Package ‘Dark’

June 2, 2016

Type Package

Title The Analysis of Dark Adaptation Data

Version 0.9.8

Date 2016-06-01

Author Jeremiah MF Kelly

Maintainer Jeremiah MF Kelly <emkayoh@mac.com>

Description The recovery of visual sensitivity in a dark environment is known as dark adaptation. In a clinical or research setting the recovery is typically measured after a dazzling flash of light and can be described by the Mahroo, Lamb and Pugh (MLP) model of dark adaptation. The functions in this package take dark adaptation data and use nonlinear regression to find the parameters of the model that 'best' describe the data. They do this by firstly, generating rapid initial objective estimates of data adaptation parameters, then a multi-start algorithm is used to reduce the possibility of a local minimum. There is also a bootstrap method to calculate parameter confidence intervals. The functions rely upon a 'dark' list or object. This object is created as the first step in the workflow and parts of the object are updated as it is processed.

License GPL-3

LazyData yes

URL https://github.com/emkayoh/Dark, http://www.nihr.ac.uk

BugReports https://github.com/emkayoh/Dark/issues

Suggests knitr, rmarkdown,testthat

Imports stats, grDevices, graphics, utils

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

`Dark-package` *Dark: A package to analyse dark adaptation data*

Description

A series of scripts to find the parameters of dark adaptation.

Details

Package:	Dark
Type:	Package
Version:	0.9.6
Date:	2015-06-12
License:	GPL-3

Dark adaptation is the recovery of visual sensitivity in a dark environment and can be described by a physiological model. This package contains a series of functions to analyse data collected during dark adaptation.

The functions use the Mahroo Lamb and Pugh (MLP) model of dark adaptation. The functions in this package take dark adaptation data and find the parameters of the model that 'best fit' the data.

The functions generate rapid initial objective estimates of data adaptation parameters, a multi-start algorithm to reduce possibility of a local minimum. There is a bootstrap method to calculate parameter confidence intervals. There are also ancillary functions to facilitate the analysis.

The functions rely upon a *dark* list or object. This object is created by the first function and parts are added to the object as it is processed.

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

Maintainer: Jeremiah MF Kelly <emkayoh@mac.com>

References

O. Mahroo and T. Lamb. Recovery of the human photopic electroretinogram after bleaching exposures: estimation of pigment regeneration kinetics. The Journal of Physiology, 554(2):417, 2004.

T. Lamb and E. Pugh. Dark adaptation and the retinoid cycle of vision. Progress in Retinal and Eye Research, 23(3):307-380, 2004.

Examples

```
set.seed(1234)
Time<- seq(0,20)
tmp<- TestData(Time)
P<-Start(tmp,1000)
MSC<-ModelSelect(tmp, P)
tmp2<-BestFit(tmp, MSC)
tmp3<-MultiStart(tmp2,10)
BootDark(tmp3,50)
```

`AICc` *Akaike information criterion*

Description

The Akaike information criterion corrected for small sample size is a measure of the relative quality of a model. The AICc is calculated from a 'dark' object.

Usage

```
AICc(obj)
```

Arguments

`obj` A dark object

This object must have at least the following elements:

obj$time	to calculate the number of observations
obj$Pn	the number of parameters in the model
obj$val	the sum of squared residual error

Value

The value returned is an indication of the information lost by fitting a particular model to the data, and is only of merit when compared to the value from another model.

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

References

See http://en.wikipedia.org/wiki/Akaike_information_criterion.

K. Burnham and D. Anderson. Model selection and multi-model inference: a practical information-theoretic approach. Springer, 2002.

Sakamoto, Y., Ishiguro, M., and Kitagawa G. (1986). Akaike Information Criterion Statistics. D. Reidel Publishing Company.

See Also

AIC

Examples

```
AICc(dark)
```

BestFit *BestFit*

Description

This script takes a dark object, a list of parameters and AICc scores from ModelSelect to chose a model and then return optimised values for the parameter estimates. Analysis of the data can be halted here if wished.

However, a MultiStart check can be useful if it is suspected that a local minimum has been found. Furthermore, BootDark will provide confidence intervals for the parameter estimates.

Usage

```
BestFit(obj, MSC, draw)
```

Arguments

obj	A dark object
MSC	A list from the function ModelSelect
draw	A flag to indicate whether a figure should be drawn.

