

# Package ‘embryogrowth’

October 6, 2014

**Type** Package

**Title** Tools to Analyze the Thermal Reaction Norm of Embryo Growth

**Version** 5.1

**Date** 2014-10-05

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

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

**Description** Tools to analyze the thermal reaction norm of embryo growth

**Depends** deSolve, parallel, phenology, Hmisc, R (>= 2.14.2)

**Suggests** entropy, numDeriv, coda, shiny

**License** GPL-2

**LazyData** yes

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-10-06 20:34:44

## R topics documented:

embryogrowth-package . . . . .	3
ChangeSSM . . . . .	4
dydt.exponential . . . . .	6
dydt.Gompertz . . . . .	7
dydt.linear . . . . .	8
embryogrowth_MHmcmc_p . . . . .	9
FormatNests . . . . .	10
GenerateAnchor . . . . .	11

GenerateConstInc . . . . .	12
GenerateTest . . . . .	13
hist.Nests . . . . .	14
hist.NestsResult . . . . .	15
info.nests . . . . .	16
likelihoodR . . . . .	17
logLik.NestsResult . . . . .	19
logLik.tsd . . . . .	20
merge.mcmcComposite . . . . .	21
minmax.periodic . . . . .	22
MovingIncubation . . . . .	23
nest . . . . .	25
plot.NestsResult . . . . .	26
plot.tsd . . . . .	28
plotR . . . . .	30
plotR_hist . . . . .	32
plot_transition . . . . .	33
predict.tsd . . . . .	34
resultNest_4p . . . . .	35
resultNest_4p_transition . . . . .	36
resultNest_4p_weight . . . . .	37
resultNest_6p . . . . .	38
resultNest_newp . . . . .	39
result_mcmc_4p . . . . .	40
result_mcmc_4p_weight . . . . .	41
result_mcmc_6p . . . . .	42
result_mcmc_newp . . . . .	44
searchR . . . . .	45
STSRE_NestingArea . . . . .	47
STSRE_TSD . . . . .	48
summary.Nests . . . . .	51
switch.transition . . . . .	52
tempConst . . . . .	53
temperature.periodic . . . . .	54
TestParallel . . . . .	55
tsd . . . . .	56
tsd_MHmcmc . . . . .	59
tsd_MHmcmc_p . . . . .	61
weightmaxentropy . . . . .	62

---

embryogrowth-package *The package embryogrowth*

---

## Description

Tools to Analyze the Thermal Reaction Norm of Embryo Growth

## Details

Fit a parametric function that describes dependency of embryo growth to temperature

Package:	embryogrowth
Type:	Package
Version:	5.1 - build 406
Date:	2014-10-05
License:	GPL (>= 2)
LazyLoad:	yes

## Author(s)

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

## References

Girondot, M. & Kaska, Y. 2014. A model to predict the thermal reaction norm for the embryo growth rate from field data. *Journal of Thermal Biology*, 45, 96-102.

## See Also

Delmas, V., Prevot-Julliard, A.-C., Pieau, C. & Girondot, M. 2008. A mechanistic model of temperature-dependent sex determination in a Chelonian, the European pond turtle. *Functional Ecology*, 22, 84-93.

Girondot, M., Ben Hassine, S., Sellos, C., Godfrey, M. & Guillon, J.-M. 2010. Modeling thermal influence on animal growth and sex determination in Reptiles: being closer of the target gives new views. *Sexual Development*, 4, 29-38.

Girondot, M. 1999. Statistical description of temperature-dependent sex determination using maximum likelihood. *Evolutionary Ecology Research*, 1, 479-486.

Girondot M. (Submitted) On the concept of embryological thermosensitive period for sex determination in reptiles.

## Examples

```
## Not run:  
library("embryogrowth")  
packageVersion("embryogrowth")  
data(nest)
```

```

formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
x <- structure(c(115.758929130522, 428.649022170996, 503.687251738993,
12.2621455821612, 306.308841227278, 116.35048615105), .Names = c("DHA",
"DHH", "DHL", "DT", "T12L", "Rho25"))
# or
x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
resultNest_4p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_4p)
pMCMC <- embryogrowth_MHmcmc_p(resultNest_4p, accept=TRUE)
# Take care, it can be very long, sometimes several days
result_mcmc_4p <- embryogrowth_MHmcmc(result=resultNest_4p,
parametersMCMC=pMCMC, n.iter=10000, n.chains = 1, n.adapt = 0,
thin=1, trace=TRUE)
data(result_mcmc_4p)
out <- as.mcmc(result_mcmc_4p)
# This out obtained after as.mcmc can be used with coda package
# plot() can use the direct output of embryogrowth_MHmcmc() function.
plot(result_mcmc_4p, parameters=1, xlim=c(0,550))
plot(result_mcmc_4p, parameters=3, xlim=c(290,320))
# summary() permits to get rapidly the standard errors for parameters
summary(result_mcmc_4p)
# The batch standard error procedure is usually thought to
# be not as accurate as the time series methods.
se <- result_mcmc_4p$BatchSE
# or
se <- result_mcmc_4p$TimeSeriesSE

## End(Not run)

```

---

ChangeSSM

---

*Generate set of parameters for Schoolfield-Sharpe-Magnuson model*


---

## Description

Generate a set of parameters for Schoolfield-Sharpe-Magnuson model

## Usage

```

ChangeSSM(temperatures = (200:350)/10,
parameters = stop("A set of parameters must be supplied"),
initial.parameters = stop("A set of parameters for new model must be supplied"),
...)

```

**Arguments**

temperatures    A vector with incubation temperatures in degrees Celsius  
 parameters     A vector of parameters for model to be converted (4 or 6 parameters)  
 initial.parameters    A vector of parameters for initial model model to be fitted (4 or 6 parameters)  
 ...             A control list to be used with optim, see ?optim

**Details**

ChangeSSM convert Schoolfield-Sharpe-Magnuson model from 4 to 6 parameters or reverse

**Value**

A vector with parameters

**Author(s)**

Marc Girondot

**Examples**

```
## Not run:
data(resultNest_6p)
x1 <- resultNest_6p$par
data(resultNest_4p)
x2 <- resultNest_4p$par
temperaturesC <- (200:350)/10
s <- ChangeSSM(temperatures=temperaturesC, parameters=x1, initial.parameters=x2)
plotR(list(resultNest_6p, resultNest_4p, s), ylim=c(0,0.3),
col=list("black", "red", "green"), lty=list(1,1,1),
legend=list("R function to mimic", "Initial new R function",
"Fitted new R function"), show.box=FALSE)
# Other example to fit anchored parameters
data(resultNest_4p)
x0 <- resultNest_4p$par
t <- hist(resultNest_4p, plot=FALSE)
x <- c(3.4, 3.6, 5.4, 5.6, 7.6, 7.5, 3.2)
names(x) <- seq(from=range(t$temperatures)[1], to=range(t$temperatures)[2], length.out=7)
newx <- ChangeSSM(temperatures = (200:350)/10, parameters = x0, initial.parameters = x,
control=list(maxit=5000))
# Example on how to generate a set of SSM parameters from anchored parameters
xanchor <- GenerateAnchor(nests=resultNest_4p)
x <- resultNest_4p$par
xanchor["294"] <- 0
xanchor["308"] <- 2.3291035
xprime <- ChangeSSM(parameters = xanchor,
initial.parameters = x, control=list(maxit=5000))
plotR(result=resultNest_4p, parameters=list(resultNest_4p$par, xprime$par),
ylim=c(0,0.3), col=c("black", "red"),
legend=list("Fitted parameters", "Constrained parameters"))
```

```
## End(Not run)
```

---

```
dydt.exponential      Return the derivative of the exponential function
```

---

### Description

Return the derivative of the exponential function  
 dydt.exponential(t, size, parms)

### Usage

```
dydt.exponential(t, size, parms)
```

### Arguments

t	The time in any unit
size	The current size
parms	A vector with alpha and K values being c(alpha=x1, K=x2). K is not used.

### Details

dydt.exponential returns the derivative of the exponential function.

### Value

A list with the derivative

### Author(s)

Marc Girondot

### Examples

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
x <- structure(c(306.174998729436, 333.708348843241,
299.856306141849, 149.046870203155),
.Names = c("DHA", "DHH", "T12H", "Rho25"))
# K or rK are not used for dydt.linear or dydt.exponential
resultNest_4p_exponential <- searchR(parameters=x, fixed.parameters=NULL,
temperatures=formatted, derivate=dydt.exponential, M0=1.7,
```

```
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)

## End(Not run)
```

---

dydt.Gompertz                      *Return the derivative of the Gompertz function*

---

## Description

Return the derivative of the Gompertz function  
dydt.Gompertz(t, size, parms)

## Usage

```
dydt.Gompertz(t, size, parms)
```

## Arguments

t	The time in any unit
size	The current size
parms	A vector with alpha and K values being c(alpha=x1, K=x2)

## Details

dydt.Gompertz returns the derivative of the Gompertz function.

## Value

A list with the derivative

## Author(s)

Marc Girondot

## Examples

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
# K or rK are not used for dydt.linear or dydt.exponential
```

```

resultNest_4p <- searchR(parameters=x, fixed.parameters=pfixed,
  temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
  test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_4p)

## End(Not run)

```

---

dydt.linear

*Return the derivative of the linear function*


---

### Description

Return the derivative of the linear function  
dydt.linear(t, size, parms)

### Usage

```
dydt.linear(t, size, parms)
```

### Arguments

t	The time in any unit
size	The current size
parms	A vector with alpha being c(alpha=x1, K=x2). Only alpha is used.

### Details

dydt.Linear returns the derivative of the linear function.

### Value

A list with the derivative

### Author(s)

Marc Girondot

### Examples

```

## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
x <- structure(c(306.174998729436, 333.708348843241, 299.856306141849,
  149.046870203155), .Names = c("DHA", "DHH", "T12H", "Rho25"))

```



```
# K or rK are not used for dydt.linear or dydt.exponential
resultNest_4p_linear <- searchR(parameters=x, fixed.parameters=NULL,
temperatures=formatted, derivate=dydt.linear, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)

## End(Not run)
```

---

embryogrowth\_MHmcmc\_p *Generates set of parameters to be used with embryogrowth\_MHmcmc()*

---

## Description

Interactive script used to generate set of parameters to be used with embryogrowth\_MHmcmc().

