

Package ‘EcoTroph’

July 2, 2014

Type Package

Title EcoTroph R package

Version 1.6

Date 2013-08-31

Author J. Guitton, M. Colleter, P. Gatti, and D. Gascuel

Maintainer Jerome Guitton <jerome.guitton@agrocampus-ouest.fr>

URL <http://sirs.agrocampus-ouest.fr/EcoTroph/>

Description EcoTroph is an approach and software for modelling marine and freshwater ecosystems. It is articulated entirely around trophic levels. EcoTroph's key displays are bivariate plots, with trophic levels as the abscissa, and biomass flows or related quantities as ordinates. Thus, trophic ecosystem functioning can be modelled as a continuous flow of biomass surging up the food web, from lower to higher trophic levels, due to predation and ontogenic processes. Such an approach, wherein species as such disappear, may be viewed as the ultimate stage in the use of the trophic level metric for ecosystem modelling, providing a simplified but potentially useful caricature of ecosystem functioning and impacts of fishing. This version contains catch trophic spectrum analysis (CTSA) function and corrected versions of the mf.diagnosis and create.ETmain functions.

License GPL

LazyLoad yes

Depends XML

NeedsCompilation no

Repository CRAN

Date/Publication 2013-09-13 15:44:12

R topics documented:

check.table	2
convert.list2tab	3
create.ETdiagnosis	4
create.ETmain	6
create.smooth	7
CTSA.catch.input	9
CTSA.forward	10
ecopath_guinee	11
E_MSY_0.1	13
plot.ETdiagnosis	14
plot.ETmain	15
plot.smooth	16
plot.Transpose	17
plot_ETdiagnosis_isopleth	18
read.ecopath.model	19
saturation	20
Transpose	21

Index	23
--------------	-----------

check.table	<i>Check Ecopath table function</i>
-------------	-------------------------------------

Description

This function enables the verification of input tables based on EwE data and used in the EcoTroph routine. A template is provided in the example: `data(ecopath_guinee)`.

Usage

```
check.table(ecopath)
```

Arguments

ecopath	is the input table used in ET. The different variables are the group name, its trophic level, biomass, production, catches, omnivory index and accessibility (fraction of the group that can be catch assuming an infinite fishing effort).
---------	---

Details

The user have the possibility to check its input table, notably the headings and values (no NA).

Value

The `check.table` function returns notifications related to the input table and its reliability to run EcoTroph (problems in the headings or values).

Author(s)

Colleter Mathieu and Guitton Jerome.

Examples

```
data(ecopath_guinee)
check.table(ecopath_guinee)
```

`convert.list2tab` *convert.list2tab*

Description

`convert.list2tab` enables to convert the list object returned by the `create.ETdiagnosis` function into a list of `data.frames`. These `data.frames` contain calculated variables by TL class and combinations of effort multipliers.

Usage

```
convert.list2tab(diagn.list)
```

Arguments

`diagn.list` is the list object returned by the `create.ETdiagnosis` function.

Value

This function returns a list of `data.frames` : B, B_acc, P, P_acc, Kin, Kin_acc, Y, Fish_mort, Fish_mort_acc and ET_Main_diagnose.

Author(s)

Gatti Paul, Colleter Mathieu, Guitton Jerome.

See Also

`create.ETdiagnosis`

Examples

```
data(ecopath_guinee)
Liste=create.ETdiagnosis(create.ETmain(ecopath_guinee))
Tab=convert.list2tab(Liste)
```

create.ETdiagnosis *ET-Diagnosis*

Description

ET-Transpose provides a picture of an ecosystem under a given fishing mortality. ET-Diagnosis is a routine simulating how this baseline ecosystem would be impacted by increasing or decreasing fishing effort. Fishing effort can be modified per fleet and/or trophic group. Ecosystem-wide effects of altering fishing effort include potential changes in biomass, accessible biomass, production, kinetics and catch trophic spectra, as well as impacts on the mean trophic level of the catch and biomass. Additionally, ET-Diagnosis constitutes a useful exploratory tool for ecosystem-based management. It simulates how reducing or increasing fishing effort and/or preferentially targeting different trophic levels could improve yield at the ecosystem scale. Lastly, ET-Diagnosis allows to view how different assumptions on ecosystem functioning (biomass input control, top-down effect) affect both trophic level specific and ecosystem-wide properties in relation to fishing.