Value

A list with the following elements:

call	the last function call on the data
time	time of observations
thrs	thresholds
resid	residuals of best model fit
fit	fitted thresholds for the optimal model and parameters
thet	seed parameters of TestData, null if not TestData
sse	sum of squared error used in TestData
val	calculated sum of squared errors

`data`	source of the data
`opt`	optimal parameter estimates
`Mod`	optimal model
`Pn`	number of parameters required by the model to fit the data
`AIC`	AICc scores for the three models fitted
`R2`	an indication of the 'goodness' of fit

Note

This function makes extensive use of `optim`.

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

See Also

`optim`

Examples

```
tmp <- TestData(0:20)
P<-Start(tmp,100)
MSC<-ModelSelect(tmp, P)
BestFit(tmp, MSC)
```

`BootDark` *BootDark*

Description

A script using bootstrap techniques to calculate confidence intervals for parameter estimates from a 'dark' object.

Usage

```
BootDark(obj, R, graph, progress = F)
```

Arguments

`obj`	A 'dark' object.
`R`	The number of repeats for the bootstrap calculations.
`graph`	A flag to indicate whether a figure should be drawn.
`progress`	A flag to indicate whether a progress bar should be drawn to the console. This might be preferred if using a large number of repeats.

Details

The script calculates bootstrap estimates of confidence intervals by sampling the residuals without replacement. The seven parameter model 'P7c' is always used. If 'P3' or 'P5c' have been found elsewhere to be a better fit then this will be confirmed by bootstrapping the 'P7c' model.

Value

Returns a list 'out'

`out$time`	times of observations
`out$thrs`	thresholds
`out$opt`	optimised parameter estimates
`out$Mod`	the name of the optimal model
`out$Pn`	number of parameters needed to describe the data
`out$AIC`	the AICc scores for the three models
`out$fit`	fitted values for the optimal parameter estimates
`out$resd`	residuals of the best fits
`out$R2`	the coefficient of determination
`out$Bootstrap`	bootstrap parameter estimates, 2.5%, 50% and 97.5%
`out$weight`	the reciprocal of the CI
`out$valid`	nn indication whether the parameter estimate is valid
`out$data`	the source of the data
`out$call`	updates the call label on the object

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

References

B. Efron. Bootstrap methods: another look at the jackknife. The Annals of Statistics, 7(1):1-26, 1979.

B. Efron. Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. Biometrika, 68(3):589, 1981.

Examples

```
set.seed(1234)
Time<- seq(0,20)
tmp<- TestData(Time)
P<-Start(tmp,1000)
MSC<-ModelSelect(tmp, P)
tmp2<-BestFit(tmp, MSC)
tmp3<-MultiStart(tmp2,10)
BootDark(tmp3,50)
```

dark *Dark adaptation data.*

Description

This data was extracted from Figure 1 in Rushton's paradox: rod dark adaptation after flash photolysis, E.N.Pugh Jr. *The Journal of Physiology,* 1975.

Usage

```
data("dark")
```

Format

`dark` is a list of 15 items, that are used or created by the functions in this package.

Details

The items are:

$time: the time of observations

- $thrs: the thresholds
- $fit: thresholds predicted by the model
- $resid: residuals between the fitted model and observed data
- $R2: the coefficient of determination
- $Bootstrap: a table of quantiles (2.5%, 50% and 97.5%) for the parameter estimates from bootstrap methods
- $weight: the parameter estimate divided by the 95% quantile range
- $valid: an integer array indicating whether the quantile range encloses zero.
- $opt: parameter estimates of the optimal model
- $Mod: a string describing the optimal model
- $Pn: the number of parameters in the optimal model
- $AIC: an array with the AICc scores for the three models
- $val: sum of residuals squared
- $call: the last function call that produced the object
- $data: the source of the data

References

E. Pugh. Rushton's paradox: rod dark adaptation after flash photolysis. The Journal of Physiology, 248(2):413, 1975.

http://onlinelibrary.wiley.com/doi/10.1113/jphysiol.1975.sp010982/abstract

http://onlinelibrary.wiley.com/doi/10.1113/jphysiol.1975.sp010982/pdf

Examples

```
data(dark)
## load(dark)
```

Declutter *Declutter*

Description

A function to remove multiple button presses, i.e. data that has multiple values for the same threshold.

