

Package ‘phenology’

August 25, 2014

Type Package

Title Tools to manage a parametric function that describes phenology

Version 3.72

Date 2014-08-24

Author Marc Girondot

Maintainer Marc Girondot <marc.girondot@u-psud.fr>

Depends fields, zoo, coda, shiny, R (>= 2.14.0)

Suggests RNetCDF, XML, maps

Description Functions to fit and test the phenology of species based on counts. The package includes also several useful more general functions.

License GPL-2

LazyData yes

LazyLoad yes

URL http://max2.esse.u-psud.fr/epc/conservation/Girondot/Publications/Marine_Turtles_Nesting_Season.html

NeedsCompilation no

Repository CRAN

Date/Publication 2014-08-25 10:21:17

R topics documented:

phenology-package	3
adapt_parameters	4
add_phenology	5
add_SD	7
as.mcmc.mcmcComposite	8
as.par.mcmcComposite	9
asc	10
barplot_errbar	10
BE_to_LBLE	12
ChangeCoordinate	13
chr	13
clean.knitr	14
compare_AIC	15
compassRose2	16
convert.ts	17
extract_result	18
fit_phenology	19
getTide	20
Gratiot	21
growlnotify	22
ind_long_lat	23
LBLE_to_BE	24
LBLE_to_L	25
likelihood_phenology	26
local.search	27
logLik.phenology	28
L_to_LBLE	29
map.scale2	30
map_Gratiot	31
map_phenology	31
MinBMinE_to_Min	33
moon_phase	34
par_init	35
phenology	36
phenology_MHmcmc	37
phenology_MHmcmc_p	38
phenology_swot	40
plot.mcmcComposite	41
plot.phenology	42
plot.phenologymap	44
plot_add	45
plot_delta	46
plot_errbar	47
plot_phi	49
print.phenology	50
print.phenologymap	51

print.phenologyout	53
read_folder	54
remove_site	55
result_Gratiot	56
result_Gratiot1	57
result_Gratiot2	58
result_Gratiot_Flat	59
result_Gratiot_mcmc	60
ScalePreviousPlot	60
shift_sinusoid	61
summary.mcmcComposite	63
summary.phenology	64
summary.phenologymap	65
summary.phenologyout	67
toggle_Min_PMin	68

Index	70
--------------	-----------

phenology-package	<i>The package phenology</i>
-------------------	------------------------------

Description

A package to fit seasonality counts

Details

Fit a parametric function that describes phenology

Package:	phenology
Type:	Package
Version:	3.72 build 379
Date:	2014-08-24
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Marc Giron dot <marc.giron dot@u-psud.fr>

References

Giron dot, M. 2010. Estimating density of animals during migratory waves: application to marine turtles at nesting site. *Endangered Species Research*, 12, 85-105.

Giron dot, M. 2010. Editorial: The zero counts. *Marine Turtle Newsletter*, 129, 5-6.

See Also

Girondot, M., Rivalan, P., Wongsopawiro, R., Briane, J.-P., Hulin, V., Caut, S., Guirlet, E. & Godfrey, M. H. 2006. Phenology of marine turtle nesting revealed by a statistical model of the nesting season. *BMC Ecology*, 6, 11.

Examples

```
## Not run:
library(phenology)
# Read a file with data
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)
data(result_Gratiot)
# Plot the phenology and get some stats
output<-plot(result_Gratiot)

## End(Not run)
```

adapt_parameters	<i>Extract the parameters from a set of parameters to be used with another dataset.</i>
------------------	-----------------------------------------------------------------------------------------

Description

The function "adapt_parameters" extracts the set of parameters to be used with a subset of data. All the unnecessary parameters are removed. It can be used when a set of beaches are fitted first and after only one of these beaches is fitted again.

Usage

```
adapt_parameters(data = stop("Datasets is mandatory for this function"),
parameters = stop("Set of parameters is mandatory for this function"))
```

Arguments

parameters	A set of parameters
data	A dataset of counts

Details

adapt_parameters get the fitted parameters from a result object.

Value

Return the set of parameters

Author(s)

Marc Girondot

Examples

```
library(phenology)
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.esse.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
refdate <- as.Date("2001-01-01")
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=refdate, format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Add unnecessary parameters to parg
parg <- c(parg, Max_dummybeach=2, Peak_dummybeach=123)
# Extract the fitted parameters
parg1<-adapt_parameters(data=data_Gratiot, parameters=parg)
```

add_phenology

Create a new dataset or add a timeserie to a previous dataset.

Description

To create a new dataset, the syntaxe is
`data<-add_phenology(add=newdata, name="Site", reference=as.Date('2001-12-31'), format='` To
add a dataset to a previous one, the syntaxe is
`data<-add_phenology(previous=previousdata, add=newdata, name='Site',`
`reference=as.Date('2001-12-31'), adjust_ref=TRUE, format='` To add several timeseries at the same
time with ' `data<-add_phenology(add=list(newdata1, newdata2), name=c('Site1', 'Site2'),`
`reference=as.Date('2001-12-31'), format=c('` The dataset to be added must include 2 or 3 columns.
The first one is the date in the format specified by
the parameter `format=`. If the number of nests is known
for an exact data, then only one date must be indicated
If the number of nests is known for a range of date, the
first and last dates must be separated but a - (dash).
For example: `1/2/2000-10/2/2000`
The second column is the number of nests observed for
this date or this range of dates.
The third column is optional and is the name of the rookery.

If only two columns are indicated, the name can be indicated as a parameter of the function with `name=`. If no name is indicated, the default name `Site` will be used, but take care, only one rookery of this name can be used.

Several rookeries can be included in the same file but in this case the rookery name is obligatory at the third column.

The simplest use of this function is just:

```
phen <- add_phenology()
```

Usage

```
add_phenology(add = file.choose(), name = NULL, reference = NULL,
  month_ref = NULL, header = NULL, format = NULL, previous = NULL,
  silent = FALSE, help = FALSE)
```

Arguments

<code>previous</code>	Name of previous data formatted with <code>add_phenology</code> or <code>NULL</code> [default] if no previous data exists
<code>add</code>	The data to be added. It can be a set of several entities that uses the same reference and date format
<code>name</code>	The name of the monitored site
<code>reference</code>	<code>as.Date('2001-12-31')</code> The date used as 1st date
<code>month_ref</code>	If no reference date is given, use this month as a reference
<code>header</code>	If the data is read from a file, can be used to force header or not
<code>format</code>	The format of the date in the file. Several format can be set and the last one that give compatible result is used
<code>silent</code>	Does information about added timeseries is shown
<code>help</code>	If <code>TRUE</code> , an help is displayed

Details

`add_phenology` creates a new dataset.

Value

Return a list of formatted data that can be used with `fit_phenology()`

Author(s)

Marc Girondot

Examples

```

## Not run:
library(phenology)
# Read a file with data
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
refdate <- as.Date("2001-01-01")
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=refdate, format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot, parametersfit=parg,
parametersfixed=NULL, trace=1)
data(result_Gratiot)
# Plot the phenology and get some stats
output<-plot(result_Gratiot)

## End(Not run)

```

add_SD

Add SD for a fixed parameter.

Description

This function is used to add standard deviation for a fixed parameter.

Usage

```
add_SD(parametersfixed = NULL, parameters = NULL, SD = NULL,
help = FALSE)
```

Arguments

parametersfixed	Set of fixed parameters
parameters	Set of current parameters
SD	Standard deviation value to be added
help	If TRUE, an help is displayed

Details

add_SD adds SD for a fixed parameter.

Value

The parameters set with the new SD value

Author(s)

Marc Girondot

Examples

```
library(phenology)
# Generate a set of fixed parameter: Flat and Min
pfixed<-c(Flat=0, Min=0)
# Add SD for the Flat parameter
pfixed<-add_SD(parametersfixed=pfixed, parameters="Flat", SD=5)
```

as.mcmc.mcmcComposite *Extract mcmc object from the result of phenology_MHmcmc to be used with coda package*

Description

Take a mcmcComposite object and create a mcmc.list object

Usage

```
## S3 method for class 'mcmcComposite'
as.mcmc(x)
```

Arguments

x A result MHmcmc search

Details

as.mcmc Extract mcmc object from the result of phenology_MHmcmc to be used with coda package

Value

A mcmc.list object

Author(s)

Marc Girondot

Examples

```
## Not run:
library(phenology)
data(result_Gratiot_mcmc)
mcmc <- as.mcmc(result_Gratiot_mcmc)

## End(Not run)
```

as.par.mcmcComposite *Extract parameters at maximum likelihood from the result of MHmcmc*

Description

Take a mcmcComposite object and create a vector object with parameter value at maximum likelihood. It also says at which iteration the maximum likelihood has been observed.

Usage

```
as.par.mcmcComposite(x)
```

Arguments

x A result from MHmcmc search

Details

as.par.mcmcComposite Extract parameters at maximum likelihood from the result of MHmcmc

Value

A vector with parameters at maximum likelihood

Author(s)

Marc Girondot

Examples

```
## Not run:  
library(phenology)  
data(result_Gratiot_mcmc)  
x <- as.par.mcmcComposite(result_Gratiot_mcmc)  
  
## End(Not run)
```

asc	<i>Return the codes (in UTF-8) of a string</i>
-----	------------------------------------------------

Description

Return the codes (in UTF-8) of a string

Usage

```
asc(x)
```

Arguments

x The string to be analyzed

Details

asc returns the codes (in UTF-8) of a string

Value

A vector with UTF-8 codes of a string

Author(s)

Based on this blog: <http://datadebrief.blogspot.com/2011/03/ascii-code-table-in-r.html>

Examples

```
asc("abcd")
asc("ABCD")
```

barplot_errbar	<i>Plot a barplot graph with error bar on y</i>
----------------	-------------------------------------------------

Description

To plot data, just use it as a normal barplot but add the `errbar.y` values or `errbar.y.minus`, `errbar.y.plus` if bars for y axis are asymmetric. Use `y.plus` and `y.minus` to set absolute limits for error bars. Note that `y.plus` and `y.minus` have priority over `errbar.y`, `errbar.y.minus` and `errbar.y.plus`.

