Package 'DEBtoolAnimal'

December 9, 2015

Type Package

Title DEB functions for an animal

Version 0.1

Date 2015-09-30

Author Goncalo M. Marques <goncalo.marques@tecnico.ulisboa.pt>

Maintainer Goncalo M. Marques <goncalo.marques@tecnico.ulisboa.pt>

Description
DEB based functions for the std (standard) and abj (with acceleration) models for animals.

License GPL

LazyData TRUE

NeedsCompilation no

R topics documented:

beta0 1
C2K 2
dget_lbarb2 3
fnget_lbarb2 3
get_lb 4
get_lbarb 5
get_lbarb2 5
get_ubarE0 6
get_ue0 7
initial_scaled_reserve 7
K2C 8
tempcorr 9

`beta0` *Particular incomplete beta function*

Description

particular incomplete beta function:

Usage

```
beta0(x0, x1)
```

Arguments

x0	scalar with lower boundary for integration
x1	scalar with upper boundary for integration

Details

Computes

$$B_{x_1}\left(\frac{4}{3},0\right) - B_{x_0}\left(\frac{4}{3},0\right) = \int_{x_0}^{x_1} t^{4/3-1}(1-t)^{-1}dt$$

To be used in the computation of the age at birth (or related quantities) for an egg.

Value

scalar with particular incomple beta function

See Also

Other miscellaneous functions: C2K; K2C

Examples

```
beta0(0.1, 0.2)
```

C2K *Conversion of Celsius to Kelvin*

Description

Converts temperature in degrees Celsius to Kelvin

Usage

```
C2K(C)
```

Arguments

C	numeric temperature in degrees Celsius

Value

temperature in Kelvin

See Also

Other miscellaneous functions: K2C; beta0

Examples

```
C2K(20)
```

dget_lbarb2 *Computes derivative d delta/dx*

Description

Obtains the derivative d delta/dx from lbarb, xb and k.

Usage

```
dget_lbarb2(x, delta, pars)
```

Arguments

x	scalar x = g/(g + e)
delta	scalar delta = x e_H/ (1 - kap)g
pars	data.frame with lbarb, xb, xb3 (xb^1/3), k

Value

scalar with derivative value d delta/ dx

See Also

Other scaled get functions: `fnget_lbarb2`; `get_lbarb2`; `get_lbarb`; `get_lb`; `initial_scaled_reserve`

Examples

```
dget_lbarb2(10^(-6), 0, c(lbarb = 0.003, xb = 10/11, xb3 = (10/11)^(1/3), k = 1))
```

fnget_lbarb2 *Computes f using the ode solver for delta(x), for finding lbarb*

Description

Computes f using the ode solver for delta(x), for finding lbarb.

Usage

```
fnget_lbarb2(lbarb, pars)
```

Arguments

lbarb	scalar with scaled length at birth (lbarb = lb/ g)
pars	data.frame with lbarb, xb, xb3 (xb^1/3), k

Value

scalar with function f which when zero indicates lbarb

See Also

Other scaled get functions: `dget_lbarb2`; `get_lbarb2`; `get_lbarb`; `get_lb`; `initial_scaled_reserve`

Examples

```
fnget_lbarb2(0.03, c(xb = 10/11, xb3 = (10/11)^(1/3), vbarHb = 0.001, k = 1))
```

`get_lb` *Computes scaled length at birth*

Description

Obtains scaled length at birth, given the scaled reserve density at birth.