Usage

```
create.ETdiagnosis(data, Mul_eff = NULL, Group = NULL,
fleet.of.interest = NULL, same.mE = NULL, B.Input=NULL,
Beta = NULL, TopD = NULL, FormD = NULL, TLpred = NULL)
```

Arguments

<code>data</code>	is the list object returned by the <code>create.ETmain</code> function.
<code>Mul_eff</code>	is a vector of fishing effort multipliers that the user wants to test. <code>Mul_eff</code> must contain the value 1 (reference state). By default, the function simulates a range of fishing effort multipliers from 0 to 5 for each fleet.
<code>Group</code>	is a character vector of trophic groups that the user specifically wants to impact by changing associated fishing efforts. By default, all trophic groups are equally impacted.
<code>fleet.of.interest</code>	is a character vector of fleet(s) that the user specifically wants to impact by changing associated fishing efforts (default =NULL). This argument is of particular interest if there are more than two fleets because it limits the mE combinations to be tested, and thus the associated computation time.
<code>same.mE</code>	is a logical argument (default=F), if TRUE the same effort multipliers are simultaneously applied to all fleets.
<code>B.Input</code>	is a logical argument (default=F), if TRUE the "Biomass input control" equation is accounted for in EcoTroph equations.
<code>Beta</code>	is a coefficient expressing the extent of the biomass input control. <code>Beta=0</code> refers to an ecosystem where all secondary production originates from grazing on primary producers, and <code>Beta=1</code> to an ecosystem where detritus and/or recruitment contribute to a major part of the biomass input (default=0.2).

TopD	is a coefficient expressing the top-down control, i.e. the fraction of the natural mortality depending on predator abundance. It varies between 0 and 1. The user can specify a numeric value, which is applied to each trophic level, or a numeric vector (of the same length as TL classes), i.e. a value for each TL (default=0.4).
FormD	is a shape parameter varying between 0 and 1. It defines the functional relationship between prey and predators. The value 1 refers to a situation where predators abundance has a linear effect on the speed of the flow of their preys. The user can specify a numeric value, which is applied to each trophic level, or a numeric vector (of the same length as TL classes), i.e. a value for each TL (default=0.5).
TLpred	is the trophic level that the user considers to be the "predator" trophic classes start. The default value is 3.5.

Details

Fleets' names used in the argument 'fleet.of.interest' are the catch column names of the Ecopath input data.frame (e.g. 'catch.1' or 'catch.ind').

Value

This function returns a list of elements referring to each simulated combination of fishing effort multipliers. Each element is a list of two types of results: - Variables characterizing the state and functioning of the modeled ecosystem: biomass, flow, kinetic, catches (total and per fleet) and fishing mortality per trophic level. - Summary statistics (contained in the ET_Main_diagnose): absolute and relative (in comparison with the reference state) total biomass, flow, catches.

Author(s)

Colleter Mathieu, Guitton Jerome and Gatti Paul.

See Also

plot.ETdiagnosis and plot.ETdiagnosis_isopleth to plot the principle graphics resulting from the create.ETdiagnosis function, create.ETmain to create a list of tables used as input in the create.ETdiagnosis function.

Examples

```
data(ecopath_guinee)
#Impacts of global changes in fishing efforts multipliers (in the range 0-5)
create.ETdiagnosis(create.ETmain(ecopath_guinee),same.mE=TRUE)
#Test of all the combinations of fishing effort multipliers per fleet
#(in the range 0-5)
create.ETdiagnosis(create.ETmain(ecopath_guinee))
#With biomass input control
create.ETdiagnosis(create.ETmain(ecopath_guinee),B.Input=TRUE)
#Impacts of changing fishing effort against Barracudas+ and Carangids groups

create.ETdiagnosis(create.ETmain(ecopath_guinee),
Mul_eff=(seq(0,5,.1)),Group=c('Barracudas+', 'Carangids'))
```

create.ETmain *Create ET-Main (ET-Transpose routine)*

Description

This function enables the creation of the ET-Main table (summarizing the principal results/variables in function of the TL classes) and other intermediate tables of the ET-Transpose routine. It provides a picture of an ecosystem under a given fishing mortality.

Usage

```
create.ETmain(ecopath, smooth_type=NULL, sigmaLN_cst=NULL,
pas=NULL, shift=NULL, smooth_param=NULL)
```

Arguments

ecopath	is the input table used in ET (possibly based on Ecopath data). The different variables are the group name, its trophic level, biomass, production, catches, omnivory index and accessibility (fraction of the group that can be catch assuming an infinite fishing effort).
smooth_type	is a parameter of the create.smooth function. It defines the type of sigma calculation for the lognormal distribution. The value for this parameter is 1, 2 or 3. By default smooth_type=1, this defines a constant sigma. By choosing smooth_type=2, the user has the possibility to put a $\sigma_{LN} = \text{smooth_param} * \ln(\text{TL} - 0.05)$, with smooth_param=0.07 and shift=0.95 by default. Smooth_type=3 corresponds to the use of the calculated Omnivory Index (OI) divided by the associated mean TL as sigmaLN.
sigmaLN_cst	is a parameter of the create.smooth function. It defines the value of the constant sigma of the lognormal distribution for smooth_type=1. By default, sigmaLN_cst=0.12.
pas	is a parameter of the create.smooth function. It defines the splitting of the TL classes.
shift	is a parameter of the create.smooth function. It defines the beginning of the smooth function and allows the subtraction of 0.05 in the sigma calculation accounting for the half interval range of the trophic class.
smooth_param	is a parameter of the create.smooth function. It defines the slope of the logarithmically increase of the TL variability with the mean trophic level of the group for smooth_type=2. $\sigma_{LN}(\text{TL})$ is thus defined as $\sigma_{LN}(\text{TL}) = \text{smooth_param} * \ln(\text{TL} - 0.05)$.