Usage

```
barplot_errbar(..., errbar.y = NULL, errbar.y.plus = NULL,
  errbar.y.minus = NULL, y.plus = NULL, y.minus = NULL,
  errbar.tick = 1/50, errbar.lwd = par("lwd"), errbar.lty = par("lty"),
  errbar.col = par("fg"), add = FALSE)
```

Arguments

...	Parameters for barplot() such as main= or ylim=
errbar.y	The length of error bars for y. Recycled if necessary.
errbar.y.plus	The length of positive error bars for y. Recycled if necessary.
errbar.y.minus	The length of negative error bars for y. Recycled if necessary.
y.plus	The absolut position of the positive error bar for y. Recycled if necessary.
y.minus	The absolut position of the nagative error bar for y. Recycled if necessary.
errbar.tick	Size of small ticks at the end of error bars defined as a proportion of total width or height graph size.
errbar.lwd	Error bar line width, see par("lwd")
errbar.lty	Error bar line type, see par("lwd")
errbar.col	Error bar line color, see par("col")
add	If true, add the graph to the previous one.

Details

plot_errbar plot a barplot with error bar on y

Value

Nothing

Author(s)

Marc Girondot

Examples

```
barplot_errbar(rnorm(10, 10, 3),
  xlab="axe x", ylab="axe y", bty="n",
  errbar.y.plus=rnorm(10, 1, 0.1), col=rainbow(10),
  names.arg=paste("Group",1:10), cex.names=0.6)
y <- rnorm(10, 10, 3)
barplot_errbar(y,
  xlab="axe x", ylab="axe y", bty="n",
  y.plus=y+2)
```

 BE_to_LBLE

Transform a set of parameters from Begin End to LengthB LengthE.

Description

This function is used to transform a set of parameters that uses Begin, Peak and End to a set of parameters that uses LengthB, Peak and LengthE.

Usage

```
BE_to_LBLE(parameters = NULL, help = FALSE)
```

Arguments

parameters	Set of current parameters
help	If TRUE, an help is displayed

Details

BE_to_LBLE transforms a set of parameters from Begin End format to LengthB LengthE.

Value

Return the set of modified parameters

Author(s)

Marc Girondot

Examples

```
## Not run:
# Read a file with data
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt",
header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
refdate <- as.Date("2001-01-01")
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=refdate, format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Change the parameters to Begin End format
parg1<-LBLE_to_BE(parameters=parg)
# And change back to LengthB LengthE.
parg2<-BE_to_LBLE(parameters=parg1)
```

ChangeCoordinate *Return a value in a changed coordinate*

Description

Return a value in a changed coordinate

Usage

```
ChangeCoordinate(x = stop("At least one value to convert must be provided"),
  initial = stop("Set of two values must be provided as references"),
  transformed = stop("Set of two transformed values must be provided"))
```

Arguments

x	value to convert
initial	Set of two values in the original system
transformed	Set of the two values in the converted system

Details

ChangeCoordinate returns a value in a changed coordinate

Value

A value in the new system

Author(s)

Marc Girondot

Examples

```
ChangeCoordinate(x=c(10, 20), initial=c(1, 100), transformed=c(0, 1))
```

chr *Return the characters defined by the codes*

Description

Return a string with characters defined by the codes

Usage

```
chr(n)
```

Arguments

n The vector with codes

Details

chr returns the characters defined by the codes

Value

A string with characters defined by the codes

Author(s)

Based on this blog: <http://datadebrief.blogspot.com/2011/03/ascii-code-table-in-r.html>

Examples

```
chr(65:75)
chr(unlist(tapply(144:175, 144:175, function(x) {c(208, x)})))
```

clean.knitr

Delete temporary files created during knitr compile

Description

Delete temporary files created during knitr compile in working directory

Usage

```
clean.knitr()
```

Details

clean.knitr deletes temporary files created during knitr compile

Value

Nothing

Author(s)

Marc Girondot

Examples

```
## Not run:
clean.knitr()

## End(Not run)
```

 compare_AIC

Compares the AIC of several outputs

Description

This function is used to compares the AIC of several outputs obtained with the same data but with different set of parameters.

The parameters must be lists with \$aic or \$AIC or \$value and \$par elements.

If several objects are within the same list, there AIC is sumed.

For example, `compare_AIC(g1=list(group), g2=list(separe1, separe2))` can be used to compare a single model onto two different sets of data against each set of data fitted with its own set of parameters.

Usage

```
compare_AIC(...)
```

Arguments

... Successive results to be compared as lists

Details

`compare_AIC` compares the AIC of several outputs obtained with the same data.

Value

A list with DeltaAIC and Akaike weight for the models.

Author(s)

Marc Girondot

Examples

```
## Not run:
# Read a file with data
library("phenology")
Gratiot<-read.delim("http://max2.es.e.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
refdate <- as.Date("2001-01-01")
data_Gratiot<-add_phenology(Gratiot, name="Complete", reference=refdate, format="%d/%m/%Y")
# Fix parameter FLat to 0
pfixed=c(Flat=0)
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=pfixed)
# Fit is done
result_Gratiot_Flat<-fit_phenology(data=data_Gratiot, parametersfit=parg2,
```

```

parametersfixed=pfixed, trace=1)
data(result_Gratiot_Flat)
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)
data(result_Gratiot)
# Compare both models
outputAIC<-compare_AIC(full=result_Gratiot, Flat=result_Gratiot_Flat)

## End(Not run)

```

compassRose2

Display a compass rose

Description

Displays a basic compass rose, usually to orient a map. `compassRose` displays a conventional compass rose at the position requested. The size of the compass rose is determined by the character expansion, as the central "rose" is calculated relative to the character size. Rotation is in degrees counterclockwise.

Usage

```

compassRose2(x, y, rot = 0, cex = 1, col = "black",
col.arrows.light = "white", col.arrows.dark = "black")

```

Arguments

<code>x</code>	The position of the center of the compass rose in user units.
<code>y</code>	The position of the center of the compass rose in user units.
<code>rot</code>	Rotation for the compass rose in degrees. See Details.
<code>cex</code>	The character expansion to use in the display.
<code>col</code>	The color of text
<code>col.arrows.light</code>	The color of lighter lines
<code>col.arrows.dark</code>	The color of darker lines

Details

`compassRose2` Display a compass rose

Value

none

Author(s)

modified from Jim Lemon; See compassRose sp

Examples

```
## Not run:
library("maps")
library("phenology")
map("world", "China")
compassRose2(x=110, y=35, col.arrows.light="grey")

## End(Not run)
```

convert.ts

Convert one Date-Time from one timezone to another

Description

Convert one Date-Time from one timezone to another. Available timezones can be shown using OlsonNames()

Usage

```
convert.ts(x, tz = Sys.timezone())
```

Arguments

x	The date-time in POSIXlt or POSIXct format
tz	The timezone

Details

convert.ts Convert one Date-Time from one timezone to another

Value

A POSIXlt date converted

Author(s)

Marc Girondot

Examples

```
d <- as.POSIXlt("2010-01-01 17:34:20", tz="UTC")
convert.ts(d, tz="America/Guatemala")
```

extract_result	<i>Extract the set of parameters from a result object.</i>
----------------	------------------------------------------------------------

Description

The function "extract_result" permits to extract the set of parameters from a result object obtained after fit_phenology.

Usage

```
extract_result(result = NULL, help = FALSE)
```

Arguments

result	A result file
help	If TRUE, an help is displayed

Details

extract_result get the fitted parameters from a result object.

Value

Return the set of fitted parameters

Author(s)

Marc Girondot

Examples

```
library(phenology)
## Not run:
# Read a file with data
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", \cr
header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
# result_Gratiot<-fit_phenology(data=data_Gratiot, parametersfit=parg,
parametersfixed=NULL, trace=1)
data(result_Gratiot)
# Extract the fitted parameters
parg1<-extract_result(result_Gratiot)

## End(Not run)
```

fit_phenology

*Fit the phenology parameters to timeseries of counts.***Description**

Function of the package phenology to fit parameters to timeseries.

To fit data, the syntaxe is :

```
Result<-fit_phenology(data=dataset, parametersfit=par, parametersfixed=pfixed, trace=1, method_incertitude=2,
zero_counts=TRUE, hessian=TRUE)
```

or if no parameter is fixed :

```
Result<-fit_phenology(data=dataset, parametersfit=par)
```

or

```
fit_phenology(help=TRUE) to have the help !
```

Add trace=1 [default] to have information on the fit progression or trace=0 to hide information on the fit progression.

method_incertitude='multinomial' [default] is the correct one from a statistical point of view.

method_incertitude='binomial' is an alternative more rapid.

method_incertitude='sum' is an aproximate method more rapid and using less memory.

zero_counts=c(TRUE, TRUE, FALSE) indicates whether the zeros have been recorded for each of these timeseries. Defaut is TRUE for all.

hessian=FALSE does not estimate se of parameters.

