Package: dietr (via r-universe)

November 25, 2024

Version 1.1.5-1

Date 2024-9-15

Title Diet Estimated Trophic Levels

Depends R (>= 4.3)

Imports rfishbase (>= 5.0)

Description Estimates fractional trophic level from quantitative and qualitative diet data and calculates electivity indices in R. Borstein (2020) <doi:10.1007/s10750-020-04417-5>.

License GPL (>= 2)

LazyData true

RoxygenNote 7.3.2

Encoding UTF-8

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

URL `https://github.com/sborstein/dietr`

BugReports `https://github.com/sborstein/dietr/issues`

NeedsCompilation no

Author Samuel R. Borstein [aut, cre]

Maintainer Samuel R. Borstein <`sam@borstein.com`>

Date/Publication 2024-09-16 11:30:06 UTC

Additional_repositories `https://cranhaven.r-universe.dev`

Config/pak/sysreqs make libicu-dev

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

RemoteUrl https://github.com/cranhaven/cranhaven.r-universe.dev

RemoteRef package/dietr

RemoteSha bf870227cfb4559a7151876f8c99f129450d57ac

Contents

`Casaux1998` *Casaux1998*

Description

Raw data from Casaux, 1998. Prey is listed in percent frequency, percent number, and percent mass consumed.

Usage

```
Casaux1998
```

Format

A data frames containing diet data for two populations (Potter Cove & Harmony Point) of Harpagifer antarcticus from the South Shetland Islands.

- Record: The record, in this case the name for the two populations, Harpagifer_antarcticus_PotterCove and Harpagifer_antarcticus_HarmonyPoint for populations from Potter Cover and Harmony Point respectively.
- Prey: Name of the prey item consumes.
- PercentOccurrence: Data for the percent occurrence of the prey consumed.
- PercentNumber: Data for the percent number of the prey consumed.
- PercentWeight: Data for the percent weight of the prey consumed.

Details

Data from Casaux, 1998 for the diet of Harpagifer antarcticus from two localities in the South Shetland Islands of Antarctica.

References

- Casaux R. 1998. The contrasting diet of Harpagifer antarcticus (Notothenioidei, Harpagiferidae) at two localities of the South Shetland Islands, Antarctica. Polar Biology 19:283-285.

`CompositeIndices` *Calculates various composite indices for studying trophic ecology*

Description

Calculates various composite indices for studying trophic ecology

Usage

```
CompositeIndices(
  DietData,
  Indices = c("IOP", "IRI", "FQ"),
  PercentNumber = NA,
  PercentOccurrence = NA,
  PercentVolWeight = NA,
  ReturnRaw = FALSE,
  PercentOnly = TRUE
)
```

Arguments

`DietData` A data frame containing the diet data. See details for information on formatting

`Indices` Character vector of indices to calculate. Options are IOP for index of preponderance, IRI for index of relative importance, and FQ for feeding quotient.

`PercentNumber` Numeric. Column number in DietData containing the percent numeric data. If calculating an index that does not require percent number data, can be left as NA.

`PercentOccurrence`
Numeric. Column number in DietData containing the percent Occurrence data. If calculating an index that does not require frequency of occurrence data, can be left as NA.

`PercentVolWeight`
Numeric. Column number in DietData containing the percent volume or weight data.

`ReturnRaw` Logical. Should the raw input data be returned with the compound indices? Default is FALSE (i.e. only the calculated indices are returned).

`PercentOnly` Logical. Should composite indices only be returned as percentages or should the raw calculated values be returned? Default is TRUE (i.e. indices are returned as percentages only).

Details

This function calculates compound indices (indices combining various diet measurements). Specifically, this function calculates three indices: The Index of Preponderance, the Index of Relative Importance, and the Feeding Quotient. These indices have numerous synonyms, and for a review of them, I recommend de Silviera et al., 2020. With this function users can select to calculate one or more of the three compound indices mentioned above. The Index of Preponderance (Natarajan & Jhingran, 1961 AKA Feeding Index Kawakami & Vazzoler, 1980) is the product of the frequency of occurrence of prey with either their volumetric or weight contribution to the diet. The Index of Relative Importance (Pinkas et al., 1971) is calculated as the the product of the sum of the weight or volume of a prey item and the percent number and the percent frequency occurrence. The Feeding Quotient (Hureau, 1970) is the product of the percent number and the percent weight. These indices can be returned as just a percentage or the percentage and the raw calculations with the PercentOnly argument.