Usage

```
get_lb(pars, eb = 1, lb0 = as.numeric(pars[3]^(1/3)))
```

Arguments

`pars`	3-vector with parameters: g, k, v_H^b
`eb`	optional scalar with scaled reserve density at birth (default eb = 1)
`lb0`	optional scalar with initial estimate for scaled length at birth (default lb0: lb for k = 1)

Details

The theory behind get_lb, get_tb and get_ue0 is discussed in http://www.bio.vu.nl/thb/research/bib/Kooy2009b.html. Solves $y(x_b) = y_b$ for l_b with explicit solution for $y(x)$

$$y(x) = \frac{xe_H}{1 - kap} = xg\frac{u_H}{l^3}$$

and $y_b = x_b g u_H^b/((1 - kap)l_b^3)$

$$\frac{d}{dx}y = r(x) - ys(x)$$

with solution $y(x) = v(x)\int_0^x r(x')/v(x')dx'$ and $v(x) = exp(-\int_0^x s(x')dx')$. A Newton Raphson scheme is used with Euler integration, starting from an optional initial value. Shooting method: get_lb2. In case of no convergence, get_lb2 is run automatically as backup. Consider the application of get_lb_foetus for an alternative initial value.

Value

scalar with scaled length at birth (lb) and indicator equals 1 if successful, 0 otherwise (info)

See Also

Other scaled get functions: `dget_lbarb2`; `fnget_lbarb2`; `get_lbarb2`; `get_lbarb`; `initial_scaled_reserve`

Examples

```
get_lb(c(g = 10, k = 1, vHb = 0.5), 1)
```

get_lbarb *Computes scaled length at birth lbarb*

Description

Obtains scaled length at birth, given the scaled reserve density at birth.

Usage

```
get_lbarb(pars, eb = 1, lbarb0 = NA)
```

Arguments

pars	3-vector with parameters: g, k, vbar_H^b
eb	optional scalar with scaled reserve density at birth (default eb = 1)
lbarb0	optional scalar with initial estimate for scaled length at birth (default lbarb0: lbarb for k = 1)

Value

scalar with scaled length at birth (lbarb) and indicator equals 1 if successful, 0 otherwise (info)

See Also

Other scaled get functions: dget_lbarb2; fnget_lbarb2; get_lbarb2; get_lb; initial_scaled_reser

Examples

```
get_lbarb(c(g = 10, k = 1, vbarHb = 0.0005), 1)
```

get_lbarb2 *Computes initial scaled reserve*

Description

Obtains scaled length at birth, given the scaled reserve density at birth. Like get_lbarb, but uses a shooting method in 1 variable.

Usage

```
get_lbarb2(pars, eb = NA)
```

Arguments

pars	3-vector with parameters: g, k, vbar_H^b
eb	optional scalar with scaled reserve density at birth (default eb = 1)

Value

scalar with scaled length at birth (lbarb) and indicator equals 1 if successful, 0 otherwise (info)

See Also

Other scaled get functions: dget_lbarb2; fnget_lbarb2; get_lbarb; get_lb; initial_scaled_reserv

Examples

```
get_lbarb2(c(g = 10, k = 1, vbarHb = 0.01), 1)
```

---

get_ubarE0 *Computes initial scaled reserve density at birth*

Description

Obtains the initial scaled reserve given the scaled reserve density at birth. Function get_ue0 does so for eggs, get_ue0_foetus for foetuses. Specification of length at birth as third input by-passes its computation, so if you want to specify an initial value for this quantity, you should use get_lb directly.

Usage

```
get_ubarE0(g = NA, k = NA, vbarHb = NA, eb = 1, lbarb = NA)
```

Arguments

g	energy investment ratio
k	maintenance ratio
vbarHb	rescaled maturity volume at birth
eb	optional scalar with scaled reserbe density at birth
lbarb	optional scalar with scaled length at birth

Value

scalar with particular incomple beta function

See Also

Other get functions: get_ue0

Examples

```
get_ubarE0(g = 10, lbarb = 0.01)
get_ubarE0(g = 10, k = 0.7, vbarHb = 5e-4)
```

`get_ue0` *Computes initial scaled reserve*

Description

Obtains the initial scaled reserve given the scaled reserve density at birth. Function get_ue0 does so for eggs, get_ue0_foetus for foetuses. Specification of length at birth as third input by-passes its computation, so if you want to specify an initial value for this quantity, you should use get_lb directly.