Usage

```
fit_phenology(data = file.choose(), parametersfit = NULL,
parametersfixed = NULL, trace = 1, maxit = 500,
method_incertitude = "multinomial", zero_counts = TRUE, hessian = TRUE,
help = FALSE, silent = FALSE, growlnotify = TRUE)
```

Arguments

data	A dataset generated by add_format
parametersfixed	Set of fixed parameters
parametersfit	Set of parameters to be fitted
trace	If 1, it shows the progression of fit; 0 is silent (don't be afraid if it is long !)
maxit	Number of iterations for search before checking if it converges. If it does not converge, it will continue to search for. Default is 500.
method_incertitude	'multinomial' [default] is the correct one from a statistical point of view; 'binomial' is an aproximate method more rapid and using less memory; 'sum' is an alternative more rapid but potentially biased (not sure).
zero_counts	example c(TRUE, TRUE, FALSE) indicates whether the zeros have been recorder for each of these timeseries. Defaut is TRUE for all.
hessian	If FALSE does not estimate se of parameters

help	If TRUE, an help is displayed
silent	If TRUE does not show any message
growlnotify	If FALSE, does not send growl notification

Details

fit_phenology fits parameters to timeseries.

Value

Return a list of with data and result

Author(s)

Marc Girondot

Examples

```
## Not run:
library(phenology)
# Read a file with data
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)
data(result_Gratiot)
# Plot the phenology and get some stats
output<-plot(result_Gratiot)

## End(Not run)
```

getTide

Annual tide calendar for one particular location

Description

The script extract tide information from <http://tides.mobilegeographics.com/> into a data.frame. The presence of XLM package is necessary for this function.

Usage

```
getTide(file = NULL, year = as.POSIXlt(Sys.time())$year + 1900,
location = 0, latitude = NA, longitude = NA, tz = "")
```

Arguments

file	An html file from the site http://tides.mobilegeographics.com/
location	Code based on http://tides.mobilegeographics.com/
year	Year to get the calendar
latitude	The latitude of the tide information
longitude	The longitude of the tide information
tz	Timezone

Details

getTide gets the annual tide calendar for one particular location.

Value

Return a data.frame with tide calendar:

Level is the tide level, Tide is the High or Low Tide information and Date.Time is the date/time in POSIXlt format.

Author(s)

Marc Girondot <marc.girondot@u-psud.fr>

Examples

```
## Not run:
library("phenology")
lat <- 5.74
long <- -54
Awala2004 <- getTide(year=2004, longitude=long, latitude=lat, tz="America/Cayenne")
with(Awala2004, plot(Date.Time, Level, bty="n", las=1, type="l",
  xlab=paste("Year", as.POSIXlt(Date.Time[1])$year+1900),
  ylab="Tide level in m"))

## End(Not run)
```

Gratiot

Leatherback nest counts from Gratiot et al. (2006) Figure 1

Description

Leatherback nest counts from Gratiot et al. (2006) Figure 1. These data have been collected by the ONG Kwata in French Guiana.

The data have been obtained from the graph of the publication (see reference).

Usage

Gratiot

Format

data.frame with the morning date in the first column and the nest counts on the second one.

Details

Leatherback nest counts from Gratiot et al. (2006) Figure 1

Author(s)

KWATA ONG -

References

Gratiot, N., Gratiot, J., de Thoisy, B. & Kelle, L. 2006. Estimation of marine turtles nesting season from incomplete data ; statistical adjustment of a sinusoidal function. *Animal Conservation*, 9, 95-102.

Examples

```
library(phenology)
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.esse.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
```

growlnotify

Send growl notification for MacOS X system.

Description

This function is used to send a notification to Mac user.

Usage

```
growlnotify(textinfo = "", help = FALSE)
```

Arguments

textinfo	Text to display in the growlnotify window
help	If TRUE, an help is displayed

Details

growlnotify send growl notification for MacOS X systems.

Value

None

Author(s)

Marc Girondot

Examples

```
# If growlnotify is used on a non-mac system, it just quits.
growlnotify("It works if you are on a Mac with GrowlNotify installed!")
```

ind_long_lat	<i>Return or the index in ncdf object from lat/longitude or inverse</i>
--------------	-------------------------------------------------------------------------

Description

Return or the index in ncdf object from lat/longitude or inverse

Usage

```
ind_long_lat(ncdf = stop("The ncdf data must be supplied"), long = NA,
  lat = NA, indice.long = NA, indice.lat = NA, name.lon = "lon",
  name.lat = "lat")
```

Arguments

ncdf	an object read from package ncdf4, ncdf or RNetCDF
long	longitude in decimal format
lat	latitude in decimal format
indice.long	Index of longitude
indice.lat	Index of latitude
name.lon	Name of argument for longitude, default is lon
name.lat	Name of argument for latitude, default is lat

Details

ind_long_lat is used to manage ncdf information

Value

Or the index in ncdf object from lat/longitude or inverse

Author(s)

Marc Girondot

Examples

```
## Not run:
url <- "ftp://ftp.cdc.noaa.gov/Datasets/noaa.oisst.v2.highres/"
url <- paste0(url, "sst.day.mean.2012.v2.nc")
dest <- paste(Sys.getenv("HOME"), "/sst.day.mean.2012.v2.nc", sep="")
download.file(url, dest)
library("ncdf4")
dta2012 <- nc_open(dest)
indices <- ind_long_lat(ncdf=dta2012, lat=5.89, long=-20.56)
coordinates <- ind_long_lat(ncdf=dta2012, indice.lat=20, indice.long=30)
# library("RNetCDF")
# dta2012 <- open.nc(dest)
# indices <- ind_long_lat(ncdf=dta2012, lat=5.89, long=-20.56)
# coordinates <- ind_long_lat(ncdf=dta2012, indice.lat=20, indice.long=30)
library("ncdf")
dta2012 <- open.ncdf(dest)
indices <- ind_long_lat(ncdf=dta2012, lat=5.89, long=-20.56)
coordinates <- ind_long_lat(ncdf=dta2012, indice.lat=20, indice.long=30)

## End(Not run)
```

LBLE_to_BE

Transform a set of parameters from LengthB LengthE to Begin End.

Description

This function is used to transform a set of parameters that uses LengthB, Peak and LengthE to a set of parameters that uses Begin, Peak and End.

Usage

```
LBLE_to_BE(parameters = NULL, help = FALSE)
```

Arguments

parameters	Set of current parameters
help	If TRUE, an help is displayed

Details

LBLE_to_BE transforms a set of parameters from LengthB LengthE to Begin End.

Value

Return a set of modified parameters

Author(s)

Marc Girondot

Examples

```
# Read a file with data
# Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", , header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
refdate <- as.Date("2001-01-01")
data_Gratiot<-add_phenology(Gratiot, name="Complete", reference=refdate, format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Change the parameters to Begin End format
parg1<-LBLE_to_BE(parameters=parg)
# And change back to LengthB LengthE
parg2<-BE_to_LBLE(parameters=parg1)
```

LBLE_to_L

Transform a set of parameters from LengthB LengthE format to Length

Description

This function is used to transform a set of parameters that uses LengthB and LengthE to a set of parameters uses Length.

Usage

```
LBLE_to_L(parameters = stop("Set of parameters must be given"))
```

Arguments

parameters Set of current parameters

Details

LBLE_to_L transforms a set of parameters from LengthB LengthE format to Length.

Value

Return the set of modified parameters

Author(s)

Marc Girondot

Examples

```
## Not run:
# Read a file with data
Gratiot<-read.delim("http://max2.es.e.u-psud.fr/epc/conservation/BI/Complete.txt",
header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
refdate <- as.Date("2001-01-01")
data_Gratiot<-add_phenology(Gratiot, name="Complete", reference=refdate, format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Change the parameters to Begin End format
parg1<-LBLE_to_L(parameters=parg)
# And change back to LengthB LengthE.
parg2<-L_to_LBLE(parameters=parg1)
```

likelihood_phenology *Estimate the likelihood of timeseries based on a set of parameters.*

Description

This function is used to estimate the likelihood based on a set of parameters.

Usage

```
likelihood_phenology(data = NULL, parametersfit = NULL,
parametersfixed = NULL, zero_counts = NULL, method_incertitude = NULL,
result = NULL)
```

Arguments

data	Dataset generated with add_format
parametersfixed	Set of fixed parameters
parametersfit	Set of parameters to be fitted
method_incertitude	2 [default] is the correct one from a statistical point of view; 0 is an approximate method more rapid; 1 is an alternative more rapid but biased.
zero_counts	example c(TRUE, TRUE, FALSE) indicates whether the zeros have been recorder for each of these timeseries. Default is TRUE for all.
result	An object obtained after fit_phenology()

Details

likelihood_phenology estimate likelihood for a set of parameters.

Value

The likelihood of the data with the parameters

Author(s)

Marc Girondot

Examples

```
## Not run:
# Read a file with data
Gratiot<-read.delim("http://max2.esse.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Estimate likelihood with this initial set of parameters
likelihood_phenology(data=data_Gratiot, parametersfit=parg, parametersfixed=NULL)

## End(Not run)
```

local.search

Return path of file searched for in local disk based on its file name

Description

Return path of file searched for in local disk based on its file name. It has been tested only with Windows XP and MacOSX.

Usage

```
local.search(pattern, directory = "", folder = "$HOME")
```

Arguments

pattern	The name of file to be searched for. Can use wildcards *
directory	The path of directory to be explored in for Windows
folder	The path of folder to be explored in for Unix based systems

Details

local.search() returns path of file searched in local disk based on its file name

Value

A vector with paths

Author(s)

Marc Girondot

Examples

```
## Not run:  
RnwFiles <- local.search("*.Rnw")  
nc.files <- local.search("*.nc", folder=paste0("'",getwd(),"'"))  
  
## End(Not run)
```

logLik.phenology *Return Log Likelihood of a fit generated by fit_phenology*

Description

Return Log Likelihood of a fit generated by fit_phenology

Usage

```
## S3 method for class 'phenology'  
logLik(object, ...)
```

Arguments

object	A result file generated by fit_phenology
...	Not used

Details

logLik.phenology Return Log Likelihood of a fit

Value

The Log Likelihood value of the fitted model and data

Author(s)

Marc Girondot

Examples

```
## Not run:  
library(phenology)  
data(result_Gratiot)  
logLik(result_Gratiot)  
AIC(result_Gratiot)  
  
## End(Not run)
```

L_to_LBLE

Transform a set of parameters from Length format to LengthB LengthE

Description

This function is used to transform a set of parameters that uses Length to a set of parameters uses LengthB and LengthE.

