

Vegetation data access and taxonomic harmonization

version 0.6.4

Florian Jansen

September 6, 2013

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 & Schaminee (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 plots 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 R naming conventions).

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)

## Loading required package: foreign
## This is vegdata 0.6.4
```

Many functions use the directory structure of Turboveg. The first time such a function is called, the internal 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()

##
## No Turbowin installation path found.
## Should I use 1) the vegdata package path, or 2) a temporary folder?
##
## #####
## Turboveg root directory is set to /tmp/RtmpokP0Ic
## If you want to change this use: options(tv_home="<path_to_your_Turbowin_root>")
## #####
```

If you want to change this, declare manually by setting option "tv_home":

```
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 is the default Taxonomic reference list in package **vegdata**. However, 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")

## Reference list used: GermanSL 1.2
##
## Taxonomic list ( species.dbf ) of reflist version GermanSL 1.2 not available.
##      TaxonUsageID LETTERCODE      TaxonName
## 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
## 8680          20096  ACHICOL   Achillea millefolium subsp. collina
## 8681          20097  ACHIPAN   Achillea millefolium subsp. pannonica
## 8682          20098  ACHIPAN   Achillea millefolium var. lanata
## 13222         26082  ACHIMIL   Achillea millefolium var. firma
## 26250         90019  ACHI*AB   Achillea millefolium agg. x nobilis
## 26251         90020  ACHIM*P   Achillea millefolium x pannonica
##
##      VernacularName SYNONYM TaxonConceptID
## 18      Artengruppe Wiesen-Schafgarbe FALSE      27
## 20      Gew+Ähnliche Wiesen-Schafgarbe FALSE      31
```

## 21	Gew+Ähnliche Wiesen-Schafgarbe i.e.S.	FALSE	32
## 22	Sudeten-Wiesenschafgarbe	FALSE	33
## 8680	<NA>	TRUE	29
## 8681	<NA>	TRUE	34
## 8682	<NA>	TRUE	34
## 13222	<NA>	TRUE	31
## 26250	<NA>	TRUE	90028
## 26251	<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 (see `tv.home()`), if it is not installed but needed. If you want to use a different list, specify `refl=<Name of your list>` according to the directory name in the Turboveg directory *Species*. Function `tax` will use the given character string, or 7 letter abbreviation or TaxonUsageID (called SPECIES_NR in Turboveg) to look for all (partially) matching species names within the reference list.

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

```
## Reference list used: GermanSL 1.2
##      TaxonUsageID LETTERCODE      TaxonName AUTHOR SYNONYM TaxonConceptID
## 20          31    ACHIMIL Achillea millefolium    L.   FALSE          31
##      TaxonConcept      VernacularName taxonRank GRUPPE    FAMILIE
## 20 Achillea millefolium Gewöhnliche Wiesen-Schafgarbe    SPE    S Asteraceae
##      IsChildTaxonOfID      IsChildTaxonOf      NACHWEIS
## 20          27 Achillea millefolium agg. BfN(Wisskirchen u. Haeupler 1998)
##      publishedInCitation HYBRID BEGRUEND EDITSTATUS
## 20 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) which can be used with option `verbose=TRUE`.

`tax` will give you all matching names by default. If you set option `strict=TRUE`, only the species with exact match to the given character string will be returned.

`syn` will give you all taxon names within the swarm of synonyms. The valid name is marked in column SYNONYM with FALSE.

```
tax("Elytrigia repens")$TaxonName
```

```
## Reference list used: GermanSL 1.2
## [1] "Elytrigia repens subsp. arenosa" "Elytrigia repens"
## [3] "Elytrigia repens var. caesia"    "Elytrigia repens var. littoralis"
## [5] "Elytrigia repens var. repens"
```

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

```
## Reference list used: GermanSL 1.2
## Name swarm of Elytrigia repens :
##      TaxonUsageID      TaxonName SYNONYM EDITSTATUS
## 4078          6541 Agropyron repens subsp. caesium    TRUE    BfN
## 4081          6544 Elymus repens subsp. repens s. l.    TRUE Korrektur
## 4791         10260 Elymus repens subsp. caesium    TRUE    BfN
## 8714         20143 Agropyron caesium    TRUE    BfN
## 8732         20167 Agropyron repens subsp. repens    TRUE    BfN
## 9890         21639 Elytrigia repens    TRUE    BfN
## 12066        24393 Triticum repens    TRUE    BfN
## 13916        27778 Elymus repens    FALSE    BfN
## 14008        27914 Agropyron repens    TRUE    BfN
```

The reference list contains information about the taxonomic hierarchy which can be used with *childs* or *parents*.

