Package: bio.infer (via r-universe)

May 31, 2026

Type Package

Title Predict Environmental Conditions from Biological Observations

Version 1.3-6

Date 2020-09-01

Author Lester L. Yuan

Maintainer Lester L. Yuan `<yuan.lester@epa.gov>`

Description Imports benthic count data, reformats this data, and computes environmental inferences from this data.

Depends R (>= 2.10), tcltk

LazyLoad FALSE

LazyData FALSE

License GPL (>= 2)

NeedsCompilation no

Repository https://lesteryuan.r-universe.dev

Date/Publication 2020-09-02 12:20:07 UTC

RemoteUrl https://github.com/cran/bio.infer

RemoteRef HEAD

RemoteSha 7bb78e99d6bcfc29b3c6b6ae7ee9866f8df2658b

Contents

`bio.infer-package` *Scripts for computing biological inferences*

Description

Reformats raw benthic count files, combines them with existing taxon-environment relationships, and computes inferences of environmental conditions.

Details

Package:	bio.infer
Type:	Package
Version:	1.0
Date:	2006-10-31
License:	GPL version 2 or newer

Use this package to import benthic count data, reformat the data, and compute inferences of environmental conditions using taxon-environment relationships.

Typically, the user will start by loading a local benthic count data file (see `bcnt.OR` for an example of the required format). This benthic count file will be merged with standardized taxonomic hierarchy information using `get.taxonomic`. Then, operational taxonomic units will be assigned, based on the number of occurrences of different taxa and based on the identity of taxa for which environmental preferences have been quantified. (See `get.otu` for the OTU assignment script, and `coef.west.wt` for an example file of environmental preferences.)

Once OTUs have been assigned, the benthic count file is reformatted as a site-taxon matrix (`makess`). This file can then be used to compute biological inferences (`mlsolve`).

This package was developed in Windows but should be fully functional in other operating systems.

Author(s)

Lester L. Yuan

`bcnt.emapw` *Benthic count data for the western United States*

Description

Benthic count data for streams in the western United States.

Usage

```
data(bcnt.emapw)
```

Format

Three fields are included.

`ID.NEW`	Unique sample code
`Taxaname`	Name of the observed taxon
`Abundance`	Abundance of the observed taxon

Source

U.S. Environmental Protection Agency Environmental Monitoring and Assessment Program

Examples

```
data(bcnt.emapw)
```

`bcnt.OR` *Benthic count data for western Oregon*

Description

Benthic count data for streams in western Oregon.

Usage

```
data(bcnt.OR)
```

Format

Three fields are included.

`SVN` Unique sample code

`Taxon` Name of the observed taxon

`CountValue` Abundance of the observed taxon

Source

Oregon Department of Environmental Quality

Examples

```
data(bcnt.OR)
```

`bcnt.otu.OR` *Benthic count data with OTU*

Description

Benthic count data from western Oregon with operational taxonomic units assigned by `get.otu`.

Usage

```
data(bcnt.otu.OR)
```

Format

Five fields are included.

- `SVN` Unique sample identifier
- `Taxon` Original taxon name, same as in `bcnt.OR`
- `CountValue` Taxon abundance
- `TNAME` Taxon name after processing by `get.taxonomic`
- `OTU` Operational taxonomic unit. Taxon name selected with associated taxon-environment data and selected to eliminate double-counting

Source

Oregon Department of Environmental Quality

Examples

```
data(bcnt.otu.OR)
```

`bcnt.tax.OR` *Benthic count with taxonomic hierarchy*

Description

Benthic count data from western Oregon with full taxonomic hierarchy assigned by `get.taxonomic`.

Usage

```
data(bcnt.tax.OR)
```

Source

Oregon Department of Environmental Quality

Examples

```
data(bcnt.tax.OR)
```

`coef.east.sed` *Regression coefficients for eastern U.S. sediment*

Description

Regression coefficients describing relationships between different taxa and percent sands/fines.

Usage

```
data(coef.west.wt)
```

Format

A list of with following four elements is provided.

- `tnames` Character vector with the names of each taxon modelled.
- `csave` Matrix of regression coefficients for each taxon. Number of rows is the same as the number of taxa modelled and the number of columns is 6.
- `xvar` Character vector with the names of variables modelled.
- `xlims` List of limits for each of the environmental variables.
- `form` Regression formula used to estimate coefficients.

