Package ‘fishdynr’

January 21, 2016

Type Package

Title fishdynr

Version 1.0

Date 2014-12-26

Author Marc Taylor

Maintainer Marc Taylor <ningkiling@gmail.com>

Description Fisheries science related population dynamics models.

License MIT + file LICENSE

URL https://github.com/marchtaylor/fishdynr

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

cohortSim *Simulation of a cohort*

Description

cohortSim simulates a single cohort

Usage

```
cohortSim(params, t_incr = 1)
```

Arguments

params	List of parameters to use in function
t_incr	Value to use a time increment (in years). Defaults to 1.

Details

params list should contain the following parameters:

- species Species name
- growthFun Name of growth function (e.g. "growth_VB" is the von Bertalanffy growth function)
- amax Maximum age
- LWa Length-weight relationship parameter a (weight~a*length^b)
- LWb Length-weight relationship parameter b (weight~a*length^b)
- M Natural mortality
- F Fishing mortality
- N0 Number of individuals at time 0
- matFun Name of maturity function (e.g. "pmat_w" is a logistic function that includes width, w, of quantiles)
- selectFun Function to use for gear selection. Determines lengths vulnerable to fishing mortality (e.g. "gillnet" and "knife_edge" functions).
- ... Other parameters for growth, maturity, and selectivity functions.

Value

A list

Examples

```
data(tilapia)
res <- cohortSim(tilapia, t_incr=.1)
plot(pcap ~ Lt, res, t="l")
plot(Lt ~ t, res, t="l")
plot(Wt ~ t, res, t="l")

plot(Bt ~ t, res, t="l")
lines(SBt ~ t, res, col=2)
```

```
plot(Bt ~ Lt, res, t="l")
lines(SBt ~ Lt, res, col=2)

plot(Yt ~ t, res, t="l")

plot(Nt ~ t, res, t="l", log="y")
lines(Nt.noF ~ t, res, col=2, lty=2)
```

`dt_growth_soVB` *Discrete seasonally oscillating von Bertalanffy growth function*

Description

`dt_growth_soVB` calculates length at time 2, given length at time 1 according to the seasonally oscillationg von Bertalanffy growth function

Usage

```
dt_growth_soVB(Linf, K, ts, C, L1, t1, t2)
```

Arguments

`Linf`	Infinite length
`K`	growth constant
`ts`	summer point. Time of year (between 0 and 1) when growth oscillation cycle begins (sine wave term becomes positive). Note that this definition differs from some interpretations of the model (see Somers 1998)
`C`	oscillation strength. Varies between 0 and 1
`L1`	Length at t1
`t1`	time 1
`t2`	time 2

References

Somers, I. F. (1988). On a seasonally oscillating growth function. Fishbyte, 6(1), 8-11.

Examples

```
# growth parameters
Linf = 100
K = 0.5
ts = 0.5
C = 0.75

# create dataframe
df <- data.frame(t = seq(0,5,0.2), L=NaN)
df$L[1] <- 10 # intial size

# run iterative model
```

```
for(i in 2:nrow(df)){
  df$L[i] <- dt_growth_soVB(
    Linf, K, ts, C,
    L1=df$L[i-1],
    t1=df$t[i-1],
    t2=df$t[i]
  )
}

# plot result
plot(L ~ t, df)
arrows(
  x0 = df$t[-nrow(df)], y0 = df$L[-nrow(df)],
  x1 = df$t[-1], y1 = df$L[-1],
  col = 8, length = 0.15
)
```

fecW_LM *Fecundity and function of weight*

Description

fecW_LM is a linear model describing fecundity (number of eggs) as a function of individual weight

Usage

```
fecW_LM(fec_a = 0, fec_b = 1, Wt = 500)
```

Arguments

fec_a	intercept
fec_b	slope
Wt	individual weight (from cohortSim function)

Value

Number of eggs per individual

Examples

```
data(tilapia)
res <- cohortSim(tilapia, t_incr=0.1)
plot(Neggst ~ Wt, res) # Number of eggs as a function of ind. weight
plot(res$t, res$Neggst*res$pmat*res$Nt) # total population fecundity by age
```

gillnet *gillnet selection*

Description

gillnet describes gillnet selection following Millar and Holst (1997). Possible selectivity distributions include normal (fixed width) and lognormal. [Note: others will be added]