## Usage

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

## Arguments

result	An object obtained after a SearchR fit
accept	If TRUE, the script does not wait user information

## Details

embryogrowth\_MHmcmc\_p generates set of parameters to be used with embryogrowth\_MHmcmc()

## Value

A matrix with the parameters

## Author(s)

Marc Girondot

## Examples

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "T12H", "DHA", "DHH", "DHL", "Rho25"
```

```

x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
resultNest_4p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_4p)
pMCMC <- embryogrowth_MHmcmc_p(resultNest_4p, accept=TRUE)

## End(Not run)

```

---

FormatNests

*Create a dataset of class Nests to be used with searchR*


---

### Description

Will create a dataset of class Nests to be used with searchR  
FormatNests(nest, previous=x) with x being a previously formatted data.  
The raw data must be organized being:  
First column is the time in minutes since the beginning of incubation  
Each column next is the trace of temperatures, one column for each nest.  
For example, for two nests:

Time	Nest1	Nest2
0	29.8	27.6
90	30.2	28.8
120	30.4	30.7
180	31.2	32.6
...		
65800	30.8	32.6
65890	30.2	
65950	30.4	

The Nest1 ends incubation at 65800 minutes whereas Nest2 ends incubation at 65950 (last row with temperature for each).

The parameter Weight is a vector: weight=c(Nest1=1, Nest2=1.2)

### Usage

```

FormatNests(data = stop("A dataset must be provided !"), previous = NULL,
simplify = TRUE, weight = NULL)

```

### Arguments

data	Data to be newly formatted
previous	Data already formatted

`simplify` If TRUE, simply the time series by removing identical time series of temperatures

`weight` Named vector with weight for likelihood

### Details

FormatNests creates a dataset of class "Nests" to be used with searchR

### Value

A list with all the nests formatted to be used with searchR.

### Author(s)

Marc Girondot

### Examples

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest, previous=NULL)
formatted <- FormatNests(nest)

## End(Not run)
```

---

GenerateAnchor      *Generate a set of anchored parameters*

---

### Description

Generate a set of anchored parameters

### Usage

```
GenerateAnchor(temperatures = NULL, nests = NULL, number.anchors = 7)
```

### Arguments

`temperatures` A vector with temperatures to serve as anchors

`nests` Formated nest data or result object obtained from searchR()

`number.anchors` Number of anchors

### Details

GenerateAnchor Generate a set of anchored parameters

**Value**

A vector with parameters

**Author(s)**

Marc Girondot

**Examples**

```
## Not run:
# Example to generate anchored parameters
newp <- GenerateAnchor()
newp <- GenerateAnchor(temperatures=seq(from=20,
  to=35, length.out=7))
newp <- GenerateAnchor(number.anchors=7)
data(nest)
formatted <- FormatNests(nest, previous=NULL)
newp <- GenerateAnchor(nests=formatted)
newp <- GenerateAnchor(nests=formatted, number.anchors=10)
data(resultNest_4p)
newp <- GenerateAnchor(nests=resultNest_4p, number.anchors=7)
newp <- GenerateAnchor(nests=resultNest_4p, temperatures=seq(from=20,
  to=35, length.out=10))
newp <- GenerateAnchor(nests=resultNest_4p, number.anchors=7)
newp <- c(newp, Scale=1)

## End(Not run)
```

---

GenerateConstInc	<i>Generate a data.frame with constant incubation temperature and incubation duration</i>
------------------	---

---

**Description**

Generate a data.frame from constant incubation temperature and incubation duration

**Usage**

```
GenerateConstInc(durations = stop("At least one incubation length must be provided"),
  temperatures = stop("At least one incubation temperature must be provided"),
  names = NULL)
```

**Arguments**

durations	A vector with incubation durations
temperatures	A vector with incubation temperatures
names	A vector of column names

**Details**

GenerateConstInc generates a data.frame with constant incubation temperature and incubation duration

**Value**

A date.frame that can be used with FormatNests()

**Author(s)**

Marc Girondot

**Examples**

```
## Not run:
temp_cst <- GenerateConstInc(durations=c(150000, 100100, 100000),
  temperatures=c(28, 30.5, 30.6),
  names=c("T28", "T30.5", "T30.6"))

## End(Not run)
```

---

GenerateTest

*Generate a data.frame that can be used as test value for searchR()*

---

**Description**

Generate a data.frame that can be used as test value for searchR()

**Usage**

```
GenerateTest(series = stop("A result object or names of series must be provided"),
  size = NULL, previous = NULL)
```

**Arguments**

series	Name of series or object from searchR()
size	Size or mass at hatching. Will be recycled if necessary
previous	Previous formatted test data

**Details**

GenerateTest Generate a data.frame that can be used as test value for searchR()

**Value**

A data.frame with size or mass at hatching for each nest

**Author(s)**

Marc Girondot

**Examples**

```
## Not run:
library(embryogrowth)
data(resultNest_4p)
testsize1 <- GenerateTest(resultNest_4p)
testsize2 <- GenerateTest(series=resultNest_4p,
size=c(Mean=39.3, SD=1.92))

## End(Not run)
```

---

hist.Nests

*Show the histogram of temperatures with set of nests*

---

**Description**

Show the histogram of temperatures with set of nests hist(data)

**Usage**

```
## S3 method for class 'Nests'
hist(x, series = "all", ...)
```

**Arguments**

x	Data formatted using formatdata.
series	Series to be used, logical (TRUE ou FALSE), numbers or names. If "all", all series are used.
...	Parameters used by hist function

**Details**

hist.Nests shows the histogram of temperatures with set of nests

**Value**

A list with an histogram object with information on histogram or NULL if no series was selected and the complete set of temperatures used.

**Author(s)**

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

**Examples**

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
h <- hist(formated, series="all")

## End(Not run)
```

---

hist.NestsResult      *Show the histogram of temperatures with set of nests*

---

**Description**

Show the histogram of temperatures with set of nests hist(data)

**Usage**

```
## S3 method for class 'NestsResult'
hist(x, series = "all", ...)
```

**Arguments**

x	Results obtained after searchR
series	Series to be used, logical (TRUE ou FALSE), numbers or names. If "all", all series are used.
...	Parameters used by hist function (example main="Title")

**Details**

hist.NestsResult shows the histogram of temperatures with set of nests

**Value**

A list with an histogram object with information on histogram or NULL if no series was selected and the complete set of temperatures used.

**Author(s)**

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

**Examples**

```
## Not run:
library(embryogrowth)
data(resultNest_4p)
h <- hist(resultNest_4p, series=c(1:5))

## End(Not run)
```

---

 info.nests

*Calculate statistics about nests*


---

## Description

Calculate statistics about nests

The `embryo.stages` is a list with stage numbers and relative size as compared to final size at the beginning of the stage.

For example `embryo.stages=list(number=21:30, size=c(8.4, 9.4, 13.6, 13.8, 18.9, 23.5, 32.2, 35.2, 35.5, 38.5)/39.33)` indicates that the stages 21 begins at the relative size of 8.4/39.33 series can be indicated as the name of the series, its number or TRUE or FALSE. "all" indicates that all series must be printed.

## Usage

```
info.nests(x, parameters = NULL, fixed.parameters = NULL, SE = NULL,
  temperatures = NULL, derivate = NULL, test = NULL, stopattest = FALSE,
  M0 = NULL, series = "all", TSP.borders = c(21, 26), progress = TRUE,
  embryo.stages = "Caretta caretta", replicate.CI = 100, ref.stage = TRUE)
```

## Arguments

<code>x</code>	A result file generated by searchR
<code>parameters</code>	A set of parameters if result is not provided.
<code>fixed.parameters</code>	Another set of parameters if result is not provided.
<code>SE</code>	Standard error for each parameter if result is not provided.
<code>temperatures</code>	Timeseries of temperatures. Will replace the one in result.
<code>derivate</code>	Function used to fit embryo growth: <code>dydt.Gompertz</code> , <code>dydt.exponential</code> or <code>dydt.linear</code>
<code>test</code>	Mean and SD of size of hatchlings
<code>stopattest</code>	True or False, does the plot stops when proxi of size reaches the mean test size.
<code>M0</code>	Measure of hatchling size proxi at laying date
<code>series</code>	The name or number of the series to be displayed. Only one series can be displayed at a time.
<code>TSP.borders</code>	The limits of TSP
<code>progress</code>	If FALSE, the progress bar is not shown (useful for use with Sweave)
<code>embryo.stages</code>	The embryo stages. At least <code>TSP.borders</code> stages must be provided if <code>show.TSP</code> is TRUE
<code>replicate.CI</code>	Number of randomizations to estimate CI
<code>ref.stage</code>	If TRUE, the reference size for stage is the observed one in modelling. If FALSE, it is the field observed size.



**Details**

Calculate statistics about nests

**Value**

A list with informations about thermosensitive period length for the series of nests analyzed

**Author(s)**

Marc Girondot

**Examples**

```
## Not run:
library(embryogrowth)
data(resultNest_4p)
info.nests(resultNest_4p, series=1,
SE=c(DHA=1.396525, DHH=4.101217, T12H=0.04330405, Rho25=1.00479))
# to get info for all nests at the same time, use
infoall <- info.nests(resultNest_4p, series="all")
# it is easier to use after dataframe conversion
infoall.df <- as.data.frame(infoall$summary)

## End(Not run)
```

---

likelihoodR

---

*Estimate the likelihood of a set of parameters for nest incubation data*


---

**Description**

Estimate the likelihood of a set of parameters for nest incubation data

**Usage**

```
likelihoodR(result = NULL, parameters = NULL, fixed.parameters = NULL,
temperatures = NULL, derivate = NULL, test = NULL, M0 = NULL,
hessian = FALSE, weight = NULL, parallel = TRUE, echo = TRUE)
```

**Arguments**

result	A object obtained after searchR or likelihoodR
parameters	A set of parameters
fixed.parameters	A set of parameters that will not be changed
temperatures	Timeseries of temperatures
derivate	Function used to fit embryo growth: dydt.Gompertz, dydt.exponential or dydt.linear
test	Mean and SD of size of hatchlings

M0	Measure of hatchling size or mass proxy at laying date
hessian	If TRUE, the hessian matrix is estimated and the SE of parameters estimated.
weight	A named vector of the weight for each nest for likelihood estimation
parallel	If true, try to use several cores using parallel computing.
echo	If FALSE, does not display the result.

## Details

likelihoodR estimates the likelihood of a set of parameters for nest incubation data

## Value

A result object

## Author(s)

Marc Girondot

## Examples

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
# K for Gompertz must be set as fixed parameter or being a constant K
# or relative to the hatchling size rK
x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
# K or rK are not used for dydt.linear or dydt.exponential
LresultNest_4p <- likelihoodR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92))
data(resultNest_4p)
LresultNest_4p <- likelihoodR(result=resultNest_4p)

## End(Not run)
```

---

logLik.NestsResult      *Return Log Likelihood of a fit generated by searchR*

---

**Description**

Return Log Likelihood of a fit generated by searchR

**Usage**

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

**Arguments**

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

**Details**

logLik.NestsResult 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(embryogrowth)  
data(resultNest_4p)  
logLik(resultNest_4p)  
AIC(resultNest_4p)  
  
## End(Not run)
```

---

`logLik.tsd`*Return Log Likelihood of a fit generated by tsd*

---

**Description**

Return Log Likelihood of a fit generated by tsd

**Usage**

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

**Arguments**

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

**Details**

logLik.tsd 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(embryogrowth)  
m <- c(10, 14, 7, 4, 3, 0, 0)  
f <- c(0, 1, 2, 4, 15, 10, 13)  
t <- c(25, 26, 27, 28, 29, 30, 31)  
result <- tsd(males=m, females=f, temperatures=t)  
logLik(result)  
AIC(result)  
  
## End(Not run)
```

---

merge.mcmcComposite    *Merge two mcmcComposite results*

---

### Description

Merge two mcmcComposite results

### Usage

```
## S3 method for class 'mcmcComposite'  
merge(x, y, ...)
```

### Arguments

x	A mcmcComposite result
y	A mcmcComposite result
...	Other parameters for merge function (not used)

### Details

merge.mcmcComposite Merge two mcmcComposite results

### Value

A mcmcComposite result

### Author(s)

Marc Girondot

### Examples

```
## Not run:  
data(result_mcmc_4p)  
# Of course, normally you should do it with two different mcmc objects !  
result_mcmc <- merge(result_mcmc_4p, result_mcmc_4p)  
  
## End(Not run)
```

---

minmax.periodic      *Search for minimum and maximum temperatures in periodic timeseries*

---

### Description

Search for minimum and maximum for periodic timeseries when only intermediate values are known.