Details

These regression coefficients were computed using logistic regression.

Source

Yuan, L.L.

Examples

```
data(coef.east.sed)
```

`coef.west.wt` *Weighted regression coefficients for western U.S.*

Description

Regression coefficients describing relationships between different taxa and percent sands/fines and stream temperature.

Usage

```
data(coef.west.wt)
```

Format

A list of with following four elements is provided.

`tnames` Character vector with the names of each taxon modelled.

`csave` Matrix of regression coefficients for each taxon. Number of rows is the same as the number of taxa modelled and the number of columns is 6.

`xvar` Character vector with the names of variables modelled.

`xlims` List of limits for each of the environmental variables.

`form` Regression formula used to estimate coefficients.

Details

These regression coefficients were computed using logistic regression weighted by the number of stream miles represented by each sample.

Source

Yuan, L.L.

Examples

```
data(coef.west.wt)
```

`correct.taxanames` *Correct unrecognized taxon names*

Description

Prompts user to correct unrecognized taxon names

Usage

```
correct.taxanames(tname.old, get.tax.env)
```

Arguments

`tname.old`	Character vector of unrecognized names
`get.tax.env`	Environment for ITIS taxon table

Details

Prompts user to enter corrections to names via a tcltk dialog box.

Value

Character vector of corrected names

Author(s)

Lester L. Yuan

`envdata.emapw` *Environmental data from the western United States*

Description

Environmental data for stream sampled in the western United States.

Usage

```
data(envdata.OR)
```

Format

A data frame with 1674 observations on the following 8 variables.

`ID.NEW`	Unique sample code
`ntl`	Total nitrogen concentration
`no3`	Nitrate concentration
`ptl`	Total phosphorus concentration
`sed`	Percent sands and fines
`tempres`	Corrected temperature
`STRMTEMP`	Grab temperature
`jday`	Julian sampling day

Source

U.S. Environmental Protection Agency Environmental Monitoring and Assessment Program.

Examples

```
data(envdata.emapw)
```

`envdata.OR` *Environmental data from western OR*

Description

Environmental data for stream sampled in western Oregon.

Usage

```
data(envdata.OR)
```

Format

A data frame with 245 observations on the following 13 variables.

`STRM.ID`	Unique sample code
`jday`	Sampling day
`lon`	Longitude in decimal degrees
`lat`	Latitude in decimal degrees
`sed`	Percent sands and fines
`elev`	Log-transformed elevation
`slope`	Slope
`area`	Log-transformed catchment area
`sed.log`	Log-transformed percent sands and fines
`temp`	Grab stream temperature
`temp.avg`	Seven day average maximum temperature
`elev.ut`	Elevation
`sp`	Estimate of unit stream power

Source

Oregon Department of Environmental Quality

Examples

```
data(envdata.OR)
```

`flist.data` *List available data*

Description

Lists available data in the package

Usage

```
flist.data()
```

Value

List of data files.

Author(s)

Lester L. Yuan

`flist.match` *Selects data files matching search string*

Description

Select data files matching search string

Usage

```
flist.match(pattern = NULL)
```

Arguments

`pattern`	Character string to be matched.

Value

List of matched data files.

Author(s)

Lester L. Yuan

`formtomat` *Analyze formula string*

Description

Analyzes formula string that defines the regression models for taxon-environment relationships. Returns a numerical representation of the formula and the derivative of the formula that can be used by `mlsolve`.

Usage

```
formtomat(a, xvar)
```

Arguments

`a`	Character string defining regression formula.
`xvar`	Character vector with variable names.

Details

Internal function to `mlsolve`.

Value

List of transformations.

Author(s)

Lester L. Yuan

`get.dupe.sel` *Get duplicate selection*

Description

Prompt user to select correct version from duplicated entries

Usage

```
get.dupe.sel(sumstr)
```

Arguments

`sumstr`	Character vector summarizing duplicated taxon names

Details

Prompts the user via a tcltk window to select appropriate name from duplicate taxon names.

Value

Returns integer indicating which names are selected.

Author(s)

Lester L. Yuan

`get.otu` *Defines operational taxonomic units (OTUs)*

Description

`get.otu` generates OTUs for benthic count data using a simple algorithm that is designed to maximize the amount of taxonomic information retained across the entire data set, while eliminating redundant taxonomic identifications.