```
childs(27, quiet = TRUE)$TaxonName

## [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")

##      TaxonUsageID LETTERCODE      TaxonName      AUTHOR SYNONYM
## 18          27    ACHI#MI Achillea millefolium agg.      <NA> FALSE
## 20644      60728    ACHI-SP      Achillea species              L. FALSE
## 20474      60463    ASTE-SP      Asteraceae species      Dumort. FALSE
## 20448      60415    ASTR-SP      Asterales species      Lindley FALSE
## 20327      60079    ASTI-SP      Asteridae species      Takht. FALSE
## 20321      60071    MAGL-SP      Magnoliopsida species      Dc. FALSE
## 20312      60049    MAGO-SP      Magnoliophytina species A. Braun & Doell FALSE
## 20286      60000    SPEA-SP      Spermatophyta species      <NA> FALSE
## 29377      94419    "GEF-SP "Gefaesspflanze" species      - FALSE
## 1           0    "GRUETW      "Gruenliches etwas"      - FALSE

##      TaxonConceptID      TaxonConcept      VernacularName taxonRank
## 18          27 Achillea millefolium agg. Artengruppe Wiesen-Schafgarbe AGG
## 20644      60728      Achillea species      Schafgarbe GAT
## 20474      60463      Asteraceae species      <NA> FAM
## 20448      60415      Asterales species      <NA> ORD
## 20327      60079      Asteridae species      <NA> UKL
## 20321      60071      Magnoliopsida species      <NA> KLA
## 20312      60049      Magnoliophytina species      <NA> UAB
## 20286      60000      Spermatophyta species      <NA> ABT
## 29377      94419      "Gefaesspflanze" species      <NA> AG2
## 1           0      "Gruenliches etwas"      <NA> ROOT

##      GRUPPE      FAMILIE IsChildTaxonOfID      IsChildTaxonOf
## 18          S Asteraceae      60728      Achillea species
## 20644      S Asteraceae      60463      Asteraceae species
## 20474      S      <NA>      60415      Asterales species
## 20448      S      <NA>      60079      Asteridae species
## 20327      S      <NA>      60071      Magnoliopsida species
## 20321      S      <NA>      60049      Magnoliophytina species
## 20312      S      <NA>      60000      Spermatophyta species
## 20286      S      <NA>      94419      "Gefaesspflanze" species
## 29377      S      <NA>      0      "Gruenliches etwas"
## 1      <NA>      <NA>      0      "Gruenliches etwas"

##      NACHWEIS      publishedInCitation HYBRID
## 18      BfN(Wisskirchen u. Haeupler 1998) BfN(Wisskirchen u. Haeupler 1998) <NA>
## 20644      BfN(Wisskirchen u. Haeupler 1998) BfN(Wisskirchen u. Haeupler 1998) <NA>
## 20474      Wisskirchen u. Haeupler 1998      Wisskirchen u. Haeupler 1998      0
## 20448      Wisskirchen u. Haeupler 1998      Wisskirchen u. Haeupler 1998      0
```

##	20327	Wisskirchen u. Haeupler 1998	Wisskirchen u. Haeupler 1998	0
##	20321	Wisskirchen u. Haeupler 1998	Wisskirchen u. Haeupler 1998	0
##	20312	Wisskirchen u. Haeupler 1998	Wisskirchen u. Haeupler 1998	0
##	20286	Wisskirchen u. Haeupler 1998	Wisskirchen u. Haeupler 1998	0
##	29377	-	[ad-hoc-Taxon für GermanSL]	0
##	1	-	[ad-hoc-Taxon für GermanSL]	0
##		BEGRUEND	EDITSTATUS	GENERATION
##	18	<NA>	BfN	1
##	20644	<NA>	BfN	2
##	20474	Abweichung zur Druckversion	Korrektur	3
##	20448	<NA>	BfN	4
##	20327	<NA>	BfN	5
##	20321	Abweichung zur Druckversion	Korrektur	6
##	20312	Abweichung zur Druckversion	Korrektur	7
##	20286	Abweichung zur Druckversion	Korrektur	8
##	29377	<NA>	Ergaenzung	9
##	1	<NA>	Ergaenzung	10

If you want to learn more about the taxonomic reference list *GermanSL* 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.

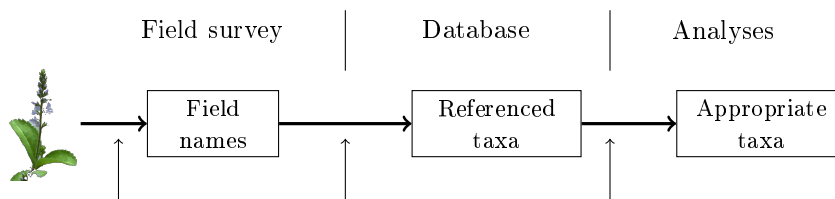
```
obs.tax <- tv.obs(db)