For each couple of value with an increasing or decreasing segment of the sinusoid function, it is possible to estimate a minimum and maximum values using analytical algebra.

Then the average and standard deviations of all minima and maxima are evaluated.

It should be noted that any extremum can be estimated at least twice, one by increasing segment and one by decreasing segment.

### Usage

```
minmax.periodic(time.minmax.daily, time.obs, temp.obs, period = 24)
```

### Arguments

time.minmax.daily	A named vector with Min and Max being the time in the day with minimum and maximum temperatures
time.obs	A vector with the time at which temperatures are recorded
temp.obs	A vector with the recorded temperatures recorded
period	The unit of day period (24 for hours, 24*60 for minutes)

### Details

minmax.periodic search for minimum and maximum temperatures in periodic timeseries

### Value

A data.frame with a column time, a column temperature and a column sd

### Author(s)

Marc Girondot

### Examples

```
## Not run:
# Generate a timeserie of time
time.obs <- NULL
for (i in 0:9) time.obs <- c(time.obs, c(0, 6, 12, 18)+i*24)
# For these time, generate a timeseries of temperatures
temp.obs <- rep(NA, length(time.obs))
temp.obs[3+(0:9)*4] <- rnorm(10, 25, 3)
temp.obs[1+(0:9)*4] <- rnorm(10, 10, 3)
```

```

for (i in 1:(length(time.obs)-1))
  if (is.na(temp.obs[i]))
    temp.obs[i] <- mean(c(temp.obs[i-1], temp.obs[i+1]))
  if (is.na(temp.obs[length(time.obs)]))
    temp.obs[length(time.obs)] <- temp.obs[length(time.obs)-1]/2

# Search for the minimum and maximum values
r <- minmax.periodic(time.minmax.daily=c(Min=2, Max=15),
time.obs=time.obs, temp.obs=temp.obs, period=24)

# Estimate all the temperatures for these values
t <- temperature.periodic(minmax=r)

plot_errbar(x=t[, "time"], y=t[, "temperature"],
errbar.y=ifelse(is.na(t[, "sd"]), 0, 2*t[, "sd"]),
type="l", las=1, bty="n", errbar.y.polygon = TRUE,
xlab="hours", ylab="Temperatures", ylim=c(0, 35),
errbar.y.polygon.list = list(col="grey"))

plot_add(x=t[, "time"], y=t[, "temperature"], type="l")

## End(Not run)

```

---

MovingIncubation

*Simulate incubation of a nest with the beginning of incubation varying*


---

## Description

Simulate incubation of a nest with the beginning varying day by day  
Temperatures must be in a data.frame with one column (Time) being the time and the second the temperatures (Temperatures). Do not use FormatNests() for this dataframe.

## Usage

```

MovingIncubation(result = stop("A result file from searchR() must be provided"),
temperatures = stop("A timeseries of temperature must be provided"),
times = stop("A timeseries of time for each temperature must be provided"),
temperatures.end.incubation = temperatures,
average.incubation.duration = 60 * 1440, skip = 1, parameters = NULL,
fixed.parameters = NULL, SE = NULL, derivate = NULL, test = NULL,
M0 = NULL, TSP.borders = c(21, 26), embryo.stages = "Caretta caretta",
replicate.CI = 1, min.length = 40 * 1440, max.length = 150 * 1440,
progress = TRUE)

```

## Arguments

result	A result file generated by searchR
temperatures	Timeseries of temperatures

times	Timeseries of times
temperatures.end.incubation	Timeseries of temperatures at the end of incubation
average.incubation.duration	The average time to complete incubation
skip	Number of data to skip between two runs
parameters	A set of parameters if result is not provided.
fixed.parameters	Another set of parameters if result is not provided.
SE	Standard error for each parameter if not present in result is not provided (not used still)
derivate	Function used to fit embryo growth: dydt.Gompertz, dydt.exponential or dydt.linear
test	Mean and SD of size of hatchlings as a vector ie test=c(Mean=xx, SD=yy)
M0	Measure of hatchling size proxi at laying date
TSP.borders	The limits of TSP
embryo.stages	The embryo stages. At least TSP.borders stages must be provided to estimate TSP length
replicate.CI	Number of randomizations to estimate CI (not used still)
min.length	Minimum length of possible incubation
max.length	Maximum length of possible incubation
progress	If TRUE, progression bar is shown (desactivate for sweave or knitr)

### Details

MovingIncubation simulate incubation of a nest with the beginning varying day by day

### Value

A dataframe with informations about thermosensitive period length and incubation length day by day of incubation

### Author(s)

Marc Girondot

### Examples

```
## Not run:
library(embryogrowth)
data(resultNest_4p)
ti <- seq(from=0, to=(60*24*100), by=60)
temp <- rnorm(length(ti), 29, 5)
out <- MovingIncubation(result=resultNest_4p, temperatures=temp,
times=ti)

## End(Not run)
```



---

nest	<i>Timeseries of temperatures for nests</i>
------	---

---

**Description**

Timeseries of temperatures for nests

**Usage**

nest

**Format**

A dataframe with raw data.

**Details**

Timeseries of temperatures for nests

**Author(s)**

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

**References**

Girondot, M. & Kaska, Y. Submitted. A model to predict temperature dependency on embryo growth rate and incubation duration from field data.

**Examples**

```
## Not run:  
library(embryogrowth)  
data(nest)  
  
## End(Not run)
```

---

plot.NestsResult      *Plot the information about a nest*

---

### Description

Plot the information about a nest

The embryo.stages is a list with stage numbers and relative size as compared to final size at the beginning of the stage.

For example for *Caretta caretta*, embryo.stages=list(number=21:30, size=c(8.4, 9.4, 13.6, 13.8, 18.9, 23.5, 32.2, 35.2, 35.5, 38.5)/39.33) indicates that the stages 21 begins at the relative size of 8.4/39.33.

The default is for the turtle "*Caretta caretta*".

Series can be indicated as the name of the series, its number or succession of TRUE or FALSE. "all" indicates that all series must be printed.

The object return is an invisible list composed of two lists: \$summary with the summary statistics for each nest and \$traces with the change of length or mass for each nest.

show.fioritures does not affect show.test option.

### Usage

```
## S3 method for class 'NestsResult'
plot(x, ..., OutPlot = NULL, parameters = NULL,
     fixed.parameters = NULL, SE = NULL, temperatures = NULL,
     derivate = NULL, test = NULL, stopattest = FALSE, M0 = NULL,
     series = 1, col.stages = "blue", col.PT = "red", col.TSP = "gray",
     col.temperatures = "green", col.S = "black", lty.temperatures = 1,
     lwd.temperatures = 2, ylimT = c(25, 35), ylimS = NULL, xlim = NULL,
     TSP.borders = c(21, 26), embryo.stages = "Caretta caretta",
     show.stages = TRUE, show.TSP = TRUE, show.third = TRUE,
     show.CI = TRUE, replicate.CI = 100, ref.stage = TRUE,
     show.fioritures = TRUE, progress = TRUE, show.temperatures = TRUE,
     show.PT = TRUE, PT = c(NA, NA), show.test = TRUE,
     lab.third = "2nd third of incubation", at.lab.third = 4, lab.PT = "PT",
     lab.stages = "Stages", mar = c(4, 5, 4, 5) + 0.3,
     xlab = "Days of incubation", ylabT = expression("Temperatures in " *
     degree * "C"), ylabS = "Embryo metric", show.plot = TRUE)
```

### Arguments

x	A result file generated by searchR
...	Parameters for plot()
OutPlot	A previous output of a plot.NestsResult() or info.nests()
parameters	A set of parameters if result is not provided.
fixed.parameters	Another set of parameters if result is not provided.
SE	Standard error for each parameter if result is not provided.

temperatures	Timeseries of temperatures. Will replace the one in result.
derivate	Function used to fit embryo growth: dydt.Gompertz, dydt.exponential or dydt.linear
test	Mean and SD of size of hatchlings ie test=c(Mean=xx, SD=yy)
stopattest	True or False, does the plot stops when proxi of size reaches the mean test size.
M0	Measure of hatchling size proxi at laying date
series	The name or number of the series to be displayed. Only one series can be displayed at a time.
col.stages	The color of the stages
col.PT	The color of the pivotal temperature
col.TSP	The color of the TSP
col.temperatures	The color of the temperatures
col.S	The color of the size or mass. Can be a vector (useful when series="all" option).
lty.temperatures	Type of line for temperatures
lwd.temperatures	Width of line for temperatures
ylimT	Range of temperatures to be displayed
ylimS	Range of size to be displayed
xlim	Range of incubation days to be displayed
TSP.borders	The limits of TSP
embryo.stages	The embryo stages. At least TSP.borders stages must be provided if show.TSP is TRUE
show.stages	True or False, does the embryo stages should be displayed
show.TSP	True or False, does the TSP borders should be displayed
show.third	True or False, does the first and second third borders should be displayed
show.CI	True or False, do the confidence intervals should be displayed
replicate.CI	Number of randomizations to estimate CI
ref.stage	If TRUE, the reference size for stage is the observed one in modelling. If FALSE, it is the field observed size.
show.fioritures	If FALSE, set show.PT, show.temperatures, show.stages, show.TSP, show.third, show.CI to FALSE
progress	If FALSE, the progress bar is not shown (useful for use with sweave or knitr)
show.temperatures	True or False, does the temperatures should be displayed
show.PT	TRUE or FALSE, does the pivotal temperature should be displayed
PT	Value for pivotal temperature
show.test	True or False, does the hatchling size should be displayed
lab.third	Label for 2nd third of incubation

at.lab.third	Position of Label for 2nd third of incubation [default=4]
lab.PT	Label for Pivotal Temperature
lab.stages	Label for Stages
mar	Parameter mar used for plot
xlab	Label for axis
ylabT	Label for temperature axis
ylabS	Label for size axis
show.plot	If FALSE, only the text output is shown

### Details

plot.NestsResult Plot the information about a nest

### Value

A list with informations about statistics for the series of nests analyzed ( $\$summary$ ) and exact times series of length changes for each nest ( $\$traces$ )