Usage

```
get.otu(bcnt, optlist = NULL, ndc = TRUE, outputFile = FALSE,
gui = FALSE)
```

Arguments

`bcnt`	benthic count data frame that includes the full taxonomic hierarchy for each taxon. Typically, `bcnt` is the output from `get.taxonomic`.
`optlist`	vector of taxon names for which trait or tolerance value is available. Specify `optlist = NULL` for cases in which OTU are computed without regard for any additional taxon specific information, e.g. for developing a RIVPACS model.
`ndc`	Logical flag indicating whether OTU should be computed that eliminate double-counting of taxa.
`outputFile`	Logical indicating whether to output summary information
`gui`	Logical indicating whether to use gui interface.

Details

`get.otu` makes decisions about OTU according to the following rule. The number samples in which a given taxonomic group (e.g. Baetidae) is observed, is compared to the number of samples in which members of the that same group are identified to a higher taxonomic level (e.g., Baetis, Fallceon, Acentrella, etc.). If more samples are observed with the coarser identification, then all identifications are downgraded to the coarse identification. If more samples are observed at the finer level of identification, then those sample are retained, and all identifications at the coarser level are omitted. Before deciding on OTU assignments, the script first reviews the species names that are listed in the benthic count file and matches them with the species names provided in `optlist`.

Value

Final OTU decisions are returned as tab-delimited text file, `sum.otu.txt`, which is best viewed using a spreadsheet. A new benthic count file is also returned in R, including the original benthic count information and a new field specifying the OTU designation for each taxon. Manual revisions to the assigned OTUs can be recorded in `sum.otu.txt` and loaded with `load.revised.otu`

Author(s)

Lester L. Yuan

See Also

`get.taxonomic`

Examples

```
data(bcnt.tax.OR)
data(coef.west.wt)
bcnt.otu.OR <- get.otu(bcnt.tax.OR, coef.west.wt)
```

`get.taxon.names` *Get taxon names from benthic count file*

Description

Get taxon names from benthic count file

Usage

```
get.taxon.names(bcnt)
```

Arguments

`bcnt`	Benthic count file with taxon names in second column

Details

Checks to see if second column is factor or character.

Value

Returns character vector of taxon names.

Author(s)

Lester L. Yuan

`get.taxonomic` *Merge benthic count data with the ITIS taxonomy table*

Description

`get.taxonomic` merges the taxon names provided in a benthic count data file with a standardized taxonomy table. The purpose of this function is to provide the complete taxonomic hierarchy for each taxon.

Usage

```
get.taxonomic(bcnt)
```

Arguments

`bcnt`	A benthic count data frame with the following three fields: a sample identifier, taxon name, and abundance. The three fields must be ordered as listed. Sample identifier is typically a character string that is a unique identifier for a given sample. Taxon name is a character string that specifies the name of the taxon observed in the sample, and abundance is the number of individuals of that taxon that were observed.

Details

`get.taxonomic` attempts to match taxon names provided in a benthic count data file with taxa listed in ITIS. The script automatically corrects for the most common conventions in taxonomic naming. For example, taxon names are converted to all capital letters, and trailing strings such as "SP." and "SPP." are deleted.

The script lists taxa remaining that are not matched to entries in the ITIS database using `fix`. The user can then enter a corrected spelling for the taxon name.

The script provides a final summary of the unmatched taxon names, and the user can continue revising names, or complete the program.

Value

A final taxonomic table is returned as tab-delimited text in the file `sum.tax.table.txt`. This file is best viewed using a spreadsheet. Within R, the script returns a benthic count file with the original taxon name associated with a full taxonomic hierarchy for that taxon. The assigned species names can be modified by hand in `sum.tax.table.txt` and the revised information loaded with `load.revised.species`.

Author(s)

Lester L. Yuan

See Also

`itis.ttable`

Examples

```
data(bcnt.OR)
data(itis.ttable)
## Not run:
bcnt.tax.OR <- get.taxonomic(bcnt.OR)

## End(Not run)
```

`get.valid.names` *Get taxon names from benthic count file*

Description

Identifies valid names from parsed name matrix.