The main input for this function is DietData, which requires minimal formatting. It is mandatory for this function that the first column contain the diet record identifier and the second column the names of the prey. The diet record identifier should be a unique name for each record, which allows one to calculate feeding indices for numerous records with a single call of the function. As the second column contains the prey identifier, if a species feeds upon three different prey, the first three rows of the dataset should have the same record identifier. The remaining columns should contain information on the the contribution of the various prey to the diet of the record required to calculate the indices. Note that the order of these columns does not matter as the frequency of occurrence, precent number, and percent volume or weight are specified with the arguments PercentNumber, PercentOccurrence, and PercentVolWeight respectively. Users should enter in the column number for these arguments that corresponds to their respective position in DietData. If you are having difficulty formatting data, I highly recommend seeing the format of the diet data used in the example below, which can be loaded by running data(Casaux1998). This example is for two unique diet records for the fish Harpagifer antarcticus from two different localities (Potter Cove and Harmony Point) and contains data on the frequency of occurrence percent number, and percent mass of the prey in the diets of the fish from these two populations.

Value

A list the length of the unique records in DietData containing data frames with the calculated diet indices. The lists will be names with the respective diet record names.

Author(s)

Samuel Borstein

References

da Silveira EL, Semmar N, Cartes JE, Tuset VM, Lombarte A, Ballester ELC, and Vaz-dos-Santos AM. 2019. Methods for Trophic Ecology Assessment in Fishes: A Critical Review of Stomach Analyses. Reviews in Fisheries Science & Aquaculture 28:71-106. 10.1080/23308249.2019.1678013

Hureau J-C. 1970. Biologie comparee de quelques poissons antarctiques (Nototheniidae). Bulletin de l'Institut Oceanographique de Monaco 68:1-244.

Kawakami E, and Vazzoler G. 1980. Metodo grafico e estimativa de indice alimentar aplicado no estudo de alimentacao de peixes. Boletim do Instituto Oceanografico 29:205-207.

Natarajan A, and Jhingran A. 1961. Index of preponderance-a method of grading the food elements in the stomach analysis of fishes. Indian Journal of Fisheries 8:54-59.

Pinkas L, Oliphant MS, and Iverson IL. 1971. Food Habits of Albacore, Bluefin Tuna, and Bonito In California Waters. Fish Bulletin 152:1-105.

Examples

```
#Load diet data from Casaux1998, which contains diet for two populations of Harpagifer
# antarcticus in percent frequency, percent number, and percent mass.
data(Casaux1998)
#Calculate all three diet indices (IOP, IRI, FQ), return the raw data, and all calculations.
CompositeIndices(DietData = Casaux1998, Indices = c("IOP","IRI","FQ"), PercentNumber = 4,
PercentOccurrence = 3, PercentVolWeight = 5, ReturnRaw = TRUE, PercentOnly = FALSE)
#Calculate all three diet indices and return only the percent of the index
CompositeIndices(DietData = Casaux1998, Indices = c("IOP","IRI","FQ"), PercentNumber = 4,
PercentOccurrence = 3, PercentVolWeight = 5, ReturnRaw = FALSE, PercentOnly = TRUE)
#Calculate Feeding Quotient and return the raw data and the all calculations.
CompositeIndices(DietData = Casaux1998, Indices = "FQ", PercentNumber = 4, PercentVolWeight = 5,
ReturnRaw = FALSE, PercentOnly = FALSE)
```

ConvertFishbaseDiet *Converts FishBase/SealifBase diet data obtained from rfishbase for use with dietr*

Description

This converts diet data from rfishbase into a format usable with dietr.

Usage

```
ConvertFishbaseDiet(ExcludeStage = NULL)
```

Arguments

ExcludeStage	a character vector, indicating which life stages to exclude. Must match stage names given by rfishbase (i.e. larvae, rec./juveniles, juv./adults, adults).