Usage

```
L_to_LBLE(parameters = stop("Set of parameters must be given"))
```

Arguments

parameters Set of current parameters

Details

L_to_LBLE transforms a set of parameters from Length format to LengthB LengthE.

Value

Return the set of modified parameters

Author(s)

Marc Girondot

Examples

```
## Not run:
# Read a file with data
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
refdate <- as.Date("2001-01-01")
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=refdate, format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Change the parameters to Begin End format
parg1<-LBLE_to_L(parameters=parg)
# And change back to LengthB LengthE.
parg2<-L_to_LBLE(parameters=parg1)
```

map.scale2

*Add Scale to Existing Unprojected Map***Description**

Adds a scale to an existing map, both as a ratio and a distance gauge. If x or y are not specified, this will be taken to be near the lower left corner of the map.

Usage

```
map.scale2(x, y, relwidth = 0.15, metric = TRUE, ratio = TRUE,
  col.line = "black", ...)
```

Arguments

x	Location of left end of distance gauge.
y	Location of left end of distance gauge.
relwidth	Proportion of width of display to be used for the scale. The default is 0.15.
metric	If TRUE, the distance gauge will be in km, otherwise miles.
ratio	If FALSE, the scale ratio of the map is not displayed.
col.line	The color of lines for the gauge.
...	Further plotting parameters may be specified as for the command text().

Details

map.scale2 Add Scale to Existing Unprojected Map

Value

The exact calculated scale is returned.

Author(s)

See map.scale maps

Examples

```
## Not run:
library("maps")
library("phenology")
map("world", "China")
map.scale2(col.line = "red", col="blue")

## End(Not run)
```

`map_Gratiot`*Likelihood map of Leatherback nest counts*

Description

Likelihood map of Leatherback nest counts from Gratiot et al. (2006) Figure 1. A intraseasonal periodic pattern was searched for varying Phi and Delta parameters.

Usage`map_Gratiot`**Format**

A list with Gratiot data and the result of the fit.

Details

Likelihood map of Leatherback nest counts from Gratiot et al. (2006) Figure 1

Author(s)

Marc Girondot <marc.girondot@u-psud.fr>

References

Gratiot, N., Gratiot, J., de Thoisy, B. & Kelle, L. 2006. Estimation of marine turtles nesting season from incomplete data ; statistical adjustment of a sinusoidal function. *Animal Conservation*, 9, 95-102.

Examples

```
library(phenology)
# Read a file with likelihood map
data(map_Gratiot)
```

`map_phenology`*Generate a likelihood map varying Phi and Delta.*

Description

This function generates a map of likelihood varying Phi and Delta.
When Delta is not given, the same precision as Phi is used.

Usage

```
map_phenology(data = NULL, parametersfit = NULL, parametersfixed = NA,
  Phi = seq(from = 0.2, to = 20, length.out = 100), Delta = NULL,
  method_incertitude = 2, zero_counts = TRUE, progressbar = TRUE,
  help = FALSE)
```

Arguments

data	dataset generated with add_format
parametersfixed	Set of fixed parameters
parametersfit	Set of parameters to be fitted
Phi	Phi values to be analyzed
Delta	Delta value to be analyzed
method_incertitude	2 [default] is the correct one from a statistical point of view; 0 is an approximate method more rapid; 1 is an alternative more rapid but biased.
zero_counts	Example c(TRUE, TRUE, FALSE) indicates whether the zeros have been recorded for each of these timeseries. Default is TRUE for all.
progressbar	If FALSE, do not show the progress bar
help	If TRUE, an help is displayed

Details

map_phenology generates a likelihood map.

Value

Display a likelihood map

Author(s)

Marc Girondot

Examples

```
library("phenology")
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
```



```

# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
## Not run:
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)

## End(Not run)
data(result_Gratiot)
# Extract the fitted parameters
parg1<-extract_result(result_Gratiot)
# Add constant Alpha and Tau values
# [day d amplitude=(Alpha+Nd*Beta)^Tau with Nd being the number of counts for day d]
pfixed<-c(parg1, Alpha=0, Tau=1)
pfixed<-pfixed[-which(names(pfixed)=="Theta")]
# The only fitted parameter will be Beta
parg2<-c(Beta=0.5, parg1["Theta"])
# Generate a likelihood map
# [default Phi=seq(from=0.1, to=20, length.out=100) but it is very long]
# Take care, it takes 20 hours ! The data map_Gratiot has the result
## Not run:
map_Gratiot<-map_phenology(data=data_Gratiot, Phi=seq(from=0.1, to=20, length.out=100),
parametersfit=parg2, parametersfixed=pfixed)

## End(Not run)
data(map_Gratiot)
# Plot the map
plot(map_Gratiot, col=heat.colors(128))
# Plot the min(-Ln L) for Phi varying at any delta value
plot_phi(map=map_Gratiot)
# Plot the min(-Ln L) for Delta varying with Phi equal to the value for maximum likelihood
plot_delta(map=map_Gratiot)
# Plot the min(-Ln L) for Delta varying with Phi the nearest to 15
plot_delta(map=map_Gratiot, Phi=15)

```

MinBMinE_to_Min

Transform a set of parameters from MinB and MinE to Min

Description

This function is used to transform a set of parameters that uses MinB and MinE to a set of parameters that uses Min.

Usage

```
MinBMinE_to_Min(parameters = stop("A set of parameters must be indicated"))
```

Arguments

parameters Set of current parameters

Details

MinBMinE_to_Min transforms a set of parameters from MinB and MinE to Min

Value

Return a set of modified parameters

Author(s)

Marc Girondot

Examples

```
# Read a file with data
# Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", , header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
refdate <- as.Date("2001-01-01")
data_Gratiot<-add_phenology(Gratiot, name="Complete", reference=refdate, format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot)
# Change the parameters to PMinB and PMinE
parg1<-MinBMinE_to_Min(parameters=parg)
```

moon_phase

Moon phase based on a date

Description

The script give an index (base 100) that represents moon phase. If the value lays between:
 0 and 1.6931595 or 98.3068405 and 100, it is full moon,
 23.3068405 and 26.6931595, last quarter,
 48.3068405 and 51.6931595, new moon,
 73.3068405 and 76.6931595, first quarter
 When phase is set to TRUE, a character representing the moon phase is returned.

Usage

```
moon_phase(date = NULL, phase = FALSE)
```

Arguments

date	A date in known format
phase	If TRUE, a vector of character with NM, FQ, FL LQ will be returned

Details

moon_phase calculates the moon phase based on a date.

Value

Return a value describing the moon phase:
 0 and 100 are full moon, 50 is new moon, 25 last quarter and 75 first quarter

Author(s)

Marc Girondot <marc.girondot@u-psud.fr>

Examples

```
library("phenology")
moon_phase(as.Date("2001-12-31"))
moon_phase(as.Date("14/04/2010", "%d/%m/%Y"))
moon_phase(as.Date("22/06/07", "%d/%m/%y"))
moon_phase(seq(from=as.Date("2012-03-01"),
to=as.Date("2012-04-15"), by="days"))
moon_phase(seq(from=as.Date("2012-03-01"),
to=as.Date("2012-04-15"), by="days"), phase=TRUE)
```

<code>par_init</code>	<i>Calculate initial set of parameters.</i>
-----------------------	---------------------------------------------

Description

This function is used to generate a first set of parameters that is expected to be not too far from the final.

Usage

```
par_init(data = stop("A dataset must be provided"), parametersfixed = NA,
help = FALSE)
```

Arguments

<code>data</code>	Dataset generated with <code>add_phenology()</code>
<code>parametersfixed</code>	Set of fixed parameters
<code>help</code>	If TRUE, an help is displayed

Details

`par_init` calculates initial set of parameters.

Value

The initial set of parameters

Author(s)

Marc Girondot

Examples

```
## Not run:
library(phenology)
# Read a file with data
Gratiot<-read.delim("http://max2.esse.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)
data(result_Gratiot)
# Plot the phenology and get some stats
output<-plot(result_Gratiot)

## End(Not run)
```

phenology

Run a shiny application for basic functions of phenology

Description

Run a shiny application for basic functions of phenology

Usage

```
phenology()
```

Details

phenology runs a shiny application for basic functions of phenology

Value

Nothing

Author(s)

Marc Girondot

Examples

```
## Not run:
library(phenology)
phenology()

## End(Not run)
```

phenology_MHmcmc *Run the Metropolis-Hastings algorithm for data*

Description

Run the Metropolis-Hastings algorithm for data.
 Deeply modified from a MCMC script by Olivier Martin (INRA, Paris-Grignon).
 The number of iterations is $n.iter+n.adapt+1$ because the initial likelihood is also displayed.
 I recommend that $thin=1$ because the method to estimate SE uses resampling.
 As initial point is maximum likelihood, $n.adapt = 0$ seems a good solution.