Usage

```
get.valid.names(df.parse, get.tax.env)
```

Arguments

`df.parse`	Parsed name matrix
`get.tax.env`	Environment for ITIS taxon table

Details

Searches for taxon names in ITIS and splits name matrix into list with first element being the matrix of valid names and the second element being the matrix of invalid names.

Value

List of two taxon name matrices.

Author(s)

Lester L. Yuan

`in.ITIS` *Check to see if taxon name is in ITIS*

Description

Checks to see if taxon names are listed in the ITIS taxon table.

Usage

```
in.ITIS(df.parse, get.tax.env, col.sel = NULL)
```

Arguments

`df.parse`	Character vector or matrix character vectors of names that need to be checked for presence in ITIS
`get.tax.env`	Environment variable where the full ITIS taxon table is available
`col.sel`	Integer allowing selection of a particular column of the matrix of names to check.

Value

Returns a list in which each element of the list is a successive character string from the parsed names.

Author(s)

Lester L. Yuan

`incorp.correct` *Incorporate taxon name revisions*

Description

Incorporates corrected taxon names into parse.list

Usage

```
incorp.correct(tname.new, parse.list)
```

Arguments

`tname.new`	Character vector of corrected taxon names
`parse.list`	List of parsed taxon names

Value

Revised parse.list.

Author(s)

Lester L. Yuan

`infergui` *Graphical User Interface for biological inferences*

Description

Provides a graphical user interface that allows users to compute biological inferences using existing taxon-environment relationships.

Usage

```
infergui()
```

Value

None. Used for its side effects.

Author(s)

Lester L. Yuan

`itis.ttable` *ITIS taxonomic hierarchy table*

Description

Full taxonomic hierarchy to genus for the Kingdom Animalia. Retrieved November 21, 2008 from the Integrated Taxonomic Information System on-line database, `http://www.itis.gov`.

Usage

```
data(itis.ttable)
```

Source

Integrated Taxonomic Information System, U.S. Department of Agriculture.

Examples

```
data(itis.ttable)
```

`load.itis` *Load ITIS taxon table*

Description

Load ITIS taxon table to the specified environment.

Usage

```
load.itis(get.tax.env)
```

Arguments

`get.tax.env`	Environment variable.

Details

Loads full ITIS taxonomic table into specified environment.

Value

None. Used for side effects.

Author(s)

Lester L. Yuan

`load.revised.otu` *Loads a user-revised OTU table*

Description

If manual correction of the OTU selections is desired, these corrections should be made on the "sum.otu.txt" file provided by `get.otu`. The corrections should be saved as tab-delimited text, and then `load.revised.otu` run.

Usage

```
load.revised.otu(bcnt.otu, otufname = "sum.otu.txt")
```

Arguments

`bcnt.otu`	Original benthic count file with OTU designations. Output from `get.otu`
`otufname`	File name of revised OTU assignments.

Details

The benthic count data frame is assumed to be the original output from `get.otu`, and therefore has a field called `TNAME`.

Value

The script returns a new benthic count file that incorporates the corrections to the OTU table.

Author(s)

Lester L. Yuan

See Also

`get.otu`

`load.revised.species` *Loads a user-revised list of species names*

Description

If manual correction of the species names assigned by `get.taxonomic` is desired, these corrections should be made on the `sum.tax.table.txt` file. The corrections should be saved as tab-delimited text, and then `load.revised.species` run.

Usage

```
load.revised.species(bcnt.tax, fname)
```

Arguments

`bcnt.tax`	benthic count data frame that includes the full taxonomic hierarchy for each taxon.
`fname`	Character string containing file name for the revised, tab-delimited, file in the format of `sum.tax.table.txt`. See `get.taxonomic`.

Details

The script prompts the user to enter the name of the text file containing the revised species names. The original "SPECIES" column in bcnt.tax is deleted and the new, revised version is merged into bcnt.tax.

Value

The script returns a new benthic count file that incorporates the corrections to the species names

Author(s)

Lester L. Yuan

See Also

`get.taxonomic`

`locate.dupes` *Locate duplicated taxon names*

Description

Locate taxon names that match multiple records in ITIS.

Usage

```
locate.dupes(fulltab)
```

Arguments

`fulltab`	Full taxonomic table for matched taxon names

Details

Finds duplicate names in fulltab

Value

Returns string that summarizes duplicates and their row numbers.