Details

As of rfishbase 3.0, the package handles returning diet data differently than in previous versions. As currently implemented, rfishbase returns two different unjoined diet tables, one with the actual diet items and their percent contribution in the diet, and another that has the metadata for the diet record. This unfortunately is a difficult format to work with and easily extract out species of interest. Additionally, in previous versions you could specify species names or numbers and only return those of interest, the only function options in the current version are specifying a server. As such, if the function is run, it will return all diet data on the site, requiring users to subset out those of interest for them. We have implemented in this function a way to join the tables for use as well as filter based on life history stage (if necessary).

Value

a list of length two, with two data frames. One containing the re-formatted diet items and their contributions and one containing the Taxonomy with species names.

Author(s)

Samuel Borstein

Examples

```
#Convert Fishbase Diet Data
my.diets <- ConvertFishbaseDiet(ExcludeStage=NULL)
#Convert Fishbase Diet Data and exclude juvenile and larval records
my.diets <- ConvertFishbaseDiet(ExcludeStage=c("recruits/juv.","larvae"))
```

ConvertFishbaseFood *Converts FishBase/SealifBase food item diet data obtained from the diet function into a usable format for dietr*

Description

Converts FishBase/SealifBase food item diet data obtained from the diet function into a usable format for dietr

Usage

```
ConvertFishbaseFood(FishBaseFood, ExcludeStage = NULL)
```

Arguments

FishBaseFood	a data frame produced by the rfishbase fooditem function
ExcludeStage	a character, indicating which life stages to exclude. Must match stage names given by rfishbase (i.e. larvae, rec./juveniles, juv./adults, adults).

Details

This converts the data frame produced by rfishbase fooditem function into a usable format for dietr.

Value

a list of length two, with two data frames. One containing the re-formatted food item data and one containing the Taxonomy with species names.

Author(s)

Samuel Borstein

Examples

```
#Get rfishbase food item data for a few species
my.food <- as.data.frame(rfishbase::fooditems(c("Lutjanus apodus","Epinephelus itajara")))
#use the ConvertFishbaseFood function to format it for dietr and exclude recruits/juveniles
cleaned.food <- ConvertFishbaseFood(FishBaseFood=my.food, ExcludeStage=c("larvae","recruits/juv."))
```

CortesPreyVals *CortesPreyVals*

Description

A data frame containing prey items and their respective trophic levels for Chondrichthyes prey from Cortes, 1999

Usage

```
CortesPreyVals
```

Format

A data frame of of 10 rows and 6 columns

- FoodI: Food category I.
- FoodII: Food category II.
- FoodIII: Food category III.
- Stage: Life history stage of the prey item.
- TL: Trophic level of the prey item.
- SE: Standard error around trophic level estimate of the prey item.

References

- Cortes E. 1999. Standardized diet compositions and trophic levels of sharks. ICES Journal of marine science 56:707-717.

dietr *Diet Estimated Trophic Levels in R*

Description

Estimates fractional trophic level from quantitative and qualitative diet data and calculates electivity indices in R.

Details

Package: dietr

Type: Package

Title: An R package to estimate trophic level from diet and food item data

Version: 1.1.4

Date: 2024-9-8

License: GPL (>= 2)

This package allows users to calculte trophic levels from proportional diet data or from food items given trophic levels of the prey items. This package calculates trophic level following the procedures from TrophLab, which was a Microsoft Access program. This implementation is faster than the original microsoft access program and also allows for a hierarchical estimation of trophic level given a corresponding "taxonomy" data frame (i.e. estimate trophic level for a individual,population,species, etc.). It also contains other functions for measuring electivity indices, compound trophic indices (ex. Feeding Quotient, Index of Preponderance), Vacuity Index, and Gastrosomatic Index.

Author(s)

Samuel Borstein Maintainer: Samuel Borstein <borstein@umich.edu>

See Also

`Electivity`,`ConvertFishbaseFood`,`DietTroph`,`FoodTroph`

DietTroph *Calculates trophic level from percentage based quantitative diet data (volumetric, weight, etc.) for species, populations, and individuals.*

Description

Calculates trophic level from percentage based diet data following the routine described in TrophLab. While FishBase data obtained from rfishbase can be used, users can also upload their own data for use in with function (see vignette for a tutorial).