Usage

```
gillnet(Lt, mesh_size, mesh_size1, select_dist, select_p1, select_p2)
```

Arguments

Lt	body size
mesh_size	mesh size
mesh_size1	smallest reference mesh size
select_dist	selectivity type ("normal_fixed", "lognormal")
select_p1	selectivity function parameter 1 (see Millar and Holst 1997)
select_p2	selectivity function parameter 2 (see Millar and Holst 1997)

References

Millar, R. B., & Holst, R. (1997). Estimation of gillnet and hook selectivity using log-linear models. ICES Journal of Marine Science: Journal du Conseil, 54(3), 471-477.

Examples

```
data(tilapia)
tilapia$selectFun="gillnet"
mesh_sizes <- c(60, 80, 100, 120)
for(i in seq(mesh_sizes)){
  tilapia$mesh_size <- mesh_sizes[i]
  res <- cohortSim(tilapia, t_incr=0.01)
  if(i == 1) plot(pcap ~ Lt, res, t="n")
  lines(pcap ~ Lt, res, col=i)
}
legend("topleft", legend=mesh_sizes,
  col=seq(mesh_sizes), lty=1,
  title="mesh size [mm]", bty="n"
)
```

`growth_soVB` *seasonally oscillating von Bertalanffy growth function*

Description

`growth_soVB` describes the growth as a function of age (t) using the seasonally oscillating von Bertalanffy growth function (Somers 1988).

Usage

```
growth_soVB(Linf, K, t, t0, ts, C)
```

Arguments

`Linf`	Infinite length
`K`	growth constant
`t`	age
`t0`	(hypothetical) age at length zero
`ts`	summer point. Time of year (between 0 and 1) when growth oscillation cycle begins (sine wave term becomes positive). Note that this definition differs from some interpretations of the model (see Somers 1998)
`C`	oscillation strength. Varies between 0 and 1.

References

Somers, I. F. (1988). On a seasonally oscillating growth function. Fishbyte, 6(1), 8-11.

Examples

```
t <- seq(0,5,0.1)
L <- growth_soVB(Linf=100, K=0.5, t=t, t0=-0.2, ts=0.5, C=0.75)
plot(t, L, t="l")
```

`growth_soVB2` *seasonally oscillating von Bertalanffy growth function (using L0)*

Description

`growth_soVB2` describes the growth as a function of age (t) using the seasonally oscillating von Bertalanffy growth function (Somers 1988). The function differs slightly from `growth_soVB` by using `L0` instead of `t0` to describe the growth function origin.

Usage

```
growth_soVB2(Linf, K, t, L0, ts, C)
```

Arguments

`Linf`	Infinite length
`K`	growth constant
`t`	age
`L0`	(hypothetical) length at time zero
`ts`	summer point. Time of year (between 0 and 1) when growth oscillation cycle begins (sine wave term becomes positive). Note that this definition differs from some interpretations of the model (see Somers 1998)
`C`	oscillation strength. Varies between 0 and 1.

Examples

```
t <- seq(0,5,0.1)
L <- growth_soVB2(Linf=100, K=0.5, t=t, L0=10, ts=0.5, C=0.5)
plot(t, L, t="l", ylim=c(0,110), xaxs="i", yaxs="i")
points(0,10); abline(h=100, lty=2, col=8)
```

`growth_VB` *von Bertalanffy growth function*

Description

`growth_VB` describes the growth as a function of age (t) using the von Bertalanffy growth function

Usage

```
growth_VB(Linf, K, t, t0)
```

Arguments

`Linf`	Infinite length
`K`	growth constant
`t`	age
`t0`	(hypothetical) age at length zero

Examples

```
t <- seq(0,5,0.1)
L <- growth_VB(Linf=100, K=0.5, t=t, t0=-0.2)
plot(t, L, t="l")
```

growth_VB2 *von Bertalanffy growth function (using L0)*

Description

growth_VB2 describes the growth as a function of age (t) using the von Bertalanffy growth function. The function differs slightly from growth_VB by using L0 instead of t0 to describe the growth function origin

Usage

```
growth_VB2(Linf, K, t, L0)
```

Arguments

Linf	Infinite length
K	growth constant
t	age
L0	(hypothetical) length at time zero

Examples

```
t <- seq(0,5,0.1)
L <- growth_VB2(Linf=100, K=0.5, t=t, L0=10)
plot(t, L, t="l", ylim=c(0,110), xaxs="i", yaxs="i")
points(0,10); abline(h=100, lty=2, col=8)
```

knife_edge *knife-edge selection*

Description

knife_edge describes knife-edge selection where probability of capture 100% after a minimum defined size. Provides a rough estimate of trawl-type selectivity.

