other species-plot differentiation you can define a combination dataframe. Two example dataframes are included in the package (`lc.0` and `lc.1`). Option `comb` has to be given as a list with first element naming the column name holding the grouping variable and as second element the name of the combination dataframe. Try

```
> data(lc.0)
> tv.veg(db, pseudo = list(lc.0, c("LAYER")), lc = "layer")
```

and check the column names:

reading observations ...

Taxonomic reference list: GermanSL 1.2

converting cover code ...

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

creating pseudo-species ...

combining occurrences using type LAYER and creating vegetation matrix ...

replacing species numbers with short names ...

```
[1] "AGRTS;P.6" "CARD#PR.6" "HIERSUG.6" "ACERPSE.5" "ACERPSE.6" "CARDPRA.6" "DACYGLO.6" "CARDPRA.6"
[9] "ACHICOL.6" "ARMEM-H" "ARMEM-E" "ARMEM-H" "CARDDEN.6" "PICEABI.2" "PICEABI.3" "GALUMOL.6"
[17] "ACHI#MI" "ARMEM-H.6" "HIERPIO" "ACHIMIL.6" "ACHIM-S.6" "PICEABI.1" "QUERROB.1" "QUERROB.2"
[25] "QUERROB.6" "ACHI-SP.6" "CARD-SP.6" "ACOR-SP.6" "ADONAES.6"
```

Separated by dots and layer numbers you can see the preserved layers. For meaning of layer numbers see Turboveg Help.

Check cover aggregation for the default layer combination.

Beside layers you can use any kind of species-plot attributes to distinguish between occurrences, for instance in a multi-temporal survey.

```
> comb <- list(data.frame(SEASON = 0:4, COMB = c(0, "Spring", "Summer", "Autumn",
+ "Winter")), "SEASON")
> tv.veg(db, tax = FALSE, pseudo = comb)
```

reading observations ...

Taxonomic reference list: GermanSL 1.2

converting cover code ...

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

creating pseudo-species ...

combining occurrences using type LAYER and creating vegetation matrix ...

replacing species numbers with short names ...

	AGRTS;P	CARD#PR	HIERSUG	ACERPSE.Spring	ACERPSE.Summer	CARDPRA	DACYGLO	CARDPRA	ACHICOL	ARMEM-H
1	3	0	0	3	13	0	3	0	2	0
2	0	0	0	0	0	0	0	0	0	0

	3	0	3	3	0	0	3	0	3	0	3	
	ARMEM-E	ARMEM-H	CARDDEN	PICEABI	GALUMOL	ACHI#MI	ARMEM-H	HIERPIO	ACHIMIL	ACHIM-S	PICEABI	QUERROB
1	0	0	0	6	3	0	0	0	38	3	0	0
2	0	0	0	0	0	18	38	3	0	0	3	70
3	3	3	3	0	0	0	0	0	0	0	0	0

	ACHI-SP	CARD-SP	ACOR-SP	ADONAES
1	3	0	0	3
2	0	0	0	0
3	0	3	3	0

```
> data(lc.1)
> veg <- tv.veg(db, lc = "sum", pseudo = list(lc.1, "LAYER"), dec = 1, quiet = TRUE)
```

```
> veg[, 1:10]
```

	AGRTS;P	HIERSUG	ACERPSE	ACERPSE.Shrub	DACYGLO	ARMEM-E	ARMEM-H	GALUMOL	PICEABI.Tree	QUERROB
1	3	0	3	13	3	0	0	3	6	0
2	0	3	0	0	0	0	38	0	3	3
3	0	3	0	0	0	3	6	0	0	0

