Vegetation data access and taxonomic harmonization version 0.6-6

Florian Jansen

January 24, 2014

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 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 conevntions).

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 0.6-6
```

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/RtmpBz5cRg"
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()
```

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                        VernacularName
18              27    ACHI#MI               Achillea millefolium agg.         Artengruppe Wiesen-Schafgarbe
20              31    ACHIMIL                    Achillea millefolium        Gew+Âhnliche Wiesen-Schafgarbe
21              32    ACHIM-M Achillea millefolium subsp. millefolium Gew+Âhnliche Wiesen-Schafgarbe i.e.S.
22              33    ACHIM-S    Achillea millefolium subsp. sudetica              Sudeten-Wiesenschafgarbe
8680         20096    ACHICOL     Achillea millefolium subsp. collina                                  <NA>
8681         20097    ACHIPAN   Achillea millefolium subsp. pannonica                                  <NA>
8682         20098    ACHIPAN        Achillea millefolium var. lanata                                  <NA>
13222        26082    ACHIMIL         Achillea millefolium var. firma                                  <NA>
26250        90019    ACHI*AB     Achillea millefolium agg. x nobilis                                  <NA>
26251        90020    ACHIM*P        Achillea millefolium x pannonica                                  <NA>
      SYNONYM TaxonConceptID
18      FALSE             27
20      FALSE             31
21      FALSE             32
22      FALSE             33
8680     TRUE             29
8681     TRUE             34
8682     TRUE             34
13222    TRUE             31
26250    TRUE          90028
26251   FALSE          90020
```

The GermanSL 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()`) or a temporyra folder, 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` can use the given species name (with option strict=FALSE also name parts), or 7 letter abbreviation or the 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         TaxonConcept
20           31    ACHIMIL Achillea millefolium     L.   FALSE             31 Achillea millefolium
                  VernacularName TaxonRank GRUPPE    FAMILIE IsChildTaxonOfID            IsChildTaxonOf
20 Gewöhnliche Wiesen-Schafgarbe       SPE      S Asteraceae               27 Achillea millefolium agg.
                            NACHWEIS                       AccordingTo HYBRID BEGRUEND EDITSTATUS
20 BfN(Wisskirchen u. Haeupler 1998) BfN(Wisskirchen u. Haeupler 1998)   <NA>     <NA>        BfN

tax("Achylleus x millefoliae", simplify = TRUE, hybrid = TRUE)

Reference list used: GermanSL 1.2
      TaxonUsageID LETTERCODE                       TaxonName                        VernacularName SYNONYM
18              27    ACHI#MI            Achill millefol agg.         Artengruppe Wiesen-Schafgarbe   FALSE
20              31    ACHIMIL                 Achill millefol        Gew+Âhnliche Wiesen-Schafgarbe   FALSE
21              32    ACHIM-M Achill millefol subsp. millefol Gew+Âhnliche Wiesen-Schafgarbe i.e.S.   FALSE
22              33    ACHIM-S  Achill millefol subsp. sudetic              Sudeten-Wiesenschafgarbe   FALSE
8680         20096    ACHICOL   Achill millefol subsp. collin                                  <NA>    TRUE
8681         20097    ACHIPAN  Achill millefol subsp. panonic                                  <NA>    TRUE
8682         20098    ACHIPAN      Achill millefol var. lanat                                  <NA>    TRUE
13222        26082    ACHIMIL       Achill millefol var. firm                                  <NA>    TRUE
26250        90019    ACHI*AB      Achill millefol agg. nobil                                  <NA>    TRUE
26251        90020    ACHIM*P         Achill millefol panonic                                  <NA>   FALSE
      TaxonConceptID                       originalTaxonName
18                27               Achillea millefolium agg.
20                31                    Achillea millefolium
21                32 Achillea millefolium subsp. millefolium
22                33    Achillea millefolium subsp. sudetica
8680              29     Achillea millefolium subsp. collina
8681              34   Achillea millefolium subsp. pannonica
8682              34        Achillea millefolium var. lanata
13222             31         Achillea millefolium var. firma
26250          90028     Achillea millefolium agg. x nobilis
26251          90020        Achillea millefolium x pannonica
```