Usage

```
Declutter(tmp, delta)
```

Arguments

`tmp`	a 'dark' object with at least two elements; tmp$time and tmp$thrs.
`delta`	The minimum time in seconds between responses. Thresholds set within two seconds of each other are discarded apart from the first threshold.

Details

In early data collected with analogue equipment it was psossible for a subject to return multiple button presses when setting just one threshold. This data is characterised by clusters of points within a very short time. This function removes the exptra presses. It is rarely need for data collected from digital equipment.

Value

Returns an object with the same elements as 'tmp' but with object$time and object$thrs altered.

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

Examples

```
set.seed(123)
Time  <-c(0, 0.02, 1, 2,  3, 3.02,  5, 6, 6.02, 7, 9, 9.02, 11,
12, 12.02, 13, 15, 15.02, 16, 18, 18.02, 20)
# with duplicated times
set.seed(1234)
tmp <- TestData(Time, sse=0.05)
## Not run: plot(tmp$time, tmp$thrs, ylim=c(-4,0))
tmp <- Declutter(tmp)
## Not run: points(tmp$time, tmp$thrs, col='red', pch=16)
```

`GetData` *GetData*

Description

A template function that gets the data from a file and converts it to a *dark* object for use by other scripts. The script can be modified to format the data. A *dark* object has time data in minutes and thresholds in log units. If the data have been collected in other units then the script should convert them.

The script defaults to returning data generated by `TestData`.

Usage

```
GetData(path, .....)
```

Arguments

`path` This is the location of the data and will usually be a file path string.

`.....` This can be any other values that might be needed to identify the data, e.g. subject number or study reference.

Details

This script can be altered in any way desired and then saved with a different name. I suggest the format ‘GetData....R‘, where the ellipsis describes the data in some way.

Value

A *dark* object with at least two elements

`time` the time elapsed after measurements begin in minutes

`thrs` the thresholds recorded in log units

other possible values include

`data` the name of the data source

`init` initial estimates of the optimal model parameters

`opt` optimal estimates of the optimal model parameters

`resid` the residuals of the data for an optimal model

`...` others to be added

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

See Also

`TestData`

Examples

```
set.seed(1234)
tmp <- GetData()
```

H *This is a simple switch function.*

Description

This function is used to transition from ’off’ to ’on’.

Usage

```
H(x, k = 100, t)
```

Arguments

`x`	is the measured time.
`k`	is the transition constant, set arbitrarily high.
`t`	is the time at which the transition occurs.

Details

This helper function used in `P5c` and `P7c` enables the `optim` function to find parameters three times as quickly than if the transitions betwen the phases are modelled by a logical function e.g. a step function.

Value

For times before ’t’ the output is less than or equal to 0.5, after this time the the output is greater than 0.5. As ’k’ grows larger the rate of transition from 0 to 1 increases.

Note

H is a logistic function that maps inputs to a values between zero and one

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

References

The logistic function: `http://en.wikipedia.org/wiki/Logistic_function`

The step function: `http://en.wikipedia.org/wiki/Heaviside_step_function`

See Also

`P5c`, `P7c`

Examples

```
x <- seq(0, 20, by=0.1)
k <- 10
t <- 10
op <- par(las=1, bty='n')
## Not run: plot(x,H(x,k,t), 'l')
par(op)
```

ModelSelect *ModelSelect*

Description

Returns a list with two elements; an array of AICc scores indexed by the number of parameters in the model considered and a matrix of parameters with three rows, one for each model.

Usage

```
ModelSelect(obj, P)
```

Arguments

`obj`	A 'dark' object.
`P`	Is a matrix with seven columns and at least one row. The values of each element can be zero.

Details

This is a *brute-force* method to make a first estimate of the optimal model parameters.

The matrix 'P' holds rows of possible parameter values. Each row is passed to the 3, 5, and 7 parameter models and the sum of residuals squared is calculated for the given times (obj$time) and thresholds (obj$thrs). So for each row in 'P' there is a score for each model. Then for each model the row which yields the lowest SSE is chosen as a starting point for optimisation. The optimised parameters are stored in 'param' and once the three parameter arrays have been found their AICc scores are found and returned as AIC.

Value

Returns a list

`AIC`	An array of seven values with AIC scores at the index of model parameter count.
`param`	A three row by seven column matrix. Each row containing the optimised parameters for each model.

