Package: FLife (via r-universe)

June 4, 2026

Type Package

Title Methods for Modelling Life History Traits

Version 3.4.0

Date 2018-07-25

Author Laurence Kell <laurie@seaplusplus.co.uk>

Maintainer Laurence Kell <laurie@seaplusplus.co.uk>

Description Many studies have shown the relationships between life history traits for processes such as growth, maturity and natural mortality. Life history has been used to develop priors in stock assesments and to parameterise ecological models. Package has a variety of methods for modelling life history traits and processes.

VignetteBuilder knitr

LazyData true

License GPL (>= 2)

Depends R(>= 3.6.0), FLCore (>= 2.6.13), ggplotFL

Imports methods, plyr, FLBRP

Suggests FLasher, FLAssess, testthat, knitr, GGally, reshape, popbio, rmarkdown

Additional_repositories http://flr-project.org/R

Collate generic.R FLife-lhPar.R FLife-stk.R data.R vonB.R gascuel.R gompertz.R richards.R ages.R len2wt.R wt2len.R knife.R logistic.R dnormal.R sigmoid.R FLife-m.R fapex.R lhEql.R lhRef.R lopt.R lh-indicators.R bench.R lh-srr-sv.R noise.R stars.R FLife-dd-m.R FLife-dd-mat.R grw-dd.R util.R leslie.R calcR.R len-index.R power.R spectra.R t0.R extendPlusGroup.R getScriptPath.R sim.R knifeEdgeMsy.R netSel.R funcs-indicators.R plot-bivariate.R reverse.R priors.R ggpairs.R steepness.R rtest.R climate.R production.R

Encoding UTF-8

RoxygenNote 7.2.1

Repository https://flr.r-universe.dev

Date/Publication 2024-03-25 11:18:28 UTC

RemoteUrl https://github.com/flr/FLife

RemoteRef HEAD

RemoteSha 16b34762fe075a44dd225756db34bc0ed69d26fd

Contents

`ages,FLQuant-method` *Fills an* `FLQuant` *with ages*

Description

Creates `FLQuant` and `FLCohort` with ages as entries

Usage

```
## S4 method for signature 'FLQuant'
ages(object)
```

Arguments

`object`	`FLQuant` or `FLCohort`
`...`	any other arguments

Value

`FLQuant` or `FLCohort` depending on what the first argument was

See Also

`knife` `gascuel` `sigmoid` `gompertz` `vonB` `dnormal` `logistic`

Examples

```
## Not run:
data(ple4)
ages(m(ple4))
## End(Not run)
```

`cas` *cas*

Description

A dataset containing lengths by year

Format

An `data.frame` object

year of capture

len at capture

n frequency

See Also

teleost

---

dnormal,FLQuant,FLPar-method

Double normal ogive

Description

Double normal ogive

Usage

```
## S4 method for signature 'FLQuant,FLPar'
dnormal(age, params, ...)
```

Arguments

`age`	FLQuant or FLCohort
`params`	`FLPar` with parameters `a1` age at maximum, `sl` SD for lefthand limb and `sr` SD for righthand limb.
`...`	any other arguments

Value

Returns an object of same class as `age` e.g. `FLQuant`

See Also

`sigmoid`, `dnormal`, `logistic`

Examples

```
## Not run:
params=FLPar(a1=4,sl=2,sr=5000)
dnormal(FLQuant(1:10,dimnames=list(age=1:10)),params)

## End(Not run)
```

`gascuel,FLQuant,FLPar-method`

Gascuel growth curve

Description

Gascuel growth equation

Usage

```
## S4 method for signature 'FLQuant,FLPar'
gascuel(age, params, ...)
```

Arguments

`age`	FLQuant, FLPar or numeric with ages
`params`	`FLPar`
`...`	any other arguments

Details

Gascuel D., Fonteneau, A., and Capisano, C. (1992). Modelisation d'une croissance en deux stances chez #l'albacore (Thunnus albacares) de l'Atlantique Est. Aquat. Living Resour. 5: 155-172.

Value

Returns a class of same type as `age` e.g. `FLQuant`

See Also

`gompertz`, `vonB` , `richards`

Examples

```
## Not run:
gascuel(10)

## End(Not run)
```

`gislason,FLQuant,numeric-method` *gislason*

Description

gislason natural mortality relatoinship estimate M as a function of length. M=a*length^b;

Usage

```
## S4 method for signature 'FLQuant,numeric'
gislason(length, params, a = 0.55, b = 1.44, c = -1.61, ...)
```

Arguments

`length`	mass at which M is to be predicted
`params`	`FLPar` with two values; i.e. a equal to M at unit mass and b a power term; defaults are a=0.3 and b=-0.288
`a`	0.55
`b`	1.44
`c`	-1.61
`...`	any other arguments