Usage

```
phenology_MHmcmc(result = stop("An output from fit_phenology() must be provided"),
  n.iter = 10000,
  parametersMCMC = stop("A model generated with phenology_MHmcmc_p() must be provided"),
  n.chains = 4, n.adapt = 0, thin = 1, trace = FALSE)
```

Arguments

<code>n.iter</code>	Number of iterations for each step
<code>parametersMCMC</code>	A set of parameters used as initial point for searching with information on priors
<code>result</code>	An object obtained after a SearchR fit
<code>n.chains</code>	Number of replicates
<code>n.adapt</code>	Number of iterations before to store outputs
<code>thin</code>	Number of iterations between each stored output
<code>trace</code>	True or False, shows progress

Details

phenology_MHmcmc runs the Metropolis-Hastings algorithm for data (Bayesian MCMC)

Value

A list with `resultMCMC` being `mcmc.list` object, `resultLnL` being likelihoods and `parametersMCMC` being the parameters used

Author(s)

Marc Girondot

Examples

```
## Not run:
library(phenology)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
  reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
  parametersfit=parg, parametersfixed=NULL, trace=1)
# Generate set of priors for Bayesian analysis
pmcmc <- phenology_MHmcmc_p(result_Gratiot, accept = TRUE)
result_Gratiot_mcmc <- phenology_MHmcmc(result = result_Gratiot, n.iter = 10000,
  parametersMCMC = pmcmc, n.chains = 1, n.adapt = 0, thin = 1, trace = FALSE)
# Get standard error of parameters
summary(result_Gratiot_mcmc)
# Make diagnostics of the mcmc results using coda package
mcmc <- as.mcmc(result_Gratiot_mcmc)
require(coda)
heidel.diag(mcmc)
raftery.diag(mcmc)
autocorr.diag(mcmc)
acf(mcmc[[1]][,"LengthB"], lag.max=200, bty="n", las=1)
acf(mcmc[[1]][,"Max_Gratiot"], lag.max=50, bty="n", las=1)
batchSE(mcmc, batchSize=100)
# The batch standard error procedure is usually thought to
# be not as accurate as the time series methods used in summary
summary(mcmc)$statistics[,"Time-series SE"]
plot(result_Gratiot_mcmc, parameters=3, las=1, xlim=c(-10, 300))

## End(Not run)
```

phenology_MHmcmc_p *Generates set of parameters to be used with phenology_MHmcmc()*

Description

Interactive script used to generate set of parameters to be used with phenology_MHmcmc().

Usage

```
phenology_MHmcmc_p(result = stop("An output from fit_phenology() must be provided"),
  accept = FALSE)
```

Arguments

result	An object obtained after a fit_phenology() fit
accept	If TRUE, does not wait for user interaction

Details

phenology_MHmcmc_p generates set of parameters to be used with MHmcmc()

Value

A matrix with the parameters

Author(s)

Marc Girondot

Examples

```
## Not run:
library(phenology)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
  reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
  parametersfit=parg, parametersfixed=NULL, trace=1)
# Generate set of priors for Bayesian analysis
pmcmc <- phenology_MHmcmc_p(result_Gratiot, accept = TRUE)
result_Gratiot_mcmc <- phenology_MHmcmc(result = result_Gratiot, n.iter = 10000,
  parametersMCMC = pmcmc, n.chains = 1, n.adapt = 0, thin = 1, trace = FALSE)
# Get standard error of parameters
summary(result_Gratiot_mcmc)
# Make diagnostics of the mcmc results using coda package
mcmc <- as.mcmc(result_Gratiot_mcmc)
require(coda)
heidel.diag(mcmc)
raftery.diag(mcmc)
autocorr.diag(mcmc)
acf(mcmc[[1]][,"LengthB"], lag.max=200, bty="n", las=1)
acf(mcmc[[1]][,"Max_Gratiot"], lag.max=50, bty="n", las=1)
batchSE(mcmc, batchSize=100)
# The batch standard error procedure is usually thought to
```

```
# be not as accurate as the time series methods used in summary
summary(mcmc)$statistics[,"Time-series SE"]
plot(result_Gratiot_mcmc, parameters=3, las=1, xlim=c(-10, 300))

## End(Not run)
```

phenology_swot

Fit a nesting season of marine turtles

Description

This function tries to make very simple the use of this package.
In most of the cases, values for header, reference and format need not to be set because they are detected automatically.

Usage

```
phenology_swot(header = NULL, reference = NULL, month_ref = NULL,
               format = NULL)
```

Arguments

header	Does the timeseries has header.
reference	Date used as reference. Is the day 1.
month_ref	Reference month. Generally will be 1 (January) or 7 (July).
format	Format for dates.

Details

phenology_swot is a simplified function for phenology package developed as part of SWOT project

Value

Return a list of with data and result

Author(s)

Marc Girondot <marc.girondot@u-psud.fr>

plot.mcmcComposite *Plot the result of a MCMC search*

Description

Plot the result of a MCMC search.
The parameters to use can be called by:
parameters="all"
parameters=1:4
parameters=c("PAR1", "PAR2", "PAR5")
parameters=c(TRUE, TRUE, FALSE, TRUE)

Usage

```
## S3 method for class 'mcmcComposite'  
plot(x, ..., chain = 1, parameters = 1,  
     scale.prior = FALSE, legend = TRUE)
```

Arguments

x	A mcmcComposite object obtained after phenology_fonctionMCMC()
chain	The chain to use
parameters	Name of parameters or their number (see description)
legend	If FALSE, the legend is not shown
scale.prior	If TRUE, the prior is scaled at the same size as posterior
...	Graphical parameters to be send to hist()

Details

plot.mcmcComposite plots the result of a MCMC search

Value

None

Author(s)

Marc Girondot

Examples

```
## Not run:
library(phenology)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
  reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
  parametersfit=parg, parametersfixed=NULL, trace=1)
# Generate set of priors for Bayesian analysis
pmcmc <- phenology_MHmcmc_p(result_Gratiot, accept = TRUE)
result_Gratiot_mcmc <- phenology_MHmcmc(result = result_Gratiot, n.iter = 10000,
  parametersMCMC = pmcmc, n.chains = 1, n.adapt = 0, thin = 1, trace = FALSE)
# Get standard error of parameters
summary(result_Gratiot_mcmc)
# Make diagnostics of the mcmc results using coda package
mcmc <- as.mcmc(result_Gratiot_mcmc)
require(coda)
heidel.diag(mcmc)
raftery.diag(mcmc)
autocorr.diag(mcmc)
acf(mcmc[[1]][,"LengthB"], lag.max=200, bty="n", las=1)
acf(mcmc[[1]][,"Max_Gratiot"], lag.max=50, bty="n", las=1)
batchSE(mcmc, batchSize=100)
# The batch standard error procedure is usually thought to
# be not as accurate as the time series methods used in summary
summary(mcmc)$statistics[,"Time-series SE"]
plot(result_Gratiot_mcmc, parameters=3, las=1, xlim=c(-10, 300))

## End(Not run)
```

plot.phenology

Plot the phenology from a result.

Description

The function plot.phenology plots the phenology graph from a result object.

Usage

```
## S3 method for class 'phenology'
plot(x, ..., data = NULL, parameters = NULL,
  parametersfixed = NA, series = "all", moon = FALSE,
  replicate.CI = 1000, progressBar = TRUE, help = FALSE,
  growlnotify = TRUE)
```

Arguments

x	A result file generated by fit_phenology
...	Parameters used by plot
data	A dataset generated by add_format
parametersfixed	Set of fixed parameters
parameters	Set of parameters to be changed
series	Number of series to be analyzed or 'all'
moon	If TRUE, the moon phase is plotted. Default is FALSE
replicate.CI	Number of replicates for estimation of confidence interval
progressbar	If FALSE, do not show the progress bar
help	If TRUE, an help is displayed
growlnotify	If False, does not send growl notification

Details

plot.phenology plots the phenology.

Value

Return A list of lists with formatted outputs
 For each site:
 \$site=name of the site
 \$estimate1=Estimation of counts not taking into account the observations
 \$sd1=The SD of estimation1
 \$estimate2=Estimation of counts taking into account the observations
 \$sd2=The SD of estimation2

Author(s)

Marc Girondot

Examples

```
## Not run:
library(phenology)
# Read a file with data
Gratiot<-read.delim("http://max2.esse.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)
```

```

data(result_Gratiot)
# Plot the phenology and get some stats
output<-plot(result_Gratiot)
# Plot only part of the nesting season
ptoutput<-plot(result_Gratiot, xlim=c(as.Date("2001-03-01"),as.Date("2001-08-31")))
# Use month names in English
Sys.setlocale(category = "LC_TIME", locale="en_GB.UTF-8")
output<-plot(result_Gratiot)
# set back the month name in local R language
Sys.setlocale(category = "LC_TIME", locale="")

## End(Not run)

```

plot.phenologymap *Plot a likelihood map with Delta and Phi varying.*

Description

This function plots a likelihood map obtained after map_phenology.

Usage

```

## S3 method for class 'phenologymap'
plot(x, ..., col = heat.colors(128), xlab = "Phi",
     ylab = "Delta", help = FALSE)

```

Arguments

x	A map generated with map_phenology.
...	not used
col	Colors could be heat.colors(128) or rainbow(64) or col=gray(c(seq(0, 1, length.out=128)))
xlab	Label for x axis
ylab	Label for y axis
help	If TRUE, an help is displayed

Details

plot.phenologymap plots a likelihood map with Delta and Phi varying.