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"                 "Elytrigia repens var. caesia"
[4] "Elytrigia repens var. littoralis" "Elytrigia repens var. repens"

syn("Elytrigia repens")

Name swarm of  :
[1] TaxonUsageID TaxonName    SYNONYM      EDITSTATUS
<0 rows> (or 0-length row.names)
```

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")

Parents of Achillea millefolium ( 31 ):
 TaxonUsageID                TaxonName TaxonRank IsChildTaxonOfID GENERATION
           27 Achillea millefolium agg.       AGG            60728          1
        60728                  Achillea       GAT            60463          2
        60463                Asteraceae       FAM            60415          3
        60415                 Asterales       ORD            60079          4
        60079                 Asteridae       UKL            60071          5
        60071             Magnoliopsida       KLA            60049          6
        60049            Magnoliophytina       UAB            60000          7
        60000             Spermatophyta       ABT            94419          8
        94419           "Gefaesspflanze"       AG2                0          9
            0        "Gruenliches etwas"      ROOT                0         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"
```

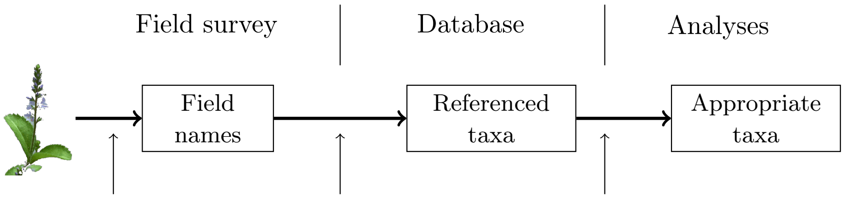
Defines the vegetation database name according to the name of the Turboveg database directory name

```
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("RELEVE_NR", "TaxonUsageID", "COVER_CODE", "LAYER", "TaxonName")])

  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
```



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

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", sink = FALSE)


Original number of names: 20
```

```
4 Synonyms found in dataset, adapted
 TaxonUsageID                               TaxonName Freq.1 TaxonConceptID                    TaxonConcept
        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
Check for monotypic taxa: 1. round.
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
  66142        FAM     61329 Acorus species        GAT
Check for monotypic taxa: 2. round.
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            IsChildTaxonOf Freq.2
           29                     Achillea collina     NA               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
2 child taxa found in dataset, adapted
 TaxonUsageID                 TaxonName Freq.1 IsChildTaxonOfID            IsChildTaxonOf Freq.2
           31      Achillea millefolium      1               27 Achillea millefolium agg.      1
           27 Achillea millefolium agg.      1            60728          Achillea species      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 harmonisation: 12

Warning: Potential pseudonyms in dataset, please check.       to_check check_No        check against TaxonUsageID
 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 TaxonUsageID                       AccordingTo
 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)
Number of taxa after validation: 12
```

The database contains 20 different names in the beginning.

Synonyms 4 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 *Asteracea* 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$OriginalName <- obs.tax$TaxonName
obs.taxval[, c("RELEVE_NR", "COVER_CODE", "TaxonName", "OriginalName")]

   RELEVE_NR COVER_CODE                           TaxonName                            OriginalName
1          2         2b                            Achillea               Achillea millefolium agg.
2          2          4                       Quercus robur                           Quercus robur
3          2          1                       Quercus robur                           Quercus robur
4          2          1                       Quercus robur                           Quercus robur
5          1          3                            Achillea                    Achillea millefolium
6          1          +                            Achillea     Achillea millefolium subsp. collina
7          1          1                            Achillea                                Achillea
8          1          1                 Acer pseudoplatanus                     Acer pseudoplatanus
9          1          2                 Acer pseudoplatanus                     Acer pseudoplatanus
10         1          1                         Picea abies                         Abies alpestris
11         1          1                            Achillea    Achillea millefolium subsp. sudetica
12         3          1     Armeria maritima subsp. halleri Armeria maritima subsp. bottendorfensis
13         3          1    Armeria maritima subsp. elongata        Armeria maritima subsp. elongata
14         3          1     Armeria maritima subsp. halleri         Armeria maritima subsp. halleri
15         1          1                         Picea abies                         Abies alpestris
16         3          1                      Acorus calamus                               Acoraceae
17         1          1                      Galium mollugo                          Galium mollugo
18         1          1                  Dactylis glomerata                      Dactylis glomerata
19         1          1                   Adonis aestivalis                       Adonis aestivalis
20         1          1 Agrostis stolonifera var. palustris     Agrostis stolonifera var. palustris
21         2          1           Hieracium subg. Pilosella                     Hieracium pilosella
22         2          3     Armeria maritima subsp. halleri                 Armeria bottendorfensis
23         3          1           Hieracium subg. Pilosella               Hieracium subg. Pilosella
24         2          1                         Picea abies                             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", sink = FALSE)
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
2                        Quercus robur   Fagaceae
3                        Quercus robur   Fagaceae
4                        Quercus robur   Fagaceae
5                 Achillea millefolium Asteraceae
6  Achillea millefolium subsp. collina Asteraceae
7                             Achillea Asteraceae
8                  Acer pseudoplatanus  Aceraceae
9                  Acer pseudoplatanus  Aceraceae
10                     Abies alpestris   Pinaceae
```