## reading observations ...

# Adding species names
species <- tax("all")

## Reference list used: GermanSL 1.2

obs.tax$TaxonName <- species$TaxonName[match(obs.tax$TaxonUsageID, species$TaxonUsageID)]
head(obs.tax[, c("RELEV_NR", "TaxonUsageID", "COVER_CODE", "LAYER", "TaxonName")])
```



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	TaxonUsageID	COVER_CODE	LAYER	TaxonName
## 1	2	27	2b	0	Achillea millefolium agg.
## 2	2	4685	4	1	Quercus robur
## 3	2	4685	1	2	Quercus robur
## 4	2	4685	1	6	Quercus robur
## 5	1	31	3	6	Achillea millefolium
## 6	1	20096	+	6	Achillea millefolium subsp. collina

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 Function `taxval`

We are using the taxonomic reference list GermanSL (Jansen & Dengler, 2008) 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, db = db, mono = "lower")

##
## Original number of names: 20
## 4 Synonyms found in dataset, adapted
## TaxonUsageID TaxonName Freq.1 TaxonConceptID
## 20096 Achillea millefolium subsp. collina 1 29
## 20583 Armeria maritima subsp. bottendorffensis 1 20585
## 25203 Abies alpestris 2 4269
## 27309 Armeria bottendorffensis 1 20585
## TaxonConcept Freq.2
## Achillea collina 0
## Armeria maritima subsp. halleri 0
## Picea abies 0
## Armeria maritima subsp. halleri 0
## 1 monotypic taxa found in dataset, set to lower rank.
## AGG_NR AGG_taxonR MEMBER_NR MEMB_NAME MEMB_taxon
## 66142 FAM 61329 Acorus species GAT
## 1 monotypic taxa found in dataset, set to lower rank.
## AGG_NR AGG_taxonR MEMBER_NR MEMB_NAME MEMB_taxon
## 61329 GAT 69 Acorus calamus SPE
## No taxa higher than ROOT found.
## 5 child taxa found in dataset, adapted
## TaxonUsageID TaxonName Freq.1 IsChildTaxonOfID
## 29 Achillea collina NA 27
## 31 Achillea millefolium 1 27
## 33 Achillea millefolium subsp. sudetica 1 31
## 27 Achillea millefolium agg. 1 60728
## 2923 Hieracium pilosella 1 12273
## IsChildTaxonOf Freq.2
## Achillea millefolium agg. 1
## Achillea millefolium agg. 1
## Achillea millefolium 1
## Achillea species 1
## Hieracium subg. Pilosella 1
## 2 child taxa found in dataset, adapted
## TaxonUsageID TaxonName Freq.1 IsChildTaxonOfID IsChildTaxonOf
## 31 Achillea millefolium 1 27 Achillea millefolium agg.
## 27 Achillea millefolium agg. 1 60728 Achillea species
## Freq.2
## 1
## 1
## 1 child taxa found in dataset, adapted
## TaxonUsageID TaxonName Freq.1 IsChildTaxonOfID IsChildTaxonOf Freq.2
## 27 Achillea millefolium agg. 1 60728 Achillea species 1
## Number of taxa after validation: 12
##
## Warning: Critical Pseudonym(s) in dataset, please check
## to_check check_No check against TaxonUsageID
## Galium mollugo 2555 Galium mollugo auct. 27395
```



```
##           publishedInCitation
## BfN(Wisskirchen u. Haeupler 1998)
## Warning: Critical species in dataset, please check
##           to_check check_No           check against TaxonUsageID
## Dactylis glomerata      1843 Dactylis glomerata s. l.      26585
## Galium mollugo         2555 Galium mollugo s. l.      26777
##           publishedInCitation
## BfN(Wisskirchen u. Haeupler 1998)
## 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 subspecies 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 explained in the next chapter will 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$TaxonName <- species$TaxonName[match(obs.taxval$TaxonUsageID, species$TaxonUsageID)]
obs.taxval[, c("RELEV_NR", "COVER_CODE", "TaxonName")]
```