Value

Return None

Author(s)

Marc Girondot

Examples

```
## Not run:
library("phenology")
# Read a file with data
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)
data(result_Gratiot)
# Extract the fitted parameters
parg1<-extract_result(result_Gratiot)
# Add constant Alpha and Tau values
# [day d amplitude=(Alpha+Nd*Beta)^Tau with Nd being the number of counts for day d]
pfixed<-c(parg1, Alpha=0, Tau=1)
pfixed<-pfixed[-which(names(pfixed)=="Theta")]
# The only fitted parameter will be Beta
parg2<-c(Beta=0.5, parg1["Theta"])
# Generate a likelihood map
# [default Phi=seq(from=0.1, to=20, length.out=100) but it is very long]
# Take care, it takes 20 hours ! The data map_Gratiot has the result
map_Gratiot<-map_phenology(data=data_Gratiot,
Phi=seq(from=0.1, to=20, length.out=100),
parametersfit=parg2, parametersfixed=pfixed)
data(map_Gratiot)
# Plot the map
plot(map_Gratiot, col=heat.colors(128))

## End(Not run)
```

plot_add

Add a plot to a previous one

Description

To plot data, just add use it as a normal plot. It will plot the new data without axes, or labels for axes.

Usage

```
plot_add(...)
```

Arguments

```
... Parameters for plot()
```

Details

plot_add adds a plot to a previous one

Value

Nothing

Author(s)

Marc Girondot

Examples

```
plot(x=1:100, y=sin(1:100), type="l", bty="n", xlim=c(1,200), xlab="x", ylab="y")
plot_add(x=1:200, y=cos(1:200), type="l", bty="n", col="red")
```

plot_delta

Plot a likelihood lineplot obtained after map_phenology.

Description

This function plots a likelihood lineplot obtained after map_phenology.

Usage

```
plot_delta(map = NULL, Phi = NULL, help = FALSE)
```

Arguments

map	A map generated with map_phenology
Phi	Phi value or NULL
help	If TRUE, an help is displayed

Details

plot_delta plots the likelihood delta for fixed Phi value.

Value

Return None

Author(s)

Marc Girondot

Examples

```
## Not run:
library("phenology")
# Read a file with data
Gratiot<-read.delim("http://max2.es.e.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)
data(result_Gratiot)
# Extract the fitted parameters
parg1<-extract_result(result_Gratiot)
# Add constant Alpha and Tau values
# [day d amplitude=(Alpha+Nd*Beta)^Tau with Nd being the number of counts for day d]
pfixed<-c(parg1, Alpha=0, Tau=1)
pfixed<-pfixed[-which(names(pfixed)=="Theta")]
# The only fitted parameter will be Beta
parg2<-c(Beta=0.5, parg1["Theta"])
# Generate a likelihood map
# [default Phi=seq(from=0.1, to=20, length.out=100) but it is very long]
# Take care, it takes 20 hours ! The data map_Gratiot has the result
map_Gratiot<-map_phenology(data=data_Gratiot,
Phi=seq(from=0.1, to=20, length.out=100),
parametersfit=parg2, parametersfixed=pfixed)
data(map_Gratiot)
# Plot the min(-Ln L) for Delta varying with Phi equal to the value for maximum likelihood
plot_delta(map=map_Gratiot)
# Plot the min(-Ln L) for Delta varying with Phi the nearest to 15
plot_delta(map=map_Gratiot, Phi=15)

## End(Not run)
```

plot_errbar

Plot a xy graph with error bar on x and/or y

Description

To plot data, just use it as a normal plot but add the `errbar.x` and `errbar.y` values or `errbar.x.minus`, `errbar.x.plus` if bars for x axis are asymmetric and `errbar.y.minus`, `errbar.y.plus` if bars for y axis are asymmetric. Use `x.plus`, `x.minus`, `y.plus` and `y.minus` to set absolute limits for error bars. Note that `x.plus` and `x.minus` have priority over `errbar.x`, `errbar.x.minus` and `errbar.x.plus` and that `y.plus` and `y.minus` have priority over `errbar.y`, `errbar.y.minus` and `errbar.y.plus`. The parameter `errbar.y.polygon=TRUE` permits to define error as an envelop for y axis.

Usage

```
plot_errbar(..., errbar.x = NULL, errbar.y = NULL, errbar.x.plus = NULL,
  errbar.x.minus = NULL, errbar.y.plus = NULL, errbar.y.minus = NULL,
  x.plus = NULL, x.minus = NULL, y.plus = NULL, y.minus = NULL,
  errbar.tick = 1/50, errbar.lwd = par("lwd"), errbar.lty = par("lty"),
  errbar.col = par("fg"), errbar.y.polygon = FALSE,
  errbar.y.polygon.list = list(NULL), add = FALSE)
```

Arguments

...	Parameters for plot() such as main= or ylim=
errbar.x	The length of error bars for x. Recycled if necessary.
errbar.x.plus	The length of positive error bars for x. Recycled if necessary.
errbar.x.minus	The length of negative error bars for x. Recycled if necessary.
errbar.y	The length of error bars for y. Recycled if necessary.
errbar.y.plus	The length of positive error bars for y. Recycled if necessary.
errbar.y.minus	The length of negative error bars for y. Recycled if necessary.
x.plus	The absolut position of the positive error bar for x. Recycled if necessary.
x.minus	The absolut position of the negative error bar for x. Recycled if necessary.
y.plus	The absolut position of the positive error bar for y. Recycled if necessary.
y.minus	The absolut position of the nagative error bar for y. Recycled if necessary.
errbar.tick	Size of small ticks at the end of error bars defined as a proportion of total width or height graph size.
errbar.lwd	Error bar line width, see par("lwd")
errbar.lty	Error bar line type, see par("lwd")
errbar.col	Error bar line color, see par("col")
errbar.y.polygon	If true, the errors are shown as a filed polygon.
errbar.y.polygon.list	List of parameters to be used for polygon.
add	If true, add the graph to the previous one.

Details

plot_errbar plot a xy graph with error bar on x and/or y

Value

Nothing

Author(s)

Marc Girondot

Examples

```
plot_errbar(1:100, rnorm(100, 1, 2),
  xlab="axe x", ylab="axe y", bty="n", xlim=c(1,100),
  errbar.x=2, errbar.y=rnorm(100, 1, 0.1))
x <- 1:100
plot_errbar(x=1:100, rnorm(100, 1, 2),
  xlab="axe x", ylab="axe y", bty="n", xlim=c(1,100),
  x.minus=x-2, x.plus=x+2)
x <- (1:200)/10
y <- sin(x)
plot_errbar(x=x, y=y, xlab="axe x", ylab="axe y", bty="n", xlim=c(1,20),
  y.minus=y-1, y.plus=y+1, ylim=c(-3, 3), type="l",
  errbar.y.polygon=TRUE,
  errbar.y.polygon.list=list(border=NA, col=rgb(0, 0, 0, 0.5)))
```

plot_phi

Plot the best likelihood for fixed Phi value.

Description

The function "plot_phi" plots the best likelihood for each Phi value.

Usage

```
plot_phi(map = NULL, help = FALSE)
```

Arguments

map	A map generated with map_phenology
help	If TRUE, an help is displayed

Details

plot_phi plots the best likelihood for fixed Phi value.

Value

Return None

Author(s)

Marc Girondot

Examples

```

library("phenology")
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
## Not run:
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)

## End(Not run)
data(result_Gratiot)
# Extract the fitted parameters
parg1<-extract_result(result_Gratiot)
# Add constant Alpha and Tau values
# [day d amplitude=(Alpha+Nd*Beta)^Tau with Nd being the number of counts for day d]
pfixed<-c(parg1, Alpha=0, Tau=1)
pfixed<-pfixed[-which(names(pfixed)=="Theta")]
# The only fitted parameter will be Beta
parg2<-c(Beta=0.5, parg1["Theta"])
# Generate a likelihood map [default Phi=seq(from=0.1, to=20, length.out=100) but it is very long]
# Take care, it takes 20 hours ! The data map_Gratiot has the result
## Not run:
map_Gratiot<-map_phenology(data=data_Gratiot,
Phi=seq(from=0.1, to=20, length.out=100),
parametersfit=parg2, parametersfixed=pfixed)

## End(Not run)
data(map_Gratiot)
# Plot the min(-Ln L) for Phi varying at any delta value
plot_phi(map=map_Gratiot)

```

print.phenology

Print the result information from a result object.

Description

The function print.phenology displays from a result.

Usage

```

## S3 method for class 'phenology'
print(x, ...)

```

Arguments

x	A result file generated by fit_phenology
...	Not used

Details

print.phenology prints the information from a result object.

Value

None

Author(s)

Marc Girondot

Examples

```
library(phenology)
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.esse.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
## Not run:
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)

## End(Not run)
data(result_Gratiot)
# Show the result
result_Gratiot
```

print.phenologymap *Print information on a phenologymap object.*

Description

print.phenologymap print information on a phenologymap object

Usage

```
## S3 method for class 'phenologymap'
print(x, ...)
```

Arguments

```
x          A map generated with map_phenology.
...        Not used
```

Value

Return None

Author(s)

Marc Girondot

Examples

```
library("phenology")
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
## Not run:
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)

## End(Not run)
data(result_Gratiot)
# Extract the fitted parameters
parg1<-extract_result(result_Gratiot)
# Add constant Alpha and Tau values
# [day d amplitude=(Alpha+Nd*Beta)^Tau with Nd being the number of counts for day d]
pfixed<-c(parg1, Alpha=0, Tau=1)
pfixed<-pfixed[-which(names(pfixed)=="Theta")]
# The only fitted parameter will be Beta
parg2<-c(Beta=0.5, parg1["Theta"])
# Generate a likelihood map
# [default Phi=seq(from=0.1, to=20, length.out=100) but it is very long]
# Take care, it takes 20 hours ! The data map_Gratiot has the result
## Not run:
map_Gratiot<-map_phenology(data=data_Gratiot,
```

```
Phi=seq(from=0.1, to=20, length.out=100), parametersfit=parg2,
parametersfixed=pfixed)

## End(Not run)
data(map_Gratiot)
# Print the information on a map
map_Gratiot
```

print.phenologyout *Print the information from a output object.*

Description

The function print.phenologyout displays the output from a plot.