Usage

```
knife_edge(Lt, knife_edge_size)
```

Arguments

Lt	body size
knife_edge_size	knife edge size. Minimum size at capture.

Examples

```
data(tilapia)
tilapia$selectFun="knife_edge"
knife_edge_sizes <- c(20, 25, 30, 35)
for(i in seq(knife_edge_sizes)){
  tilapia$knife_edge_size <- knife_edge_sizes[i]
  res <- cohortSim(tilapia, t_incr=0.01)
  if(i == 1) plot(pcap ~ Lt, res, t="n")
  lines(pcap ~ Lt, res, col=i)
}
legend("topleft", legend=knife_edge_sizes,
  col=seq(knife_edge_sizes), lty=1,
  title="min. size", bty="n"
)
```

`logisticSelect` *logistic selection*

Description

`logisticSelect` describes knife-edge selection where probability of capture 100% after a minimum defined size. Provides a rough estimate of trawl-type selectivity.

Usage

```
logisticSelect(Lt, L50, wqs)
```

Arguments

`Lt`	body size
`L50`	length at first capture (i.e. where prob. of capture equals 50%)
`wqs`	width of quartiles between 25% and 75% probability of capture

Examples

```
data(tilapia)
tilapia$selectFun="logisticSelect"
tilapia$L50 <- 20
WQS <- c(0, 3, 6)
COL <- rainbow(3)
for(i in seq(WQS)){
  tilapia$wqs <- WQS[i]
  res <- cohortSim(tilapia, t_incr=0.01)
  if(i == 1) plot(pcap ~ Lt, res, t="n")
  lines(pcap ~ Lt, res, col=COL[i])
}
legend("bottomright",
       legend=paste(WQS, c(" ('knife-edge')", "", ""), sep=""),
       col=COL, lty=1,
       title="Width between 0.25 and 0.75 quantiles", bty="n"
)
lines(c(0,tilapia$L50,tilapia$L50), c(0.5,0.5,0), col=1, lty=2)
```

```
text(20,.5,bquote("L"[50]), pos=4, col=1, font=1)
```

`optim.stockSim` *Optimization of fishing mortality*

Description

Sets up an exploration of optimum fishing policy using the function `stockSim` (i.e. optimal time series of fishing mortalities, `Ft`, in order to maximize time series yield, `Yt`). The routine is described in further detail in Walters and Martell (2004).

Usage

```
optim.stockSim(Ft = 0, params, nyears = 100, env_at = 1, env_bt = 1,
  opt = "sum", disc.rate = 0.02)
```

Arguments

`Ft` time series vector for fishing mortality. If a single value, then the function assumes a constant fishing mortality for the entire simulation (default=0)

`params` List of parameters to for the population. Applies a single cohort simulation for the initial population state using `cohortSim` function. See `stockSim` for details.

`nyears` number of years in the simulation

`env_at` time series vector for environmental effects to maximum recruitment (e.g. `srrFecBH_a` in `srrFecBH`) (default=1).

`env_bt` time series vector for environmental effects to maximum recruitment (e.g. `srrFecBH_b` in `srrFecBH`) (default=1).

`opt` what should be optimized ("sum", "sum.log", "sum.disc")

`disc.rate` discount rate (default = 0.02)

References

Walters, C. J., Martell, S. J., 2004. Fisheries ecology and management. Princeton University Press.