Value

This function returns a list containing: the ET-Main table, intermediate matrices (biomass, accessible biomass, flowP...) and a list of matrices corresponding to the different fisheries catches.

Author(s)

Colleter Mathieu, Guitton Jerome and Gatti Paul.

See Also

plot.ETmain to create the principle graphics resulting from the create.ETmain function, create.smooth to create the Smooth table used in this function, Transpose to convert data referring to groups into data referring to TL classes.

Examples

```
data(ecopath_guinee)
create.ETmain(ecopath_guinee)
#Use of the second smooth type
create.ETmain(ecopath_guinee,smooth_type=2)
```

create.smooth

Create Smooth Function

Description

create.smooth is used to create a smooth function. This function enables the conversion of data pertaining to specific taxa or functional groups into data by trophic classes. The main assumption of this Smooth function is that the distribution of the biomass (or catch...) of a trophic group around its mean trophic level follows a lognormal curve. The curve is defined by a mean (the mean TL of the trophic group) and a standard deviation (sigma), which is a measure of the trophic level variability within the group. The distribution is then defined by the lognormal function LN(mean TL, sigma).

Usage

```
create.smooth(tab_input, smooth_type=NULL, sigmaLN_cst=NULL,
pas=NULL, shift=NULL, smooth_param=NULL)
```

Arguments

tab_input	is the input table based on Ecopath data or on independent data. The different variables are the group name, its trophic level, biomass, production on biomass ratio, catches, omnivory index and accessibility (fraction of the group that can be catch assuming an infinite fishing effort) if the input table corresponds to an EwE model. In other case, to simply build trophic spectra, only the group names and their trophic levels are necessary.
smooth_type	defines the type of sigma calculation for the lognormal distribution. Values of this parameter are 1, 2 or 3. By default smooth_type=1, this defines a constant sigma. By choosing smooth_type=2, the user has the possibility to implement a $\sigma_{LN} = \text{smooth_param} * \ln(\text{TL} - 0.05)$, with the parameter smooth_param=0.07 and shift=0.95 by default. Smooth_type=3 corresponds to the use of the omnivory index (OI) in the sigmaLN calculation ($\sigma_{LN} = \text{OI}/\text{TL}$).

<code>sigmaLN_cst</code>	defines the value of the constant sigma of the lognormal distribution in case of <code>smooth_type=1</code> . By default, <code>sigmaLN_cst=0.12</code> .
<code>pas</code>	defines the splitting of the TL classes. By default, <code>pas=0.1</code> .
<code>shift</code>	defines the beginning of the smooth function and allows the subtraction of 0.05 in the sigma calculation accounting for the half interval range of the trophic class. By default, with a constant <code>sigmaLN</code> (<code>smooth_type=1</code>), <code>shift=1.8</code> ; with a function defined <code>sigmaLN</code> (<code>smooth_type=2</code>), <code>shift=0.95</code> ; and with <code>sigmaLN=OI/TL</code> (<code>smooth_type=3</code>), <code>shift=0</code> .
<code>smooth_param</code>	defines the slope of the log-linear increase of the TL variability with the mean trophic level of the group. <code>SigmaLN(TL)</code> is thus defined as <code>sigmaLN(TL)=smooth_param*ln(TL-0.05)</code> . By default, <code>smooth_param=0.07</code> .

Details

The user has the possibility to define `sigmaLN` for each trophic group and also adjust the LN distribution with the `smooth_type`, `sigmaLN_cst`, `smooth_param`, `shift` and `pas` parameters. Different choices are available : a constant sigma, a function defined sigma (log-linear increase) , or a sigma equal to the omnivory index divided by the associated mean TL.

Value

`create.smooth` returns a table of the TL distribution within a trophic class. This table enables the calculation of Trophic Spectra, it is used in the Transpose function.

Author(s)

Colleter Mathieu and Guitton Jerome.

See Also

`plot.smooth` to plot the Smooth function, `Transpose` to build trophic spectra, `plot.Transpose` to plot the trophic spectra.