Usage

```
DietTroph(
  DietItems,
  PreyValues,
  Taxonomy,
  PreyClass = c("FoodI", "FoodII", "FoodIII", "Stage"),
  SumCheck = TRUE
)
```

Arguments

`DietItems` A data frame with rows as individual entries and each row consisting of a prey classification and a corresponding diet percentages. The column names for prey classification of the diets should match those of PreyValues. The first column should contain the identifier of the individual and be named "Individual", The last column should contain the percent of the prey in the diet and be labelled "Percent".

`PreyValues` a data frame with rows as prey item names and columns containing the trophic level of the prey item and the standard error of that trophic item. The column names of PreyValues except for TL and SE should match those in DietItems.

`Taxonomy` a data frame starting with the least inclusive level progressing to more inclusive moving towards the right.

`PreyClass` Column names of the PreyValues used for matching between DietItems and PreyValues, exclusive of TL and SE. Default is those of FishBase.

`SumCheck` Logical. Should the sum of diet items be checked, and, if not equal to 100, recalculated?

Value

A list length of the columns in taxonomy, each containing trophic level estimation at the respective taxonomic level.

Author(s)

Samuel Borstein

Examples

```
###EXAMPLE USING RAW DATA FROM Magalhaes et al., 2015###
data(Herichthys)#load data
#Subset out individuals with diet data
HMdat<-Herichthys[Herichthys$total==100,]
#Make a data frame of the individuals, lake by year, and lake.
HMtax<-cbind.data.frame(HMdat$individual,paste(HMdat$lake,HMdat$year),HMdat$lake)
#Name the data frame
colnames(HMtax)<-c("Individual","Lake x Year","Lake (all years)")
#To calculate trophic level for the entire species, add a vecotr to the data frame of
#the species name
```

```
HMtax$species<-"Herichthys minckleyi"
HMdat<-HMdat[,c("individual","X.Gastrop","X.Insect","X.Fish","X.Zoopl"
,"X.plants","X.algae", "X.detritus")]
#Repeat the individual name the number of unique prey types (6)
Inds<-rep(x = HMdat$individual, times=6)[order(rep(x = HMdat$individual, times=6))]
#Repeat the number of food typed the length of the number of individuals
FoodTypes<-rep(x = colnames(HMdat[2:7]),times=length(unique(HMtax$Individual)))
#Make a data frame, the length of the individuals with three columns
HM.mat<-as.data.frame(matrix(nrow = length(Inds),ncol = 3))
#Name these columns
colnames(HM.mat)<-c("Individual","FoodItem","Percent")
#Populate the dataframes first column with the individual and the second column with the prey type
HM.mat$Individual<-Inds
HM.mat$FoodItem<-FoodTypes
#Run this for loop to find the diet data based on the individual and then match the
#diet percentage based on the name of the prey type
for(i in 1:nrow(HMdat)){
rows2match<-which(HM.mat$Individual==HMdat$individual[i])
HM.mat$Percent[rows2match]<-as.vector(as.numeric(HMdat[i,2:7]))
}
#Remove prey that do not contribute to diets
HM.mat<-HM.mat[!HM.mat$Percent==0,]
#Create a empty data frame for prey values
PreyMat<-as.data.frame(matrix(ncol = 3,nrow = 6))
#Name the columns something useful
colnames(PreyMat)<-c("FoodItem","TL","SE")
#Add in the prey name to the PreyMat
PreyMat[,1]<-unique(FoodTypes)
#Add in the trophic levels of the prey
PreyMat[,2]<-c(2.37,2.2,3.5,2.1,2,2)
#Add in the SE of the prey
PreyMat[,3]<-c(.58,.4,.8,.3,0,0)
HM.TL<-DietTroph(DietItems = HM.mat,PreyValues = PreyMat, PreyClass = "FoodItem",
Taxonomy = HMtax, SumCheck = TRUE)
###EXAMPLE USING DATA FROM FISHBASE###
#Get some food item data from rfishbase
library(rfishbase)
#convert FishBase data into data for trophic calculation using dietr
converted.diet <- ConvertFishbaseDiet(ExcludeStage=NULL)
#Subset three studies out, as this contains all  studies from FishBase
my.diets <- converted.diet$DietItems[1:26,]
my.taxonomy <- converted.diet$Taxonomy[1:3,]
#Load Prey Values
data(FishBasePreyVals)
#Calculate Trophic Levels
my.TL <- DietTroph(DietItems = my.diets,PreyValues = FishBasePreyVals, Taxonomy =
my.taxonomy, PreyClass=c("FoodI","FoodII","FoodIII","Stage"))
```

Electivity *Calculates a variety of electivity indices and foraging ratios*

Description

This function calculates the forage ratio and a variety of electivity indices. Included indices include Ivlev's (1961), Strauss' (1979), Jacob's Q and D (1974), Chesson's (1983)(Which is similar to Manly's Alpha (1974)), and Vanderploeg & Scavia (1979).