Author(s)

Lester L. Yuan

`make.fulltab1` *Make full taxonomic table*

Description

Merge full ITIS taxonomy with names in benthic count file

Usage

```
make.fulltab1(df.parse, get.tax.env)
```

Arguments

`df.parse`	Matrix with valid taxon names in the second column
`get.tax.env`	Environment variable where the full ITIS taxon table is available

Details

Initial merge of taxon names with full ITIS table.

Value

Returns the current list of valid taxon names merged with the full taxonomic hierarchy.

Author(s)

Lester L. Yuan

`make.species` *Generate species names*

Description

Generate species names from possibilities in the parsed taxa name table.

Usage

```
make.species(df.parse, fulltab)
```

Arguments

`df.parse`	Matrix with valid taxon names in the second column
`fulltab`	Full taxonomic table for matched taxon names

Details

Generates likely species names from matched taxon names that are identified as Genus level names.

Value

Returns full taxonomic table with SPECIES field appended.

Author(s)

Lester L. Yuan

`makess` *Make a site-OTU matrix*

Description

`makess` generates a site-OTU matrix based on a benthic count file. The script assumes that the benthic count file has a field entitled 'OTU' on which the matrix is based.

Usage

```
makess(bcnt, tname = "OTU", plothist=FALSE, prints = FALSE, nview = 0)
```

Arguments

`bcnt`	a benthic count file that results from running `get.otu` or `load.revised.otu`.
`tname`	Character string specifying the field in bcnt in which the name of the taxon is stored.
`plothist`	Set to `TRUE` to plot a histogram of the proportion of abundance at each site that is included in the OTU designations
`prints`	Set to `TRUE` to view statistics on the proportion of abundance at each site that is included in the OTU designations
`nview`	Selects the number of sites to view in which the proportion of abundance included is low

Value

A site-OTU matrix.

Author(s)

Lester L. Yuan

See Also

`get.otu`, `load.revised.otu`

Examples

```
data(bcnt.otu.OR)
ss.OR <- makess(bcnt.otu.OR)
```

`mlsolve` *Maximum likelihood inferences*

Description

`mlsolve` uses maximum likelihood estimation to infer environmental conditions from biological observations.

Usage

```
mlsolve(ss, coef0, site.sel = "all", bruteforce = FALSE)
```

Arguments

`ss`	A site-species matrix, with sample identifiers in the first column, and abundances. Typically, the result of `makess`.
`site.sel`	Character vector of sample id's at which inferences should be computed. Default selection of 'all' forces script to compute inferences for all samples in the matrix.
`coef0`	Coefficient file with model specifications and regression coefficients defining taxon-environment relationships. See `taxon.env` for more details.
`bruteforce`	Logical flag. Select `TRUE` to compute solution by brute force rather than by conjugate gradients. Provides a means of examining likelihood contours when needed.

Details

`mlsolve` uses maximum likelihood estimation to infer environmental conditions from biological observations. Maximum likelihood estimation in this context is a constrained optimization problem, in which we wish find the point at which the likelihood function is maximized, constrained by the range of the environmental variables in the calibration data. `mlsolve` formulates the likelihood function and calls `optim` to solve the optimization problem.

The option `bruteforce` can be used to check the performance of the iterative solver. When `bruteforce` is set to be `TRUE`, the script also computes likelihood values for a uniformly-spaced grid that spans the ranges of the environmental variables. This is a time-consuming calculation, and therefore should only be attempted on a few samples. Once the grid is computed, though, contours of the likelihood surface are plotted.

`mlsolve` requires that the taxa included in `ss` each have associated taxon-environment information in `coef0`. The best way to ensure that this condition is satisfied is to make sure that `get.otu` is run with the same `coef0` as used in `mlsolve`, and that the results from `get.otu` are used to generate `ss` (see `makess`).

Value

The script returns a dataframe containing inferences for each sample as well as a logical flag indicating whether solutions computing with four different starting locations are consistent with one another.