Examples

```
data(ecopath_guinee)
create.smooth(ecopath_guinee)
create.smooth(ecopath_guinee,sigmaLN_cst=0.11)
create.smooth(ecopath_guinee,smooth_type=2,pas=0.2)
```

CTSA.catch.input	<i>Catch input for CTSA</i>
------------------	-----------------------------

Description

CTSA.catch.input is used to create inputs for the CTSA.forward function. It is a list of data.frames referring to catches per fleet formatted with TL classes in rows and trophic groups in columns.

Usage

```
CTSA.catch.input(catch.group, smooth_type=NULL, sigmaLN_cst=NULL,
pas=NULL, shift=NULL, smooth_param=NULL)
```

Arguments

catch.group	is a data.frame containing: a column group_name, column(s) referring to the catches of each fleet (named 'catch.1', 'catch.2' ...), a column TL specifying the mean TL of each group, and optionally a column OI (omnivory index) used for smooth_type=3.
smooth_type	is a parameter of the create.smooth function. It defines the type of sigma calculation for the lognormal distribution. Values of this parameter are 1, 2 or 3. By default smooth_type=1, this defines a constant sigma. By choosing smooth_type=2, the user has the possibility to implement a $\sigma_{LN} = \text{smooth_param} * \ln(\text{TL} - 0.05)$, with the parameter smooth_param=0.07 and shift=0.95 by default. Smooth_type=3 corresponds to the use of the omnivory index (OI) in the sigmaLN calculation ($\sigma_{LN} = \text{OI}/\text{TL}$).
sigmaLN_cst	is a parameter of the create.smooth function. It defines the value of the constant sigma of the lognormal distribution in case of smooth_type=1. By default, sigmaLN_cst=0.12.
pas	is a parameter of the create.smooth function. It defines the splitting of the TL classes. By default, pas=0.1.
shift	is a parameter of the create.smooth function. It defines the beginning of the smooth function and allows the subtraction of 0.05 in the sigma calculation accounting for the half interval range of the trophic class. By default, with a constant sigmaLN (smooth_type=1), shift=1.8; with a function defined sigmaLN (smooth_type=2), shift=0.95; and with $\sigma_{LN} = \text{OI}/\text{TL}$ (smooth_type=3), shift=0.
smooth_param	is a parameter of the create.smooth function. It defines the slope of the log-linear increase of the TL variability with the mean trophic level of the group. $\sigma_{LN}(\text{TL})$ is thus defined as $\sigma_{LN}(\text{TL}) = \text{smooth_param} * \ln(\text{TL} - 0.05)$. By default, smooth_param=0.07.

Value

CTSA.catch.input returns a list of data.frames, referring to catches per fleet formatted with TL classes in rows and trophic groups in columns.

Author(s)

Gatti Paul, Colleter Mathieu, Guitton Jerome.

See Also

create.smooth, Transpose and CTSA.forward.

Examples

```
data(ecopath_guinee)
catch.group=ecopath_guinee[,c("group_name","TL","catch.1","catch.2")]
Y_test <- CTSA.catch.input(catch.group)
Y_test
```

CTSA.forward

Catch Trophic Spectrum Analysis (CTSA) - Forward

Description

CTSA.forward enables to estimate biomass, biomass flow, kinetic of an ecosystem starting from a biomass or biomass flow at TL=1, catches, transfer efficiencies, and kinetics at virgin state.

Usage

```
CTSA.forward(catch,Prod.TL1=NULL,Biomass.TL1=NULL,TE=NULL,
Kin=NULL,Kin_acc=NULL,temp=NULL,Selec=NULL,TL50=NULL,
asymptote=NULL,slope=NULL,TopD=NULL,FormD=NULL)
```

Arguments

catch	is a formatted list of catch per fleet. It should be a data.frame with TL classes in rows and trophic groups in columns.
Prod.TL1	is the value of biomass flow at TL=1 used to initiate CTSA computations.
Biomass.TL1	is the value of biomass at TL=1 used to initiate CTSA computations. Only one of the arguments Biomass.TL1/Prod.TL1 should be implemented.
TE	is the value of transfer efficiency (in
Kin	is the kinetic of biomass flow at virgin state.
Kin_acc	is the kinetic of accessible biomass flow at virgin state.
temp	is the mean ecosystem temperature used to estimate Kin and/or Kin_acc, if unknown, based on empirical equations function of temperature and TL (see Gasciel et al. (2008)).
Selec	is the selectivity vector used to calculate accessible biomass from total biomass. If unknown, a selectivity curve can be computed using a logistic function which takes as input parameters TL50, asymptote and slope.
TL50	is the TL at first catch, parameter of the logistic selectivity curve.

asymptote	is the asymptote value of the logistic selectivity curve.
slope	is the slope value of the logistic selectivity curve.
TopD	is a coefficient expressing the top-down control, i.e. the fraction of the natural mortality depending on predator abundance. It varies between 0 and 1. The user can specify a numeric value, which is applied to each TL class, or a numeric vector (of the same length as TL classes), i.e. a value for each different TL class (default=0.4).
FormD	is a shape parameter varying between 0 and 1. It defines the functional relationship between prey and predators. The value 1 refers to a situation where predators abundance has a linear effect on the speed of the flow of their preys. The user can specify a numeric value, which is applied to each TL class, or a numeric vector (of the same length as TL classes), i.e. a different value for each TL class (default=0.5).