6.3 Combine species by hand

Beside semi-automatic taxon harmonization with function `taxval` there are two possibilities to change Taxonomy manually. If you decide to interpret a certain species name in your database different than stored in the standard view of the taxonomic reference you can replace species numbers within the observational dataframe and run `taxval` later on.

```
> obs.tax$SPECIES_NR[obs.tax$SPECIES_NR == 27] <- 31
```

will replace all occurrences of *Achillea millefolium agg.* with *Achillea millefolium* which might be adequate for your survey and will prevent a too coarse taxon grouping. For a longer list of replacements you can use a dataframe.

```
> taxon.repl <- data.frame(old = c(27), new = c(31))
> obs.tax$SPECIES_NR <- replace(obs.tax$SPECIES_NR, match(taxon.repl$old, obs.tax$SPECIES_NR),
+   taxon.repl$new)
```

The second possibility is to use function `comb.species` on vegetation matrices.

```
> comb.species(veg, sel = c("QUERROB", "QUERROB.Tree"))
```

The following names are combined to new name QUERROB

	"QUERROB"	"QUERROB.Tree"
[1]	"QUERROB"	"QUERROB.Tree"

	AGRTS;P	HIERSUG	ACERPSE	ACERPSE.Shrub	DACYGLO	ARMEM-E	ARMEM-H	GALUMOL	PICEABI.Tree	ACHI-SP
1	3	0	3	13	3	0	0	3	6	46
2	0	3	0	0	0	0	38	0	3	18
3	0	3	0	0	0	3	6	0	0	0

	CARD-SP	ACOR-SP	ADONAES	QUERROB
1	0	0	3	0
2	0	0	0	74
3	15	3	0	0

will use the first name ('QUERROB') for the replacement column with the sum of the selected columns.

7 Site data

`tv.site` will load the site (header) data and does some basic corrections caused by Turbovegs dBase format.

```
> site <- tv.site(db)
```

The following columns contain no data and are omitted

```
[1] REFERENCE TABLE_NR NR_IN_TAB PROJECT AUTHOR SYNTAXON UTM ALTITUDE
[9] 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 COV_LITTER
[9] COV_WATER COV_ROCK TREE_HIGH TREE_LOW SHRUB_HIGH SHRUB_LOW HERB_HIGH HERB_LOW
[17] 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 is quite straightforward. After loading the file *tvhabita.dbf* from the specified database folder, warnings are given for plots without specified relevé area or date and 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.

8 VegetWeb, the National German vegetation plot repository

`vw.obs` and `vw.site` provide possibilities to access the data stored in VegetWeb, the German national vegetation database.

```
> library(RMySQL)
```

Installation of RMySQL package under Windows OS Whereas Linux OS normally include drivers to access MySQL databases this needs some installation and configuration for MS Windows operational systems. Installation under Windows requires installation of MySQL headers, eg. MySQL Server or ODBC Connector. See <http://biostat.mc.vanderbilt.edu/wiki/Main/RMySQL>

Tip: If loading fails under MS Windows despite installed and running MySQL-Server (Error message 'load-Namespace' failed) try to insert `MYSQL_HOME="C:/Programme/MySQL/MySQLServer 5.0"` (Please change to the correct path of your MySQL Installation) into a file named "Renviron.site" at <R-Installation-path>/etc and restart R or execute options(`MYSQL_HOME =< MYSQL - PATH >`) before loading RMySQL.

```
> vw.site()
```

No query string specified.

You can select vegetation plots from VegetWeb with queries like

```
query="Projekt='T271'"
```

This will select all relevés from project T271, i.e Dengler 2007 Tuexenia.

If you want to see which selection parameters are available, try:

```
con <- vw.con()
```

```
dbListFields(con, "beobachtung")
summary(dbGetQuery(con, "SELECT pH FROM beobachtung"))
dbGetQuery(con, "SELECT * FROM projekt")
etc.
```

vw.site() and vw.veg() are the user interface to retrieve data from VegetWeb respectively an open access mirror of the original BfN Server at the University of Greifswald (updated every Sunday).

```
> con <- vw.con()
> dbListTables(con)
> url.show("http://planto.de/OekoArt/ModellLog.php")
```

Get names of VegetWeb tables and look for the Relationship-Diagram.