Usage

```
Electivity(
  Diet,
  Available,
 Indices = c("ForageRatio", "Ivlev", "Strauss", "JacobsQ", "JacobsD", "Chesson",
    "VanderploegScavia"),
  LogQ = TRUE,
  CalcAbundance = FALSE,
  Depleting = FALSE
)
```

Arguments

`Diet`	Data frame with data corresponding to consumed resources found in the diet. See details for formatting.
`Available`	Data frame with data corresponding to the available resources. See details for formatting.
`Indices`	Character vector containing the names of the desired indices to calculate. See description for information on available indices.
`LogQ`	Logical. If true, should Jacob's Q be logged? This is the recommendation of Jacob, 1974.Default is TRUE, following the recommendation.
`CalcAbundance`	Logical. If true, are the data raw and would you like to calculate the relative abundance? Relative abundance will be calculated for both diet and available given the row sums of the supplied Diet and Available parameter inputs. Default is False, i.e. relative abundances are not calculated.
`Depleting`	Logical. If true, calculates Chesson's Case 2, where food depletion occurs and the available food is not constant. Default is False.

Details

This function calculates one or multiple electivity indices for one or more diet records for which one or more records of prey availability exists.For example, it is possible to calculate multiple indices for multiple diet records that may be from a number of sites with different prey availability all in one call of the function (see example which). Specifically, this function measures the following indices (and their input for the Indices argument) Ivlev's (1961) Forage Ratio ("ForageRatio") and electivity ("Ivlev"), Strauss'(1979) ("Strauss"), Jacobs (1974) Q ("JacobsQ") and D ("JacobsD"), Chesson (1983)(Which is similar to Manly's Alpha (1974))("Chesson"), and Vanderploeg & Scavia (1979) ("VanderploegScavia"). For those wishing to calculate Vanderploeg and Scavia's selectivity coefficient (W), please select "Chesson" as an argument for indices, which will calculate Chesson's alpha, which is identical to Vanderploeg and Scavia's selectivity coefficient (W).

The function takes two data frames as input. The first argument, Diet, should be formatted as followed. Each row in the data frame should be a diet record. The first column should contain the name of the record to be calculated. The second column should contain the name linking the consumed prey in Diet to that in Available (example, name of the different habitats, note in our example it is "Year"), which will be described below. All remaining columns should contain the abundance or relative abundance of the prey in the diet. These columns should also be named so they can be matched to those in Available. The second data frame, Available should be formatted similar to Diet where each row describes a unique record for available prey. The remaining columns should contain the abundance or relative abundance of the prey that are available to be consumed. These columns should also be named so they can be matched to Those in Diet. These indices typically utilize relative abundances and the abundances with relative abundance represented by a decimal (i.e sum to 1). Users can supply raw data that are not in relative abundances and use the CalcAbundance option to calculate relative abundances if they wish to do so. By default, CalcAbundance is set to FALSE and assumes the supplied data are already in relative abundance, so users wishing to calculate relative abundance should use CalcAbundance = TRUE.

Indices are bounded by the following values. Ivlev's, Strauss', and Jacobs' D, and Vanderploeg & Scavia's indices are bounded between -1 and 1, with items closer to -1 representing avoided items, 0 randomly feeding, and 1 preferential items. Forage ratio values range between 1 and infinity for preferred items while values between 0 and 1 represent avoided prey. Similar to forage ratio, Jacobs' Q ranges between 1 and infinity for preferred items and 0 and 1 for avoided prey, however log10(Q) is the preferred as it provides the advantage of equal ranges and ranges from -infinity to +infinity for avoidance and preference respectively. This option can be selected in the function with the logQ argument, which by default is set to TRUE. Finally, Chesson's index ranges between 0 and 1 and preference is typically assessed using 1/n, where n is the number of prey types. The value of 1/n represents random feeding while values above and below 1/n represent preference and avoidance respectively. For Chesson's index, users can also specify if the available resources are depleting, in which case the equation from case 2 of Chesson, 1983 is calculated. Note, this takes the log of (p-r)/p) and values of 0 or negatives will return NaN.