Usage

```
## S3 method for class 'phenologyout'
print(x, ...)
```

Arguments

x	An output generated by plot_phenology
...	Not used

Details

print.phenologyout prints the information from a result object.

Value

None

Author(s)

Marc Girondot

Examples

```
library(phenology)
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
```

```

reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
## Not run:
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)

## End(Not run)
data(result_Gratiot)
# Plot the phenology and get some stats
output<-plot(result_Gratiot)
# Show the output
output

```

read_folder	<i>Reads all files present in a folder and creates a list with the content of these files</i>
-------------	-----------------------------------------------------------------------------------------------

Description

To create a list, the syntax is
`datalist<-read_folder(folder=".", read=read.delim, header=FALSE)`
 Return NULL with a warning if the folder does not exist or is empty.
 The names of the elements of the list are the filenames.

Usage

```
read_folder(folder = try(file.choose()), silent = TRUE), wildcard = "*.*",
read = read.delim, ...)
```

Arguments

folder	Where to search for files; can be or a file path or a folder path
wildcard	Define which files are to be read (examples: "*.*", "*.xls", "essai*.txt")
read	Function used to read file. Ex: read.delim or read.xls from gdata package
...	Parameters send to the read function

Details

read_folder reads all files present in a folder

Value

Return a list of the data in the files of the folder (directory for windows users)

Author(s)

Marc Girondot

Examples

```
## Not run:  
library(phenology)  
# Read all the files from a folder/directory  
Gratiot<-read_folder(".")  
  
## End(Not run)
```

remove_site	<i>Removes site information from a set of parameters.</i>
-------------	-----------------------------------------------------------

Description

This function is used to remove the information of the site from a set of parameters. It can be used to other timeseries after.

Usage

```
remove_site(parameters = NULL, help = FALSE)
```

Arguments

parameters	Set of parameters
help	If TRUE, an help is displayed

Details

remove_site removes beach information from a set of parameters.

Value

Return a set of modified parameters

Author(s)

Marc Girondot

Examples

```
library(phenology)
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
## Not run:
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)

## End(Not run)
data(result_Gratiot)
# Extract parameters from result
parg<-extract_result(result_Gratiot)
# Remove site information
parg1<-remove_site(parg)
```

result_Gratiot

Result of the fit of Leatherback nest counts

Description

Result of the fit of Leatherback nest counts from Gratiot et al. (2006) Figure 1 The phenology has been fitted with MinE, MinB, Max, Flat, LengthB, LengthE, Peak, Theta.

Usage

```
result_Gratiot
```

Format

A list with Gratiot data and the result of the fit.

Details

Result of the fit of Leatherback nest counts from Gratiot et al. (2006) Figure 1

Author(s)

Marc Girondot <marc.girondot@u-psud.fr>

References

Gratiot, N., Gratiot, J., de Thoisy, B. & Kelle, L. 2006. Estimation of marine turtles nesting season from incomplete data ; statistical adjustment of a sinusoidal function. *Animal Conservation*, 9, 95-102.

Examples

```
library(phenology)
# Read a file with result
data(result_Gratiot)
```

result_Gratiot1	<i>Result of the fit of Leatherback nest counts</i>
-----------------	-----------------------------------------------------

Description

Result of the fit of Leatherback nest counts from Gratiot et al. (2006) Figure 1 The phenology has been fitted with MinE, MinB, Max, Flat, LengthB, LengthE, Peak, Theta, Alpha, Beta, Tau, Phi, Delta

Usage

```
result_Gratiot1
```

Format

A list with Gratiot data and the result of the fit.

Details

Result of the fit of Leatherback nest counts from Gratiot et al. (2006) Figure 1

Author(s)

Marc Girondot <marc.girondot@u-psud.fr>

References

Gratiot, N., Gratiot, J., de Thoisy, B. & Kelle, L. 2006. Estimation of marine turtles nesting season from incomplete data ; statistical adjustment of a sinusoidal function. *Animal Conservation*, 9, 95-102.

Examples

```
library(phenology)
# Read a file with result
data(result_Gratiot1)
```

result_Gratiot2	<i>Result of the fit of Leatherback nest counts</i>
-----------------	-----------------------------------------------------

Description

Result of the fit of Leatherback nest counts from Gratiot et al. (2006) Figure 1 The phenology has been fitted with MinE, MinB, Max, Flat, LengthB, LengthE, Peak, Theta, Alpha, Beta, Tau, Phi, Delta, Alpha1, Beta1, Tau1, Phi1, Delta1.

Usage

```
result_Gratiot2
```

Format

A list with Gratiot data and the result of the fit.

Details

Result of the fit of Leatherback nest counts from Gratiot et al. (2006) Figure 1

Author(s)

Marc Girondot <marc.girondot@u-psud.fr>

References

Gratiot, N., Gratiot, J., de Thoisy, B. & Kelle, L. 2006. Estimation of marine turtles nesting season from incomplete data ; statistical adjustment of a sinusoidal function. *Animal Conservation*, 9, 95-102.

Examples

```
library(phenology)
# Read a file with result
data(result_Gratiot2)
```

result_Gratiot_Flat *Result of the fit of Leatherback nest counts*

Description

Result of the fit of Leatherback nest counts from Gratiot et al. (2006) Figure 1. The phenology has been fitted with MinE, MinB, Max, LengthB, LengthE, Peak, Theta. The Flat parameter is set to 0 and is not fitted.

Usage

```
result_Gratiot_Flat
```

Format

A list with Gratiot data and the result of the fit.

Details

Result of the fit of Leatherback nest counts from Gratiot et al. (2006) Figure 1

Author(s)

Marc Girondot <marc.girondot@u-psud.fr>

References

Gratiot, N., Gratiot, J., de Thoisy, B. & Kelle, L. 2006. Estimation of marine turtles nesting season from incomplete data ; statistical adjustment of a sinusoidal function. *Animal Conservation*, 9, 95-102.

Examples

```
library(phenology)
# Read a file with result
data(result_Gratiot_Flat)
```

result_Gratiot_mcmc *Result of the mcmc for Leatherback nest counts*

Description

Result of the mcmc for Leatherback nest counts from Gratiot et al. (2006) Figure 1 The phenology has been fitted with MinE, MinB, Max, Flat, LengthB, LengthE, Peak, Theta.

Usage

```
result_Gratiot_mcmc
```

Format

A mcmcComposite object with mcmc result.

Details

Result of the mcmc for Leatherback nest counts from Gratiot et al. (2006) Figure 1

Author(s)

Marc Girondot <marc.girondot@u-psud.fr>

References

Gratiot, N., Gratiot, J., de Thoisy, B. & Kelle, L. 2006. Estimation of marine turtles nesting season from incomplete data ; statistical adjustment of a sinusoidal function. *Animal Conservation*, 9, 95-102.

Examples

```
library(phenology)
# Read a file with result
data(result_Gratiot_mcmc)
```

ScalePreviousPlot *Return the scale of the previous plot*

Description

Return a list with the limits of the previous plot, the center, the range, and the position of label on this axe.

Usage

```
ScalePreviousPlot()
```

Details

ScalePreviousPlot returns the scale of the previous plot

Value

A list with xlim and ylim

Author(s)

Marc Girondot

Examples

```
par(xaxs="i", yaxs="i")
plot(x=1:100, y=sin(1:100), type="l", bty="n", xlim=c(1,200), xlab="x", ylab="y")
xlim= ScalePreviousPlot()$xlim[1:2]
ylim= ScalePreviousPlot()$ylim[1:2]
par(xaxs="r", yaxs="i")
plot(x=1:100, y=sin(1:100), type="l", bty="n", xlim=c(1,200), xlab="x", ylab="y")
xlim= ScalePreviousPlot()$xlim[1:2]
ylim= ScalePreviousPlot()$ylim[1:2]
# Here is an example of the use of the label output
plot(x=1:100, y=sin(1:100), type="l", bty="n", xlim=c(1,200), xlab="", ylab="")
text(x=ScalePreviousPlot()$xlim["label"], y=ScalePreviousPlot()$ylim["center"],
     xpd=TRUE, "Legend for Y axes", pos=3, srt=90)
text(x=ScalePreviousPlot()$xlim["center"], y=ScalePreviousPlot()$ylim["label"],
     xpd=TRUE, "Legend for X axes", pos=1)
```

 shift_sinusoid

Shift sinusoid information.

Description

This function is used to shift sinusoid parameters from ", '1' or '2'.

Usage

```
shift_sinusoid(parameters = NULL, from = "", to = "1", help = FALSE)
```

Arguments

parameters	set of parameters
from	The number of series to change
to	The number of series to change
help	If TRUE, an help is displayed

Details

shift_sinusoid shift sinusoid information.