```
> con <- vw.con()
> dbListFields(con, "beobachtung")
```

[1] "Beobachtungs_ID"	"Beobachtungscode"	"Plotcode"
[4] "Vorbeobachtung"	"Projekt"	"DatumVon"
[7] "Datum"	"Moosidentifikation"	"Flechtenidentifikation"
[10] "Deckungsmethode"	"Deckungsmethode2"	"Nutzung"
[13] "Anteil Streuschicht"	"Anteil offene Wasserfl\xe4che"	"Anteil Fels"
[16] "Anteil Skelett"	"Anteil offener Boden"	"Deckung Baumschicht"
[19] "Deckung Strauchschicht"	"Deckung Feldschicht"	"Deckung Kryptogamenschicht"
[22] "Deckung Schwimmblattschicht"	"Deckung Wasserpflanzenschicht"	"H\xe4ufl. Baumschicht"
[25] "H\xe4ufl. Strauchschicht"	"H\xe4ufl. Feldschicht"	"H\xe4ufl. Kryptogamenschicht"
[28] "Tiefe Wasserpflanzen"	"Bemerkung"	"Erheber"
[31] "Digitalisierer"	"Zitat"	"Zitattabelle"
[34] "Zitataufnahme"	"Verband"	"Assoziation"
[37] "Gesellschaftsbezeichnung"	"Subassoziation"	"Sukzessionsstatus"
[40] "Bestandsalter"	"Bioz. \xe4kologische Aspekte"	"Hydrologie"
[43] "Grundwasserflurabstand"	"Uferentfernung"	"Bodenart"
[46] "Bodentyp"	"Humusform"	"pH"
[49] "Phosphor"	"Kalium"	"Magnesium"
[52] "N-Gehalt"	"Biotoptyp"	"Pflegema\xdfnahmen"
[55] "D\xe4ngung"	"Schutz"	"Temperatur"
[58] "User"	"Modified"	

```
> dbGetQuery(con, "SELECT * FROM projekt")[1:2]
```

	Projekt	Projektname
1	Bohn	Vegetationsaufnahmen bodensaurer Buchenw\xe4lder
2	BgWd	Bergwald - Datenbank
3	T252	T\xfcxenia 25 Becker
4	T251	T\xfcxenia 25
5	T253	T\xfcxenia 25 B\xfccktmann
6	T262	T\xfcxenia 26
7	T264	T\xfcxenia 26
8	T254	T\xfcxenia 25 Fuchs
9	T255	T\xfcxenia 25 Gehlken
10	T269	T\xfcxenia 26
11	T261	T\xfcxenia 26
12	T268	T\xfcxenia 26
13	T256	T\xfcxenia 25 Otte und Maul
14	T265	T\xfcxenia 26
15	T266	T\xfcxenia 26
16	T282	Die Schwermetall-Vegetation des Harzes

```

17 T292      Populationsstruktur und Vergesellschaftung von Dictamnus albus L.
18 T275      T\xfcxenia 27 Huntke
19 T274      T\xfcxenia 27 Kr\xe4mer u. Fartmann
20 T273      T\xfcxenia 27 Brandes
21 T272      T\xfcxenia 27 Klauck
22 T271      T\xfcxenia 27 Dengler
23 T263      T\xfcxenia 26
24 T276      T\xfcxenia 27 Becker
25 T281      Die Allmendeweide \x84NSG Kanzelstein bei Eibach\x93
26 T291 Verbreitung, Vergesellschaftung und \xd6kologie von Lathraea squamaria
27 T293      Trittgesellschaften der nordrhein-westf\xe4lischen D\xf6rfer
28 T294      Succession and management of calcareous dry grasslands
29 SYPF10     Synopsis der Pflanzengesellschaften Deutschlands
30 vNWR      Naturwaldreservate Bayerns
31 LANUV     LINFOS-Daten Landesamt f\xfcr Natur, Umwelt und Verbraucherschutz NRW
32 T302      T\xfcxenia 30 Schmitt et al.

```

8.1 Plot locations in Google Earth

```
> query <- "Projekt='T302'"
```

Since several years all authors of **Tuexenia** are committed to give their published data to VegetWeb. Therefore we can quite easily download the data from e.g.:

Schmitt, Fartmann, Hoelzel 2010 Phytosociology and ecology of *Gladiolus palustris* in southern Bavaria, Tuexenia 30, p. 105-128.