See Also

`lorenzen`

Examples

```
## Not run:
params=lhPar(FLPar(linf=111))
len=FLQuant(c( 1.90, 4.23, 7.47,11.48,16.04,20.96,26.07,31.22,
               36.28,41.17,45.83,50.20,54.27,58.03,61.48,64.62),
             dimnames=list(age=1:16))
gislason(length,params)

## End(Not run)
```

`gompertz,FLQuant,FLPar-method` *Gompertz growth equation*

Description

gompertz growth equation

Usage

```
## S4 method for signature 'FLQuant,FLPar'
gompertz(age, params, ...)
```

Arguments

`age`	FLQuant, FLPar or numeric with ages
`params`	`FLPar` with parameters for `linf`, `a`, `k`
`...`	any other arguments

Value

Returns an object of same class as `age` e.g. `FLQuant`

See Also

`gascuel`, `vonB`, `richards`

Examples

```
## Not run:
params=FLPar(linf=100,a=2,b=.4)
age=FLQuant(1:10,dimnames=list(age=1:10))
gompertz(age,params)

## End(Not run)
```

`grwdd,FLQuant,FLPar-method` *grwdd*

Description

Lorenzen natural mortality relationship where M is a function of weight, modified to explicitly included M as a function of numbers in a cohort, i.e. density dependence

Usage

```
## S4 method for signature 'FLQuant,FLPar'
grwdd(age, params, scale, k = 1, fn = vonB)
```

Arguments

`age`	mass at which length is to be predicted
`params`	an `FLPar` with two values; i.e. a equal to M at unit mass and b a power term; defaults are a=0.3 and b=-0.288
`scale`	reference
`k`	rate of change in density dependence
`fn`	function with growth model, with args age params
`...`	other arguments, such as scale, e.g. stock numbers now relative to a reference level, e.g. at virgin biomass and k steepness of relationship

Details

@details

The Lorenzen natural mortality relationship is a function of mass-at-age i.e. M=a*wt^b

The relationship can be explained by population density, since as fish grow they also die and so there is potentially less competition for resources between larger and older fish. Density dependence can be modelled by a logistic function, a sigmoid curve (or S shaped) curve, with equation

f(x)=L/(1+exp(-k(x-x0)))

where e is the natural logarithm base (also known as Euler's number), x0 is the x-value of the sigmoid's midpoint, L is the curve's maximum value, and k the steepness of the curve.

Combining the two functions gives

M=aL/(1+exp(-k(n-ref)))*wt^b;

See Also

`vonB`

Examples

```
## Not run:
library(FLBRP)
library(FLife)

data(teleost)
par=teleost[,"Hucho hucho"]
par=lhPar(par)
hutchen=lhEql(par)

scale=stock.n(hutchen)[,25]%*%stock.wt(hutchen)
scale=(stock.n(hutchen)%*%stock.wt(hutchen)%-%scale)%/%scale

grw=grwdd(wt2len(stock.wt(hutchen),par),params=par,scale,k=.2)

ggplot(as.data.frame(grw))+
   geom_line(aes(age,data,col=factor(year)))+
   theme(legend.position="none")+
   scale_x_continuous(limits=c(0,15))

## End(Not run)
```

`knife,FLQuant,FLPar-method`

knife edge ogive

Description

A method to simulate a knife edge ogive where at a given age the proportion chnages from 0 to 1

Usage

```
## S4 method for signature 'FLQuant,FLPar'
knife(age, params, ...)
```

Arguments

`age`	FLQuant, FLPar or numeric with ages
`params`	`FLPar`
`...`	any other arguments

Details

The knife ogive is an S-shaped or knife curve or knifeal functions, Verhulst hypothesizes that small populations increase geometrically, because the supply of resources exceeds demand. Then, as supply and demand balance, population growth is constant. Finally, as demand exceeds supply, population growth decreases at the same rate that it had increased. Verhulst describes this process with an equation that enables him to predict when a population will reach any given size (see Verhulst's Figure):

Value

returns an object of same type as `age` e.g. `FLQuant`

See Also

`sigmoid`, `dnormal`, `logistic`

Examples

```
## Not run:
params=FLPar(a1=4)
age=FLQuant(1:10,dimnames=list(age=1:10))
knife(age,params)

## End(Not run)
```

`len2wt,FLQuant,FLPar-method`
Length to weight conversion

Description

Converts length to weight based on \$W=aL^b\$

Usage

```
## S4 method for signature 'FLQuant,FLPar'
len2wt(length, params)
```

Arguments

`length`	age FLQuant, FLPar or numeric
`params`	`FLPar`
`...`	any other arguments

Value

Returns a class of same type as `length` e.g. `FLQuant`

See Also

`wt2len`

Examples

```
## Not run:
params=FLPar(a=1,b=3)
len2wt(FLQuant(10),params)

## End(Not run)
```

`leslie,FLBRP-method` *Leslie matrix*

Description

Creates a Leslie Matrix from a `FLBRP` object that represents a population at equilibrium