### Author(s)

Marc Girondot

### Examples

```
## Not run:
library(embryogrowth)
data(resultNest_4p)
plot(resultNest_4p, xlim=c(0,70), ylimT=c(22, 32), ylimS=c(0,45), series=1,
SE=c(DHA=1.396525, DHH=4.101217, T12H=0.04330405, Rho25=1.00479))
# to plot all the nest at the same time, use
plot(resultNest_4p, xlim=c(0,70), ylimT=c(22, 32), ylimS=c(0,45),
series="all", show.fioritures=FALSE)
# to use color different for series
plot(resultNest_4p, xlim=c(0,70), ylimT=c(22, 32), ylimS=c(0,45),
series="all", show.fioritures=FALSE, col.S=c(rep("black", 5), rep("red", 6)))

## End(Not run)
```

---

plot.tsd

*Plot results of tsd() that best describe temperature-dependent sex determination*

---

### Description

Plot the estimates that best describe temperature-dependent sex determination

**Usage**

```
## S3 method for class 'tsd'
plot(x, ..., se = NULL, l = 0.05, males.freq = TRUE,
     las.x = 1, las.y = 1, lab.PT = "Pivotal temperature",
     lab.TRT = paste0("Transitional range of temperatures l=", l * 100, "%"),
     col.TRT = "gray", col.TRT.CI = rgb(0.8, 0.8, 0.8, 0.5),
     col.PT.CI = rgb(0.8, 0.8, 0.8, 0.5), show.CI = TRUE, replicate = 1000,
     range.CI = 0.95, limit.low.TRT.minimum = 5, limit.high.TRT.maximum = 90,
     print = TRUE)
```

**Arguments**

x	A result file generated by tsd()
...	Parameters for plot()
se	Standard error for parameters
l	The limit to define TRT (see Girondot, 1999)
males.freq	Should the graph uses males frequency [TRUE] or females [FALSE]
las.x	las parameter for x axis
las.y	las parameter for y axis
lab.PT	Label to describe pivotal temperature
lab.TRT	Label to describe transitional range of temperature
col.TRT	The color of TRT
col.TRT.CI	The color of CI of TRT based on range.CI
col.PT.CI	The color of CI of PT based on range.CI
show.CI	Do the CI for the curve should be shown
replicate	Number of replicate to estimate SE of TRT
range.CI	The range of confidence interval for estimation, default=0.95
limit.low.TRT.minimum	Minimum lower limit for TRT
limit.high.TRT.maximum	Maximum higher limit for TRT
print	Do the results must be printed at screen? TRUE or FALSE

**Details**

plot.tsd plot result of tsd() that best describe temperature-dependent sex determination

**Value**

A list the pivotal temperature, transitional range of temperatures and their SE

**Author(s)**

Marc Girondot

## References

Girondot, M. 1999. Statistical description of temperature-dependent sex determination using maximum likelihood. *Evolutionary Ecology Research*, 1, 479-486.

Godfrey, M.H., Delmas, V., Girondot, M., 2003. Assessment of patterns of temperature-dependent sex determination using maximum likelihood model selection. *Ecoscience* 10, 265-272.

Hulin, V., Delmas, V., Girondot, M., Godfrey, M.H., Guillon, J.-M., 2009. Temperature-dependent sex determination and global change: are some species at greater risk? *Oecologia* 160, 493-506.

Girondot M., Submitted. On the concept of embryological thermosensitive period for sex determination in reptiles.

## Examples

```
## Not run:
CC_AtlanticSW <- subset(STSRE_TSD, RMU=="Atlantic, SW" &
                        Species=="Caretta caretta" & Sexed!=0)
par(mar=c(4,4,5,1)+0.1)
tsdL <- with (CC_AtlanticSW, tsd(males=Males, females=Females,
                                temperatures=Incubation.temperature-Correction.factor,
                                equation="logistic"))
plot(tsdL)

## End(Not run)
```

---

plotR

*Show the fitted growth rate dependent on temperature*

---

## Description

To show the growth rate, the syntaxe is  
`plotR(result=res)`

## Usage

```
plotR(result = NULL, parameters = NULL, fixed.parameters = NULL,
       col = "black", legend = NA, SE = NULL, set.par = 1, size = NA,
       xlim = c(20, 35), scaleY = "auto", lty = 1, ltyCI = 3, lwd = 1,
       lwdCI = 1, xlimR = xlim, replicate.CI = 100, show.box = TRUE,
       local.box = "topleft", ...)
```

## Arguments

<code>result</code>	A result object or a list of result objects
<code>parameters</code>	Indicate some parameters if the result object is not supplied
<code>fixed.parameters</code>	Indicate some parameters if the result object is not supplied
<code>col</code>	The color to use for a list of colors if several results

legend	Text to show in bottom right legend or a list of text if several results
SE	The standard error for the parameters or a list of SE if several results. Use NA to force not use SE
set.par	1 or 2 or a list of 1 or 2 to designate with set of parameters to show
size	If indicated, will show the growth rate for this size
xlim	Range of values for x-axis
scaleY	Scaling factor for y axis or "auto"
lty	The type of line to use if several results as a list
ltyCI	The type of line to use for confidence interval as a list
lwd	The type of line to use if several results as a list
lwdCI	The type of line to use for confidence interval as a list
xlimR	Range of values to be displayed for R curve; can be a list if a list of results is used
replicate.CI	Number of randomizations to estimate CI
show.box	If TRUE show a box with "mean" and "confidence interval"
local.box	Position of the box with "mean" and "confidence interval", default="topleft"
...	Parameters for plot() such as main= or ylim=

### Details

plotR shows the fitted growth rate dependent on temperature

### Value

Nothing

### Author(s)

Marc Girondot

### Examples

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
resultNest_4p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
```

```

test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_4p)
pMCMC <- embryogrowth_MHmcmc_p(resultNest_4p, accept=TRUE)
# Take care, it can be very long, sometimes several days
result_mcmc_4p_80 <- embryogrowth_MHmcmc(result=resultNest_4p,
parametersMCMC=pMCMC, n.iter=10000, n.chains = 1, n.adapt = 0,
thin=1, trace=TRUE)
data(result_mcmc_4p)
plotR(result=resultNest_4p, SE=result_mcmc_4p$TimeSeriesSE,
ylim=c(0,0.3))
x <- structure(c(115.758929130522, 428.649022170996, 503.687251738993,
12.2621455821612, 306.308841227278, 116.35048615105), .Names = c("DHA",
"DHH", "DHL", "DT", "T12L", "Rho25"))
pfixed <- c(rK=2.093313)
resultNest_6p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_6p)
plotR(list(resultNest_4p, resultNest_6p),
col=c("black", "red"), legend=c("4 parameters", "6 parameters"))
#####
# new formulation of parameters
data(resultNest_newp)
plotR(resultNest_newp)

## End(Not run)

```

---

plotR\_hist

*Shows the histogram of temperatures with set of nests and the R function superimpose*


---

## Description

Shows the histogram of temperatures with set of nests and the R function superimpose `plotR_hist(data)`

## Usage

```
plotR_hist(x, ..., ylimH = NULL, ylabH = "Frequency of temperatures")
```

## Arguments

x	Result data
...	Parameters used by hist or plotR functions
ylimH	Scale of histogram using <code>ylimH=c(min, max)</code>
ylabH	Label for histogram scale

## Details

`plotR_hist` shows the histogram of temperatures with set of nests and the R function superimpose



**Value**

Nothing

**Author(s)**

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

**Examples**

```
## Not run:  
library(embryogrowth)  
data(resultNest_4p)  
plotR_hist(resultNest_4p)  
  
## End(Not run)
```

---

plot_transition	<i>Show fonction used for transition</i>
-----------------	--

---

**Description**

Plot the transition function

**Usage**

```
plot_transition(result = NULL, parameters = NULL, sizes = c(0, 40), ...)
```

**Arguments**

result	A result object
parameters	Set of parameters. If both result and parameters are indicated, parameters have priority.
sizes	The range of possible sizes
...	Parameters for plot() such as main= or ylim=

**Details**

plot\_transition show fonction used for transition

**Value**

Nothing

**Author(s)**

Marc Girondot

**Examples**

```

## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
data(resultNest_4p)
# Get a set of parameters without transition
x1 <- resultNest_4p$par
# Generate a set of parameters with transition
x2 <- switch.transition(x1)
x2 <- x2[names(x2)!="transition_P"]
x2["transition_S"] <- 4
pfixed <- c(rK=2.093313, transition_P=20)
resultNest_4p_transition <- searchR(parameters=x2, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_4p_transition)
# show the model for smallest size
plotR(resultNest_4p_transition, ylim=c(0,0.3))
# show the model for larger sizes
plotR(resultNest_4p_transition, set.par=2, ylim=c(0,0.3))
# plot model for both together
plotR(resultNest_4p_transition, set.par=c(1,2), ylim=c(0,0.3),
      col=c("red", "black"), legend=list("Initial", "End"))
plot_transition(result=resultNest_4p_transition, las=1, sizes=c(0,40))
compare_AIC(one.model=list(resultNest_4p), two.models=list(resultNest_4p_transition))
# Note that the model with fitted transition_P is trivial. Embryos grow fast until
# they reach hatchling size and then growth rate becomes null!

## End(Not run)

```

---

predict.tsd

*Estimate sex ratio according to constant incubation temperature*


---

**Description**

Estimate sex ratio according to constant incubation temperature The list has the following components: \$sexratio \$CI.minus.sexratio \$CI.plus.sexratio \$range.CI if range.CI is set to NULL, the confidence interval is not estimated

**Usage**

```

## S3 method for class 'tsd'
predict(object, temperatures = NULL, durations = NULL,
      range.CI = 0.95, ...)

```

**Arguments**

object	A result file generated by tsd
temperatures	A vector of temperatures
durations	A vector of durations
range.CI	The range of confidence interval for estimation, default=0.95
...	Not used

**Details**

predict.tsd Estimate sex ratio according to constant incubation temperature

**Value**

A list with informations about sex-ratio

**Author(s)**

Marc Girondot

**Examples**

```
## Not run:  
library(embryogrowth)  
m <- c(10, 14, 7, 4, 3, 0, 0)  
f <- c(0, 1, 2, 4, 15, 10, 13)  
t <- c(25, 26, 27, 28, 29, 30, 31)  
result <- tsd(males=m, females=f, temperatures=t)  
predict(result, temperatures=c(25, 31))  
  
## End(Not run)
```

---

resultNest\_4p

*Fit using the nest database*

---

**Description**

Fit using the nest database

**Usage**

```
resultNest_4p
```

**Format**

A list with fitted information about data(nest)

**Details**

Result of the fit using the nest database

**Author(s)**

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

**References**

Girondot, M. & Kaska, Y. In press. A model to predict the thermal reaction norm for the embryo growth rate from field data. *Journal of Thermal Biology*. In press

**Examples**

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
resultNest_4p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)

## End(Not run)
```

---

resultNest\_4p\_transition

*Result of the fit using the nest database using transition*

---

**Description**

Fit using the nest database using transition

**Usage**

```
resultNest_4p_transition
```

**Format**

A list with fitted information about data(nest)