Value

Return a set of modified parameters

Author(s)

Marc Girondot

Examples

```
# Read a file with data
library("phenology")
## Not run:
Gratiot<-read.delim("http://max2.es.e.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Fix parameter Flat to 0
pfixed=c(Flat=0)
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=pfixed)
# Fit is done
## Not run:
result_Gratiot_Flat<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=pfixed, trace=1)

## End(Not run)
data(result_Gratiot_Flat)
parg<-extract_result(result_Gratiot_Flat)
# Add data for one sinusoid superimposed
# [day d amplitude=(Alpha+Nd*Beta)^Tau with Nd being the number of counts for day d]
parg<-c(parg, Alpha=0.5, Beta=0.8, Delta=3, Phi=15)
# Tau is fixed to 1
pfixed=c(Flat=0, Tau=1)
# Run the optimisation
## Not run:
result_Gratiot1<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=pfixed, trace=1)
# Plot the phenology
output1<-plot(result_Gratiot1, moon=TRUE)
#'
## End(Not run)
data(result_Gratiot1)
# Extract the fitted parameters
parg1<-extract_result(result_Gratiot1)
# Shift sinusoid information to the '1'
```

```

parg2<-shift_sinusoid(parameters=parg1, from="", to="1")
# Tau is fixed to 1
pfixed=c(Flat=0, Tau1=1, Tau=1)
# Add data for another sinusoid superimposed
# [day d amplitude=(Alpha+Nd*Beta)^Tau with Nd being the number of counts for day d]
parg<-c(parg2, Alpha=0.5, Beta=0.8, Delta=3, Phi=10)
# Run the optimisation
## Not run:
result_Gratiot2<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=pfixed, trace=1)
# Plot the phenology
output2<-plot(result_Gratiot2, moon=TRUE)

## End(Not run)
data(result_Gratiot2)

```

summary.mcmcComposite *Summarize the result of a MCMC search*

Description

Summary for the result of a MCMC search

Usage

```
## S3 method for class 'mcmcComposite'
summary(object, ..., chain = NULL)
```

Arguments

object	A mcmcComposite object obtained after MHmcmc()
...	Internal use
chain	The chain to use

Details

summary.mcmcComposite get info on the result of a MCMC search

Value

A summary of the result

Author(s)

Marc Girondot

Examples

```
## Not run:
library(phenology)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
  reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
result_Gratiot<-fit_phenology(data=data_Gratiot,
  parametersfit=parg, parametersfixed=NULL, trace=1)
# Generate set of priors for Bayesian analysis
pmcmc <- phenology_MHmcmc_p(result_Gratiot, accept = TRUE)
result_Gratiot_mcmc <- phenology_MHmcmc(result = result_Gratiot, n.iter = 10000,
  parametersMCMC = pmcmc, n.chains = 1, n.adapt = 0, thin = 1, trace = FALSE)
# Get standard error of parameters
summary(result_Gratiot_mcmc)
# Make diagnostics of the mcmc results using coda package
mcmc <- as.mcmc(result_Gratiot_mcmc)
require(coda)
heidel.diag(mcmc)
raftery.diag(mcmc)
autocorr.diag(mcmc)
acf(mcmc[[1]][,"LengthB"], lag.max=200, bty="n", las=1)
acf(mcmc[[1]][,"Max_Gratiot"], lag.max=50, bty="n", las=1)
batchSE(mcmc, batchSize=100)
# The batch standard error procedure is usually thought to
# be not as accurate as the time series methods used in summary
summary(mcmc)$statistics[,"Time-series SE"]
plot(result_Gratiot_mcmc, parameters=3, las=1, xlim=c(-10, 300))

## End(Not run)
```

```
summary.phenology      Print the result information from a result object.
```

Description

The function `print.phenology` displays from a result.

Usage

```
## S3 method for class 'phenology'
summary(object, ...)
```

Arguments

<code>object</code>	A result file generated by <code>fit_phenology</code>
<code>...</code>	Not used

Details

summary.phenology prints the information from a result object.

Value

None

Author(s)

Marc Girondot

Examples

```
library(phenology)
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
## Not run:
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)

## End(Not run)
data(result_Gratiot)
# Display information from the result
summary(result_Gratiot)
```

summary.phenologymap *Print information on a phenologymap object.*

Description

summary.phenologymap print information on a phenologymap object

Usage

```
## S3 method for class 'phenologymap'
summary(object, ...)
```

Arguments

object A map generated with map_phenology.
 ... Not used

Value

Return None

Author(s)

Marc Girondot

Examples

```
library("phenology")
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
## Not run:
result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)

## End(Not run)
data(result_Gratiot)
# Extract the fitted parameters
parg1<-extract_result(result_Gratiot)
# Add constant Alpha and Tau values
# [day d amplitude=(Alpha+Nd*Beta)^Tau with Nd being the number of counts for day d]
pfixed<-c(parg1, Alpha=0, Tau=1)
pfixed<-pfixed[-which(names(pfixed)=="Theta")]
# The only fitted parameter will be Beta
parg2<-c(Beta=0.5, parg1["Theta"])
# Generate a likelihood map
# [default Phi=seq(from=0.1, to=20, length.out=100) but it is very long]
# Take care, it takes 20 hours ! The data map_Gratiot has the result
## Not run:
map_Gratiot<-map_phenology(data=data_Gratiot,
Phi=seq(from=0.1, to=20, length.out=100),
parametersfit=parg2, parametersfixed=pfixed)

## End(Not run)
data(map_Gratiot)
```

```
# Print the information on a map
summary(map_Gratiot)
```

summary.phenologyout *Print the summary information from a output object.*

Description

The function summary.phenologyout displays the output from a plot.

Usage

```
## S3 method for class 'phenologyout'
summary(object, ...)
```

Arguments

object	An output generated by plot_phenology
...	Not used

Details

summary.phenologyout prints the information from a result object.

Value

None

Author(s)

Marc Girondot

Examples

```
library(phenology)
# Read a file with data
## Not run:
Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", header=FALSE)

## End(Not run)
data(Gratiot)
# Generate a formatted list named data_Gratiot
data_Gratiot<-add_phenology(Gratiot, name="Complete",
reference=as.Date("2001-01-01"), format="%d/%m/%Y")
# Generate initial points for the optimisation
parg<-par_init(data_Gratiot, parametersfixed=NULL)
# Run the optimisation
## Not run:
```

```

result_Gratiot<-fit_phenology(data=data_Gratiot,
parametersfit=parg, parametersfixed=NULL, trace=1)

## End(Not run)
data(result_Gratiot)
# Plot the phenology and get some stats
output<-plot(result_Gratiot)
# Show the output
summary(output)

```

toggle_Min_PMin	<i>Transform a set of parameters from Min, MinB or MinE to PMin, PminB or PminE, or reverse</i>
-----------------	-------------------------------------------------------------------------------------------------

Description

This function is used to transform a set of parameters that uses Min, MinB or MinE to a set of parameters that uses PMin, PminB or PminE, or reverse.

Usage

```
toggle_Min_PMin(parameters = stop("A set of parameters must be indicated"))
```

Arguments

parameters Set of current parameters

Details

toggle_Min_PMin transforms a set of parameters from Min, MinB or MinE to PMin, PminB or PminE, or reverse

Value

Return a set of modified parameters

Author(s)

Marc Girondot

Examples

```

# Read a file with data
# Gratiot<-read.delim("http://max2.ese.u-psud.fr/epc/conservation/BI/Complete.txt", , header=FALSE)
data(Gratiot)
# Generate a formatted list named data_Gratiot
refdate <- as.Date("2001-01-01")
data_Gratiot<-add_phenology(Gratiot, name="Complete", reference=refdate, format="%d/%m/%Y")
# Generate initial points for the optimisation

```

```
parg<-par_init(data_Gratiot)
# Change the parameters to PMinB and PMinE
parg1<-toggle_Min_PMin(parameters=parg)
# And change back to MinB and MinE
parg2<-toggle_Min_PMin(parameters=parg1)
```

Index

- *Topic **Ecology**
 - phenology-package, 3
- *Topic **Lunar**
 - moon_phase, 34
- *Topic **Lune**
 - moon_phase, 34
- *Topic **Moon**
 - moon_phase, 34
- *Topic **Phenology**
 - phenology-package, 3
- *Topic **Seasonality**
 - phenology-package, 3
- *Topic **Tide**
 - getTide, 20
- *Topic **datasets**
 - Gratiot, 21
 - map_Gratiot, 31
 - result_Gratiot, 56
 - result_Gratiot1, 57
 - result_Gratiot2, 58
 - result_Gratiot_Flat, 59
 - result_Gratiot_mcmc, 60

adapt_parameters, 4
add_phenology, 5
add_SD, 7
as.mcmc.mcmcComposite, 8
as.par.mcmcComposite, 9
asc, 10

barplot_errbar, 10
BE_to_LBLE, 12

ChangeCoordinate, 13
chr, 13
clean.knitr, 14
compare_AIC, 15
compassRose2, 16
convert.ts, 17
extract_result, 18

fit_phenology, 19
getTide, 20
Gratiot, 21
growlnotify, 22

ind_long_lat, 23

L_to_LBLE, 29
LBLE_to_BE, 24
LBLE_to_L, 25
likelihood_phenology, 26
local.search, 27
logLik.phenology, 28

map.scale2, 30
map_Gratiot, 31
map_phenology, 31
MinBMinE_to_Min, 33
moon_phase, 34

par_init, 35
phenology, 36
phenology-package, 3
phenology_MHmcmc, 37
phenology_MHmcmc_p, 38
phenology_swot, 40
plot.mcmcComposite, 41
plot.phenology, 42
plot.phenologymap, 44
plot_add, 45
plot_delta, 46
plot_errbar, 47
plot_phi, 49
print.phenology, 50
print.phenologymap, 51
print.phenologyout, 53

read_folder, 54
remove_site, 55
result_Gratiot, 56

result_Gratiot1, [57](#)
result_Gratiot2, [58](#)
result_Gratiot_Flat, [59](#)
result_Gratiot_mcmc, [60](#)

ScalePreviousPlot, [60](#)
shift_sinusoid, [61](#)
summary.mcmcComposite, [63](#)
summary.phenology, [64](#)
summary.phenologymap, [65](#)
summary.phenologyout, [67](#)

toggle_Min_PMin, [68](#)