Usage

```
## S4 method for signature 'FLBRP'
leslie(object, fbar = FLQuant(0), numbers = TRUE, ...)
```

Arguments

`object`	`FLBRP`
`fbar`	`numeric` F at whicj survival calculated
`numbers`	boolean numbers or biomass, numbers bt default
`...`	any other arguments

Value

`matrix`

See Also

`lhRef`, `lhPar`, `lhEql`

Examples

```
## Not run:
eql=lhEql(lhPar(FLPar(linf=100)))
leslie(eql)

## End(Not run)
```

`lhEql,FLPar-method` *Derives an* `FLBRP` *from life history parameters*

Description

Takes an `FLPar` object with life history and selectivity parameters and generates a corresponding `FLBRP` object. Can use a range of functional forms.

Usage

```
## S4 method for signature 'FLPar'
lhEql(
  params,
  growth = FLife::vonB,
  m = "gislason",
  sr = "bevholt",
  mat = logistic,
  sel = dnormal,
  range = c(min = 0, max = 40, minfbar = 1, maxfbar = 40, plusgroup = 40),
  spwn = 0,
  fish = 0.5,
  midyear = 0.5,
  ...
)
```

Arguments

`params`	an `FLPar` object with life history parameters
`growth`	function that takes an `FLPar` object with parameters, by default `vonB`
`m`	`character` takes the natural mortality model name, by default gislason
`sr`	`character` value, "bevholt" by default
`mat`	function that takes an `FLPar` object with parameters, by default `logistic`
`sel`	function that takes an `FLPar` object with parameters, by default `dnormal`
`range`	`numeric` with age range by default from 0 to 40
`spwn`	`numeric` give propotion of year when spawning occurs, by default is params["a50"]-floor(params["a50"])
`fish`	`numeric` give propotion of year when fishing occurs, by default 0.5
`midyear`	when growth measured, default 0.5
`...`	any other arguments
`units`	`character` for vectors in `FLBRP` returned by method

Value

`FLBRP` object

See Also

`lhPar`, `lhRef`

`vonB` `lorenzen` `sigmoid`

Examples

```
## Not run:
data(teleost)
alb=teleost[,"Thunnus alalunga"]
eql=lhEql(lhPar(alb))

## End(Not run)
```

`lhPar` *Generates life history parameters*

Description

Uses life history theory to derive parameters for biological relationships, i.e. or growth, maturity, natural mortality. Selectivity by default is set so age at peak selectivity is the same as age at 50% mature (a50) As a minimum all ‘lhPar‘ requires is ‘linf‘ the asymptotic length of the von Bertalannfy growth equation.

Uses life history theory to derive parameters for biological relationships, i.e. or growth, maturity, natural mortality. Selectivity by default is set so age at peak selectivity is the same as age at 50% mature (a50) As a minimum all ‘lhPar‘ requires is ‘linf‘ the asymptotic length of the von Bertalannfy growth equation.

Usage

```
lhPar(
  ...,
  m = list(model = "gislason", params = c(m1 = 0.55, m2 = -1.61, m3 = 1.44)),
  k = function(params, a = 3.15, b = -0.64) a * params["linf"]^b,
  t0 = function(params, a = -0.3922, b = -0.2752, c = -1.038) -exp(a - b *
    log(params$linf) %-% (c * log(params$k))),
  l50 = function(params, a = 0.72, b = 0.93) a * params["linf"]^b
)

lhStk(
  ...,
  k = function(params, a = 3.15, b = -0.64) a * params["linf"]^b,
  t0 = function(params, a = -0.3922, b = -0.2752, c = -1.038) -exp(a - b *
    log(params$linf) %-% (c * log(params$k))),
  l50 = function(params, a = 0.72, b = 0.93) a * params["linf"]^b,
```

```
   gowth = vonB,
   mat = logistic,
   sel = dnormal,
   sr = "bevholt",
   m = list(model = "gislason", params = c(m1 = 0.55, m2 = -1.61, m3 = 1.44)),
  fmult = function(x) refpts(x)["msy", "harvest"] %*% FLQuant(seq(0, 2, length.out =
    100)),
   range = c(min = 0, max = 40, minfbar = 1, maxfbar = 40, plusgroup = 40),
   spwn = 0,
   fish = 0.5,
   midyear = 0.5
 )
```