**Details**

Result of the fit using the nest database using transition

**Author(s)**

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

**References**

Girondot, M. & Kaska, Y. In press. A model to predict the thermal reaction norm for the embryo growth rate from field data. *Journal of Thermal Biology*. In press

**Examples**

```
## Not run:  
library(embryogrowth)  
data(nest)  
formatted <- FormatNests(nest)  
data(resultNest_4p_transition)  
  
## End(Not run)
```

---

resultNest\_4p\_weight *Fit using the nest database with weight*

---

**Description**

Fit using the nest database with weight

**Usage**

```
resultNest_4p_weight
```

**Format**

A list with fitted information about data(nest)

**Details**

Result of the fit using the nest database with weight

**Author(s)**

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

**References**

Girondot, M. & Kaska, Y. In press. A model to predict the thermal reaction norm for the embryo growth rate from field data. *Journal of Thermal Biology*. In press

## Examples

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
w <- weightmaxentropy(formatted, control_plot=list(xlim=c(20,36)))
x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
# K or rK are not used for dydt.linear or dydt.exponential
resultNest_4p_weight <- searchR(parameters=x,
fixed.parameters=pfixed, temperatures=formatted,
derivate=dydt.Gompertz, M0=1.7, test=c(Mean=39.33, SD=1.92),
method = "BFGS", maxiter = 200, weight=w)
data(resultNest_4p_weight)
plotR(resultNest_4p_weight, ylim=c(0,0.50), xlim=c(15, 35))

## End(Not run)
```

---

resultNest\_6p

*Fit using the nest database*

---

## Description

Fit using the nest database

## Usage

```
resultNest_6p
```

## Format

A list with fitted information about data(nest)

## Details

Result of the fit using the nest database

## Author(s)

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

## References

Girondot, M. & Kaska, Y. In press. A model to predict the thermal reaction norm for the embryo growth rate from field data. *Journal of Thermal Biology*. In press

**Examples**

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
x <- structure(c(115.758929130522, 428.649022170996, 503.687251738993,
12.2621455821612, 306.308841227278, 116.35048615105), .Names = c("DHA",
"DHH", "DHL", "DT", "T12L", "Rho25"))
pfixed <- c(rK=2.093313)
resultNest_6p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)

## End(Not run)
```

---

resultNest\_newp

*Fit using the nest database with anchored parameters*


---

**Description**

Fit using the nest database with anchored parameters

**Usage**

```
resultNest_newp
```

**Format**

A list with fitted information about data(nest) with anchored parameters

**Details**

Result of the fit using the nest database with anchored parameters

**Author(s)**

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

**References**

Girondot, M. & Kaska, Y. In press. A model to predict the thermal reaction norm for the embryo growth rate from field data. *Journal of Thermal Biology*. In press

## Examples

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
newp <- GenerateAnchor(nests=formatted, number.anchors=7)
pfixed <- c(rK=2.093313)
resultNest_newp <- searchR(parameters=newp, fixed.parameters=pfixed,
  temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
  test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_newp)

## End(Not run)
```

---

result\_mcmc\_4p

*Result of the mcmc using the nest database*

---

## Description

Fit using the nest database

## Usage

```
result_mcmc_4p
```

## Format

A list of class mcmcComposite with mcmc result for data(nest) with 4 parameters and Gompertz model of growth

## Details

Result of the mcmc using the nest database

## Author(s)

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

## References

Girondot, M. & Kaska, Y. In press. A model to predict the thermal reaction norm for the embryo growth rate from field data. *Journal of Thermal Biology*. In press



**Examples**

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
resultNest_4p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_4p)
pMCMC <- embryogrowth_MHmcmc_p(resultNest_4p, accept=TRUE)
# Take care, it can be very long, sometimes several days
result_mcmc_4p <- embryogrowth_MHmcmc(result=resultNest_4p,
parametersMCMC=pMCMC, n.iter=10000, n.chains = 1, n.adapt = 0,
thin=1, trace=TRUE)
data(result_mcmc_4p)
plot(result_mcmc_4p, parameters="T12H", main="", xlim=c(290, 320), bty="n")
plotR(resultNest_4p, SE=result_mcmc_4p$TimeSeriesSE, ylim=c(0,0.3), las=1)

## End(Not run)
```

---

result\_mcmc\_4p\_weight *Result of the mcmc using the nest database*

---

**Description**

Fit using the nest database

**Usage**

```
result_mcmc_4p_weight
```

**Format**

A list of class mcmcComposite with mcmc result for data(nest) with 4 parameters and Gompertz model of growth weighted to maximized entropy

**Details**

Result of the mcmc using the nest database with weight

**Author(s)**

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

**References**

Girondot, M. & Kaska, Y. Submitted. A model to predict temperature dependency on embryo growth rate and incubation duration from field data.

**Examples**

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
w <- weightmaxentropy(formatted, control_plot=list(xlim=c(20,36)))
x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
# K or rK are not used for dydt.linear or dydt.exponential
resultNest_4p_weight <- searchR(parameters=x,
fixed.parameters=pfixed, temperatures=formatted,
derivate=dydt.Gompertz, M0=1.7, test=c(Mean=39.33, SD=1.92),
method = "BFGS", maxiter = 200, weight=w)
data(resultNest_4p_weight)
pMCMC <- embryogrowth_MHmcmc_p(resultNest_4p_weight, accept=TRUE)
# Take care, it can be very long, sometimes several days
result_mcmc_4p_weight <- embryogrowth_MHmcmc(result=resultNest_4p_weight,
parametersMCMC=pMCMC, n.iter=10000, n.chains = 1, n.adapt = 0,
thin=1, trace=TRUE)
data(result_mcmc_4p_weight)
plot(result_mcmc_4p_weight, parameter="T12H", main="", xlim=c(290, 320), bty="n")
plotR(resultNest_4p_weight, SE=result_mcmc_4p_weight$TimeSeriesSE,
ylim=c(0,0.3), las=1)
data(resultNest_4p)
data(result_mcmc_4p)
plotR(list(resultNest_4p_weight, resultNest_4p),
SE=list(result_mcmc_4p_weight$TimeSeriesSE, result_mcmc_4p$TimeSeriesSE),
ylim=c(0,0.3), las=1, col=list("red", "black"),
legend=list("Maximum entropy weighted", "Not weighted"))

## End(Not run)
```

---

result\_mcmc\_6p

*Result of the mcmc using the nest database*

---

**Description**

Fit using the nest database

**Usage**

```
result_mcmc_6p
```

**Format**

A list of class `mcmcComposite` with mcmc result for `data(nest)` with 6 parameters and Gompertz model of growth

**Details**

Result of the mcmc using the nest database

**Author(s)**

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

**References**

Girondot, M. & Kaska, Y. In press. A model to predict the thermal reaction norm for the embryo growth rate from field data. *Journal of Thermal Biology*. In press

**Examples**

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
x <- structure(c(115.758929130522, 428.649022170996, 503.687251738993,
12.2621455821612, 306.308841227278, 116.35048615105), .Names = c("DHA",
"DHH", "DHL", "DT", "T12L", "Rho25"))
pfixed <- c(rK=2.093313)
resultNest_6p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_6p)
pMCMC <- embryogrowth_MHmcmc_p(resultNest_6p, accept=TRUE)
# Take care, it can be very long, sometimes several days
result_mcmc_6p <- embryogrowth_MHmcmc(result=resultNest_6p,
parametersMCMC=pMCMC, n.iter=10000, n.chains = 1, n.adapt = 0,
thin=1, trace=TRUE)
data(result_mcmc_6p)
plot(result_mcmc_6p, parameters="T12L", main="", xlim=c(290, 320), bty="n")

## End(Not run)
```

---

result_mcmc_newp	<i>Result of the mcmc using the nest database with anchored parameters</i>
------------------	--

---

**Description**

Fit using the nest database with anchored parameters

**Usage**

```
result_mcmc_newp
```

**Format**

A list of class mcmcComposite with mcmc result for data(nest) with anchored parameters

**Details**

Result of the mcmc using the nest database with anchored parameters

**Author(s)**

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

**References**

Girondot, M. & Kaska, Y. In press. A model to predict the thermal reaction norm for the embryo growth rate from field data. *Journal of Thermal Biology*. In press

**Examples**

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
newp <- GenerateAnchor(nests=formatted, number.anchors=7)
pfixed <- c(rK=2.093313)
resultNest_newp <- searchR(parameters=newp, fixed.parameters=pfixed,
  temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
  test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_newp)
pMCMC <- embryogrowth_MHmcmc_p(resultNest_newp, accept=TRUE)
# Take care, it can be very long, sometimes several days
result_mcmc_newp <- embryogrowth_MHmcmc(result=resultNest_newp,
  parametersMCMC=pMCMC, n.iter=1000, n.chains = 1, n.adapt = 0,
  thin=1, trace=TRUE)
data(result_mcmc_newp)
data(resultNest_4p)
newp <- GenerateAnchor(nests=resultNest_4p, number.anchors=7)
# Here the confidence interval is built based on anchored parameters
```

```

plotR_hist(resultNest_4p, parameters=newp, SE=result_mcmc_newp$TimeSeriesSE,
  ylim=c(0,0.4), ylimH=c(0,0.4))
# Here the confidence interval is built based on parametric SSM equation
plotR_hist(resultNest_4p, SE=result_mcmc_4p$TimeSeriesSE,
  ylim=c(0,0.4), ylimH=c(0,0.4))
plot(result_mcmc_newp, las=1, xlim=c(0,30), parameters="294")
plot(result_mcmc_newp, las=1, xlim=c(0,30), parameters="296.333333333333")
plot(result_mcmc_newp, las=1, xlim=c(0,30), parameters=3)

## End(Not run)

```

---

searchR

*Fit the parameters that best represent nest incubation data.*


---

## Description

Fit the parameters that best represent data.

## Usage