```

> site <- vw.site(query)
> site.coord <- site[!is.na(site$"Geografische Breite"), ]

```

Plot coordinates of vegetation relevés into an interactive Google Map

```

> library(googleVis)
> site.coord$loc <- paste(site.coord[, 29], site.coord[, 28], sep = ":")

```

If you do not have geodesic coordinates as used in Google Earth (EPSG-Code 4326), you can convert coordinates with R packages rgdal

```

> library(rgdal)
> coord <- data.frame(HW = as.numeric(site.coord$Nordkoordinate), RW = as.numeric(site.coord$Ostkoordinate))
> coordinates(coord) <- c("RW", "HW")
> proj4string(coord) <- CRSargs(CRS("+init=epsg:31468"))
> coord <- spTransform(coord, CRS("+init=epsg:4326"))
> site.coord$long <- coordinates(coord)[, 1]
> site.coord$lat <- coordinates(coord)[, 2]
> site.coord$loc <- paste(site.coord$lat, site.coord$long, sep = ":")

```

To give some information we will create Tips:

```

> site.coord$tip <- paste(paste("Releve_NR:", site.coord$RELEVE_NR), paste("Table:",
+   site.coord$TABLE_NR), paste("Nr. in table:", site.coord$NR_IN_TAB), site.coord$DATE,
+   landuse = site.coord$NUTZUNG, site.coord$Assoziation, site.coord$Erheber,
+   paste("Locality:", site.coord[, 26]), paste("Longitude:", site.coord[, 28]),
+   paste("Latitude:", site.coord[, 29]), sep = "<BR>")

```

and the produced map will be open in your standard web browser.

```

> places <- gvisMap(site.coord, "loc", "tip", options = list(showTip = TRUE, showLine = FALSE,
+   enableScrollWheel = TRUE, mapType = "hybrid", useMapTypeControl = TRUE,
+   width = 1000, height = 800))
> plot(places)

```

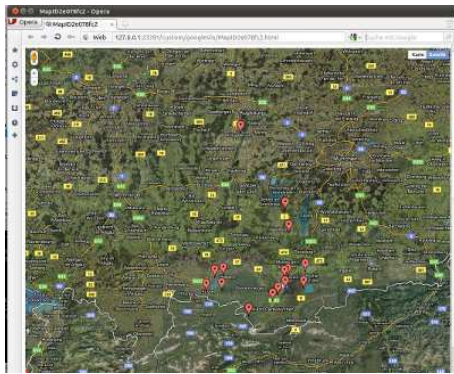



Figure 2: Spatial distribution of vegetation plots from a VegetWeb project. The map is interactive and zoomable.

9 Additional functions

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

9.1 Combine different taxonomic reference lists

If you have to combine different taxonomic reference lists, functions `tv.compRef1` might be a starting point, comparing species numbers and/or species names of both lists.

```
> tv.compRef1("taxref1", "taxref2")
```

10 Vegetation analyses

The package *vegdata* serves mostly as a helper for the analysis of vegetation data. Several powerful R packages like *vegan* and others exist, to provide a very broad range of possibilities.

10.1 Frequency tables

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

```
> data(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")
```

We can e.g. look at the relative frequency of all species with more than 40% at least in one column, according to the height of the groundwater table (low or high) and the amplitude of the groundwater table fluctuations (high or low deviations from the mean).