Arguments

`t0`	of von Bertalanffy. This is a default that isnt normally derived from life history theory, as are the following args.
`sr`	obsolete now replaced by sel3
`params`	`FLPar` object with parameters for life history equations and selection pattern. Need Linfinity to estimate other parameters, if any other parameters supplied in code then these are not provided by the algorithm
`a`	coefficient of length weight relationship
`b`	exponent of length weight relationship
`ato95`	age at which 95% of fish are mature, offset to age at which 50% are mature
`s`	steepness of stock recruitment relationship
`v`	virgin biomass
`sel1`	selectivity-at-age parameter for double normal, age at maximum selectivity by default set to same as age at 100% mature
`sel2`	selectivity-at-age parameter for double normal, standard deviation of lefthand limb of double normal, by default 5
`sel3`	selectivity-at-age parameter for double normal, standard deviation of righthand limb of double normal, by default 5000
`sl`	obsolete now replaced by sel2
`m1`	m-at-age parameter by default for Gislason empirical relationship
`m2`	m-at-age parameter, by default for Gislason empirical relationship
`m3`	m-at-age parameter, by default for Gislason empirical relationship

Value

object of class `FLPar` with missing parameters calculated from life history theory

object of class `FLPar` with missing parameters calculated from life history theory

See Also

`loptAge`, `lhRef`, `lhPar`, `lhEql`

`loptAge`, `lhRef`, `lhPar`, `lhEql`

Examples

```
## Not run:
#COMPARE with output of FLife::lhPar

x <- as(lhpar(linf=100), 'list')
x <- x[sort(names(x))]
y <- as(lhPar(FLPar(linf=100)), 'list')
y <- y[sort(names(y))]

all.equal(x,y)

for(i in seq(length(x)))
   cat(names(x[i]), ":", unlist(x[i]), "-", names(y[i]), ":", unlist(y[i]), "\n")

 # CALL with iters
 lhpar(FLPar(linf=100), v=rnorm(100, 300, 200))

 lhPar(FLPar(linf=rnorm(100, 80, 10)))
 lhPar(FLPar(linf=100, v=rnorm(100, 300, 200)))
 lhPar(FLPar(linf=100), FLPar(v=rnorm(100, 300, 200)))
 lhPar(FLPar(linf=100, v=rnorm(100, 300, 200)), t0=-1, data.frame(a=1,b=7))

 attributes(lhpar(FLPar(linf=100), v=rnorm(100, 300, 200)))$mmodel

## End(Not run)

## Not run:
```

`lhRef` *Reference points based on life histories*

Description

`lhRef` calculates a variety of reference points i.e. population growth rate at small population sizs (r), and at B~MSY (rc), ratio of virgin biomass to B~MSY (sk), life time reproductive output (srp0) and reproductive output at B~MSY (sprmsy)

Usage

```
lhRef(
  params,
 m = function(length, params) exp(0.55) * (length^-1.61) %*% (params["linf"]^1.44)
    %*% params["k"],
  sr = "bevholt",
  range = c(min = 0, max = 40, minfbar = 1, maxfbar = 40, plusgroup = 40),
  what = c("r", "rc", "msy", "lopt", "sk", "spr0", "sprmsy"),
  msy = "msy"
)
```

Arguments

`params`	`FLPar`
`m`	function for natural mortality
`sr`	`character` with stock recruitment relationship, e.g. "bevholt","ricker",...
`range`	ages used i.e. c(min=0,max=40,minfbar=1,maxfbar=40,plusgroup=40)
`what`	quantities to calculate "r","lopt","rc","sk","spr0","sprmsy"
`msy`	`character` with "msy", "f0.1", ...

Value

object of class `FLPar` with reference points, i.e r, rc, sk, lopt,

See Also

`lhPar`, `lhEql`

Examples

```
## Not run:
library(FLBRP)
params=FLPar(linf=100,t0=0,k=.4)
params=lhPar(params)
lhRef(params)

## End(Not run)
```

`logistic,FLQuant,FLPar-method`

logistic

Description

logistic function

Usage

```
## S4 method for signature 'FLQuant,FLPar'
logistic(age, params, ...)
```

Arguments

`age`	`FLQuant` with ages
`params`	`FLPar` with parameters a50,ato95 and asym
`...`	other arguments

See Also

`gompertz`

Examples

```
## Not run:
params=FLPar(a50=4,ato95=1,asym=1.0)
age=FLQuant(1:10,dimnames=list(age=1:10))
logistic(age,params)

## End(Not run)
```

`lopt,FLPar-method` *Lopt*

Description

Lopt, the length at which a cohort achives its maximum biomass, can be used as a reference point to identify growth over- or underfishing. Since taking fish below or above this size results in potential loss of yield. The total biomass of a cohort changes through time as a result of gains due to an increase in mean size-at-age and losses due to natural mortality. Lopt can therefore be estimated from the natural mortality and weight-at-age vectors.