```

searchR(parameters = stop("Initial set of parameters must be provided"),
  fixed.parameters = NULL,
  temperatures = stop("Temperature data must be provided !"),
  derivate = dydt.Gompertz, test = c(Mean = 39.33, SD = 1.92), M0 = 1.7,
  method = "BFGS", maxiter = 200, saveAtMaxiter = FALSE,
  fileName = "intermediate", weight = NULL, hessian = TRUE,
  parallel = (.Platform$OS.type == "unix"))

```

## Arguments

parameters	A set of parameters used as initial point for searching
fixed.parameters	A set of parameters that will not be changed
temperatures	Timeseries of temperatures after formatted using FormatNests()
derivate	Function used to fit embryo growth: dydt.Gompertz, dydt.exponential or dydt.linear
test	Mean and SD of size of hatchlings
M0	Measure of hatchling size or mass proxi at laying date
method	Method uses for searching. Can be any method from optim function
maxiter	After maxiter iteration, the value of parameters is displayed but it continues if convergence is not acheived
saveAtMaxiter	If True, each time number of interation reach maxiter, current data are saved in file with filename name
fileName	The intermediate results are saved in file with fileName.Rdata name
weight	A named vector of the weight for each nest for likelihood estimation
hessian	If TRUE, the hessian matrix is estimated and the SE of parameters estimated.
parallel	If true, try to use several cores using parallel computing. Must be FALSE in Windows.

**Details**

searchR fits the parameters that best represent nest incubation data.

**Value**

A result object

**Author(s)**

Marc Girondot

**Examples**

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
# The initial parameters value can be:
# "T12H", "DHA", "DHH", "Rho25"
# Or
# "T12L", "DT", "DHA", "DHH", "DHL", "Rho25"
# K for Gompertz must be set as fixed parameter or being a constant K
# or relative to the hatchling size rK
x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
resultNest_4p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_4p)
plot(resultNest_4p, xlim=c(0,70), ylimT=c(22, 32), ylimS=c(0,45), series=1)
x <- structure(c(115.758929130522, 428.649022170996, 503.687251738993,
12.2621455821612, 306.308841227278, 116.35048615105), .Names = c("DHA",
"DHH", "DHL", "DT", "T12L", "Rho25"))
pfixed <- c(rK=2.093313)
resultNest_6p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_6p)
pMCMC <- embryogrowth_MHmcmc_p(resultNest_6p, accept=TRUE)
# Take care, it can be very long, sometimes several days
result_mcmc_6p <- embryogrowth_MHmcmc(result=resultNest_6p,
parametersMCMC=pMCMC, n.iter=10000, n.chains = 1, n.adapt = 0,
thin=1, trace=TRUE)
data(result_mcmc_6p)
# compare_AIC() is a function from the package "phenology"
outputAIC<-compare_AIC(list(test1=resultNest_4p, test2=resultNest_6p))
##### with new parametrization
data(resultNest_4p)
x0 <- resultNest_4p$par
t <- hist(resultNest_4p, plot=FALSE)
```

```
temperatures <- seq(from=floor(range(t$temperatures)[1]+273.15-1),
to=floor(range(t$temperatures)[2]+273.15+1), length.out=7)
newx <- embryogrowth:::SSM(temperatures, x0)[[1]]*1E5
names(newx) <- temperatures
pfixed <- c(rK=2.093313)
resultNest_newp <- searchR(parameters=newx, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.Gompertz, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
plotR_hist(resultNest_newp, ylim=c(0,0.3), xlimR=c(23, 34), ylimH=c(0, 0.3))
outputAIC<-compare_AIC(list(test4p=resultNest_4p, test6p=resultNest_6p,
testAnchor=resultNest_newp))

## End(Not run)
```

---

STSRE\_NestingArea      *Database of RMU for marine turtles*

---

**Description**

Database of RMU for marine turtles

**Usage**

STSRE\_NestingArea

**Format**

A dataframe with raw data.

**Details**

Database of RMU for marine turtles

**Author(s)**

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

Maria Sousa Martins <maria.esmartins@gmail.com>

**References**

Wallace, B. P., DiMatteo, A. D., Hurley, B. J., Finkbeiner, E. M., Bolten, A. B., Chaloupka, M. Y., Hutchinson, B. J., Abreu-Grobois, F. A., Amorcho, D., Bjordal, K. A., Bourjea, J., Bowen, B. W., Duenas, R. B., Casale, P., Choudhury, B. C., Costa, A., Dutton, P. H., Fallabrino, A., Girard, A., Girondot, M., Godfrey, M. H., Hamann, M., Lopez-Mendilaharsu, M., Marcovaldi, M. A., Mortimer, J. A., Musick, J. A., Nel, R., Seminoff, J. A., Troeng, S., Witherington, B. & Mast, R. B. 2010. Regional management units for marine turtles: a novel framework for prioritizing conservation and research across multiple scales. *Plos One*, 5, e15465.

**Examples**

```
## Not run:
library(embryogrowth)
data(STSRE_NestingArea)

## End(Not run)
```

STSRE\_TSD

*Database of TSD information for marine turtles***Description**

Database of TSD information for turtles

The columns are:

- \* Species: Name of the species in binominal nomenclature
- \* Country: From which country the eggs come from
- \* Area: Name of the beach or region the eggs come from
- \* RMU: Name of the RMU for this population; see Wallace, B.P., DiMatteo, A.D., Hurley, B.J., Finkbeiner, E.M., Bolten, A.B., Chaloupka, M.Y., Hutchinson, B.J., Abreu-Grobois, F.A., Amoroch, D., Bjorndal, K.A., Bourjea, J., Bowen, B.W., Dueñas, R.B., Casale, P., Choudhury, B.C., Costa, A., Dutton, P.H., Fallabrino, A., Girard, A., Girondot, M., Godfrey, M.H., Hamann, M., López-Mendilaharsu, M., Marcovaldi, M.A., Mortimer, J.A., Musick, J.A., Nel, R., Seminoff, J.A., Troëng, S., Witherington, B., Mast, R.B., 2010. Regional management units for marine turtles: a novel framework for prioritizing conservation and research across multiple scales. Plos One 5, e15465.
- \* Incubation.temperature: Nominal incubation temperature
- \* Fluctuation: How much the temperature could fluctuate around nominal temperature
- \* Precision: What is the precision of the measure of temperature
- \* Correction.factor: Difference between the incubator temperature and the eggs temperature
- \* IP.min: Shorter incubation period
- \* IP.max: Longer incubation period
- \* IP.amplitude: IP.max-IP.min if available
- \* IP.mean: Mean incubation periods
- \* IP.mode: Mode for incubation periods
- \* IP.SD: Standard deviation for incubation periods
- \* Total: Total number of eggs incubated
- \* Hatched: Number of hatchlings
- \* Intersexes: Number of individuals intersexes or ambiguous for sex phenotype
- \* Males: Number of individuals identified as males
- \* Females: Number of individuals identified as females
- \* Sexed: Number of sexed individuals
- \* Clutch: Identity of clutch
- \* Reference: Bibliographic reference
- \* Note: Diverse information for this incubation
- \* Version: Date of the version of this database



**Usage**

STSRE\_TSD

**Format**

A dataframe with raw data.

**Details**

Database of TSD information for marine turtles

**Author(s)**

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

Maria Sousa Martins <maria.esmartins@gmail.com>

**References**

- Binckley C.A., Spotila J.R., Wilson K.S. and Paladino F.V. (1998) Sex determination and sex ratios of Pacific Leatherback Turtles, *Dermochelys coriacea*. *Copeia*, 1998(2), 291-300.
- Crastz F. (1982) Embryological stages of the marine turtle *Lepidochelys olivacea* (Eschscholtz). *Rev. Biol. Trop.*, 30, 113-120.
- De Souza, R.R., Vogt, R.C., 1994. Incubation temperature influences sex and hatchling size in the neotropical turtle *Podocnemis unifilis*. *Journal of Herpetology* 28, 453-464.
- Girondot M. 1999 Statistical description of temperature-dependent sex determination using maximum likelihood. *Evolutionary Ecology Research*, 1(3), 479-486.
- Hulin, V., Delmas, V., Girondot, M., Godfrey, M. H. and Guillon, J.-M. 2009. Temperature-dependent sex determination and global change: are some species at greater risk? *Oecologia*, 160, 493-506.
- Limpus C.J., Reed P.C. and Miller J.D. (1985) Temperature dependent sex determination in Queensland sea turtles: Intraspecific variation in *Caretta caretta*. In Grigg G., Shine R. and Ehmann H. (eds) *Biology of Australian frogs and reptiles*. New South Wales, Australia: Royal Zoological Society, pp 343-351.
- López Correa, J.Y., 2010. Diferenciación gonádica en crias de *Lepidochelys olivacea* (Eschscholtz, 1829) (Testudinata: Cheloniidae), Instituto Politécnico Nacional, Centro Interdisciplinaria de Ciencias Marinas, La Paz, BCS, p. 108.
- Marcovaldi M.A., Godfrey M.H. and Mrosovsky N. (1997) Estimating sex ratios of loggerhead turtles in Brazil from pivotal incubation temperatures. *Canadian Journal of Zoology-Revue Canadienne De Zoologie*, 75, 755-770.
- McCoy C.J., Vogt R.C. and Censky E.J. (1983) Temperature-controlled sex determination in the sea turtle *Lepidochelys olivacea*. *J. Herpetol.*, 17(4), 404-406.
- Merchant-Larios, H, Díaz-Hernández, V, Marmolejo-Valencia A. 2010. Gonadal morphogenesis and gene expression in reptiles with temperature-dependent sex determination. *Sexual Development* 4:50-61.

- Merchant-Larios, H., Villalpando-Fierro, I. & Centeno-Urruiza, B. 1989. Gonadal morphogenesis under controlled temperature in the sea turtle *Lepidochelys olivacea*. *Herpetol. Monographs*, 3, 43-61.
- Merchant-Larios, H., Ruiz-Ramirez, S., Moreno-Mendoza, N. & Marmolejo-Valencia, A. 1997. Correlation among thermosensitive period, estradiol response, and gonad differentiation in the sea turtle *Lepidochelys olivacea*. *General and Comparative Endocrinology*, 107, 373-385.
- Michel-Morfinu J.E., Gomez Munoz V.M. and Navarro Rodriguez C. (2001) Morphometric model for sex assessment in hatchling olive ridley sea turtles. *Chelonian Conservation and Biology*, 40(1), 53-58.
- Mrosovsky N., Kamel S., Rees A.F. and Margaritoulis D. (2002) Pivotal temperature for loggerhead turtles (*Caretta caretta*) from Kyparissia Bay, Greece. *Canadian Journal of Zoology-Revue Canadienne De Zoologie*, 80(12), 2118-2124.
- Mrosovsky N., Dutton P.H. and Whitmore C.P. (1984) Sex ratios of two species of sea turtle nesting in Suriname. *Can. J. Zool.*, 62, 2227-2239.
- Mrosovsky N., Bass A., Corliss L.A., Richardson J.I. and Richardson T.H. (1992) Pivotal and beach temperature for hawksbill turtles nesting in Antigua. *Can. J. Zool.*, 70, 1920-1925.
- Rimblot F., Fretey J., Mrosovsky N., Lescure J. and Pieau C. (1985) Sexual differentiation as a function of the incubation temperature of eggs in the sea-turtle *Dermochelys coriacea* (Vandelli, 1761). *Amphibia-Reptilia*, 85(6), 83-92.
- Rimblot-Baly F., Lescure J., Fretey J. and Pieau C. (1986-1987) Sensibilite a la temperature de la differenciation sexuelle chez la tortue Luth, *Dermochelys coriacea* (Vandelli, 1761); application des donnees de l'incubation artificielle a l'etude de la sex-ratio dans la nature. *Annales des Sciences Naturelles, Zoologie*, 8, 277-290.
- Ruiz García N.A. (2014) Efectos de la temperatura sobre el desarrollo embrionario y el desempeño de crías de la tortuga golfina, *Lepidochelys olivacea*. Tesis Nivel Maestria.
- Valenzuela, N., 2001. Constant, shift, and natural temperature effects on sex determination in *Podocnemis expansa*. *Ecology* 82, 3010-3024.
- Tokunaga, S., Iwakiri, Y., Nakajima, Y., (1999). Temperature-dependent sex determination of a sea turtle, *Caretta caretta*, from Miyazaki, Japan. *Bull. Kitakyushu Mus. Nat. Hist.* 18, 147-156.
- Wibbels T., Rostal D.C. and Byles R. (1998) High pivotal temperature in the sex determination of the olive ridley sea turtle, *Lepidochelys olivacea*, from Playa Nancite, Costa Rica. *Copeia*, 1998(4), 1086-1088.
- Yntema C.L. and Mrosovsky N. (1980) Sexual differentiation in hatchling loggerheads (*Caretta caretta*) incubated at different controlled temperatures. *Herpetologica*, 36(1), 33-36.
- Yntema C.L. and Mrosovsky N. (1982) Critical periods and pivotal temperatures for sexual differentiation in loggerhead sea turtles. *Canadian Journal of Zoology-Revue Canadienne De Zoologie*, 60(5), 1012-1016.