Value

This function returns a list object containing the catches per fleet, the ET-Main table containing all the variables characterizing the ecosystem state. This list object can be used as an input for the create.ETdiagnosis function in order to test fishing impacts on the "reconstituted ecosystem".

Author(s)

Gatti Paul, Colleter Mathieu, Guitton Jerome.

See Also

CTSA.catch.input to create input data for the CTSA.forward function, and create.ETdiagnosis to run the ET-Diagnosis routine.

Examples

```
data(ecopath_guinee)
catch.group=ecopath_guinee[,c("group_name","TL","catch.1","catch.2")]
Y_test <- CTSA.catch.input(catch.group)
B.TL1=create.ETmain(ecopath_guinee)$ET_Main$B[1]
results <- CTSA.forward(Y_test,Biomass.TL1=B.TL1,
asymptote=.7,TL50=3,slope=5,temp=28,TE=10)
results
```

ecopath_guinee

EcoTroph example dataset: Guinean data

Description

This example dataset is extracted from the 2004 Guinean Ecopath model (Gascuel et al., 2009). It provides a template for the input table formatting, the wanted variables names and the different capabilities of this package (used in function's examples).

Usage

```
data(ecopath_guinee)
```

Format

A data.frame with 35 observations on the following 8 variables.

`group_name` a character vector corresponding to the names of the trophic groups used in the Eco-path model. Has obligatory to be written 'group_name'.

`TL` a numeric vector corresponding to the trophic level of the associated trophic groups. Has obligatory to be written 'TL'.

`biomass` a numeric vector corresponding to the biomass of the associated trophic groups. Has obligatory to be written 'biomass'.

`prod` a numeric vector corresponding to the production on biomass ratio. For the Detritus groups (no P/B value entered in Ecopath), put 0 as a value. Has obligatory to be written 'prod'.

`catch.1` a numeric vector corresponding to the catch of the artisanal fleet. A value must be entered for all groups, with a 0-value if no catch are made. Has obligatory to be written 'catch.something'.

`catch.2` a numeric vector corresponding to the catch of the industrial fleet. A value must be entered for all groups, with a 0-value if no catch are made. Has obligatory to be written 'catch.something'.

`accessibility` a numeric vector corresponding to the fraction of the trophic group that can be catch assuming an infinite fishing effort. Has obligatory to be written 'accessibility'.

`OI` a numeric vector corresponding to the omnivory index calculated by Ecopath for each trophic group. Has obligatory to be written 'OI'.

Details

No NA are accepted in the dataset (0 for the P/B of the detritus groups, 0 for the catch...). Follow the instructions stated in the variables descriptions. Different fleets can be entered in the model using the following system: `catch.1`, `catch.2`, `catch.whatyouwant...` If there is only one fleet, you just have to put `catch` as a variable name.

Source

Gascuel et al. (2009) Impact de la peche sur l'ecosysteme marin de Guinee - Modelisation EwE 1985/2005 -

Examples

```
data(ecopath_guinee)
ecopath_guinee
names(ecopath_guinee)
```

E_MSY_0.1

*E_MSY_0.1: Exploitation indices***Description**

E_MSY_0.1 computes two indices of exploitation: Emsy or Fmsy (maximum sustainable yield), and E0.1 or F0.1 ("start" of full exploitation) per TL class.

Usage

```
E_MSY_0.1(data, Mul_eff=NULL, B.Input=NULL, Beta=NULL,
TopD=NULL, FormD=NULL, TLpred=NULL, maxTL=NULL)
```