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

5.2 Implementing other taxon views

If you wish to use another taxonomic concept (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", sink = FALSE)
```

However, writing files which contain the necessary changes is tidious and other taxon concept based systems have to be developed and used to cover the challenges of different taxon views.

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 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 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)

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

tail(obs)
```

```
   RELEVE_NR TaxonUsageID COVER_CODE LAYER DET_CERT SEASON MICROREL FLOWER COVERSCALE COVER_PERC
19         1           76          1     6        0      0     <NA>      0         01          3
20         1        10024          1     6        0      0     <NA>      0         01          3
21         2         2923          1     0        0      0     <NA>      0         02          3
22         2        27309          3     6        0      0     <NA>      0         02         38
23         3        12273          1     6        0      0     <NA>      0         01          3
24         2         4269          1     1        0      0     <NA>      0         02          3
```

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*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

```r
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"   "ARMEM-E"
 [9] "ARMEM-H"   "PICEABI.2" "PICEABI.3" "GALUMOL.6" "ACHI#MI"   "ARMEM-H.6" "HIERPIO"   "ACHIMIL.6"
[17] "ACHIM-S.6" "PICEABI.1" "QUERROB.1" "QUERROB.2" "QUERROB.6" "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.

```r
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 ...
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"        "ACHICOL"
 [7] "ARMEM-H"        "ARMEM-E"        "ARMEM-H"        "PICEABI"        "GALUMOL"        "ACHI#MI"
[13] "ARMEM-H"        "HIERPIO"        "ACHIMIL"        "ACHIM-S"        "PICEABI"        "QUERROB"
[19] "ACHI-SP"        "ACOR-SP"        "ADONAES"
```

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

4 Synonyms found in dataset, adapted.
1 monotypic taxa found in dataset, set to species rank.
1 monotypic taxa found in dataset, set to species rank.
5 child taxa found in dataset, adapted.
2 child taxa found in dataset, adapted.
1 child taxa found in dataset, adapted.
Warning:  Potential pseudonyms in dataset, please check.
Warning:  Critical species in dataset, please check
   Information is written to /tmp/RtmpBz5cRg/file3c685338edd4txt.
```

```
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 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.
1 monotypic taxa found in dataset, set to species rank.
1 monotypic taxa found in dataset, set to species rank.
5 child taxa found in dataset, adapted.
2 child taxa found in dataset, adapted.
1 child taxa found in dataset, adapted.
Warning:  Potential pseudonyms in dataset, please check.
Warning:  Critical species in dataset, please check
```

```
Number of taxa after validation: 12

Information is written to /tmp/RtmpBz5cRg/file3c684f8a5dd9txt.

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

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 ACHI-SP ACOUCAL ADONAES
1       3       0       3            13       3       0       0       3            6      43       0       3
2       0       3       0             0       0       0      38       0            3      18       0       0
3       0       3       0             0       0       3       6       0            0       0       3       0
  QUERROB
1       0
2      72
3       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)


 The following columns contain no data and are omitted
 [1] REFERENCE  TABLE_NR   NR_IN_TAB  PROJECT    AUTHOR     SYNTAXON   UTM        ALTITUDE   EXPOSITION
[10] 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 COV_WATER
[10] COV_ROCK   TREE_HIGH  TREE_LOW   SHRUB_HIGH SHRUB_LOW  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 `http://www.vegetweb.de`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 address:

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

9 ESVeg, a XML exchange format for vegetation data

Some preliminary functions to load vegetation data from ESVeg formatted XML files are now implemented.