Examples

```
set.seed(1)
data(tilapia)
params <- tilapia
params$knife_edge_size <- 20
params$N0 <- 1e9
nyears <- 50
Ft <- rep(0.5, nyears)
set.seed(1)
env_at <- runif(nyears, min=0.5, max=1.5)
env_bt <- rep(1, nyears); env_bt[20:35] <- 0.5
```

```
# Optimization of Ft (will take some time to reach cost function mimimum)
out <- optim(
  par = Ft,  # initial guess
  fn = optim.stockSim,
  params = params,
  nyears = nyears,
  env_at = env_at,
  env_bt = env_bt,
  opt = "sum.log", # optimize the sum of log yield
  method = "L-BFGS-B",
  lower = 0,
  upper = 2,
  control = list(fnscale=-1, trace=4)
)

# optimum Ft series
plot(out$par, t="l")

# optimum Yt series
tmp <- stockSim(params, nyears=nyears, Ft=out$par, env_at=env_at, env_bt=env_bt)
plot(Yt ~ t, tmp, t="l")
sum(tmp$Yt/1e6, na.rm=TRUE)

# optimum yield versus stock biomass
plot(Yt ~ Bt, tmp)
fit <- lm(Yt ~ Bt, tmp)
abline(fit)
summary(fit)

op <- par(mar=c(4,4,1,4))
plot(Bt ~ t, tmp, t="l", ylim=c(0, max(tmp$Bt)))
par(new=TRUE)
plot(Yt ~ t, tmp, col=2, t="l", axes=FALSE, xlab="", ylab="")
axis(4, col=2)
mtext("Yt", side=4, line=3, col=2)
par(op)
```

pmat_w *Probability of maturity logistic function using quartile width*

Description

pmat_w describes the probability of maturity as a function of the size when 50% of the individuals in a population are mature and the width (in size) between 25% and 75% probability of maturity quartiles (after Heino et al. 2002).

Usage

```
pmat_w(Lt, Lmat, wmat)
```

Arguments

`Lt` size for probability of maturity calculation

`Lmat` size at 50% probability of maturity (i.e. "massive maturity")

`wmat` width (in size) between 25% and 75% probability of maturity quartiles

References

Heino M, Dieckmann U, Godo OR (2002) Measuring probabilistic reaction norms for age and size at maturation. Evolution 56: 669-678.

Examples

```
L <- seq(1,20,0.1)
pmat1 <- pmat_w(L, Lmat=10, wmat=5)
pmat2 <- pmat_w(L, Lmat=10, wmat=2)
plot(L, pmat1, t="l", ylab="prob. of maturity")
lines(L, pmat2, lty=2)
legend("bottomright", legend=c("Lmat=10; wmat=5", "Lmat=10; wmat=2"),
 col=1, lty=1:2, bty="n")
```

`srrBH` *Stock-recruitment relationship (Beverton-Holt type)*

Description

`srrBH` describes stock-recruitment relationship as the number of recruits resulting from a given spawning biomass.

Usage

```
srrBH(rmax = 1000, beta = 500, SB = 500)
```

Arguments

`rmax` maximum recruitment level

`beta` parameter describing steepness of relationship. Specifically, `beta` describes the point where the number of spawned eggs results in half the maximum number of recruits, `rmax`

`SB` spawning biomass of the adult population

Value

Number of recruited individuals

References

Beverton, R. J., Holt, S. J., 1957. On the dynamics of exploited fish populations

Examples

```
SB <- seq(0,1e11,,100)
rmax = 2e8
beta = 1e10
Nrecr <- srrBH(rmax, beta, SB)
plot(SB, Nrecr, t="l", ylim=c(0, rmax))
abline(h=rmax, lty=2, col=8)
lines(x=c(0, beta, beta), y=c(rmax/2, rmax/2, 0), lty=2, col=8)
text(x=0, y=rmax*0.95, labels="rmax", col=8, pos=4)
text(x=beta, y=rmax/2, labels="rmax/2", pos=4, col=8)
text(x=beta, y=0, labels="beta", pos=4, col=8)
```

srrFecBH *Stock-recruitment relationship using fecundity (Beverton-Holt type)*

Description

srrFecBH describes stock-recruitment relationship as the number of recruits resulting from a given number of eggs produced by the population.