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

References

http://en.wikipedia.org/wiki/Brute-force_search

Examples

```
set.seed(1234)
tmp<- TestData(0:20)
P<-Start(tmp)
ModelSelect(tmp,P)
```

MultiStart *MultiStart*

Description

Given a *dark* object, obj, this function repeatedly optimises the parameters in the vicinity of the seed array. The width of the search is dependent upon the value of spread.

Usage

```
MultiStart(obj, repeats, draw, spread, debug)
```

Arguments

obj	A *dark* object containing at least;

obj$time	time
obj$thrs	thresholds
obj$init	an initial estimate of the parameters of dark adaptation.

repeats	The number of times the algorithm is repeated
draw	A flag indicating whether a figure should be drawn.
spread	The amount by which the seed array should be varied. A larger value gives a greater range of possible starting points.
debug	A flag used in debugging the software.

Details

To reduce the possibility of selecting non-optimal parameter estimates, the optimisation is repeated in the region of initial estimates. The

Value

Returns a list;

time	times of threshold setting
out$thrs	observed thresholds
out$resid	residuals
out$fit	optimal fitted values

`out$thet`	seed parameters if test data
`out$sse`	sum of squared residuals if test data
`out$data`	source of the data
`out$opt`	optimal parameter estimates of the chosen model
`out$Mod`	name of the optimal model
`out$Pn`	the number of parameters needed to describe the data
`out$AIC`	array of AICc scores
`out$val`	calculated sum of squared residuals
`out$R2`	the coefficient of determination
`out$warning`	if none of the nearby values converge
`out$call`	updates the function call label

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

References

Nelder, J.A.; Mead, R. 1965: A simplex for function minimization. Comput. J. 7, 308-313

Examples

```
set.seed(1234)
Time<- seq(0,20)
tmp<- TestData(Time)
P<-Start(tmp,1000)
MSC<-ModelSelect(tmp, P)
tmp2<-BestFit(tmp, MSC)
tmp3<-MultiStart(tmp2,10)
```

`P3` *Three parameter model.*

Description

The three parameter model. A simple exponential decay.

Usage

```
P3(a, X)
```

Arguments

`a` An array of parameters;

a:	parameter	Units
a[1]:	cone threshold	(log lum)
a[2]:	cone coefficient	(log lum)
a[3]:	cone time constant	(minutes)

X The times when the model predicts thresholds.

Details

This function has three roles, to calculate the thresholds for given parameters a and times X. If missing X, then the function calls the values x and y from the .Globalenv and calculates the sum of residuals squared error (SSE) for a. If a is an array of length 1L or boolean then a description of the model is returned.

The use of the function H rather than an impulse function gives a three-fold increase is speed for the optim function.

Value

The output depends upon the input. If the input is an array of length 1L or a boolean then a list is returned

Pn	number of parameters
Mod	name of the model

If a parameter array is passed then the sum of residuals squared is calculated. This is used by optim to optimise the parameter estimates.

Passing a parameter array and a series of putative times causes the function to return predicted thresholds.

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

References

O. Mahroo and T. Lamb. Recovery of the human photopic electroretinogram after bleaching exposures: estimation of pigment regeneration kinetics. The Journal of Physiology, 554(2):417, 2004.

T. Lamb and E. Pugh. Dark adaptation and the retinoid cycle of vision. Progress in Retinal and Eye Research, 23(3):307-380, 2004.

See Also

P5c,P6c,P7c

Examples

```
set.seed(1234)
x <- 0:20
a <- c(-1.00,  1.00,  1.00, -0.24,  6.00,  0.20, 13.00)
tmp <- TestData(x, a)
y <- tmp$resid

P3(TRUE)
# Describes the model

P3(a)
# The sum of squared residuals
```

```
P3(a,x)
# The fitted thresholds for given parameters 'theta' and times 'x'
```

P5c *Five parameter model.*

Description

The five parameter model. An exponential decay followed by a linear phase.

Usage

```
P5c(a,X)
```

Arguments

`a` An array of parameters;

a:	parameter	Units
a[1]:	cone threshold	(log lum)
a[2]:	cone coefficient	(log lum)
a[3]:	cone time constant	(minutes)
a[4]:	S2	(log lum / minute)
a[5]:	alpha point	(minutes)

`X` The times when the model predicts thresholds.

Details

This function has three roles, to calculate the thresholds for given parameters `a` and times `X`. If missing `X`, then the function calls the values `x` and `y` from the `.Globalenv` and calculates the sum of residuals squared error (SSE) for `a`. If `a` is an array of length `1L` or `boolean` then a description of the model is returned.