Arguments

data	is the list object returned by the create.ETmain function.
Mul_eff	is a parameter of the create.ETdiagnosis function. It is a vector of fishing effort multipliers that the user wants to test. Mul_eff must contain the value 1 (reference state). By default, the function simulates a range of fishing effort multipliers from 0 to 5 for each fleet.
B.Input	is a parameter of the create.ETdiagnosis function. It is a logical argument (default=F), if TRUE the "Biomass input control" equation is accounted for in EcoTroph equations.
Beta	is a parameter of the create.ETdiagnosis function. It is a coefficient expressing the extent of the biomass input control. Beta=0 refers to an ecosystem where all secondary production originates from grazing on primary producers, and Beta=1 to an ecosystem where detritus and/or recruitment contribute to a major part of the biomass input (default=0.2).
TopD	is a parameter of the create.ETdiagnosis function. It is a coefficient expressing the top-down control, i.e. the fraction of the natural mortality depending on predator abundance. It varies between 0 and 1. The user can specify a numeric value, which is applied to each trophic level, or a numeric vector (of the same length as TL classes), i.e. a value for each TL (default=0.4).
FormD	is a parameter of the create.ETdiagnosis function. It is a shape parameter varying between 0 and 1. It defines the functional relationship between prey and predators. The value 1 refers to a situation where predators abundance has a linear effect on the speed of the flow of their preys. The user can specify a numeric value, which is applied to each trophic level, or a numeric vector (of the same length as TL classes), i.e. a value for each TL (default=0.5).
TLpred	is a parameter of the create.ETdiagnosis function. It is the trophic level that the user considers to be the "predator" trophic classes start. The default value is 3.5.
maxTL	is a numeric string indicating the maximum TL for which indices are computed.

Details

For any TL class, if E0.1 and/or Emsy value(s) is(are) equal to the maximum effort multiplier tested (max(Mul_eff)), then E/F0.1 and/or E/Fmsy are set equal to NA.

Value

The E_MSX_0.1 function returns a data.frame containing Fmsy, Emsy, F0.1 and E0.1 per TL class.

Author(s)

Gatti Paul, Colleter Mathieu, Guitton Jerome.

Examples

```
data(ecopath_guinee)
E_MSX_0.1(create.ETmain(ecopath_guinee))
```

plot.ETdiagnosis *Plot Diagnosis Function*

Description

This function enables the creation of the principle graphics resulting from the create.ETdiagnosis function.

Usage

```
## S3 method for class 'ETdiagnosis'
plot(x,scale=NULL,maxrange=NULL,legend.cex=NULL,ask=interactive(),...)
```

Arguments

x	is the list object returned by the create.ETdiagnosis function.
scale	is the scale parameter of the Biomass Trophic Spectra, can be log or by default the standard scale of results.
maxrange	is the maximum TL wanted for the x-axis. By default maxrange = 5.5.
legend.cex	defines the value of the cex for the legend.
ask	default value is interactive. Parameter used to enable the user to control the display of each graph.
...	plot other arguments

Details

The scale parameter controls the scale of the BTS y-axis, it corresponds by default to the scale of results, but it's usually more practical to use a log scale.

Value

The function returns the principal plots of the global ET-Diagnosis routine: the graphics of the biomass, accessible biomass...rates for the different effort multipliers, the Biomass Trophic Spectra (BTS) for the different effort multipliers, the B/Bref(mE=1) and Y/Yref graphs for the main TL classes and the Catch Trophic Spectra (CTS) (global and per fleet).

Author(s)

Colleter Mathieu, Guitton Jerome and Gatti Paul.

Examples

```
data(ecopath_guinee)
diagn.list<-create.ETdiagnosis(create.ETmain(ecopath_guinee),same.mE=TRUE)
plot(diagn.list)
```

plot.ETmain

Plot ET-Main

Description

This function enables the display of the principle plots resulting from the create.ETmain function: Biomass Trophic Spectra, Accessible Biomass Trophic Spectra, Catch by fleet Trophic spectra, Total Catch Trophic Spectra and other summary plots.

Usage

```
## S3 method for class 'ETmain'
plot(x,scale1=NULL,scale2=NULL,scale3=NULL,legend.cex=NULL,
ask=interactive(),...)
```

Arguments

x	is the list of tables returned by the create.ETmain function.
scale1	defines the scale of the Biomass plots y-axis: can be log or not.
scale2	defines the scale of the Accessible Biomass plots y-axis: can be log or not.
scale3	defines the scale of the Catch by fleet plots y-axis: can be log or not.
legend.cex	defines the value of the cex for the legend.
ask	default value is interactive. Parameter used to enable the user to control the display of each graph.
...	plot other arguments

Value

The function returns the principal graphics of the global ET-Transpose routine: the Biomass Trophic Spectra, the Accessible Biomass Trophic Spectra and other graphics, notably the Catch Trophic Spectra.

Author(s)

Colleter Mathieu and Guitton Jerome.

See Also

create.smooth function to create the Smooth, Transpose to calculate the data transposition into trophic spectra, create.ETmain to create a list of table including the ET-Main table.

Examples

```
data(ecopath_guinee)
plot(create.ETmain(ecopath_guinee), scale1=log)
plot(create.ETmain(ecopath_guinee), scale1=log, scale3=log)
```

plot.smooth

Plot Smooth Function

Description

plot.smooth is used to plot the Smooth function. This function enables the user to see the TL distributions around their mean trophic levels.