##	RELEV_NR	COVER_CODE	TaxonName
## 1	2	2b	Achillea species
## 2	2	4	Quercus robur
## 3	2	1	Quercus robur
## 4	2	1	Quercus robur
## 5	1	3	Achillea species
## 6	1	+	Achillea species
## 7	1	1	Achillea species
## 8	1	1	Acer pseudoplatanus

```
## 9      1      2      Acer pseudoplatanus
## 10     1      1      Picea abies
## 11     1      1      Achillea species
## 12     3      1      Armeria maritima subsp. halleri
## 13     3      1      Armeria maritima subsp. elongata
## 14     3      1      Armeria maritima subsp. halleri
## 15     1      1      Picea abies
## 16     3      1      Acorus calamus
## 17     1      1      Galium mollugo
## 18     1      1      Dactylis glomerata
## 19     1      1      Adonis aestivalis
## 20     1      1      Agrostis stolonifera var. palustris
## 21     2      1      Hieracium subg. Pilosella
## 22     2      3      Armeria maritima subsp. halleri
## 23     3      1      Hieracium subg. Pilosella
## 24     2      1      Picea abies
```

Critical Pseudonyms Taxon misapplication is maybe the greatest danger in using survey data. Known misapplications of names (.auct) are embedded within 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.

Coarsening to a specific taxonomic level If you want only taxa of e.g. level "species" in your analyses but no other taxonomic level, use `taxval(obs, ag='adapt', rank='SPE')`. All hierarchical levels below the species level (including the above specified monotypic subspecies) are set to species level in this case.

```
tmp <- taxval(obs.tax, refl = "GermanSL 1.2", ag = "adapt", rank = "FAM")
tmp$oldTaxon <- tax(obs.tax$TaxonUsageID, refl = "GermanSL 1.2")$TaxonName
tmp$newTaxon <- tax(tmp$TaxonUsageID, refl = "GermanSL 1.2")$TaxonName
```

```
head(tmp[, c("oldTaxon", "newTaxon")], 10)

##              oldTaxon              newTaxon
## 1      Achillea millefolium agg. Asteraceae species
## 2              Quercus robur      Fagaceae species
## 3              Quercus robur      Fagaceae species
## 4              Quercus robur      Fagaceae species
## 5      Achillea millefolium Asteraceae species
## 6 Achillea millefolium subsp. collina Asteraceae species
## 7      Achillea species Asteraceae species
## 8      Acer pseudoplatanus  Aceraceae species
## 9      Acer pseudoplatanus  Aceraceae species
## 10     Abies alpestris      Pinaceae species
```

Check `?taxval` and `args(taxval)` for more options.

5.2 Implementing other taxon views

If you wish to use another taxonomic concept (see Berendsohn (1995)) than the default, you can use a conversion table to change synonymy etc. to catch your needs.

```
newconcept <- taxval(obs, db = db, concept = "korneck1996")
```

6 Vegetation matrices

At the moment there exists 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 relevés 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 information about the performance of a species within the survey area, often given in some kind of alphanumeric code for cover percentage within the survey plot. Different code systems are 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* and the entries in the header data column **COVERSCALE**.