Value

Object of class Electivity which is a list containing data frames for each electivity index selected.

Author(s)

Samuel Borstein

References

Chesson, J. 1983. The estimation and analysis of preference and its relatioship to foraging models. Ecology 64:1297-1304.

Ivlev, U. 1961. Experimental ecology of the feeding of fish. Yale University Press, New Haven.

Jacobs, J. 1974. Quantitative measurement of food selection. Oecologia 14:413-417.

Manly, B. 1974. A model for certain types of selection experiments. Biometrics 30:281-294.

Strauss, R. E. 1979. Reliability Estimates for Ivlev's Electivity Index, the Forage Ratio, and a Proposed Linear Index of Food Selection. Transactions of the American Fisheries Society 108:344-352.

Vanderploeg, H., and D. Scavia. 1979. Two electivity indices for feeding with special reference to zooplankton grazing. Journal of the Fisheries Board of Canada 36:362-365.

See Also

PlotElectivity

Examples

```
#Load Electivity Data from Horn 1982
data(Horn1982)
#Run all electivity indices
my.indices <- Electivity(Diet = Horn1982$Consumed, Available = Horn1982$Available, Indices =
c("ForageRatio","Ivlev","Strauss","JacobsQ","JacobsD","Chesson","VanderploegScavia"),LogQ = TRUE,
CalcAbundance = FALSE, Depleting = FALSE)
```

FishBasePreyVals *FishBasePreyVals*

Description

A data frame containing prey items and their respective trophic levels following FishBase and TrophLab.

Usage

```
FishBasePreyVals
```

Format

A data frame of of 384 rows and 6 columns

- FoodI: Food category I.
- FoodII: Food category II.
- FoodIII: Food category III.
- Stage: Life history stage of the prey item.
- TL: Trophic level of the prey item.
- SE: Standard error around trophic level estimate of the prey item.

References

- Froese R, and Pauly D. 2018. FishBase. http://www.fishbase.org/2018).
- Pauly D, Froese R, Sa-a P, Palomares M, Christensen V, and Rius J. 2000. TrophLab manual. ICLARM, Manila, Philippines.

FoodTroph *Calculates trophic level from qualitative food item data for species, populations, and individuals using a random subsampling routine.*

Description

Calculates trophic level from food items where quantitative contribution is unknown. Follows the routine described in TrophLab. While FishBase data obtained from rfishbase can be used, users can also upload their own data for use with the function.

Usage

```
FoodTroph(
  FoodItems,
  PreyValues,
  Taxonomy,
  PreyClass = c("FoodI", "FoodII", "FoodIII", "Stage"),
  Iter = 100,
  SE.Type = "TrophLab"
)
```

Arguments

FoodItems a data frame with rows as individuals and each row consisting of a prey item name.

PreyValues a data frame with rows as prey item names and columns containing the trophic level of the prey item and the standard error of that trophic item.

Taxonomy a data frame with the least inclusive level in the leftmost column progressing to more inclusive with columns to the right.Can be a single column.

PreyClass Column names of the PreyValues used for matching between FoodItems and PreyValues, exclusive of TL and SE. Default is those of FishBase.

Iter Numeric representing how many iterations of the subsampling routine should be performed. Default is 100 (same as in TrophLab)

SE.Type Type of SE to perform. Can be either TrophLab, Sims, or Both. Default is TrophLab. See details for a more in-depth description of these options.

Details

Users can set the number of iterations for the subsampling routine for the simulated ranking of trophic items used in calculating trophic levels. We have this set currently to match how it is implemented in TrophLab, at 100. We find this performs relatively well and is quite quick to do, but users could also increase this value with minimal impacts on run time.

For calculating the SE (SE.Type parameter) around the estimated trophic level users have a few options. The current default follows TrophLab and uses the SE around the trophic levels of prey to estimate the SE which they also refer to as the omnivory index. Alternatively, users can select the Sims option, in which case the SE is estimated based on the estimated TL of the n number of subsampling routines.Users can also select "Both" if they would like to calculate both types of SE.