The use of the function `H` rather than an impulse function gives a three-fold increase is speed for the `optim` function.

Value

If the input is an array of length `1L` or a `boolean` then a list is returned

`Pn`	number of parameters
`Mod`	name of the model

If a parameter array is passed then the sum of residuals squared is calculated. This is used by `optim` to optimise the parameter estimates.

Passing a parameter array and a series of putative times causes the function to return predicted thresholds.

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

References

O. Mahroo and T. Lamb. Recovery of the human photopic electroretinogram after bleaching exposures: estimation of pigment regeneration kinetics. The Journal of Physiology, 554(2):417, 2004.

T. Lamb and E. Pugh. Dark adaptation and the retinoid cycle of vision. Progress in Retinal and Eye Research, 23(3):307-380, 2004.

See Also

P3,P6c,P7c

Examples

```
set.seed(1234)
x <- 0:20
a <- c(-1.00,  1.00,  1.00, -0.24,  6.00,  0.20, 13.00)
tmp <- TestData(x, a)
y <- tmp$resid

P5c(TRUE)
# Describes the model

P5c(a)
# The sum of squared residuals

P5c(a,x)
# The fitted thresholds for given parameters 'theta' and times 'x'
```

P6c *A six parameter model*

Description

An exponential decay followed by a second exponential decay.

Usage

```
P6c(a, X)
```

Arguments

a An array of parameters;

a:	parameter	Units
a[1]:	absolute (rod) threshold	(log lum)
a[2]:	cone coefficient	(log10 lum)

a[3]:	1/cone time constant	(1/minutes)
a[4]:	alpha point	(minutes)
a[5]:	absolute threshold offset	(log10 lum)
a[6]:	1/rod time constant	(1/minutes)

note that the cone threshold is a[1] + a[5] log10(lum)

`X` The times in minutes when the model predicts thresholds.

Details

This function has three roles, to calculate the thresholds for given parameters `a` and times `X`. If missing `X`, then the function calls the values `x` and `y` from the `.Globalenv` and calculates the sum of residuals squared error (SSE) for `a`. If `a` is an array of length `1L` or `boolean` then a description of the model is returned.

The use of the function `H` rather than an impulse function gives a three-fold increase is speed for the `optim` function.

Value

The output depends upon the input. If the input is an array of length `1L` or a `boolean` then a list is returned

`Pn` number of parameters

`Mod` name of the model

If a parameter array is passed then the sum of residuals squared is calculated. This is used by `optim` to optimise the parameter estimates.

Passing a parameter array and a series of putative times causes the function to return predicted thresholds.

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

References

O. Mahroo and T. Lamb. Recovery of the human photopic electroretinogram after bleaching exposures: estimation of pigment regeneration kinetics. The Journal of Physiology, 554(2):417, 2004.

T. Lamb and E. Pugh. Dark adaptation and the retinoid cycle of vision. Progress in Retinal and Eye Research, 23(3):307-380, 2004.

See Also

`P3`,`P5c`,`P6c`

Examples

```
set.seed(1234)
x <- 0:20
a <- c(-2, 2, 1/2, 10, 3, 1/8)

# P6c(TRUE)
# Describes the model

# P6c(a)
# The sum of squared residuals

# P6c(a,x)
# The fitted thresholds for given parameters 'a' and times 'x'
```

P7c *Seven parameter model*

Description

The seven parameter model. An exponential decay followed by two linear phases.

Usage

```
P7c(a, X)
```

Arguments

a An array of parameters;

a:	parameter	Units
a[1]:	cone threshold	(log lum)
a[2]:	cone coefficient	(log lum)
a[3]:	cone time constant	(minutes)
a[4]:	S2	(log lum/ minute)
a[5]:	alpha point	(minutes)
a[6]:	-(S2 + S3)	(log luminance)
a[7]:	beta point	(log luminance)

X The times when the model predicts thresholds.

Details

This function has three roles, to calculate the thresholds for given parameters `a` and times `X`. If missing `X`, then the function calls the values `x` and `y` from the `.Globalenv` and calculates the sum of residuals squared error (SSE) for `a`. If `a` is an array of length `1L` or `boolean` then a description of the model is returned.