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

## reading observations ...

obs <- tv.coverperc(db, obs)

## Warning: 1 relevés without date. Not converted from factor to date format.

##
## Cover code used: 01 Braun/Blanquet (old)
##      SCH1 SCH2 SCH3 SCH4 SCH5 SCH6 SCH7
## code r   +   1   2   3   4   5
## perc 1   2   3  13  38  68  88
##
## Cover code used: 02 Braun/Blanquet (new)
##      SCH1 SCH2 SCH3 SCH4 SCH5 SCH6 SCH7 SCH8 SCH9
## code r   +   1   2m  2a  2b  3   4   5
## perc 1   2   3   4   8  18  38  68  88

head(obs)

##  RELEVÉ_NR TaxonUsageID 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          31          3     6         0      0      <NA>      0         01
## 6         1        20096          +     6         0      0 Schlenke      1         01
## COVER_PERC
## 1         18
## 2         68
## 3          3
## 4          3
## 5         38
## 6          2
```

A few simple possibilities for percentage cover transformations are directly included in the `tv.veg` code, 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 ...
## creating pseudo-species ...
## combining occurrences using type LAYER and creating vegetation matrix ...
## replacing species numbers with short names ...
## Reference list used: GermanSL 1.2
## [1] "AGRTS;P.6" "HIERSUG.6" "ACERPSE.5" "ACERPSE.6" "DACYGLO.6" "ACHICOL.6" "ARMEM-H"
## [8] "ARMEM-E" "ARMEM-H" "PICEABI.2" "PICEABI.3" "GALUMOL.6" "ACHI#MI" "ARMEM-H.6"
## [15] "HIERPIO" "ACHIMIL.6" "ACHIM-S.6" "PICEABI.1" "QUERROB.1" "QUERROB.2" "QUERROB.6"
## [22] "ACHI-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 (`data(lc.1)`) 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")
names(tv.veg(db, tax = FALSE, pseudo = comb, quiet = TRUE))

## reading observations ...
```

```
## Taxonomic reference list: GermanSL 1.2
## converting cover code ...

## Warning: 1 releves without date. Not converted from factor to date format.

## creating pseudo-species ...
## combining occurrences using type LAYER and creating vegetation matrix ...
## replacing species numbers with short names ...
## Reference list used: GermanSL 1.2
## [1] "AGRTS;P"      "HIERSUG"      "ACERPSE.Spring" "ACERPSE.Summer" "DACYGLO"
## [6] "ACHICOL"      "ARMEM-H"      "ARMEM-E"        "ARMEM-H"        "PICEABI"
## [11] "GALUMOL"      "ACHI#MI"      "ARMEM-H"        "HIERPIO"        "ACHIMIL"
## [16] "ACHIM-S"      "PICEABI"      "QUERROB"        "ACHI-SP"        "ACOR-SP"
## [21] "ADONAES"
```

```
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
## 1	3	0	3	13	3	0	0	3	6
## 2	0	3	0	0	0	0	38	0	3
## 3	0	3	0	0	0	3	6	0	0
## QUERROB									
## 1	0								
## 2	3								
## 3	0								

6.3 Combine species manually

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$TaxonUsageID[obs.tax$TaxonUsageID == 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$TaxonUsageID <- replace(obs.tax$TaxonUsageID, match(taxon.repl$old, obs.tax$TaxonUsageID),
  taxon.repl$new)
```

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

```
veg <- tv.veg("taxatest", quiet = TRUE)
```

```
## reading observations ...
## Taxonomic reference list: GermanSL 1.2
##
## Original number of names: 20
## 4 Synonyms found in dataset, adapted
## No taxa higher than ROOT found.
## 5 child taxa found in dataset, adapted
## 2 child taxa found in dataset, adapted
## 1 child taxa found in dataset, adapted
## Number of taxa after validation: 12
##
## Warning: Critical Pseudonym(s) in dataset, please check
## Warning: Critical species in dataset, please check
## converting cover code ...

## Warning: 1 releves without date. Not converted from factor to date format.

## creating pseudo-species ...
## combining occurrences using type LAYER and creating vegetation matrix ...
## replacing species numbers with short names ...
## Reference list used: GermanSL 1.2

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

## The following names are combined to the new name: QUERROB
## [1] "QUERROB" "QUERROB.Tree"
## AGRTS;P HIERSUG ACERPSE ACERPSE.Shrub DACYGLO ARMEM-E ARMEM-H GALUMOL PICEABI.Tree
## 1 3 0 3 13 3 0 0 3 6
## 2 0 3 0 0 0 0 38 0 3
## 3 0 3 0 0 0 3 6 0 0
## ACHI-SP ACOR-SP ADONAES QUERROB
## 1 43 0 3 0
## 2 18 0 0 72
## 3 0 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 Turboveg dBase format.

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

## Warning: 1 releves without date. Not converted from factor to date format.

##
## 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] X_COORD Y_COORD
## 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('X_COORD', 'Y_COORD')])
```