Usage

```
## S3 method for class 'smooth'
plot(x,...)
```

Arguments

x is the table returned by the create.smooth function.
... plot other arguments

Value

The function returns a plot of the Smooth with the TL classes in abscissa.

Author(s)

Colleter Mathieu and Guitton Jerome.

See Also

create.smooth function to create the Smooth, Transpose to calculate the data transposition into trophic spectra.

Examples

```
data(ecopath_guinee)
plot(create.smooth(ecopath_guinee))
plot(create.smooth(ecopath_guinee, smooth_type=2))
```

plot.Transpose *Plot Transpose Function*

Description

This function returns the two principal plots of the Transpose function : a plot by group and the associated Trophic Spectra (CTS, BTS...).

Usage

```
## S3 method for class 'Transpose'
plot(x, title = NULL, scale = NULL, legend.cex, ...)
```

Arguments

x	is the table returned by the Transpose function.
title	defines the title of the graph.
scale	defines the scale of the y-axis: can be log or not.
legend.cex	defines the value of the cex for the legend.
...	plot other arguments

Value

The function returns the principal plots of the Transpose function : the different Trophic Spectra (CTS, BTS...) depending on the transposed column, and a plot by group.

Author(s)

Colleter Mathieu and Guitton Jerome.

See Also

create.smooth function to create the Smooth, plot.smooth to plot the smooth function, Transpose to calculate the data transposition into trophic spectra.

Examples

```
data(ecopath_guinee)
smoothed<-create.smooth(ecopath_guinee)
plot(Transpose(smoothed,ecopath_guinee,"biomass"),scale=log)
plot(Transpose(smoothed,ecopath_guinee,"catch.1"),
title="Small Scale Fishery Catch")
```

```
plot_ETdiagnosis_isopleth
```

Diagnosis Isopleth Plot Function

Description

This function enables to plot the mixed impacts of changes in fishing effort for two fleets (or groups of fleets).

Usage

```
plot_ETdiagnosis_isopleth(x,fleet1,fleet2,var=NULL,n.level=NULL,
relative=NULL,name.fleet1=NULL,name.fleet2=NULL,color=NULL,
ask=interactive())
```

Arguments

x	is the list object returned by the create.ETdiagnosis function.
fleet1	is a character vector of fleets for which fishing efforts are equally changed.
fleet2	is a second character vector of fleets for which fishing efforts are equally changed. Fishing efforts remain unchanged for fleets not assigned in fleet1 or fleet2. If fleet2 is NULL, all fleets not assigned in fleet1 are assigned in fleet2. If the argument 'fleet.of.interest' has been assigned in the function create.ETdiagnosis, fleet1=fleet.of.interest and thus fleet2 is composed of the remaining fleet(s) (not assigned in fleet.of.interest).
var	is a character vector of plotted variables (TOT_biomass,TOT_biomass_acc,Y,Y_fleet1, Y_fleet2,TL_TOT_biomass,TL_TOT_biomass_acc,TL_Y,TL_Y_fleet1,TL_Y_fleet2). All the variables are plotted by default.
n.level	is a numeric string, specifying the number of plotted isopleth areas (7 is the default value).
relative	is a logical string (by default relative=F), specifying if the variables have to be plotted in absolute or relative values (in comparison with reference state, Mul_eff=1). Note that if relative=TRUE, mean trophic level in biomass or catches (TL_TOT_biomass,TL_Y,...) are not plotted.
name.fleet1	is a character string used to implement x-axis name. By default name.fleet1='fleet 1'.
name.fleet2	is a character string used to implement y-axis name. By default name.fleet2='fleet 2'. If the argument fleet.of.interest has been assigned in the function create.ETdiagnosis, name.fleet1 = 'fleet of interest' and name.fleet2 = 'other fleets'.

color	is a vector of colors, the length of this vector should be equal to the value of levels. By default, color=rainbow(n=levels).
ask	default value is interactive. Parameter used to enable the user to control the display of each graph.

Details

Fleets' names used in the arguments 'fleet1' and 'fleet2' are the catch column names of ecopath input dataframe (e.g. 'catch.1' or 'catch.ind').

Value

The function plot.ETdiagnosis_isopleth plots isopleth areas of different variables (biomass, accessible biomass, total catch, mean trophic level of the total biomass...) with fishing effort multipliers for fleet1 in x-axis and for fleet2 in y-axis. This enables to display mixed impacts of changing fishing efforts of two groups of fleets.

Author(s)

Gatti Paul, Colleter Mathieu, Guitton Jerome.