Usage

```
## S4 method for signature 'FLPar'
lopt(
  params,
 m = function(length, params) exp(0.55) * (length^-1.61) %*% (params["linf"]^1.44)
    %*% params["k"],
  growth = FLife::vonB,
  ...
)
```

Arguments

`params`	an `FLPar` object with parameter values for the natural mortality and growth functions, and the exponent b of the length/weight relationship.
`m`	natural mortality function, by default Gislason
`growth`	length or weight-at-age function, by default von Bertalanffy
`...`	any other arguments

Details

Lopt is a function of growth and natural mortality-at-age and there are several approximations such as $2/3L_\infty$ and $L_\infty\frac{3}{3+k/m}$. If the life history parameters and relationships are known then L_opt can be found by finding the time (t) and hence length at which the maximum biomass is achieved i.e. $L(T)^a e^{\int_0^T m(t)}$ where $m(t)$ can be found from the relationship of mortality at length using the relationship of Gislason, assuming the von Bertalanffy growth curve.

Value

`FLPar` with L_opt the length at which a cohort achives its maximum biomass

See Also

`gislason`, `vonB`, `lhRef`, `lhPar`, `lhEql`,

Examples

```
## Not run:
params=lhPar(FLPar(linf=100,k=0.1,t0=-0.1,b=3))
lopt(params)

## End(Not run)
```

`loptAge,FLPar-method` *Age at maximum biomass*

Description

Finds length at maximum biomass

Usage

```
## S4 method for signature 'FLPar'
loptAge(
  params,
 m = function(length, params) params["m1"] %*% (exp(log(length) %*% params["m2"])),
  growth = vonB,
  ...
)
```

Arguments

`params`	FLPar
`m`	A function, i.e. gislason
`growth`	A function, i.e. vonB
`...`	any other arguments

Details

There are several ways to calculate L_{opt}, i.e. i) $2/3^{rds} L_{\infty}$ ii) $L_{\infty}\frac{3}{3+k/m}$ iii) by maximising the biomass of iv) from an FLBRP object by fishing at F=0 and finding age where biomass is a maximum

Value

`FLPar` with length at maximum biomass of a cohort

See Also

`loptAge`, `lhRef`, `lhPar`, `lhEql`,

Examples

```
## Not run:
params=lhPar(FLPar(linf=100))
loptAge(params)

## End(Not run)
```

`lorenzen,FLQuant,FLPar-method`

lorenzen

Description

Lorenzen natural mortality relationship estimate M as a function of weight. M=a*wt^b;

Usage

```
## S4 method for signature 'FLQuant,FLPar'
lorenzen(wt, params, ...)
```

Arguments

`wt`	mass at which M is to be predicted
`params`	an `FLPar` with two values; i.e. a equal to M at unit mass and b a power term; defaults are a=0.3 and b=-0.288
`...`	any other arguments

See Also

`gislason`

Examples

```
## Not run:
mass=FLQuant(c( 1.90, 4.23, 7.47,11.48,16.04,20.96,26.07,31.22,
               36.28,41.17,45.83,50.20,54.27,58.03,61.48,64.62),
             dimnames=list(age=1:16))
lorenzen(mass)

## End(Not run)
```

`matdd,FLQuant,FLPar-method` *matdd*

Description

Logistic ogive for proportion mature-at-age, modified to explicitly included maturity as a function of numbers in a cohort, i.e. density dependence

Usage

```
## S4 method for signature 'FLQuant,FLPar'
matdd(age, params, scale, k = 1, flagAge = TRUE)
```

Arguments

`age`	ages
`params`	an `FLPar` with two values; i.e. a equal to M at unit mass and b a power term; defaults are a=0.3 and b=-0.288
`scale`	reference
`k`	rate of change in density dependence
`flagAge`	default is FALSE, i.e. density dependence is based on length ratherv than age
`...`	other arguments, such as scale, e.g. stock numbers now relative to a reference level, e.g. at virgin biomass and k steepness of relationship

Details

The relationship can be explained by population density, since as fish grow they also die and so there is potentially less competition for resources between larger and older fish. Density dependence can be modelled by a logistic function, a sigmoid curve (or S shaped) curve, with equation

f(x)=L/(1+exp(-k(x-x0)))

where e is the natural logarithm base (also known as Euler's number), x0 is the x-value of the sigmoid's midpoint, L is the curve's maximum value, and k the steepness of the curve.

Combining the two functions gives

O=aL/(1+exp(-k(n-ref)))*wt^b;

See Also

logistic,mdd

Examples

```
## Not run:
library(FLBRP)
library(FLife)

data(teleost)
par=teleost[,"Hucho hucho"]
par=lhPar(par)
hutchen=lhEql(par)

scale=stock.n(hutchen)[,25]%*%stock.wt(hutchen)
scale=(stock.n(hutchen)%*%stock.wt(hutchen)%-%scale)%/%scale

mat=matdd(ages(scale),par,scale,k=.5)

ggplot(as.data.frame(mat))+
   geom_line(aes(age,data,col=factor(year)))+
   theme(legend.position="none")+
   scale_x_continuous(limits=c(0,15))


## End(Not run)
```

`mdd,FLQuant,FLPar-method`
mdd

Description

Lorenzen natural mortality relationship where M is a function of weight, modified to explicitly included M as a function of numbers in a cohort, i.e. density dependence