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

To prevent incompatibilities with Windows users who want to use Turboveg data but do not want to use VegetWeb data or are not able to install R package RMySQL, I excluded the VegetWeb functions from package vegdata. They can still be downloaded from the following website

```
source("http://geobot.botanik.uni-greifswald.de/download/r_package/vegetweb.r")
```

9 ESVeg, a XML exchange format for vegetation data

First preliminary functions to load vegetation data from ESVeg formatted XML files are now implemented. As soon as there is a software to use Veg-X, the international TDWG exchange standard for vegetation data, I will implement functions for this.

```
destfile <- tempfile()
download.file("http://geobot.botanik.uni-greifswald.de/download/data/T302.xml", destfile)
T302.site <- ESVeg.site(destfile)

## User defined plot attributes: BEOBACHTUN PLOTCODE PROJEKT NUTZUNG ERHEBER USER MODIFIED PLOT_ID P
T302.site <- T302.site[!is.na(T302.site$LONGITUDE), ]
```

10 Additional functions

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

10.1 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")
```

10.2 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.

```
elbaue <- tv.veg("elbaue")
elbaue.env <- tv.site("elbaue")
```

```
clust <- vector("integer", nrow(elbaue.env))
clust[elbaue.env$MGL < -50 & elbaue.env$SDGL < 50] <- 1 # dry sites, low deviation
clust[elbaue.env$MGL < -50 & elbaue.env$SDGL >= 50] <- 2 # dry sites, high deviation
clust[elbaue.env$MGL >= -50 & elbaue.env$SDGL >= 50] <- 3 # wet sites, high deviation
clust[elbaue.env$MGL >= -50 & elbaue.env$SDGL < 50] <- 4 # wet sites, low deviation
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). Additionally we can use 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) to order the syntaxon table. Together with Ellenberg indicator values with will get a comprehensive view into our data.

```
require(indicspecies)

## Loading required package: indicspecies
## Loading required package: permute

traits <- tv.traits()

##
## Changing character fields into logical, integer or numericals if appropriate:
##
## Class of OEK_L changed to integer
## Class of OEK_T changed to integer
## Class of OEK_K changed to integer
## Class of OEK_F changed to integer
## Class of OEK_R changed to integer
## Class of OEK_N changed to integer
## Class of OEK_S changed to integer
## Class of Mahdvertra changed to integer
## Class of Weidevertr changed to integer
## Class of Trittvertr changed to integer
## Class of Futterwert changed to integer
## Class of Futter_Dam changed to integer

trait <- data.frame(EIV_F = traits$OEK_F, EIV_N = traits$OEK_N)
rownames(trait) <- traits$ABBREVIAT
st <- syntab(elbaue, clust, mupa = TRUE, fullnames = TRUE)

## Number of clusters: 4
## Cluster frequency 7 10 5 11
## Reference list used: GermanSL 1.2
```



```
print(st, limit = 30, trait = trait)
```

##		dry.ld	dry.hd	wet.hd	wet.ld	index	stat	p.value	EIV_F	EIV_N
##	Cirsium arvense	43	.	.	.	1	0.64	0.010	NA	7
##	Deschampsia cespitosa	57	.	.	.	1	0.72	0.005	7	3
##	Euphorbia esula	43	.	.	.	1	0.65	0.005	4	NA
##	Galium verum agg.	71	.	.	.	1	0.83	0.005	4	3
##	Lathyrus pratensis	43	.	.	.	1	0.59	0.025	6	6
##	Vicia tetrasperma	57	.	.	.	1	0.71	0.010	5	5
##	Alopecurus geniculatus	.	.	60	.	3	0.65	0.030	8	7
##	Rorippa amphibia	.	.	60	.	3	0.77	0.005	10	8
##	Caltha palustris	.	.	.	36	4	0.60	0.035	9	6
##	Agrostis canina	.	.	.	36	4	0.60	0.035	9	2
##	Carex vesicaria	.	.	.	55	4	0.74	0.010	9	5
##	Carex acuta	.	.	.	82	4	0.87	0.005	9	4
##	Ranunculus flammula	.	.	.	55	4	0.74	0.005	9	2
##	Carex praecox	43	70	.	.	5	0.77	0.005	3	4
##	Elymus repens	57	90	.	.	5	0.87	0.005	NA	7
##	Alopecurus pratensis	71	90	.	.	5	0.88	0.005	6	7
##	Rumex thyrsoiflorus	43	60	.	.	5	0.73	0.020	3	4
##	Taraxacum sect. Ruderalia	57	60	.	.	5	0.72	0.015	5	8
##	Cardamine pratensis	43	.	.	55	7	0.69	0.020	6	NA
##	Glyceria maxima	.	.	80	45	10	0.75	0.020	10	9
##	Sium latifolium	.	.	40	45	10	0.66	0.045	10	7

11 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.