Usage

```
srrFecBH(srrFecBH_a, srrFecBH_b, neggs)
```

Arguments

srrFecBH_a	long-term mean survival parameter
srrFecBH_b	carrying capacity parameter
neggs	number of eggs spawned

Value

Number of recruited individuals

References

Beverton, R. J., Holt, S. J., 1957. On the dynamics of exploited fish populations

Examples

```
n <- 100
neggs <- seq(0,1e8,,n)
srrFecBH_a = 0.2
srrFecBH_b = 1e7/srrFecBH_a
Nrecr <- srrFecBH(srrFecBH_a, srrFecBH_b, neggs)
plot(neggs, Nrecr, t="l", ylim=c(0, srrFecBH_b*srrFecBH_a))
abline(0, srrFecBH_a, col=8, lty=2)
abline(h=srrFecBH_b*srrFecBH_a, lty=2, col=8)
```

stockSim *Simulation of a stock*

Description

stockSim simulates a stock using variable fishing mortality and stock recruitment relationship

Usage

```
stockSim(params, nyears = 100, Ft = 0, env_at = 1, env_bt = 1)
```

Arguments

params	List of parameters for the population. Applies a single cohort simulation for the initial population state using cohortSim function
nyears	number of years in the simulation
Ft	time series vector for fishing mortality. If a single value, then the function assumes a constant fishing mortality for the entire simulation (default=0)
env_at	time series vector for environmental effects to maximum recruitment (e.g. srrFecBH_a in srrFecBH) (default=1).
env_bt	time series vector for environmental effects to half maximum recruitment parameter (e.g. srrFecBH_b in srrFecBH) (default=1).

Details

params list should contain the following parameters:

- species Species name
- growthFun Name of growth function (e.g. "growth_VB" is the von Bertalanffy growth function)
- amax Maximum age
- LWa Length-weight relationship parameter a (weight~a*length^b)
- LWb Length-weight relationship parameter b (weight~a*length^b)
- M Natural mortality
- F Fishing mortality
- N0 Number of individuals at time 0
- matFun Name of maturity function (e.g. "pmat_w" is a logistic function that includes width, w, of quantiles)
- selectFun Function to use for gear selection. Determines lengths vulnerable to fishing mortality (e.g. "gillnet" and "knife_edge" functions).
- srrFun Stock-recruitment relationship function (e.g. "srrFecBH").
- fec Number of eggs produced per weight [g] of mature female (For use in srrFecFun).
- ... Other parameters for growth, maturity, and selectivity functions.

For fitting an optimal time series of fishing mortalities, Ft, see optim.stockSim (Walters and Martell, 2004).

Value

A list

- Btc matrix. Stock biomass by time (rows) and cohort (columns).
- Ytc matrix. Fishery yield by time (rows) and cohort (columns).
- Bt vector. Stock biomass by time.
- Yt vector. Fishery yield by time.
- Nt vector. Stock size (in numbers) by time.
- Ct vector. Fishery catch (in numbers) by time.

References

Walters, C. J., Martell, S. J., 2004. Fisheries ecology and management. Princeton University Press.

Examples

```
data(tilapia)
params <- tilapia
params$knife_edge_size <- 20
params$N0 <- 1e9
nyears <- 50
Ft <- rep(0.5, nyears)
set.seed(1)
env_at <- runif(nyears, min=0.5, max=1.5)
env_bt <- rep(1, nyears); env_bt[20:35] <- 0.5
tmp <- stockSim(Ft=Ft, params=params, nyears=nyears, env_at=env_at, env_bt=env_bt)
plot(tmp$Bt, t="l")
plot(tmp$Yt, t="l")
sum(tmp$Yt/1e6, na.rm=TRUE)
```

tilapia *Parameter list for Oreochromis niloticus (Nile tilapia)*

Description

The tilapia data set contains a list of parameters used by other functions in the fishdynr package.