Author(s)

Lester L. Yuan

Examples

```
data(coef.west.wt)
data(ss.OR)
inf.out <- mlsolve(ss.OR, coef.west.wt)
```

`modalDialog` *Modal dialog box using TclTK*

Description

`modalDialog` generates a dialog box prompting the user to enter a list of text before the process continues. Based on scripts provided in http://bioinf.wehi.edu.au/~wettenhall/RTclTkExamples/

Usage

```
modalDialog(title,itemlist,entryInit,entryWidth=20,
                        returnValOnCancel="ID_CANCEL")
```

Arguments

`title`	Title for the dialog box
`itemlist`	Character vector of names that the user must correct
`entryInit`	Default string to fill in the entries
`entryWidth`	Width of entry string
`returnValOnCancel`	Value for script to return if canceled

Value

Character vector of user entries.

Author(s)

Lester L. Yuan

`output.tax.table` *Output final taxonomic table*

Description

Output final taxonomic table as tab-delimited text

Usage

```
output.tax.table(finaltab, tlevs)
```

Arguments

`finaltab`	Merged taxonomic table
`tlevs`	Character vector of taxonomic levels in ITIS

Details

Outputs final table to file.

Value

None. Used for side effects.

Author(s)

Lester L. Yuan

`parse.taxon.name` *Parse taxon name*

Description

Parses taxon names from benthic count file to facilitate matching with ITIS.

Usage

```
parse.taxon.name(tname.orig)
```

Arguments

`tname.orig`	Character vector of taxon names

Details

Parses compound names typically found in benthic count files into a list of strings for each name.

Value

Returns a matrix in which each row corresponds with a different element of the provided character vector, and each column corresponds to distinct strings in each element.

Author(s)

Lester L. Yuan

`remove.dupes` *Remove unselected duplicates from taxon table*

Description

Remove unselected duplicate taxon names from taxon table.

Usage

```
remove.dupes(fulltab, dupe.list, dupe.sel)
```

Arguments

`fulltab`	Full taxon table for matched taxon names.
`dupe.list`	List of duplicated entries.
`dupe.sel`	Character vector of taxa selected by user.

Details

Deletes rows from fulltab that are not selected by user

Value

Returns revised fulltab.

Author(s)

Lester L. Yuan

`resolve.mult` *Resolve cases in which two taxa are found in ITIS*

Description

Resolves cases in which two taxa listed on the same entry are both found in ITIS.

Usage

```
resolve.mult(parse.list, get.tax.env)
```

Arguments

`parse.list`	List of taxa that are found in ITIS (position 1)and taxa that are still unresolved (position 2).
`get.tax.env`	Environment variable where the full ITIS taxon table is available

Details

The most likely reason for two valid taxa being listed in the same entry is that the taxonomist could only identify a specimen down to two possibilities. In these cases, the script downgrades the identification to a coarser level that is the same for both taxa.

Value

Updates parse.list so that resolved multiple listings are included in position 1.

Author(s)

Lester L. Yuan

`sel.coeffile` *Select coefficient file*

Description

Prompt user to select coefficient file from those available in data directory

Usage

```
sel.coeffile(fstring = "coef")
```

Arguments

`fstring`	Character string for filtering files in data directory

Details

Provides the list of data files available to the user.

Value

Selected file name.

Author(s)

Lester L. Yuan

`ss.OR` *site-OTU matrix for western Oregon*

Description

Matrix with each row representing a distinct sample and each colum representing a different OTU for western Oregon benthic count data.

Usage

```
data(ss.OR)
```

Source

Oregon Department of Environmental Quality

Examples

```
data(ss.OR)
```

`taxon.env` *Estimate taxon-environment relationships*

Description

`taxon.env` estimates taxon-environment relationships using logistic regression.

Usage

```
taxon.env(form, bcnt, envdata, bcnt.siteid, bcnt.abndid,
env.siteid,tlevs = "all", dumpdata = FALSE)
```

Arguments

`form`	Regression formula for model. Format is identical to that used for `glm`.
`bcnt`	A benthic count file with a full taxonomic hierarchy that results from running `get.taxonomic`.
`envdata`	Environmental data file
`bcnt.siteid`	Field name for `bcnt` unique site identifier.
`bcnt.abndid`	Field name for `bcnt` taxon abundance.
`env.siteid`	Field name for `envdata` unique site identifier.
`tlevs`	Taxonomic levels to compute taxon-environment relationships.
`dumpdata`	Logical field specifying whether data used to estimate taxon-environment relationships should be included in the output.