11.1 Plot coordinates of vegetation relevés into an interactive Google Map

We will use data from VegetWeb (Schmitt, Fartmann, Hoelzel 2010 Phytosociology and ecology of *Gladiolus palustris* in southern Bavaria, Tuexenia 30, p. 105-128.) to make an interactive map of plot locations. The data has already been loaded through the ESVeg functionality 9.

A database with 140 plots with plot locations.

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

```
library(rgdal)
library(googleVis)
coord <- data.frame(lat = T302.site$LATITUDE, long = T302.site$LONGITUDE)
coordinates(coord) <- c("long", "lat")
proj4string(coord) <- CRSargs(CRS("+init=epsg:31468")) # GK, 4. Stripe
coord <- spTransform(coord, CRS("+init=epsg:4326")) # WGS 84, geographical coordinates, decimal deg
T302.site$long <- coordinates(coord)[, 1]
T302.site$lat <- coordinates(coord)[, 2]
```

To give some information in the interactive map, we will Provide Hyperlink Tips:

```
T302.site$loc <- paste(T302.site$LATITUDE, T302.site$LONGITUDE, sep = ":")
T302.site$tip <- paste(paste("Releve_NR:", T302.site$plotCode), paste("Table:", T302.site$referenceT
```

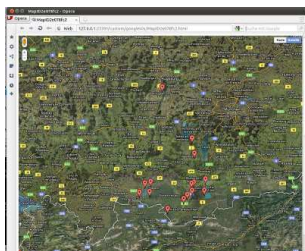


Figure 2: Spatial distribution of vegetation plots from a VegetWeb project. The map is interactive and scalable.

```
paste("Nr. in table:", T302.site$referencePlot), paste("Date:", T302.site$obsEndDate),
paste("Landuse:", T302.site$NUTZUNG), paste("Author:", T302.site$ERHEBER), paste("Locality:",
T302.site$LOKALIT_T), paste("Longitude:", T302.site$LONGITUDE), paste("Latitude:",
T302.site$LATITUDE), paste("geogr. Uncertainty:", T302.site$GENAUIGKEI), sep = "<BR>")
```

and the produced map will open in your web browser.

```
places <- gvisMap(T302.site[, c("loc", "tip")], "loc", "tip", options = list(showTip = TRUE,
showLine = FALSE, enableScrollWheel = TRUE, mapType = "hybrid", useMapTypeControl = TRUE,
width = 1000, height = 800))
plot(places)
```

11.2 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).

```
## Data analyses
library(vegan)

## This is vegan 2.0-8

veg.nmms <- metaMDS(elbaue, distance = "bray", trymax = 5, autotransform = FALSE, noshare = 1,
expand = TRUE, trace = 2)
mT.F <- meanTraits("OEK_F", elbaue)
mT.N <- meanTraits("OEK_N", elbaue)
env <- envfit(veg.nmms, data.frame(mT.F, mT.N))
```

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

```
library(labds)

## Loading required package: mgcv
## This is mgcv 1.7-24. For overview type 'help("mgcv-package")'.
## Loading required package: MASS
##
## Attaching package: 'labds'
```

```
##
## Das folgende Objekt ist maskiert from 'package:stats':
##
##      density

library(akima)
color = function(x) rev(topo.colors(x))
nm.ds.plot <- function(ordi, site, var1, var2, disp, plottitle = "NMDS", env = NULL, ...) {
  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)
      if (!is.null(env))
        plot(env, col = "red")
    }, ...)
}
```

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 in 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.
- 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. & Schaminee, 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 fuer 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.

```
nmds.plot(veg.nmds, elbaue.env, disp = "species", var1 = "MGL", var2 = "SDGL", env = env, plottitle =
```