Value

a list of data frames containing estimated trophic levels from food items at each taxonomic level provided by the user.

Author(s)

Samuel Borstein

Examples

```
#Get some food item data from rfishbase
my.food <- rfishbase::fooditems(c("Lutjanus apodus","Epinephelus itajara"))
#convert FishBase data into data for trophic calculation using TrophicLevelR
converted.foods <- ConvertFishbaseFood(my.food)
#Load Prey Values
data(FishBasePreyVals)
#Calculate Trophic Levels
my.TL <- FoodTroph(FoodItems = converted.foods$FoodItems,PreyValues = FishBasePreyVals, Taxonomy =
converted.foods$Taxonomy,PreyClass=c("FoodI","FoodII","FoodIII","Stage"))
```

GastrosomaticIndex — *Calculates Gastro-somatic Index (AKA Fullness Index, Index of fullness)*

Description

This function calculates Gastro-somatic fullness, the relationship between stomach weight and body weight to assess fullness.

Usage

```
GastrosomaticIndex(StomachData, Calc.Vacuity = FALSE)
```

Arguments

`StomachData`	Object of class data frame containing stomach data. The data frame must be set up as follows. The first column should contain the specimen identifier, the second column the stomach weight, and the third column the weight of the specimens.
`Calc.Vacuity`	Logical. Should the vacuity index also be calculated? Default is FALSE.

Value

Object of Class List of length 2 or 3. If Calc.Vacuity = FALSE, a list of length two is returned containing a named vector of individual specimen gastro-somatic index as well as a numeric vector length of 1 containing the mean gastro-somatic index. If Calc.Vacuity = TRUE, an additional numeric vector of length 1 is returned containing the calculated vacuity index.

Author(s)

Samuel Borstein

References

Hureau J-C. 1970. Biologie comparee de quelques poissons antarctiques (Nototheniidae). Bulletin de l'Institut Oceanographique de Monaco 68:1-244.

See Also

VacuityIndex

Examples

```
data(SebastesStomachs)#load example data
#Calculate the Gastro-somatic Index
my.GSI <- GastrosomaticIndex (StomachData = SebastesStomachs)
#Calculate the Gastro-somatic Index and Vacuity Index
my.GSI <- GastrosomaticIndex (StomachData = SebastesStomachs, Calc.Vacuity = TRUE)
```

Herichthys *Herichthys*

Description

Raw supplementary data from Magalhaes et al., 2015 for the diets of Herichthys minckleyi used in a tutorial for the vignette.

Usage

```
Herichthys
```

Format

A data frame of of 519 rows and 40 columns

References

- Magalhaes IS, Ornelas-Garcia CP, Leal-Cardin M, Ramirez T, and Barluenga M. 2015. Untangling the evolutionary history of a highly polymorphic species: introgressive hybridization and high genetic structure in the desert cichlid fish Herichtys minckleyi. Mol Ecol 24:4505-4520. 10.1111/mec.13316

`Horn1982` *Horn1982*

Description

Raw data from Horn, 1982. Prey consumption and availability are relative abundance by weight of macroalgae species. Note that values do not sum to 1 as dataset only includes macroalgae and not other prey items. See Table 4 and Table 4 in Horn et al., 1982.

Usage

```
Horn1982
```

Format

A list of length 2 containing two data frames.

- Available A data frame with 2 rows and 18 columns containing data on percent weight relative abundance of macroalgae prey availability in two different years.
- Consumed A data frame with four rows and 19 columns containing data on percent weight relative abundance of macroalgae prey consumption by two species in two different years.

Details

Data from Horn, 1982 for the diet of two species of Stichaeidae and prey availability over two years.

References

- Horn M, Murray S, and Edwards T. 1982. Dietary selectivity in the field and food preferences in the laboratory for two herbivorous fishes (Cebidichthys violaceus and Xiphister mucosus) from a temperate intertidal zone. Marine Biology 67:237-246.

`PlotElectivity` *Creates Electivity plots*

Description

This function create a plot of electivity values calculated by the function Electivity.

Usage

```
PlotElectivity(
  Electivity.Calcs,
  Indices = c("Ivlev", "Strauss", "JacobsD", "VanderploegScavia"),
  BarColor = NULL,
  NameSize = 1,
  AxisFontSize = 1,
  BorderCol = "Black",
  LegendFontSize = 1,
  LegendTitle = "Record"
)
```

Arguments

`Electivity.Calcs`
An object of class Electivity produced by the function Electivity.

