

Vegetation data access and evaluation

Version 0.3.1

Florian Jansen

July 1, 2011

Abstract

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

1 Preliminary notes

Most *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)). For the same reason this is not a strict vignette, i.e. it is build with Sweave and uses executable *vegdata* code, but requires the installation of Turboveg and the existence of taxonomic reference list GermanSL, which can be downloaded from <http://geobot.botanik.uni-greifswald.de/reflist>.

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 changes 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, i.e. 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 revaluation

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 `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 a unique species performance platform is needed which will mostly 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.

2.3 Layer aggregation

The most frequently used sample unit in vegetation science is a plot based vegetation 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) at a specific time. It contains (at least is intended to contain) a complete list of photoautotrophic 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 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 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

Turbogveg stores relevés as a list of occurrences (s. above) but almost all functions and programs for vegetation analyses use plot-species crosstables 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 lettercodes (default) or full names (with underscores instead of blanks).

3 Examples

Maybe the best way to introduce 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 a subset of the taxonomic reference list Germansl 1.2, which can be loaded with option `sysPath=TRUE`. If you want to learn more about the taxonomic reference list for Germany, please look at Jansen & Dengler (2008). You can download the list from '<http://geobot.botanik.uni-greifswald.de/portal/reflist>'.

The package functions try to guess as many things as possible. The path of your Turboveg installation is searched at package loading time on MS-Windows OS 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 or several Turboveg installations, please specify option `tv_home` manually (`options('tv_home=<>')`).

Using the default function settings you might only have to specify a Turboveg database name. That is the name also occurring in Turbowin dialogues and can be found below the directory "Data" in your Turboveg installation path. 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 vegetation data access from Turboveg, taxonomic amendment, cover unification, layer combination and the generation of a vegetation matrix.

`tv.site()` will load the site (header) data and does some basic corrections caused by Turboveg's DBASE 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”, stored in the database directory which can then be loaded by:

```
> tv.metainfo(db1)
```

3.2 Site data

```
> site <- tv.site(db1)
```

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 `tv.site()` 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.

3.3 Vegetation data

Now we care about the species occurrence data.

The simple loading of species observations from Turboveg (*tvabund.dbf*) is done by function `tv.obs()`

```
> obs <- tv.obs(db1)
```

```
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

In Turboveg data is stored as a flat table of occurrences, that is one species-plot occurrence per row. Field *RELEVE_NR* contains the plot number, *SPECIES_NR* the taxon name 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")
```

```
[1] "Using Lettercodes."
```

```
  SPECIES_NR LETTERCODE  ABBREVIAT  NATIVENAME SYNONYM VALID_NR
6           8    ACERNEG Acer negundo Eschen-Ahorn   FALSE      8
```

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

```
  SPECIES_NR LETTERCODE  ABBREVIAT AUTHOR SYNONYM VALID_NR
18           27    ACHI#MI Achillea millefolium agg.  <NA>   FALSE      27
      VALID_NAME                NATIVENAME GRUPPE RANG  AGG      AGG_NAME
18 Achillea millefolium agg. Artengruppe Wiesen-Schafgarbe      S  AGG 60728 Achillea species
      NACHWEIS                SECUNDUM HYBRID BEGRUEND EDITSTATUS
18 BfN(Wisskirchen u. Haeupler 1998) BfN(Wisskirchen u. Haeupler 1998)  <NA>    <NA>      BfN
```

As stated in the beginning the care about the taxonomic integrity of your database should be the beginning of your vegetation analyses. For Turboveg databases referenced with taxonomic list *GermanSL* (versions 0.9 or higher) this can be done semi-automatically.

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

```
> obs <- taxval(obs, refl = "GermanSL 1.1")
```

Original number of names: 20

4 Synonyms found in dataset, adapted

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

VALID_NAME	Freq_Agg
Armeria maritima subsp. halleri	0
Achillea collina	0
Picea abies	0
Armeria maritima subsp. halleri	0

No taxa higher than ROOT found.

5 child taxa found in dataset, adapted

SPECIES_NR	ABBREVIAT	AGG	AGG_NAME
31	Achillea millefolium	27	Achillea millefolium agg.
29	Achillea collina	27	Achillea millefolium agg.
33	Achillea millefolium subsp. sudetica	31	Achillea millefolium
27	Achillea millefolium agg. 60728		Achillea spec.
2923	Hieracium pilosella 12273	Hieracium subg. Pilosella	

2 child taxa found in dataset, adapted

SPECIES_NR	ABBREVIAT	AGG	AGG_NAME
27	Achillea millefolium agg. 60728		Achillea spec.
31	Achillea millefolium	27	Achillea millefolium agg.