The use of the function `H` rather than an impulse function gives a three-fold increase is speed for the `optim` function.

Value

The output depends upon the input. If the input is an array of length `1L` or a `boolean` then a list is returned

`Pn`	number of parameters
`Mod`	name of the model

If a parameter array is passed then the sum of residuals squared is calculated. This is used by `optim` to optimise the parameter estimates.

Passing a parameter array and a series of putative times causes the function to return predicted thresholds.

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

References

O. Mahroo and T. Lamb. Recovery of the human photopic electroretinogram after bleaching exposures: estimation of pigment regeneration kinetics. The Journal of Physiology, 554(2):417, 2004.

T. Lamb and E. Pugh. Dark adaptation and the retinoid cycle of vision. Progress in Retinal and Eye Research, 23(3):307-380, 2004.

See Also

`P3`,`P5c`,`P6c`

Examples

```
set.seed(1234)
x <- 0:20
a <- c(-1.00,  1.00,  1.00, -0.24,  6.00,  0.20, 13.00)
tmp <- TestData(x, a)
y <- tmp$resid

# P7c(TRUE)
# Describes the model

# P7c(a)
# The sum of squared residuals

# P7c(a,x)
# The fitted thresholds for given parameters 'a' and times 'x'
```

Start *Start*

Description

A function to build an array of starting parameters from a *dark* object.

Usage

```
Start(obj, Reps)
```

Arguments

`obj`	A *dark* object
`Reps`	The number of rows in the array.

Details

The array of starting parameters is built from the time and threshold data in the object, `obj`.

Each parameter is assumed to have a possible range given the data.

Each range is constructed as follows; the time points; *alpha* (cone-rod transition α minutes) and *beta* (rod-rod transition β minutes) are assumed to fall in the first and second halves of the time data respectively (`obj$time`).

The *cone threshold* is assumed to be in the upper half of the threshold data (`obj$thrs` log units). The *cone coefficient* (log units) or threshold at time zero is presumed to be positive and the same values are used for the *time constants* (tau minutes).

The rate of rod recovery *S2*, and the combined parameter *-(S2 + S3)* are ranged between -0.6 and 0 log units/minute.

These ranges of possible values are complied into an array by sampling without replacement from each range for each parameter.

Value

Output is an array of seven columns and number of rows = 'Reps'.

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

Examples

```
set.seed(1234)
tmp<-TestData(0:20)
Start(tmp, 10)
```

TestData *Data that can be used to test other scripts.*

Description

This script creates data in the form of a *dark* object from specified times, parameters, and level of variability. It is used for testing and developing other scripts. Presently only the seven parameter model and its subsets are implemented.

Usage

```
TestData(x, theta, sse, repeatable)
```

Arguments

`x` the times at which observations are made.

`theta` the parameters of the Mahroo Lamb Pugh model

theta :	parameter	default	Units
theta[1]:	cone threshold	-1	(log lum)
theta[2]:	cone coefficient	1	(log lum)
theta[3]:	cone time constant	1	(minutes)
theta[4]:	S2	-0.24	(log lum/ minute)
theta[5]:	alpha point	6	(minutes)
theta[6]:	-(S2 + S3)	0.20	(log luminance)
theta[7]:	beta point	13	(log luminance)

`sse` the variability of the data

`repeatable` a boolean flag that ensures that the function can return the same values each time it is called.

Details

The parameters values chosen as defaults are entirely arbitrary. The sixth parameter is the negative sum of the rates of rod recovery called *S2* and *S3*

Value

The function returns a *dark* object with the following components;

`call` a label describing the last call the object was subject to

`time` the time observations

`thrs` thresholds

`resid` residuals

`fit` thresholds predicted in the absence of noise

`thet` parameters passed to the function

`sse` the sum of squared residuals value used to describe the variability in the data

`val`	the actual sse of the generated data
`data`	the name of the data source

Author(s)

Jeremiah MF Kelly

Faculty of Life Sciences, The University of Manchester, M13 9PL, UK

References

L. Patryas, N. R. Parry, D. Carden, D. H. Baker, J. M. Kelly, T. Aslam, and I. J. Murray. Assessment of age changes and repeatability for computer-based rod dark adaptation. Graefe's Archive for Clinical and Experimental Ophthalmology, pages 1-7, 2013.

Examples

```
set.seed(1234)
x <- seq(0,20)
tmp <- TestData(x)
tmp
```

Index