Usage

```
## S4 method for signature 'FLQuant,FLPar'
mdd(object, params, scale, k = 1, m = gislason)
```

Arguments

`object`	mass at which M is to be predicted
`params`	an `FLPar` with two values; i.e. a equal to M at unit mass and b a power term; defaults are a=0.3 and b=-0.288
`scale`	reference
`k`	rate of change in density dependence

`m`	function with mortality model, by default gisalson
`...`	other arguments, such as scale, e.g. stock numbers now relative to a reference level, e.g. at virgin biomass and k steepness of relationship

Details

@details

The Lorenzen natural mortality relationship is a function of mass-at-age i.e. M=a*wt^b

The relationship can be explained by population density, since as fish grow they also die and so there is potentially less competition for resources between larger and older fish. Density dependence can be modelled by a logistic function, a sigmoid curve (or S shaped) curve, with equation

f(x)=L/(1+exp(-k(x-x0)))

where e is the natural logarithm base (also known as Euler's number), x0 is the x-value of the sigmoid's midpoint, L is the curve's maximum value, and k the steepness of the curve.

Combining the two functions gives

M=aL/(1+exp(-k(n-ref)))*wt^b;

See Also

lorenzen

Examples

```
## Not run:
library(FLBRP)
library(FLife)

data(teleost)
par=teleost[,"Hucho hucho"]
par=lhPar(par)
hutchen=lhEql(par)

scale=stock.n(hutchen)[,25]%*%stock.wt(hutchen)
scale=(stock.n(hutchen)%*%stock.wt(hutchen)%-%scale)%/%scale

m=mdd(wt2len(stock.wt(hutchen),par),params=par,scale,k=.9)

ggplot(as.data.frame(m))+
   geom_line(aes(age,data,col=factor(year)))+
   theme(legend.position="none")+
   scale_x_continuous(limits=c(0,15))

m=mdd(stock.wt(hutchen),params=FLPar(m1=3,m2=-0.288),scale,k=1.2,m=lorenzen)

 library(FLife)

## End(Not run)
```

`r,FLPar,missing-method` *Population growth rate*

Description

Estimates population growth rate for a Leslie matrix

Usage

```
## S4 method for signature 'FLPar,missing'
r(m, fec, ...)
```

Arguments

`m`	`FLPar`
`fec`	`missing`
`...`	any other arguments

Value

`FLPar` with growth rate a small population size

See Also

`leslie`, `lhRef`

Examples

```
## Not run:
library(popbio)
eql=lhEql(lhPar(FLPar(linf=100)))
L=leslie(eql)
lambda(L[drop=TRUE])

## End(Not run)
```

`richards,FLQuant,FLPar-method` *Richards growth curve*

Description

Richards growth equation

Usage

```
## S4 method for signature 'FLQuant,FLPar'
richards(age, params, ...)
```

Arguments

`age`	FLQuant, FLPar or numeric object with values corresponding to ages
`params`	`FLPar` object with parameters `linf`, `k` and `t0`
`...`	other arguments

Value

Returns an object of same class as `age` e.g. `FLQuant`

See Also

`vonB`, `gompertz`, `gascuel`

Examples

```
## Not run:
age=FLQuant(1:10,dimnames=list(age=1:10))
len=richards(age,FLPar(linf=100,k=.4,b=.1,m=2))

## End(Not run)
```

`rlnoise` *Random noise with different frequencies*

Description

A noise generator for lognormal errors

Usage

```
## S4 method for signature 'numeric,FLQuant'
rlnoise(
  n = n,
  len = len,
  sd = 0.3,
  b = 0,
  burn = 0,
  trunc = 0,
  what = c("year", "cohort", "age")
)
```

Arguments

`n`	number of iterations
`len`	an `FLQuant`
`sd`	standard error for simulated series
`b`	autocorrelation parameter a real number in [0,1]
`burn`	gets rid of 1st values i series
`trunc`	get rid of values > abs(trunc)
`what`	returns time series for year, cohort or age"
`...`	any other parameters

Value

A `FLQuant` with autocorrelation equal to B.