1 child taxa found in dataset, adapted

SPECIES_NR	ABBREVIAT	AGG	AGG_NAME
27	Achillea millefolium agg. 60728		Achillea spec.

No (more) monotypic taxa found.

Number of taxa after validation: 12

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 & 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 & Haeupler 1998)
Galium mollugo	2555	Galium mollugo s. l.	26777 BfN (Wisskirchen & Haeupler 1998)

Have a look at `?taxval` or `args(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 (Jansen & Dengler, 2008). If your database is not referenced with GermanSL or a reference list with equal structure you can not use `tv.taxval()` and you have to execute `tv.veg()` with option `tax=FALSE` or you can try to convert your database to GermanSL (Export to XML in Turboveg and re-import choosing the new GermanSL) assuming you have a central european database.

3.3.2 Differing taxon views

German SL is based upon **taxon views** (Berendsohn (1995)) of available standard 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.

```
> tax1 <- tax(unique(taxval(db = "taxatest"), "SPECIES_NR"))
> tax2 <- tax(unique(taxval(db = "taxatest", concept = "korneck1996"), "SPECIES_NR"))
```

	SPECIES_NR		ABBREVIAT	VALID_NR
20641	60728		Achillea species	60728
2883	4685		Quercus robur	4685
10	15		Acer pseudoplatanus	15
2609	4269		Picea abies	4269
9050	20585	Armeria maritima subsp. halleri		20585
9049	20584	Armeria maritima subsp. elongata		20584
21653	66142	Acoraceae species		66142
1571	2555	Galium mollugo		2555
1118	1843	Dactylis glomerata		1843
46	76	Adonis aestivalis		76
4687	10024	Agrostis stolonifera var. palustris		10024
6005	12273	Hieracium subg. Pilosella		12273

	SPECIES_NR		ABBREVIAT	VALID_NR
20641	60728		Achillea species	60728
2883	4685		Quercus robur	4685
10	15		Acer pseudoplatanus	15
2609	4269		Picea abies	4269
9048	20583	Armeria maritima subsp. bottendorffensis		20585
9049	20584	Armeria maritima subsp. elongata		20584
9050	20585	Armeria maritima subsp. halleri		20585
21653	66142	Acoraceae species		66142
1571	2555	Galium mollugo		2555
1118	1843	Dactylis glomerata		1843
46	76	Adonis aestivalis		76
4687	10024	Agrostis stolonifera var. palustris		10024
6005	12273	Hieracium subg. Pilosella		12273

3.3.3 Cover values

Cover is coded in Turboveg as an 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.coverperc(db1, 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)
```

	RELEVÉ_NR	SPECIES_NR	COVER_CODE	LAYER	DET_CERT	SEASON	MICROREL	FLOWER	COVERSCALE	COVER_PERC
1	2	60728	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	60728	3	6	0	0	<NA>	0	01	38
6	1	60728	+	6	0	0	Schlenke	1	01	2

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

3.3.4 Pseudo-species, layer combinations and vegetation 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, the inflation of a vegetation matrix and the handling of species name 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 e.g. mean, max, min or first value. If we assume an independent occurrence of a species in different layers, we can do the calculations with option `lc = 'layer'` (the default). E.g. a tree with a cover of 50% in a tree layer and 50% in herb layer can be accounted with an overall cover of 75% (i.e. 50% overlap).

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

```
> 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)
```

```
> 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

```

3.4 Additional functions

`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
[1] "Using Lettercodes."
```

	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 acuta	14	.	40	82	4
Carex vesicaria	.	.	.	55	4
Elymus caninus	.	.	.	36	4
Ranunculus flammula	.	.	.	55	4
Alopecurus pratensis	71	90	20	36	1+2
Carex praecox	43	70	.	.	1+2
Elymus repens	57	90	.	.	1+2
Rumex thyrsiflorus	43	60	.	.	1+2
Taraxacum sect. Alpina, Hamata et Ruderalia	57	60	.	18	1+2
Cardamine pratensis	43	10	.	55	1+4
Glyceria maxima	.	.	80	45	3+4
Sium latifolium	.	.	40	45	3+4

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

3.5 Vegetation analyses

The package *vegdata* serves only 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.