## Examples

```
## Not run:
library(embryogrowth)
data(STSRE_TSD)
STSRE_TSD$Version[1]
totalIncubation_Lo <- subset(STSRE_TSD, Species=="Lepidochelys olivacea" & Sexed!=0)
```

```
tot_Lo <- with(totalIncubation_Lo, tsd(males=Males, females=Females,
  temperatures=Incubation.temperature, par=c(P=29, S=-0.01), xlim=c(25, 35)))
predict(tot_Lo)

## End(Not run)
```

---

summary.Nests	<i>Summarize the information from a Nests object.</i>
---------------	---

---

### Description

Summarize the information from a Nests object: - Name of the nests, total incubation length and average temperature

### Usage

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

### Arguments

object	A object obtained after FormatNests()
...	Not used

### Details

summary.Nests Summarize the information from a Nests object

### Value

None

### Author(s)

Marc Girondot

### Examples

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest, previous=NULL)
summary(formatted)

## End(Not run)
```

---

switch.transition      *Add a transition parameter on a set of parameters or remove it*

---

**Description**

Add a transition parameter on a set of parameters or remove it

**Usage**

```
switch.transition(parameters = stop("A set of parameters must be supplied"))
```

**Arguments**

parameters      A vector with parameters

**Details**

switch.transition Add a transition parameter on a set of parameters or remove it

**Value**

A vector with parameters

**Author(s)**

Marc Girondot

**Examples**

```
## Not run:
data(resultNest_6p)
# Get a set of parameters without transition
x1 <- resultNest_6p$par
# Generate a set of parameters with transition
x2 <- switch.transition(x1)
# Generate a set of parameters without transition
x3 <- switch.transition(x3)

## End(Not run)
```

tempConst

*Timeseries of constant temperatures for nests***Description**

Timeseries of temperatures for nests

**Usage**

```
tempConst
```

**Format**

A dataframe with raw data.

**Details**

Timeseries of constant temperatures for nests

**Author(s)**

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

**References**

Girondot, M. & Kaska, Y. Submitted. A model to predict temperature dependency on embryo growth rate and incubation duration from field data.

**Examples**

```
## Not run:
library(embryogrowth)
# Same as:
# GenerateConstInc(durations = rep(104*60*24, 11),
# temperatures = 25:35,
# names = paste0("T",25:35))
data(tempConst)
tempConst_f <- FormatNests(tempConst)
x <- structure(c(118.768297442004, 475.750095909406, 306.243694918151,
116.055824800264), .Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
resultNest_4p <- searchR(parameters=x, fixed.parameters=pfixed,
temperatures=formatted, derivate=dydt.linear, M0=1.7,
test=c(Mean=39.33, SD=1.92), method = "BFGS", maxiter = 200)
data(resultNest_4p)
# Use the fitted parameters from resultNest_4p with
# the constant incubation temperatures:
plot(resultNest_4p, temperatures=tempConst_f,
```

```

stopattest=TRUE, series="all", xlim=c(0,120),
ylimT=c(22, 32), show.stages=FALSE, show.PT=FALSE,
show.temperatures=FALSE, show.TSP=FALSE)

## End(Not run)

```

---

temperature.periodic *Estimate temperatures in periodic timeseries based on anchored minimum and maximum*

---

### Description

Estimate temperatures in periodic timeseries based on anchored minimum and maximum. The data.frame minmax can be generated manually. It should have two columns, one time and one temperature with all the successive minimum and maximum temperatures.

### Usage

```
temperature.periodic(minmax, time = NULL, replicates = 100)
```

### Arguments

minmax	A data.frame returned by minmax.periodic
time	The time at which produced the estimate
replicates	Number of replicates to estimate sd

### Details

temperature.periodic estimate temperatures in periodic timeseries based on anchored minimum and maximum

### Value

A data.frame with a column time and a column temperature

### Author(s)

Marc Girondot

### Examples

```

## Not run:
# Generate a timeserie of time
time.obs <- NULL
for (i in 0:9) time.obs <- c(time.obs, c(0, 6, 12, 18)+i*24)
# For these time, generate a timeseries of temperatures
temp.obs <- rep(NA, length(time.obs))
temp.obs[3+(0:9)*4] <- rnorm(10, 25, 3)
temp.obs[1+(0:9)*4] <- rnorm(10, 10, 3)

```

```

for (i in 1:(length(time.obs)-1))
  if (is.na(temp.obs[i]))
    temp.obs[i] <- mean(c(temp.obs[i-1], temp.obs[i+1]))
  if (is.na(temp.obs[length(time.obs)]))
    temp.obs[length(time.obs)] <- temp.obs[length(time.obs)-1]/2

# Search for the minimum and maximum values
r <- minmax.periodic(time.minmax.daily=c(Min=2, Max=15),
time.obs=time.obs, temp.obs=temp.obs, period=24)

# Estimate all the temperatures for these values
t <- temperature.periodic(minmax=r)

plot_errbar(x=t[, "time"], y=t[, "temperature"],
errbar.y=ifelse(is.na(t[, "sd"]), 0, 2*t[, "sd"]),
type="l", las=1, bty="n", errbar.y.polygon = TRUE,
xlab="hours", ylab="Temperatures", ylim=c(0, 35),
errbar.y.polygon.list = list(col="grey"))

plot_add(x=t[, "time"], y=t[, "temperature"], type="l")

## End(Not run)

```

---

TestParallel	<i>Estimate the likelihood of a set of parameters for nest incubation data with or without parallel computing option</i>
--------------	--

---

### Description

Estimate the likelihood of a set of parameters for nest incubation data with or without parallel computing option

### Usage

```
TestParallel(result = stop("A ResultNest object must be provided"))
```

### Arguments

result            A object obtained after searchR or likelihoodR

### Details

TestParallel estimates the likelihood of a set of parameters for nest incubation data with or without parallel computing option

### Value

Nothing

**Author(s)**

Marc Girondot

**Examples**

```
## Not run:
library(embryogrowth)
data(resultNest_4p)
TestParallel(resultNest_4p)

## End(Not run)
```

---

tsd	<i>Estimate the parameters that best describe temperature-dependent sex determination</i>
-----	---

---

**Description**

Estimate the parameters that best describe temperature-dependent sex determination

**Usage**

```
tsd(df = NULL, males = NULL, females = NULL, N = NULL,
    temperatures = NULL, durations = NULL, l = 0.05,
    parameters.initial = c(P = NA, S = -0.5, K = 0, K1 = 1, K2 = 0),
    males.freq = TRUE, las.x = 1, las.y = 1,
    lab.PT = "Pivotal temperature",
    lab.TRT = paste0("Transitional range of temperatures l=", l * 100, "%"),
    col.TRT = "gray", col.TRT.CI = rgb(0.8, 0.8, 0.8, 0.5),
    col.PT.CI = rgb(0.8, 0.8, 0.8, 0.5), show.CI = TRUE,
    equation = "logistic", replicate = 1000, range.CI = 0.95,
    limit.low.TRT.minimum = 5, limit.high.TRT.maximum = 90, print = TRUE,
    show.plot = TRUE, ...)
```

**Arguments**

df	A dataframe with at least two columns named males, females or N and temperatures, Incubation.temperature or durations column
males	A vector with male numbers
females	A vector with female numbers
N	A vector with total numbers
temperatures	The constant incubation temperatures used to fit sex ratio
durations	The duration of incubation or TSP used to fit sex ratio
l	The limit to define TRT (see Girondot, 1999)
parameters.initial	Initial values for P, S or K search as a vector, ex. c(P=29, S=-0.3)



males.freq	Should the graph uses males frequency [TRUE] or females [FALSE]
las.x	las parameter for x axis
las.y	las parameter for y axis
lab.PT	Label to describe pivotal temperature
lab.TRT	Label to describe transitional range of temperature
col.TRT	The color of TRT
col.TRT.CI	The color of CI of TRT based on range.CI
col.PT.CI	The color of CI of PT based on range.CI
show.CI	Do the CI for the curve should be shown
equation	Could be "logistic", "Hill", "Richards", "Hulin", "Double-Richards" or "GSD"
replicate	Number of replicate to estimate SE of TRT
range.CI	The range of confidence interval for estimation, default=0.95
limit.low.TRT.minimum	Minimum lower limit for TRT
limit.high.TRT.maximum	Maximum higher limit for TRT
print	Do the results must be printed at screen? TRUE (default) or FALSE
show.plot	Do the plot must be shown? TRUE (default) or FALSE
...	Graphical parameters for plot(), exemple xlab="", ylab="", main=""

### Details

tsd estimates the parameters that best describe temperature-dependent sex determination

### Value

A list the pivotal temperature, transitional range of temperatures and their SE

### Author(s)

Marc Girondot

### References

- Girondot, M. 1999. Statistical description of temperature-dependent sex determination using maximum likelihood. *Evolutionary Ecology Research*, 1, 479-486.
- Godfrey, M.H., Delmas, V., Girondot, M., 2003. Assessment of patterns of temperature-dependent sex determination using maximum likelihood model selection. *Ecoscience* 10, 265-272.
- Hulin, V., Delmas, V., Girondot, M., Godfrey, M.H., Guillon, J.-M., 2009. Temperature-dependent sex determination and global change: are some species at greater risk? *Oecologia* 160, 493-506.
- Girondot M., Submitted. On the concept of embryological thermosensitive period for sex determination in reptiles.

## Examples

```
## Not run:
CC_AtlanticSW <- subset(STSRE_TSD, RMU=="Atlantic, SW" &
                        Species=="Caretta caretta" & Sexed!=0)
par(mar=c(4,4,5,1)+0.1)
tsdL <- with (CC_AtlanticSW, tsd(males=Males, females=Females,
                                temperatures=Incubation.temperature-Correction.factor,
                                equation="logistic"))
tsdH <- with (CC_AtlanticSW, tsd(males=Males, females=Females,
                                temperatures=Incubation.temperature-Correction.factor,
                                equation="Hill"))
tsdR <- with (CC_AtlanticSW, tsd(males=Males, females=Females,
                                temperatures=Incubation.temperature-Correction.factor,
                                equation="Richards"))
gsd <- with (CC_AtlanticSW, tsd(males=Males, females=Females,
                                temperatures=Incubation.temperature-Correction.factor,
                                equation="GSD"))
compare_AIC(Logistic_Model=tsdL, Hill_model=tsdH, Richards_model=tsdR,
            GSD_model=gsd)
#####
eo <- subset(STSRE_TSD, Species=="Emys orbicularis", c("Males", "Females",
            "Incubation.temperature"))

eo_Hill <- with(eo, tsd(males=Males, females=Females,
                        temperatures=Incubation.temperature,
                        equation="Hill"))

eo_logistic <- tsd(eo)
eo_Richards <- with(eo, tsd(males=Males, females=Females,
                            temperatures=Incubation.temperature,
                            equation="Richards"))