```
> syntab(elbaue, clust, limit = 40)
```

```

Number of clusters: 4
Cluster frequency 7 10 5 11
      dry.ld dry.hd wet.hd wet.ld
AGRTCAP      57      30      .      18
ALOPGEN       .      20     60      9
ALOPPRA      71     90     20     36
ANTXODO      43     10      .     27
CARDPRA      43     10      .     55
CAREACU      14      .     40     82
CAREPRA      43     70      .      .
CAREVES       .      .      .     55
CIRSARV      43      .      .      9
DESCCES      57      .      .     18
ELYMREP      57     90      .      .
EUPHESU      43      .      .      .
GALUPAL      29     30     60     64
GALU#VE      71     20      .      .
GLYCMAX       .      .     80     45
HOLCLAN      43      .      .     36
JUNUEFF      14      .     20     45
LATYPRA      43      .      .      9
PHALARU      14     40     80     64
POA PAL       29     60     20     45
POA #PR       57     60     20     27
POA T-T       14     30     20     45
RANCFLA       .      .      .     55
RANCREP      29     60     40     73
RORIAMP       .      .     60      9
RUMEACE      43      .      .     27
RUMETHY      43     60      .      .
SIUMLAT       .      .     40     45
STELPAU      14     20      .     64
TARA/AN      57     60      .     18
VICICRA      43     10      .     18
VICITET      57     10      .      .

```

Or we can calculate the affiliation of species to abiotic clusters with the help of package `indicspecies`, which calculates species indicator values for one or several cluster (De Cáceres *et al.*, 2010).

```
> syntab(elbaue, clust, mupa = TRUE, fullnames = TRUE)
```

```

Number of clusters: 4
Cluster frequency 7 10 5 11
      dry.ld dry.hd wet.hd wet.ld  cl
Cirsium arvense      43      .      .      9  1
Deschampsia cespitosa  57      .      .     18  1
Euphorbia esula      43      .      .      .  1
Galium verum agg.    71     20      .      .  1
Lathyrus pratensis   43      .      .      9  1
Vicia tetrasperma    57     10      .      .  1
Alopecurus geniculatus .     20     60      9  3
Rorippa amphibia     .      .     60      9  3
Carex gracilis      14      .     40     82  4
Carex vesicaria      .      .      .     55  4
Agropyron caninum    .      .      .     36  4
Ranunculus flammula   .      .      .     55  4
Alopecurus pratensis  71     90     20     36 1+2
Carex praecox agg.   43     70      .      . 1+2

```

Agropyron repens subsp. caesium	57	90	.	. 1+2
Rumex thyrsiflorus	43	60	.	. 1+2
Taraxacum officinale agg.	57	60	.	18 1+2
Cardamine nemorosa	43	10	.	55 1+4
Plantago major subsp. intermedia	.	40	20	. 2+3
Rorippa sylvestris	.	40	40	9 2+3
Glyceria maxima	.	.	80	45 3+4
Sium latifolium	.	.	40	45 3+4

10.2 Multivariate Ordinations

With the functions shown above we are now ready to do some example analyses in the wide area of vegetation analyses.

We can do, for instance, 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(elbaue, distance = "bray", trymax = 5, autotransform = FALSE,
+   noshare = 1, expand = TRUE, trace = 2)
```

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

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

The first axis of our NMDS plot show the influence of mean groundwater level on the patterns of the dataset. *Glyceria maxima* is marking the wet side of the gradient, whereas *Cnidium dubium* *Agrostis capillaris* or *Galium verum agg.* occur only at low mean groundwater level. The second axis can be assigned to the fluctuation of water levels measured as standard deviation of mean groundwater level. Species indicating high water fluctuation are *Agrostis stolonifera* or *Alopecurus geniculatus* whereas *Carex vesicaria* occurs only at more balanced situations.