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.nmnds <- 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))
> nm.ds.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

- Berendsohn, W.G. (1995). The concept of "potential taxa" in databases. *Taxon*, 44, 207–212.
- Cox, T.F. & Cox, M.A.A. (1994, 2001). *Multidimensional Scaling*. Chapman & Hall.
- De Cáceres, M., Legendre, P. & Moretti, M. (2010). Improving indicator species analysis by combining groups of sites. *Oikos*, 119, 1674–1684.
- Hennekens, S.M. & Schaminée, J.H.J. (2001). Turboveg, a comprehensive data base management system 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.
- Jansen, F. & Dengler, J. (2010). Plant names in vegetation databases - a neglected source of bias. *Journal of Vegetation Science*, 21, 1179–1186.
- 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*.

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

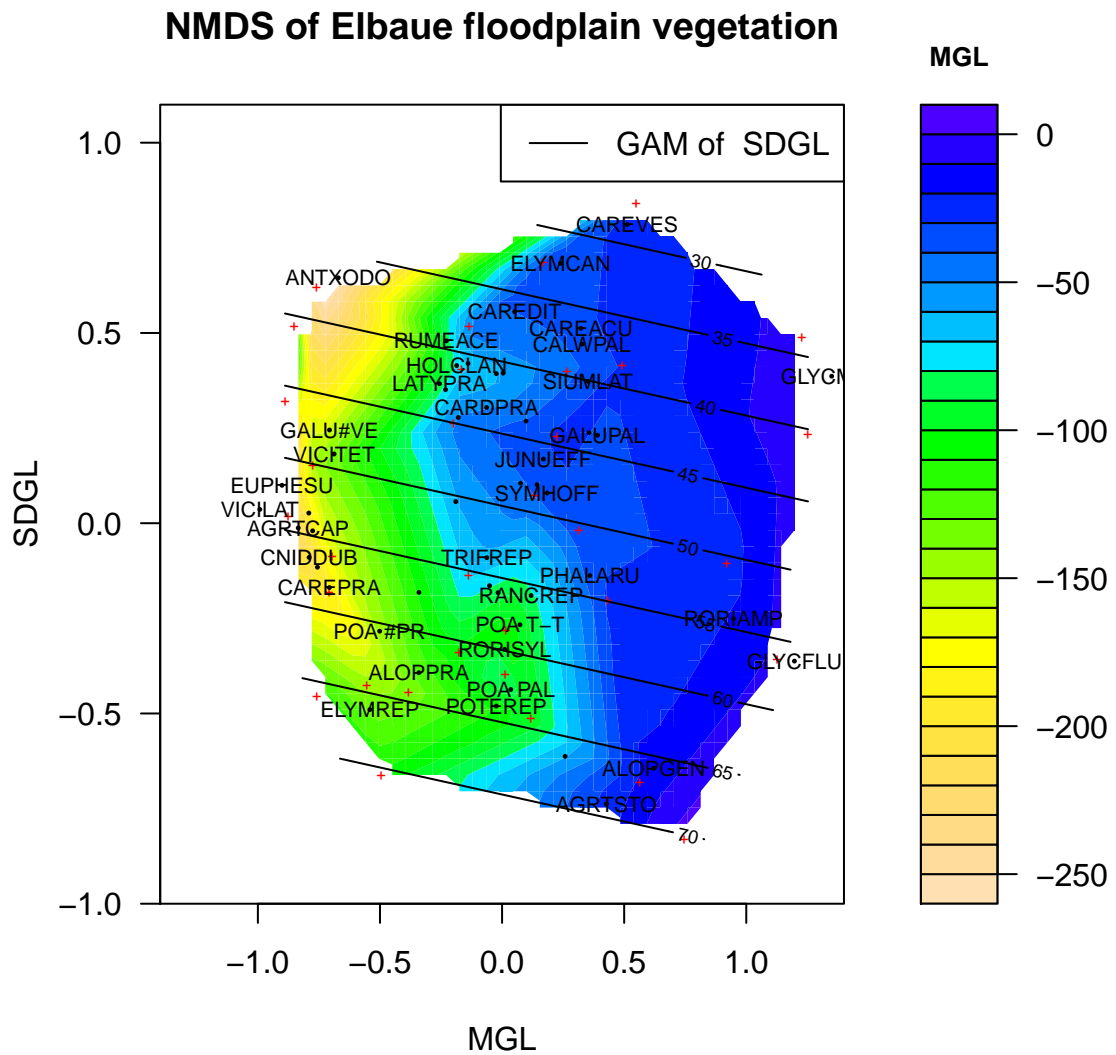


Figure 1: Nonmetric multidimensional scaling of the elbaue vegetation data with an overlay of mean ground-water table (colors) and standard deviation of groundwater level fluctuations (lines).