### The Hulin model is a modification of Richards (See Hulin et al. 2009)
### limit.low.TRT and limit.high.TRT must be setup for Hulin equation
par <- eo_Richards$par
names(par)[which(names(par)=="K")] <- "K2"
par <- c(par, K1=0)
eo_Hulin <- with(eo, tsd(males=Males, females=Females,
                        parameters.initial=par,
                        temperatures=Incubation.temperature,
                        equation="Hulin",
                        limit.low.TRT.minimum=25,
                        limit.high.TRT.maximum=35))

### The Double-Richards model is a Richards model with K1 and K2 using the two values
### below and above P
par <- eo_Richards$par
names(par)[which(names(par)=="K")] <- "K2"
par <- c(par, K1=as.numeric(par["K2"])-0.1)
par["K1"] <- par["K1"]-0.1
eo_Double_Richards <- with(eo, tsd(males=Males, females=Females,
                                parameters.initial=par,
                                temperatures=Incubation.temperature,
                                equation="Double-Richards"))

### Example with only points and mean curve
```

```

eo_Double_Richards <- with(eo, tsd(males=Males, females=Females,
                                parameters.initial=par,
                                temperatures=Incubation.temperature,
                                equation="Double-Richards",
                                col.TRT="white", col.TRT.CI="white", col.PT.CI="white"))
compare_AIC(Logistic=eo_logistic, Hill=eo_Hill, Richards=eo_Richards,
            Hulin=eo_Hulin, Double_Richards=eo_Double_Richards)
### Note the asymmetry of the Double-Richards model
predict(eo_Double_Richards,
        temperatures=c(eo_Double_Richards$par["P"]-0.2, eo_Double_Richards$par["P"]+0.2))
predict(eo_Double_Richards)

## End(Not run)

```

tsd\_MHmcmc

*Run the Metropolis-Hastings algorithm for tsd***Description**

Run the Metropolis-Hastings algorithm for tsd.  
 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.  
 If initial point is maximum likelihood,  $n.adapt = 0$  is a good solution.  
 Note that `resultLnL` has all the likelihoods, not only those defined by  $n.adapt$  and  $thin$ .  
 To get the SE from `result_mcmc <- embryogrowth_MHmcmc(result=try)`, use:  
`result_mcmc$BatchSE` or `result_mcmc$TimeSeriesSE`  
 The batch standard error procedure is usually thought to be not as accurate as the time series methods.  
 Based on Jones, Haran, Caffo and Neath (2005), the batch size should be equal to  $\sqrt{n.iter}$ .  
 Jones, G.L., Haran, M., Caffo, B.S. and Neath, R. (2006) Fixed Width Output Analysis for Markov chain Monte Carlo, *Journal of the American Statistical Association*, 101:1537-1547.  
 coda package is necessary for this function.

**Usage**

```

tsd_MHmcmc(result = stop("An output from tsd() must be provided"),
           n.iter = 10000,
           parametersMCMC = stop("Priors must be given. Use tsd_MHmcmc_p()"),
           n.chains = 1, n.adapt = 0, thin = 1, trace = FALSE,
           batchSize = sqrt(n.iter))

```

**Arguments**

<code>result</code>	An object obtained after a SearchR fit
<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

n.chains	Number of replicates
n.adapt	Number of iterations before to store outputs
thin	Number of iterations between each stored output
trace	True or False, shows progress
batchSize	Number of observations to include in each batch fo SE estimation

### Details

tsd\_MHmcmc runs the Metropolis-Hastings algorithm for tsd (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(embryogrowth)
eo <- subset(STSRE_TSD, Species=="Emys orbicularis", c("Males", "Females",
                                                    "Incubation.temperature"))

eo_logistic <- tsd(eo)
pMCMC <- tsd_MHmcmc_p(eo_logistic, accept=TRUE)
# Take care, it can be very long; several days
result_mcmc_tsd <- tsd_MHmcmc(result=eo_logistic,
                             parametersMCMC=pMCMC, n.iter=10000, n.chains = 1,
                             n.adapt = 0, thin=1, trace=TRUE)
# summary() permits to get rapidly the standard errors for parameters
summary(result_mcmc_tsd)
# They are store in the result also. Two SE are estimated using or
# batch method or time-series SE:
# The batch standard error procedure is usually thought to be not
# as accurate as the time series methods.
se1 <- result_mcmc_tsd$BatchSE
se2 <- result_mcmc_tsd$TimeSeriesSE
plot(result_mcmc_tsd, parameters="S", scale.prior=TRUE, xlim=c(-3, 3), las=1)
plot(result_mcmc_tsd, parameters="P", scale.prior=TRUE, xlim=c(25, 35), las=1)
plot(eo_logistic, se=se2)

## End(Not run)
```

---

tsd_MHmcmc_p	<i>Generates set of parameters to be used with tsd_MHmcmc()</i>
--------------	---

---

### Description

Interactive script used to generate set of parameters to be used with tsd\_MHmcmc().

### Usage

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

### Arguments

result	An object obtained after a tsd fit
accept	If TRUE, the script does not wait user information

### Details

tsd\_MHmcmc\_p generates set of parameters to be used with embryogrowth\_MHmcmc()

### Value

A matrix with the parameters

### Author(s)

Marc Girondot

### Examples

```
## Not run:
library(embryogrowth)
eo <- subset(STSRE_TSD, Species=="Emys orbicularis", c("Males", "Females",
             "Incubation.temperature"))
eo_logistic <- with(eo, tsd(males=Males, females=Females,
             temperatures=Incubation.temperature))
pMCMC <- tsd_MHmcmc_p(eo_logistic, accept=TRUE)

## End(Not run)
```

---

weightmaxentropy	<i>Search for the weights of the nests which maximize the entropy of nest temperatures distribution</i>
------------------	---

---

### Description

Search for the weights of the nests which maximize the entropy of nest temperatures distribution. Entropy is measured by Shanon index.

Entropy method must be entropy.empirical because it is the only method insensitive to scaling.

If no weight is given, the initial weight is uniformly distributed.

Use control\_optim=list(trace=0) for not show progress of search report.

### Usage

```
weightmaxentropy(temperatures = stop("Temperature data must be provided !"),
  weight = NULL, entropy.method = entropy::entropy.empirical, plot = TRUE,
  control_optim = list(trace = 0, maxit = 500), control_plot = NULL,
  control_entropy = NULL, col = c("black", "red"))
```

### Arguments

temperatures	Timeseries of temperatures formated using FormatNests()
weight	A named vector of the initial weight search for each nest for likelihood estimation
entropy.method	Entropy function, for example entropy::entropy.empirical. See package entropy for description
plot	Do the plot of temperatures before and after weight must be shown ? TRUE or FALSE
control_optim	A list with control paramaters for optim function
control_plot	A list with control paramaters for plot function
control_entropy	A list with control paramaters for entropy function
col	Colors for unweighted and weighted distributions

### Details

Search for the weights of the nests which maximize the entropy of nest temperatures distribution

### Value

A named vector of weights

### Author(s)

Marc Girondot

**Examples**

```
## Not run:
library(embryogrowth)
data(nest)
formatted <- FormatNests(nest)
w <- weightmaxentropy(formatted, control_plot=list(xlim=c(20,36)))
x <- structure(c(120.940334922916, 467.467455887442,
306.176613681557, 117.857995419495),
.Names = c("DHA", "DHH", "T12H", "Rho25"))
# pfixed <- c(K=82.33) or rK=82.33/39.33
pfixed <- c(rK=2.093313)
# K or rK are not used for dydt.linear or dydt.exponential
resultNest_4p_weight <- searchR(parameters=x,
fixed.parameters=pfixed, temperatures=formatted,
derivate=dydt.Gompertz, M0=1.7, test=c(Mean=39.33, SD=1.92),
method = "BFGS", maxiter = 200, weight=w)
data(resultNest_4p_weight)
plotR(resultNest_4p_weight, ylim=c(0,0.50), xlim=c(15, 35))
# Standard error of parameters can use the embryogrowth_MHmcmc() function

## End(Not run)
```

# Index

- \*Topic **Ecology**
  - embryogrowth-package, 3
- \*Topic **Embryo**
  - embryogrowth-package, 3
- \*Topic **Gompertz**
  - embryogrowth-package, 3
- \*Topic **Growth**
  - embryogrowth-package, 3
- \*Topic **TSD**
  - embryogrowth-package, 3
- \*Topic **Temperature**
  - embryogrowth-package, 3
- \*Topic **datasets**
  - nest, 25
  - result\_mcmc\_4p, 40
  - result\_mcmc\_4p\_weight, 41
  - result\_mcmc\_6p, 42
  - result\_mcmc\_newp, 44
  - resultNest\_4p, 35
  - resultNest\_4p\_transition, 36
  - resultNest\_4p\_weight, 37
  - resultNest\_6p, 38
  - resultNest\_newp, 39
  - STSRE\_NestingArea, 47
  - STSRE\_TSD, 48
  - tempConst, 53
- ChangeSSM, 4
- dydt.exponential, 6
- dydt.Gompertz, 7
- dydt.linear, 8
- embryogrowth-package, 3
- embryogrowth\_MHmcmc\_p, 9
- FormatNests, 10
- GenerateAnchor, 11
- GenerateConstInc, 12
- GenerateTest, 13
- hist.Nests, 14
- hist.NestsResult, 15
- info.nests, 16
- likelihoodR, 17
- logLik.NestsResult, 19
- logLik.tsd, 20
- merge.mcmcComposite, 21
- minmax.periodic, 22
- MovingIncubation, 23
- nest, 25
- plot.NestsResult, 26
- plot.tsd, 28
- plot\_transition, 33
- plotR, 30
- plotR\_hist, 32
- predict.tsd, 34
- result\_mcmc\_4p, 40
- result\_mcmc\_4p\_weight, 41
- result\_mcmc\_6p, 42
- result\_mcmc\_newp, 44
- resultNest\_4p, 35
- resultNest\_4p\_transition, 36
- resultNest\_4p\_weight, 37
- resultNest\_6p, 38
- resultNest\_newp, 39
- searchR, 45
- STSRE\_NestingArea, 47
- STSRE\_TSD, 48
- summary.Nests, 51
- switch.transition, 52
- tempConst, 53
- temperature.periodic, 54
- TestParallel, 55



tsd, [56](#)

tsd\_MHmcmc, [59](#)

tsd\_MHmcmc\_p, [61](#)

weightmaxentropy, [62](#)