`Indices` Character. Which index should be plotted. Options can include any single index or combination of Ivlev, Strauss, JacobsD, and VanderploegScavia.

`BarColor` Character vector. Color to be assigned for each record in Electivity.Calcs. If specified, it should be the sames as the number of rows/records in Electivity.Calcs. If not specified, colors are randomly selected from the rainbow color palette.

`NameSize` Numeric. Controls the font size for names of prey on the x-axis. Default = 1.

`AxisFontSize` Numeric. Contols the font size of values on the y-axis. Default = 1.

`BorderCol` Character or Logical. specifying color for the border. If users wish to not include a border between bars, they should set this to FALSE. Default = Black.

`LegendFontSize` Numeric. Controls the font size of the legend. Default = 1.

`LegendTitle` Character. Title of the legend. Default is called "Record" following the formatting returned by the Electivity function, but if records are distinct species, could be for example set to "species".

Value

One or more plots. Length of the plots will equal the length of the Index parameter

Author(s)

Samuel Borstein

See Also

`Electivity`

Examples

```
#Load Electivity Data from Horn 1982
data(Horn1982)
#Run all electivity indices
my.indices <- Electivity(Diet = Horn1982$Consumed, Available = Horn1982$Available,
Indices = c("Ivlev","Strauss","JacobsD","VanderploegScavia"),LogQ = TRUE, Depleting = FALSE)

#Plot All Indices with default colors
PlotElectivity(Electivity.Calcs = my.indices)
#Plot only VanderploegScavia and set colors. Note, there are four records, so we need
#to set four colors
PlotElectivity(Electivity.Calcs = my.indices, Indices = "VanderploegScavia",
BarColor = c("Red","Purple","Black","Grey"))
```

SebastesStomachs *SebastesStomachs*

Description

Raw data from NOAA containing information on the stomach weight and specimen weight of eleven Sebastes flavidus caught in the Gulf of Alaska in 2011.

Usage

```
SebastesStomachs
```

Format

A data frames containing three columns with specimen identifier, specimen stomach weight, and specimen weight for eleven Sebastes flavidus.

- SpecimenSpecimen identifier. Follows the pattern of number, month, day, and year.
- StomachWeight: Weight of stomach contents in grams.
- PredatorWeight: Weight of the specimen in grams.

Details

Stomach and specimen weights for eleven Sebastes flavidus from the Gulf of Alaska by NOAA in 2011 (available at: https://access.afsc.noaa.gov/REFM/REEM/WebDietData/DietDataIntro.php).

References

- Livingston PA, Aydin K, Buckley TW, Lang GM, Yang M-S, and Miller BS. 2017. Quantifying food web interactions in the North Pacific - a data-based approach. Environmental Biology of Fishes 100:443-470. 10.1007/s10641-017-0587-0.
- .

VacuityIndex *Calculates the vacuity index, the proportion of fish which contain food in their stomach.*

Description

This function calculates the vacuity index, the frequency of fish which lack food in their stomach.

Usage

```
VacuityIndex(StomachData)
```

Arguments

`StomachData`	Object of class data frame containing stomach data. The data frame must be set up as follows. The first column should contain the specimen identifier, the second column the stomach weight.

Details

This function calculates the vacuity index (Hureau, 1970). This is simply the percentage of empty stomachs. The function takes a data frame of stomach data, which should be formatted as follows. The first column should contain the specimen identifier and the second column should contain the weight of the stomach contents. If users also have information on fish weight, they can calculate the Gastro-somatic index, using the function GastrosomaticIndex, which has an option to calculate the vacuity index utilizing this function.

Value

Numeric vector of length one with the calculated vacuity index.

Author(s)

Samuel Borstein

References

Hureau J-C. 1970. Biologie comparee de quelques poissons antarctiques (Nototheniidae). Bulletin de l'Institut Oceanographique de Monaco 68:1-244.

See Also

`GastrosomaticIndex`

Examples

```
data(SebastesStomachs)#load example data
My.Vacuity <- VacuityIndex(StomachData = SebastesStomachs)
```

Index