```
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 PLOTCODE_1 PROJEKT_1 FL__CHE

T302.site <- T302.site[!is.na(T302.site$LONGITUDE), ]
```

As soon as there is a software to use `Veg-X`, the international TDWG exchange standard for vegetation data, I will try to implement functions for this.

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.compRefl` might be a starting point, comparing species numbers and/or species names of both lists.

```
tv.compRefl("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")

5 Synonyms found in dataset, adapted.
1 monotypic taxa found in dataset, set to species rank.
Warning:  Critical species in dataset, please check
   Information is written to /tmp/RtmpBz5cRg/file3c68795311edtxt.

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 syntaxonical 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 ofOEK_Lchanged to integer
    Class ofOEK_Tchanged to integer
    Class ofOEK_Kchanged to integer
    Class ofOEK_Fchanged to integer
    Class ofOEK_Rchanged to integer
    Class ofOEK_Nchanged to integer
    Class ofOEK_Schanged to integer
    Class ofMahdvertrachanged to integer
    Class ofWeidevertrchanged to integer
    Class ofTrittvertrchanged to integer
    Class ofFutterwertchanged to integer
    Class ofFutter_Damchanged 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      .      .      9     1 0.64   0.025    NA     7
Deschampsia cespitosa      57      .      .     18     1 0.72   0.020     7     3
Euphorbia esula            43      .      .      .     1 0.65   0.005     4    NA
Galium verum agg.          71     20      .      .     1 0.83   0.005     4     3
Vicia tetrasperma          57     10      .      .     1 0.71   0.005     5     5
Alopecurus geniculatus      .     20     60      9     3 0.65   0.040     8     7
Rorippa amphibia            .      .     60      9     3 0.77   0.005    10     8
Caltha palustris            .      .      .     36     4 0.60   0.020     9     6
Agrostis canina             .      .      .     36     4 0.60   0.045     9     2
Carex vesicaria             .      .      .     55     4 0.74   0.010     9     5
Carex acuta                14      .     40     82     4 0.87   0.005     9     4
Ranunculus flammula         .      .      .     55     4 0.74   0.015     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     20     36     5 0.88   0.005     6     7
Rumex thyrsiflorus         43     60      .      .     5 0.73   0.005     3     4
Cardamine pratensis        43     10      .     55     7 0.69   0.035     6    NA
Glyceria maxima             .      .     80     45    10 0.75   0.035    10     9
Sium latifolium             .      .     40     45    10 0.66   0.025    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 urlhttp://www.vegetweb.deVegetWeb (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.

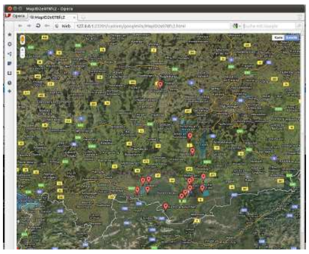


Figure 2: Spatial distribution of vegetation plots from a VegetWeb project. The map is interactive and scalable.

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)
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 degrees
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$plotCode), paste("Table:", T302.site$referenceTable), paste("Nr. in tab
    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)

Loading required package:  lattice
This is vegan 2.1-42

veg.nmds <- 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.nmds, 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(labdsv)

Loading required package:  mgcv
Loading required package:  nlme
This is mgcv 1.7-27.  For overview type 'help("mgcv-package")'.
Loading required package:  MASS
    Attaching package:  'labdsv'
    Das folgende Objekt ist maskiert from 'package:stats':
        density

library(akima)
color = function(x) rev(topo.colors(x))
nmds.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.