References

Cox, T.F. & Cox, M.A.A. (1994, 2001). *Multidimensional Scaling*. Chapman & Hall.

```
> print(nmms.plot(veg.nmms, elbaue.env, disp = "species", var1 = "MGL", var2 = "SDGL",
+   plottitle = "NMDS of Elbaue floodplain vegetation"))
```

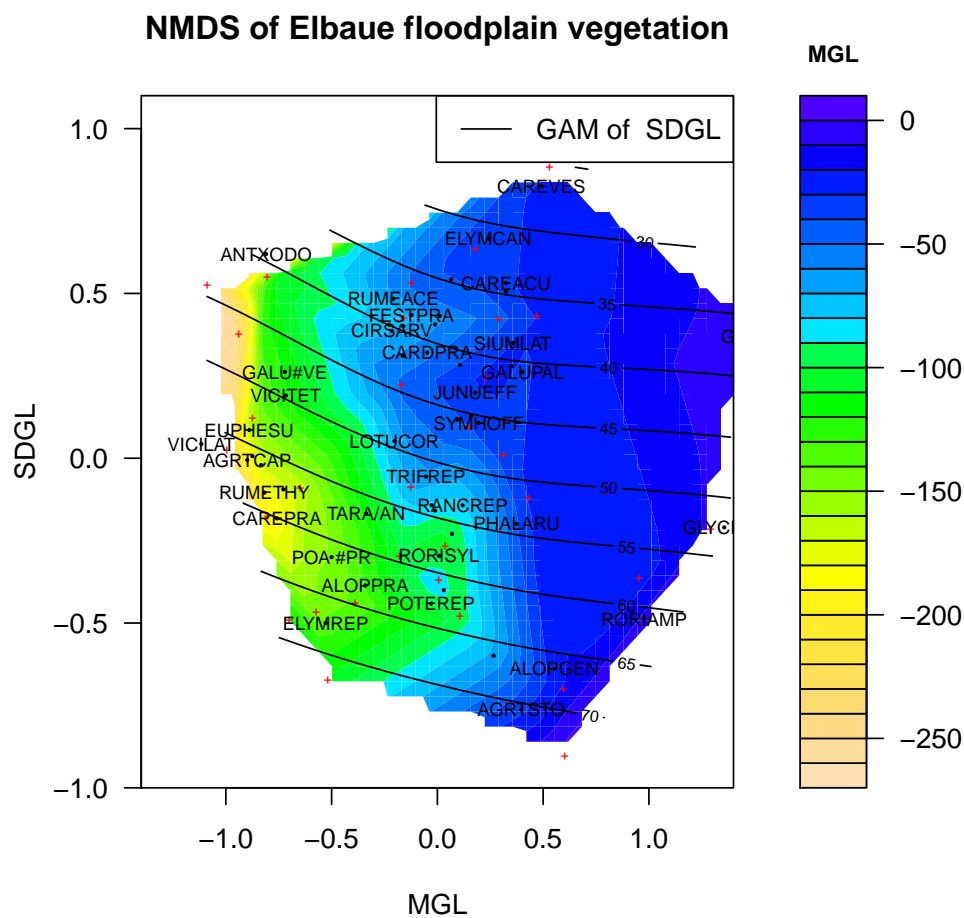


Figure 3: Non-metric multidimensional scaling of the elbaue vegetation data with an overlay of mean ground-water table (colors) and standard deviation of groundwater level fluctuations (lines).

- De Cáceres, M., Legendre, P. & Moretti, M. (2010). Improving indicator species analysis by combining groups of sites. *Oikos*, 119, 1674–1684.
- Dengler, J., Jansen, F., Glöckler, F., Peet, R., De Cáceres, M., Chytrý, M., Ewald, J., Oldeland, J., Lopez-Gonzalez, G., Finckh, M. & Others (2011). The Global Index of Vegetation-Plot Databases (GIVD): a new resource for vegetation science. *Journal of Vegetation Science*, 22, 582–597.
- Hennekens, S.M. & Schaminée, J.H.J. (2001). Turboveg, a comprehensive data base management system for vegetation data software 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.
- Jansen, F. & Dengler, J. (2010). Plant names in vegetation databases - a neglected source of bias. *Journal of Vegetation Science*, 21, 1179–1186.
- 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*.