References

Ranta and Kaitala 2001 Proc. R. Soc. vt = b * vt-1 + s * sqrt(1 - b^2) s is normally distributed random variable with mean = 0 b is the autocorrelation parameter

Examples

```
## Not run:
flq=FLQuant(1:100)
white <- rnoise(1000,flq,sd=.3,b=0)
plot(white)
acf(white)

red <- rlnoise(1000,flq,sd=.3,b=0.7)
plot(red)
acf(red)

data(ple4)
res=rnoise(1000,log(flq),sd=.3,b=0)

ggplot()+
```

```
geom_point(aes(year,age,size= data),
           data=subset(as.data.frame(res),data>0))+
geom_point(aes(year,age,size=-data),
           data=subset(as.data.frame(res),data<=0),colour="red")+
scale_size_area(max_size=4, guide="none")+
facet_wrap(~iter)

res=rlnoise(4,log(m(ple4)),burn=10,b=0.9,cohort=TRUE)
ggplot()+
geom_point(aes(year,age,size= data),
         data=subset(as.data.frame(res),data>0))+
geom_point(aes(year,age,size=-data),
         data=subset(as.data.frame(res),data<=0),colour="red")+
scale_size_area(max_size=4, guide="none")+
facet_wrap(~iter)


## End(Not run)
```

`rnoise,numeric,FLQuant-method`

Random noise with different frequencies

Description

A noise generator

Usage

```
## S4 method for signature 'numeric,FLQuant'
rnoise(
  n = n,
  len = len,
  sd = 0.3,
  b = 0,
  burn = 0,
  trunc = 0,
  what = c("year", "cohort", "age")
)
```

Arguments

`n`	number of iterations
`len`	an `FLQuant`
`sd`	standard error for simulated series
`b`	autocorrelation parameter a real number in [0,1]
`burn`	gets rid of 1st values i series

`trunc`	get rid of values > abs(trunc)
`what`	returns time series for year, cohort or age"
`...`	any other parameters

Value

A `FLQuant` with autocorrelation equal to B.

References

Ranta and Kaitala 2001 Proc. R. Soc. vt = b * vt-1 + s * sqrt(1 - b^2) s is normally distributed random variable with mean = 0 b is the autocorrelation parameter

Examples

```
## Not run:
flq=FLQuant(1:100)
white <- rnoise(1000,flq,sd=.3,b=0)
plot(white)
acf(white)

red <- rnoise(1000,flq,sd=.3,b=0.7)
plot(red)
acf(red)

data(ple4)
res=rnoise(1000,flq,sd=.3,b=0)

ggplot()+
geom_point(aes(year,age,size= data),
            data=subset(as.data.frame(res),data>0))+
geom_point(aes(year,age,size=-data),
            data=subset(as.data.frame(res),data<=0),colour="red")+
scale_size_area(max_size=4, guide="none")+
facet_wrap(~iter)

res=rnoise(4,m(ple4),burn=10,b=0.9,cohort=TRUE)
ggplot()+
geom_point(aes(year,age,size= data),
          data=subset(as.data.frame(res),data>0))+
geom_point(aes(year,age,size=-data),
          data=subset(as.data.frame(res),data<=0),colour="red")+
scale_size_area(max_size=4, guide="none")+
facet_wrap(~iter)


## End(Not run)
```

`rod,FLQuant-method` *rod*

Description

Regime shifts Evidence for regime shifts are explored using a a sequential t-test algorithm (STARS; *rodionov2004sequential*) as modified by Szuwalski et al., (submitted)

Usage

```
## S4 method for signature 'FLQuant'
rod(object, ...)
```

Arguments

`object`	an object of class `FLQuant`
`...`	any other arguments

Details

Returns a data.frame

Examples

```
## Not run:
 object=rlnorm(1,FLQuant(0,dimnames=list(year=1:30)),.3)
 pg=rod(object)
 ggplot(object) +
     geom_polygon(aes(year,data,group=regime),
         fill="lavender",col="blue",
         lwd=.25,data=pg,alpha=.75)+
     geom_point(aes(year,data))+
     geom_line(aes(year,data))

## End(Not run)
```

`roff` *Natural mortality*

Description

Methods to provide estimates of natural mortality based on growth and reproduction parameters

Usage

```
roff(params, ...)
```

Arguments

`params`	`FLPar`
`...`	any other arguments

Details

Natural Mortality For larger species securing sufficient food to maintain a fast growth rate may entail exposure to a higher natural mortality @gislason2008does. While many small demersal species seem to be partly protected against predation by hiding, cryptic behaviour, being flat or by possessing spines have the lowest rates of natural mortality @griffiths2007natural. Hence, at a given length individuals belonging to species with a high

$$L_\infty$$

may generally be exposed to a higher M than individuals belonging to species with a low

$$L_\infty$$

.

$$log(M) = 0.55 - 1.61log(L) + 1.44log(L_\infty) + log(k)$$

Functional forms

Many estimators have been propose for M, based on growth and reproduction,

Age at maturity

$$M = \frac{1.521}{a_{50}^{0.72}} - 0.155$$

$$M = \frac{1.65}{a_{50}}$$

Growth

$$M = 1.5k$$

$$M = 1.406W_\infty^{-0.096}k^{0.78}$$

$$M = 1.0661L_\infty^{-0.1172}k^{0.5092}$$

Growth and length at maturity

$$M = 3kL_\infty\frac{(1-\frac{L_{50}}{L_\infty})}{L_{50}}$$

$$M = \frac{\beta k}{e^{k(a_{50}-t_0)}-1}$$

Varing by length, weight or age

Value

returns an object of `FLQuant`

See Also

gislason, lorenzen

Examples

```
## Not run:
params=FLPar(FLPar(linf=120,k=.15,t0=-0.1,l50=60,a=0.0001,b=3))
age=FLQuant(1:10,dimnames=list(age=1:10))


roff(params)
rikhter(params)
rikhter2(params)
griffiths(params)
djababli(params)
jensen(params)
jensen2(params)

## End(Not run)
```