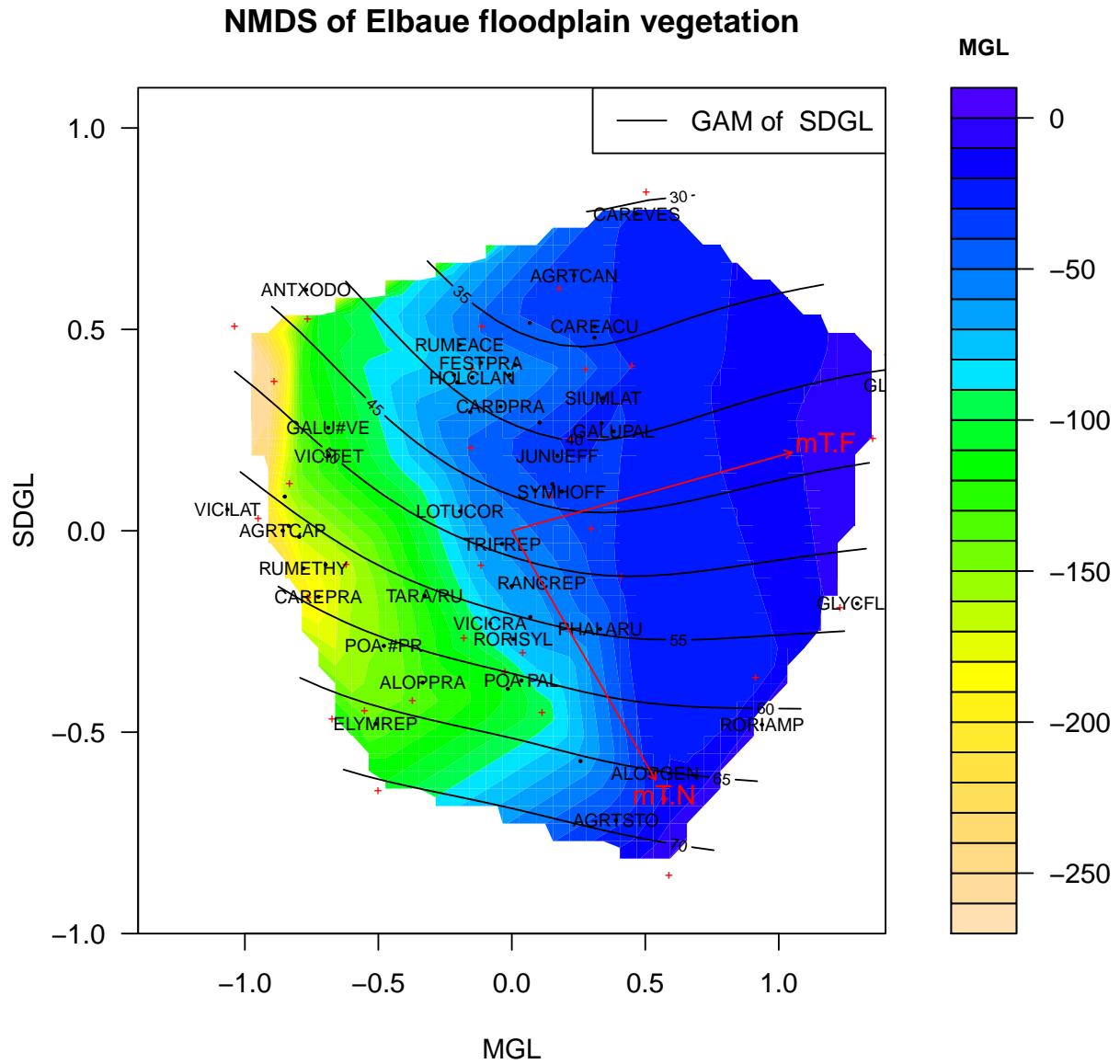


Figure 3: Non-metric multidimensional scaling of the elbaue vegetation data with an overlay of mean ground-water table (colors) and standard deviation of groundwater level fluctuations (lines). Arrows show direction of increasing mean Ellenberg F resp. N

Leyer, I. & Wesche, K. (2007). *Multivariate Statistik in der Oekologie*. Springer, Berlin.

Oksanen, J., Kindt, R., Legendre, P., O'Hara, B., Simpson, G.L. & Stevens, M.H.H. (2008). *vegan: Community Ecology Package*.