- species. Species name
- growthFun. Name of growth function ("growth_VB" is von Bertalanffy growth function)
- K. Growth constant (for use in von Bertalanffy growth function))
- Linf. Infinite length (for use in von Bertalanffy growth function)
- t0. (hypothetical) time when length equals zero (for use in von Bertalanffy growth function)
- amax. Maximum age
- LWa. Length-weight relationship parameter a (weight~a*length^b)
- LWb. Length-weight relationship parameter b (weight~a*length^b)
- M. Natural mortality
- F. Fishing mortality

- N0. Number of individuals at time 0
- matFun. Name of maturity function ("pmat_w" is a logistic function that includes width, w, of quantiles)
- Lmat. Length at maturity (i.e. where probability of being mature is 50 (for use in "pmat_w" function)
- wmat. Width of length between 25 of transition from immature to mature (for use in "pmat_w" function)
- fec. Number of eggs produced per weight [g] of mature female
- selectFun. Function to use for gear selection. Determines lengths vulnerable to fishing mortality.
- select_p1. Parameter 1 used in "gillnet" selectivity function
- select_p2. Parameter 2 used in "gillnet" selectivity function
- mesh_size. Mesh size [streched length in mm] used in "gillnet" selectivity function
- mesh_size1. Mesh size of smallest reference net used in fitting "gillnet" selectivity function
- select_dist. Distribution type used in "gillnet" selectivity function
- knife_edge_size. Minimum length selected by "knife_edge" selectivity function
- srrFun. Function used for stock-recruitment relationship
- rmax. Maximum recruitment parameter used in "srrBH" stock-recruitment (Beverton-Holt)
- beta. "beta" parameter used in "srrBH" stock-recruitment (Beverton-Holt)

Usage

```
data(tilapia)
```

Format

A list containing parameters used on other `fishdynr` functions

Examples

```
data(tilapia)
res <- cohortSim(tilapia, t_incr=0.1)
plot(Lt ~ t, res, t="l")
```

ypr *Yield-per-recruit*

Description

Sets up an yield-per-recruit exploration by adjusting fishing mortality and selectivity using the `cohortSim` function

Usage

```
ypr(params, adj.params)
```

Arguments

`params` List of parameters for the population. Applies a single cohort simulation for the initial population state using `cohortSim` function

`adj.params` List of 2 parameters combinations to use for yield per recruit analysis. In the case of trawl-type selectivity, these should be levels for F and knife-edge length at capture. For gillnet-type selectivity, these should be levels for F and mesh_size.

Value

A list of parameter combinations used in the yield-per-recruit exploration, plus matrix output for yield (`Y`) and spawning stock biomass (`SB`) given the parameter combinations (`adj.params.comb`).

Examples

```
# Trawl-type
data(tilapia)
n <- 30
adj.params <- list(F=seq(0,3,,n), knife_edge_size=seq(0,tilapia$Linf,,n))
res <- ypr(params=tilapia, adj.params)
pal <- colorRampPalette(c(
 rgb(1,0.5,0.5), rgb(1,1,0.5), rgb(0.5,1,1), rgb(0.5,0.5,1)
))
op <- par(mfcol=c(1,2))
# Yield
image(x=res$F, y=res$knife_edge, z=res$Y, col=pal(100))
contour(x=res$F, y=res$knife_edge, z=res$Y, add=TRUE)
mtext("Yield", line=0.5, side=3)
# Relative spawning biomass
SB_F0 <- res$SB[which(res$F==0),1]
image(x=res$F, y=res$knife_edge, z=res$SB/SB_F0, col=pal(100))
contour(x=res$F, y=res$knife_edge, z=res$SB/SB_F0, add=TRUE)
mtext("Rel. Spawning Biomass", line=0.5, side=3)
par(op)


# Gillnet-type
data(tilapia)
tilapia$selectFun <- "gillnet"
n <- 30
adj.params <- list(F=seq(0,3,,n), mesh_size=seq(60,160,,n))
res <- ypr(params=tilapia, adj.params)
pal <- colorRampPalette(c(
 rgb(1,0.5,0.5), rgb(1,1,0.5), rgb(0.5,1,1), rgb(0.5,0.5,1)
))
op <- par(mfcol=c(1,2))
# Yield
image(x=res$F, y=res$mesh_size, z=res$Y, col=pal(100))
contour(x=res$F, y=res$mesh_size, z=res$Y, add=TRUE)
mtext("Yield", line=0.5, side=3)
# Relative spawning biomass
SB_F0 <- res$SB[which(res$F==0),1]
image(x=res$F, y=res$mesh_size, z=res$SB/SB_F0, col=pal(100))
contour(x=res$F, y=res$mesh_size, z=res$SB/SB_F0, add=TRUE)
mtext("Rel. Spawning Biomass", line=0.5, side=3)
```

```
par(op)
```

Index