Examples

```
data(ecopath_guinee)
diagn.list=create.ETdiagnosis(create.ETmain(ecopath_guinee))
plot_ETdiagnosis_isopleth(diagn.list,fleet1='catch.1',fleet2='catch.2')
plot_ETdiagnosis_isopleth(diagn.list,fleet1='catch.1',fleet2='catch.2',
relative=TRUE)
```

read.ecopath.model *Input data import function (from an xml file)*

Description

This function loads input data from an xml file created by the user, or exported from the EwE EcoTroph plug-in, or from a web service associated to a database populated with parameters of several EwE models.

Usage

```
read.ecopath.model(filename)
```

Arguments

filename is the address of the file the user wants to import.

Value

This function returns a data.frame containing all the column needed to run the EcoTroph R package.

Author(s)

Colleter Mathieu and Guitton Jerome.

See Also

check.table to control the reliability of the dataset.

saturation

Sigma Saturation Function

Description

This function enables an other calculation for the sigma of the create.smooth function. Sigma is calculated on the base of a saturation function reflecting a biological reasoning about the variability of the TL within trophic classes: the variability increases with the TL and reaches a plateau after a certain TL.

Usage

```
saturation(sigma_inf = NULL, coeff = NULL, pas = NULL)
```

Arguments

sigma_inf	defines the value of the curve's plateau.
coeff	defines the value of the slope.
pas	defines the splitting of the TL classes.

Details

By default sigma is constant. This function enables an other user choice reflecting a different reasoning.

Value

saturation returns a vector of values for the sigma used in the create.smooth function.

Author(s)

Colleter Mathieu and Guitton Jerome.

See Also

create.smooth function to create the Smooth, plot.smooth to plot the smooth function.

Examples

```
plot(saturation())
lines(saturation(0.2))
text(48,0.18,"sigma_inf=0.2")
lines(saturation(coeff=0.5))
text(48,0.35,"coeff=0.5")
```

Transpose

ET Transpose Function

Description

Transpose enables the conversion of data pertaining to specific taxa or functional groups into data by trophic class. Data can represent catches, biomasses or production in order to produce continuous distributions of those variables over trophic levels.

Usage

```
Transpose(tab_smooth, tab_input, column)
```

Arguments

<code>tab_smooth</code>	is the table returned by the <code>create.smooth</code> function.
<code>tab_input</code>	is the input table based on Ecopath data or on independent data. The different variables are the group name, its trophic level, biomass, production on biomass ratio, catches, omnivory index and accessibility (fraction of the group that can be catch assuming an infinite fishing effort) if the input table corresponds to an EwE model. In other case, to simply build trophic spectra, only the group names, their trophic levels and related variables are necessary.
<code>column</code>	is the <code>tab_input</code> table column name of the variable the user wants to transpose (for example "biomass" or "catch").

Value

The function returns a table of the choosen variable split into the defined trophic classes per group.

Author(s)

Colleter Mathieu and Guitton Jerome.

See Also

`create.smooth` function to create the `Smooth`, `plot.smooth` to plot the smooth function, `plot.Transpose` to plot the associated trophic spectra.

Examples

```
data(ecopath_guinee)
Transpose(create.smooth(ecopath_guinee),ecopath_guinee,"biomass")
Transpose(create.smooth(ecopath_guinee),ecopath_guinee,"catch.1")
```

Index

*Topic **\textasciitildeCTSA**

CTSA.catch.input, 9
CTSA.forward, 10

*Topic **\textasciitildeDiagnosis**

convert.list2tab, 3
create.ETdiagnosis, 4
E_MSY_0.1, 13
plot.ETdiagnosis, 14
plot_ETdiagnosis_isopleth, 18

*Topic **\textasciitildeEcoTroph**

check.table, 2
convert.list2tab, 3
create.ETdiagnosis, 4
create.ETmain, 6
create.smooth, 7
CTSA.catch.input, 9
CTSA.forward, 10
E_MSY_0.1, 13
ecopath_guinee, 11
plot.ETdiagnosis, 14
plot.ETmain, 15
plot.smooth, 16
plot.Transpose, 17
plot_ETdiagnosis_isopleth, 18
read.ecopath.model, 19
saturation, 20
Transpose, 21

*Topic **\textasciitildeSmooth**

create.smooth, 7
plot.smooth, 16
saturation, 20

*Topic **\textasciitildeTranspose**

create.ETmain, 6
plot.ETmain, 15
plot.Transpose, 17
Transpose, 21

*Topic **\textasciitildedatasets**

ecopath_guinee, 11

check.table, 2

convert.list2tab, 3
create.ETdiagnosis, 4
create.ETmain, 6
create.smooth, 7
CTSA.catch.input, 9
CTSA.forward, 10

E_MSY_0.1, 13
ecopath_guinee, 11

plot.ETdiagnosis, 14
plot.ETmain, 15
plot.smooth, 16
plot.Transpose, 17
plot_ETdiagnosis_isopleth, 18

read.ecopath.model, 19

saturation, 20

Transpose, 21