Usage

```
get_ue0(pars, eb = 1, lb0 = NA)
```

Arguments

`pars`	1 or 3 -vector with parameters g, k_J/ k_M, v_H^b, see get_lb
`eb`	optional scalar with scaled reserbe density at birth (default: eb = 1)
`lb0`	optional scalar with scaled length at birth (default: lb is optained from get_lb)

Value

uE0 scalar with scaled reserve at t=0: $U_E^0 g^2 k_M^3/ v^2$ with $U_E^0 = M_E^0/ {J_EAm}$, lb scalar with scaled length at birth and info indicator equals 1 if successful, 0 otherwise

See Also

Other get functions: `get_ubarE0`

Examples

```
get_ue0(pars = c(0.42, 1, 0.066), eb = 1, lb0 = 0.4042)
```

`initial_scaled_reserve` *Gets initial scaled reserve*

Description

Gets initial scaled reserve.

Usage

```
initial_scaled_reserve(f, pars, Lb0 = NA)
```

Arguments

`f`	n-vector with scaled functional responses
`pars`	5-vector with parameters: VHb, g, kJ, kM, v
`Lb0`	optional n-vector with lengths at birth

Value

n-vector with initial scaled reserve: M_E^0/ J_EAm (U0), n-vector with length at birth (Lb) and n-vector with 1's if successful, 0's otherwise (info)

See Also

Other scaled get functions: `dget_lbarb2`; `fnget_lbarb2`; `get_lbarb2`; `get_lbarb`; `get_lb`

Examples

```
initial_scaled_reserve(f = c(1, 0.9), pars = c(VHb = .8, g = .42, kJ = 1.7, kM = 1.7, v =
```

K2C *Conversion of Kelvin to Celsius*

Description

Converts temperature in Kelvin to degrees Celsius

Usage

```
K2C(K)
```

Arguments

`K`	numeric temperature in degrees Kelvin

Value

temperature in Kelvin

See Also

Other miscellaneous functions: `C2K`; `beta0`

Examples

```
K2C(293.15)
```

`tempcorr` *Temperature correction*

Description

Calculates the factor with which physiological rates should be multiplied to go from a reference temperature to a given temperature

Usage

```
tempcorr(Temp, T_1, T_A, T_L = NA, T_AL = NA, T_H = NA, T_AH = NA)
```

Arguments

`Temp`	vector with temperatures (in Kelvin)
`T_1`	scalar with reference temperature (in Kelvin)
`T_A`	scalar with Arrhenius temperature (in Kelvin)
`T_L`	optional scalar with lower boundary of temperature range (in Kelvin)
`T_AL`	optional scalar with Arrhenius temperature for lower boundary of temperature range (in Kelvin)
`T_H`	optional scalar with upper boundary of temperature range (in Kelvin)
`T_AH`	optional scalar with Arrhenius temperature for upper boundary of temperature range (in Kelvin)

Details

Temperature impacts metabolic rates. This impact, in its most simplest way (1 parameter), is modeled by multiplying all the time-dependent parameters by a correction factor:

$$\exp\left(\frac{T_A}{T_1} - \frac{T_A}{T}\right)$$

For a more detailed modeling one can multiply with an extra fraction $s(T_1)/s(T)$ with (3 parameters):

$$s(T) = 1 + \exp\left(\frac{T_{AL}}{T} - \frac{T_{AL}}{T_L}\right)$$

or (5 parameters)

$$s(T) = 1 + \exp\left(\frac{T_{AL}}{T} - \frac{T_{AL}}{T_L}\right) + \exp\left(\frac{T_{AH}}{T_H} - \frac{T_{AH}}{T}\right)$$

Value

vector with temperature correction factors that affect all rates

Examples

```
tempcorr(c(330, 331, 332), 320, T_A = 12000, T_L = 277, T_H = 331, T_AL = 20000, T_AH = 1
```