`sigmoid,FLQuant,FLPar-method`

Sigmoid ogive

Description

sigmoid ogive

Usage

```
## S4 method for signature 'FLQuant,FLPar'
sigmoid(age, params, ...)
```

Arguments

`age`	FLQuant, FLPar or numeric with ages
`params`	`FLPar` with age at whick probability is 0.5 (`a50`) and offset to 0.95 (`ato95`)
`...`	any other arguments

Details

The sigmoid ogive is an S-shaped or sigmoid curve or sigmoidal functions, Verhulst hypothesizes that small populations increase geometrically, because the supply of resources exceeds demand. Then, as supply and demand balance, population growth is constant. Finally, as demand exceeds supply, population growth decreases at the same rate that it had increased. Verhulst describes this process with an equation that enables him to predict when a population will reach any given size (see Verhulst's Figure):

Value

returns an object of same type as age e.g. `FLQuant`

See Also

`knife`, `dnormal`, `logistic`

`dnormal`,`knife`

Examples

```
## Not run:
params=FLPar(a50=4,ato95=1)
age=FLQuant(1:10,dimnames=list(age=1:10))
sigmoid(age,params)

## End(Not run)
```

---

sv,FLPar,character-method

Calculates steepness and virgin biomass

Description

Calculates steepness and virgin biomass given a and b for a Beverton and Holt SRR

Usage

```
## S4 method for signature 'FLPar,character'
sv(x, model, spr0 = NA)
```

Arguments

`x`	`FLPar` with a and b
`model`	`character` with name of stock recruitment relationship, by default "bevholt"
`spr0`	spawner per recruit at F=0
`...`	any other arguments

Value

`FLPar` with values for steepness (s) and virgin biomass (v)

Examples

```
## Not run:
#bug
params=FLPar(a=37.8,b=8.93)
sv(params,"bevholt",.4)

## End(Not run)
```

`teleost` *teleost*

Description

A dataset containing life history parameters for a range of teleost species

Format

A data frame with 139 rows (i.e. species) and 13 variables:

species,genus,family,order,class

linf asymptotic length of the Von Bertalanffy growth equation

k rate at which length reaches linf

t0 adjusts the growth equation for the initial size at birth

l50 length at which 50% are mature

a scaling factor of the length weight relationship

b exponent of the length weight relationship

temp temperature

habit e.g. pelagic or demeresal

See Also

`cas`

`vonB,FLQuant,FLPar-method` *von Bertalanffy growth curve*

Description

Von Bertalanffy growth equation

Usage

```
## S4 method for signature 'FLQuant,FLPar'
vonB(age, params, ...)
```

Arguments

`age`	FLQuant, FLPar or numeric object with values corresponding to ages
`params`	`FLPar` object with parameters `linf`, `k` and `t0`
`...`	other arguments

Value

Returns an object of same class as age e.g. `FLQuant`

See Also

`gompertz`, `gascuel`, `richards`

Examples

```
## Not run:
params=FLPar(linf=100,t0=0,k=.4)
age=FLQuant(1:10,dimnames=list(age=1:10))
len=vonB(age,params)

#inverse growth curve
vonB(params=params,length=len)

## End(Not run)
```

`wklife` *wklife*

Description

A dataset containing life history parameters for a range of teleost species

Format

A data frame with 15 rows (i.e. species) and 13 variables:

species

name

area

stock

sex

a scaling factor of the length weight relationship

b exponent of the length weight relationship

linf asymptotic length of the Von Bertalanffy growth equation

k rate at which length reaches linf

t0 adjusts the growth equation for the initial size at birth

lmax asymptotic length of the Von Bertalanffy growth equation

l50 length at which 50% are mature

a50 age at which 50% are mature

See Also

`teleost`

`wt2len,FLQuant,FLPar-method`

Mass to length conversion

Description

converts weight to length

Usage

```
## S4 method for signature 'FLQuant,FLPar'
wt2len(wt, params, ...)
```

Arguments

`wt`	FLQuant, FLPar or numeric with length
`params`	`FLPar` with a and b, i.e. condition and scaling factors
`...`	any other arguments

Value

Returns an object of same class as `wt` e.g. `FLQuant`

See Also

`len2wt`

Examples

```
## Not run:
params=FLPar(a=0.1,b=3)
wt2len(FLQuant(10),params)

## End(Not run)
```

Index