

Package: EcoTroph (via r-universe)

September 6, 2024

Encoding UTF-8

Type Package

Title An implementation of the EcoTroph Ecosystem modelling approach

Version 1.6.1

Date 2022-04-13

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 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, utils, stats, graphics, grDevices

RoxygenNote 7.1.2

Repository <https://polehalieutique.r-universe.dev>

RemoteUrl <https://github.com/polehalieutique/ecotroph>

RemoteRef HEAD

RemoteSha d59c3e098d69e02178028ca7b92c8f35bdb21ba9

Contents

check.table	2
convert.list2tab	3
create.ETdiagnosis	3
create.ETmain	6
create.smooth	7
CTSA.catch.input	8
E0.1	10
ecopath_guinee	10
Emsy	11
E_MSY_0.1	12
mf.diagnosis	13
plot.ETdiagnosis	14
plot.ETmain	15
plot.smooth	16
plot.Transpose	17
plot_ETdiagnosis_isopleth	18
read.ecopath.model	19
regPB	20
regPB.ac	20
saturation	20
Transpose	21

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

check.table	<i>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.</i>
-------------	---

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

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

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).
---------	--

Examples

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

convert.list2tab	<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

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

Arguments

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

Examples

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

create.ETdiagnosis	<i>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.</i>
--------------------	--

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

data	is the list object returned by the create.ETmain function.
Mul_eff	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.
Group	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.
fleet.of.interest	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.
same.mE	is a logical argument (default=F), if TRUE the same effort multipliers are simultaneously applied to all fleets.
B.Input	is a logical argument (default=F), if TRUE the "Biomass input control" equation is accounted for in EcoTroph equations.
Beta	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 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.

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	<i>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.</i>
---------------	--

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 log-linearly 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.

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 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	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	defines the value of the constant sigma of the lognormal distribution in case of smooth_type=1. By default, sigmaLN_cst=0.12.
pas	defines the splitting of the TL classes. By default, pas=0.1.
shift	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 sigmaLN=OI/TL (smooth_type=3), shift=0.
smooth_param	defines the slope of the log-linear increase of the TL variability with the mean trophic level of the group. SigmaLN(TL) is thus defined as sigmaLN(TL)=smooth_param*ln(TL-0.05). By default, smooth_param=0.07.

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.

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. 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 σ_{LN} .
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 log-linearly 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

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.

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
```

E0.1	<i>E0.1</i>
------	-------------

Description

E0.1

Usage

E0.1(TL, Y, Fish_mort)

Arguments

TL	Trophic level
Y	Catches
Fish_mort	Fishing mortality

ecopath_guinee	<i>EcoTroph example dataset: Guinean data</i>
----------------	---

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 Ecopath 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)
```

Emsy

E_MSY

Description

E_MSY

Usage

```
Emsy(TL, Y, Fish_mort)
```

Arguments

TL	Trophic level
Y	Catches
Fish_mort	Fishing mortality

E_MSY_0.1	<i>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.</i>
-----------	--

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 ($\max(\text{Mul_eff})$), then E/F0.1 and/or E/Fmsy are set equal to NA.

Value

The E_MSY_0.1 function returns a data.frame containing Fmsy, Emsy, F0.1 and E0.1 per TL class.

Examples

```
data(ecopath_guinee)
E_MSY_0.1(create.ETmain(ecopath_guinee))
```

mf.diagnosis	<i>Effort multiplier diagnosis</i>
--------------	------------------------------------

Description

Effort multiplier diagnosis

Usage

```
mf.diagnosis(
  x,
  ET_Main,
  catch.list,
  TL_out,
  fleet,
  n.fleet,
  Fish_mort_ref,
  Fish_mort_acc_ref,
  B.Input,
  Beta,
  TopD,
  FormD,
  TLpred,
  n.TL,
  range.TLpred,
  lim.high.TL,
  range.highTL
)
```

Arguments

x	is the list object returned by the create.ETdiagnosis function.
ET_Main	is the list object returned by the create.ETmain function.
catch.list	catches for each fleet

TL_out	Maximum TL to consider
fleet	List of available fleet
n.fleet	Number of fleet
Fish_mort_ref	Fishing mortality
Fish_mort_acc_ref	Accessible Fissing mortality
B.Input	Biomass
Beta	Beta parameters of the formula
TopD	Parameters of the formula
FormD	Parameters of the formula
TLpred	Trophic level for predators
n.TL	Number of trophic levels
range.TLpred	Range of trophic level for predators
lim.high.TL	Limit for high trophic levels
range.highTL	Range for High trophic levels

plot.ETdiagnosis	<i>This function enables the creation of the principle graphics resulting from the create.ETdiagnosis function. 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).</i>
------------------	---

Description

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

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

Examples

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

plot.ETmain	<i>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.</i>
-------------	--

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.

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

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

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

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 = NULL, ...)
```

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

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

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

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 = NULL,
  fleet2 = NULL,
  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) 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').

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.

See Also

check.table to control the reliability of the dataset.

regPB	<i>function used to compute pB for the highest trophic levels @param compteur @param pb.mf @param TL_out @param range.highTL</i>
-------	--

Description

function used to compute pB for the highest trophic levels @param compteur @param pb.mf @param TL_out @param range.highTL

Usage

```
regPB(compteur, pb.mf, TL_out, range.highTL)
```

regPB.ac	<i>function used to compute pB for the highest trophic levels and accessible biomass @param compteur @param pb.mf @param TL_out @param range.highTL</i>
----------	---

Description

function used to compute pB for the highest trophic levels and accessible biomass @param compteur @param pb.mf @param TL_out @param range.highTL

Usage

```
regPB.ac(compteur, pb.mf.ac, TL_out, range.highTL)
```

saturation	<i>Sigma Saturation Function</i>
------------	----------------------------------

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

<code>sigma_inf</code>	defines the value of the curve's plateau.
<code>coeff</code>	defines the value of the slope.
<code>pas</code>	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.

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

Transpose enables the conversion of data pertaining to specific taxa or fonctionnal 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.

Description

Transpose enables the conversion of data pertaining to specific taxa or fonctionnal 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").

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

* **~EcoTroph**

ecopath_guinee, [10](#)

* **~datasets**

ecopath_guinee, [10](#)

check.table, [2](#)

convert.list2tab, [3](#)

create.ETdiagnosis, [3](#)

create.ETmain, [6](#)

create.smooth, [7](#)

CTSA.catch.input, [8](#)

E0.1, [10](#)

E_MSY_0.1, [12](#)

ecopath_guinee, [10](#)

Emsy, [11](#)

mf.diagnosis, [13](#)

plot.ETdiagnosis, [14](#)

plot.ETmain, [15](#)

plot.smooth, [16](#)

plot.Transpose, [17](#)

plot_ETdiagnosis_isopleth, [18](#)

read.ecopath.model, [19](#)

regPB, [20](#)

regPB.ac, [20](#)

saturation, [20](#)

Transpose, [21](#)