Details

This script generates logistic regression models for taxa at various levels of taxonomic resolution (as specified by the user). The script computes the degrees of freedom specified by the model (df) and selects taxa that occur in at least 10 times df and is absent from at least 10 times df sites. Then, logistic regression models based on the formula specified by the user are computed for each taxon using `glm`. The user can select the taxonomic levels (e.g., CLASS, ORDER, SPECIES, etc.) at which taxon-environment relationships are computed using the parameter `tlevs`. The default value for `tlevs` computes relationships at all levels.

Value

Regression coefficient file that can be used with `mlsolve` to infer environmental conditions.

`tnames`	Taxon names for which models are calculated
`csave`	Matrix of regression coefficients
`xvar`	Character vector with names of explanatory variables
`xlims`	Range of each explanatory variable
`form`	Model formula
`roc`	Area under receiver operator curve for model for each taxon.
`raw.data`	Data used to estimate relationships. Only included if `dumpdata = T`.

Author(s)

Lester L. Yuan

See Also

`get.taxonomic`

Examples

```
data(envdata.OR)
data(bcnt.tax.OR)
coef <- taxon.env(~sed + sed^2, bcnt.tax.OR, envdata.OR, "SVN",
"CountValue", "STRM.ID")
```

`tklist.modal` *Modal select list dialog using TclTk*

Description

`tklist.modal` generates a dialog box that prompts the user to select from a list of options. This script uses TclTk tools that should be operable on different operating systems.

Usage

```
tklist.modal(title, elements0, returnValOnCancel="ID_CANCEL",selectmode =
"single")
```

Arguments

`title`	Title for the list box.
`elements0`	Character vector of elements listing different choices
`returnValOnCancel`	Character string returned by the script if the dialog box is canceled
`selectmode`	Set to multiple to allow for multiple selections.

Value

A TclTk dialog box.

Author(s)

Lester L. Yuan

Examples

```
## Not run:
tklist.modal("Select a color", c("red", "green", "blue"))

## End(Not run)
```

`trait.feeding` *Feeding traits for benthic invertebrates*

Description

Feeding classifications for different benthic invertebrates.

Usage

```
data(trait.feeding)
```

Format

A data frame with two columns.

`TAXON` Character vector with the names of each taxon.

`TRAITVAL` Value of trait classification

Details

These trait classifications were summarized from the USGS data set prepared by Viera et al. `http://pubs.usgs.gov/ds/ds187/`.

Source

Yuan, L.L.

Examples

```
data(trait.feeding)
```

`trait.habit` *Habit traits for benthic invertebrates*

Description

Habit classifications for different benthic invertebrates.

Usage

```
data(trait.habit)
```

Format

A data frame with two columns.

`TAXON` Character vector with the names of each taxon.

`TRAITVAL` Value of trait classification

Details

These trait classifications were summarized from the USGS data set prepared by Viera et al. http://pubs.usgs.gov/ds/ds187/.

Source

Yuan, L.L.

Examples

```
data(trait.habit)
```

`trait.stat` *Compute trait metrics*

Description

Compute trait metrics from benthic count data.

Usage

```
trait.stat(bcnt.otu, coefs)
```

Arguments

`bcnt.otu`	Benthic count file with OTU defined consistently with selected trait file
`coefs`	Trait file

Details

OTUs appropriate for the selected trait file must first be assigned using `get.otu`. A list of trait files provided in the library can be obtained using `sel.coeffile`. Script will compute richness, relative richness, and relative abundance of each trait group. Note that many taxa in a benthic count file are often not assigned to traits, and so relative richness and relative abundance values within a given sample may not sum to 1.

Value

Data frame of trait metrics.

Author(s)

Lester L. Yuan

`view.te` *View taxon-environment relationships*

Description

Viewer for taxon-environment relationships.

Usage

```
view.te(coef, plotform = "pdf")
```

Arguments

`coef`	Taxon-environment coefficient file (e.g. `coef.west.wt`)
`plotform`	Specify output plot format. Options include pdf and windows.

Details

Generates contour plots that show taxon-environment relationships represented by taxon-environment coefficients. Models with a single explanaotry variable are displayed as line plots, and models with two explanatory variables are displayed as contour plots. Plots are returned returned the file `taxon.env.pdf` is `plotform = "pdf"`.

Value

None. Used for side effects.

Author(s)

Lester L. Yuan

Examples

```
## Not run:
data(coef.west.wt)
view.te(coef.west.wt)
## End(Not run)
```

Index