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

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

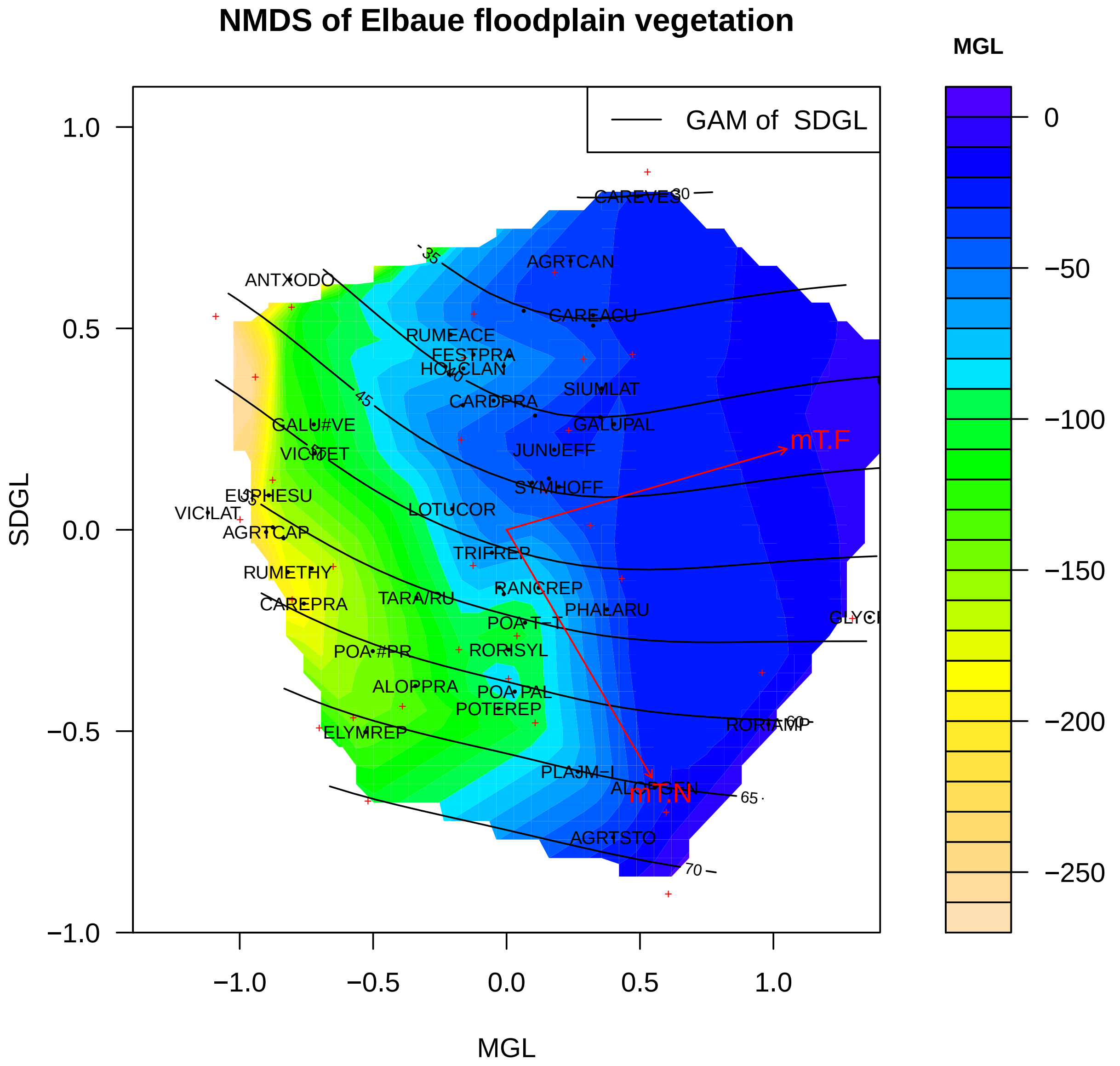


Figure 3: Non-metric multidimensional scaling of the elbaue vegetation data with an overlay of mean groundwater table (colors) and standard deviation of groundwater level fluctuations (lines). Arrows show direction of increasing mean Ellenberg F resp. N

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