

# Package ‘lhmixr’

May 5, 2017

**Title** Fits Sex-Specific Life History Models With Missing Classifications

**Version** 0.0.0.9000

**Author** Coilin Minto, John Hinde, Rui Coelho

**Maintainer** Coilin Minto <coilin.minto@gmit.ie>

**Description** lhmixr fits sex-specific life-history models for fish and other taxa where some of the individuals have unknown sex.

**Depends** R (>= 3.2.0)

**License** GPL (>= 3)

**URL** <https://github.com/mintoc/lhmixr>

**LazyData** true

**RoxygenNote** 6.0.0

## R topics documented:

Epusillus . . . . .	2
Espinax . . . . .	2
get_growth_post_prob . . . . .	3
get_maturity_post_prob . . . . .	4
maturity_mix . . . . .	4
sim_mat_data . . . . .	6
sim_vb_data . . . . .	7
vb_bind_gr . . . . .	8
vb_bind_nll . . . . .	9
vb_growth_mix . . . . .	10
vb_lengths . . . . .	11
<b>Index</b>	<b>13</b>

---

Epusillus	<i>Etmopterus pusillus data</i>
-----------	---------------------------------

---

**Description**

Growth data for deepwater smooth lantern shark *Etmopterus pusillus* from Coelho and Erzini (2007). Data are cross-sectional with one observation (row) per individual.

**Usage**

Epusillus

**Format**

A data frame with five variables:

species Full species name

sex Sex of the animal: Female (F) or Male (M)

age Age in years

length Total length in centimetres

maturity Maturation status: immature or mature

**Source**

Coelho, R. & Erzini, K. (2007). Population parameters of the smooth lantern shark, *Etmopterus pusillus*, in southern portugal (NE Atlantic). *Fisheries Research*, 86, 42–57.

---

Espinax	<i>Etmopterus spinax data</i>
---------	-------------------------------

---

**Description**

Growth data for deepwater velvet belly lantern shark *Etmopterus spinax* from Coelho and Erzini (2008). Data are cross-sectional with one observation (row) per individual.

**Usage**

Espinax

**Format**

A data frame with five variables:

species Full species name

sex Sex of the animal: Female (F) or Male (M)

age Age in years

length Total length in centimetres

maturity Maturation status: immature or mature

**Source**

Coelho, R. and Erzini, K. (2008). Life history of a wide-ranging deepwater lantern shark in the north-east Atlantic, *Etmopterus spinax* (chondrichthyes: Etmopteridae), with implications for conservation. *Journal of Fish Biology*, 73, 1419–1443.

---

get\_growth\_post\_prob    *Posterior probability of sex being female based on growth alone*

---

**Description**

get\_growth\_post\_prob returns the probability of the observation(s) arising from the female component given a set of parameters and an assumed distribution (normal currently). The component probability is given by Bayes' theorem. Used internally.

**Usage**

```
get_growth_post_prob(mixprop, muF, muM, sigmaF, sigmaM, data, distribution)
```

**Arguments**

mixprop	Numeric scalar of mixing proportion (overall sex ratio)
muF	Numeric vector with predicted female lengths
muM	Numeric vector with predicted male lengths
sigmaF	Numeric scalar for female residual standard deviation
sigmaM	Numeric scalar for male residual standard deviation
data	A data.frame with column "length". Note predicted means "muF" and "muM" must come from corresponding ages.
distribution	Character with options: "normal" or "lognormal".

**Value**

Numeric vector of the posterior probability of being female.

**Examples**

```
get_growth_post_prob(mixprop = 0.5, muF = 4, muM = 6, sigmaF = 1,
                     sigmaM = 1, data = data.frame(length = 4.5), distribution = "normal")
```

---

```
get_maturity_post_prob
```

*Posterior probability of sex being female based on maturity alone*

---

### Description

get\_growth\_post\_prob returns the probability of the observation(s) arising from the female component given a set of parameters and an assumed distribution (binomial). The component probability is given by Bayes' theorem. Used internally.

### Usage

```
get_maturity_post_prob(mixprop, pF, pM, data)
```

### Arguments

mixprop	Numeric scalar of mixing proportion (overall sex ratio)
pF	Numeric vector with predicted female maturity probability.
pM	Numeric vector with predicted male maturity probability.
data	A data.frame with binary column "maturity".

### Value

Numeric vector of the posterior probability of being female.

### Examples

```
get_maturity_post_prob(mixprop = 0.5, pF = 0.5, pM = 0.5, sigmaF = 1,
  data = data.frame(maturity = 1))
```

---

```
maturity_mix
```

*Fit finite mixture maturity model.*

---

### Description

maturity\_mix fits sex-specific maturity models where some of the animals are of unknown sex. Optimization is via the Expectation-Maximisation algorithm.

### Usage

```
maturity_mix(start.list, data, binding = NULL, maxiter.em = 1000,
  reltol = 1e-08, plot.fit = FALSE, verbose = TRUE,
  estimate.mixprop = TRUE)
```

**Arguments**

<code>start.list</code>	A list with a list called <code>par</code> containing starting values for: "mixprop", "betaF" (female parameters) and "betaM" (male parameters) (see Examples).
<code>data</code>	A <code>data.frame</code> with columns: "maturity" (binary), "x" (single explanatory covariate currently), and "obs.sex". "obs.sex" must have values "female", "male", "unclassified".
<code>binding</code>	A (kx2) parameter index matrix (not implemented yet)
<code>maxiter.em</code>	Integer for maximum number of EM iterations (1e3 default).
<code>reltol</code>	Relative tolerance for EM observed data log likelihood convergence (1e-8 default).
<code>plot.fit</code>	Logical, if TRUE fit plotted per iteration (binary maturity jittered for visualisation). Red and blue circles are used for known females and males, respectively. Unclassified animals are plotted as triangle with the colour indicating the expected probability of being female or male (FALSE default).
<code>verbose</code>	Logical, if TRUE iteration and observed data log-likelihood printed.
<code>estimate.mixprop</code>	Logical, if TRUE the mixing proportion is estimated, otherwise fixed at the starting value.

**Value**

List containing the components:

<code>logLik.vec</code>	Observed data log-likelihood at each iteration.
<code>logLik</code>	Observed data log-likelihood on the last EM iteration.
<code>complete_data</code>	Data frame of the data (re-ordered) with component probabilities (tau).
<code>coefficients</code>	Parameter estimates (on logit scale) by sex and associated standard errors (not currently implemented).
<code>vcov</code>	Estimated variance covariance matrix of the parameters estimated on the real line. Can be used to obtain parameter standard errors on the natural scale (not currently implemented).
<code>convergence</code>	Binary with a "0" denoting convergence of the EM algorithm.

**Examples**

```
set.seed(1011)
sim.dat<-sim_mat_data(nfemale = 30, nmale = 30, shapes = c(3, 3),
  mat_parF = c(x50 = 0.5, xr = 0.1),
  mat_parM = c(x50 = 0.4, xr = 0.1),
  xcutoff = c(female = 0.4, male = 0.3))

start.list <- list(par = list(
  mixprop = 0.5,
  betaF = matrix(c(-5, 10)),
  betaM = matrix(c(-5, 10))
))

mat.fit <- maturity_mix(start.list = start.list, data = sim.dat)
```

---

sim_mat_data	<i>Simulate sex-specific maturity data with missing classifications.</i>
--------------	--------------------------------------------------------------------------

---

## Description

sim\_mat\_data simulates sex-specific maturity data according to a logistic maturity model.

## Usage

```
sim_mat_data(nfemale, nmale, shapes, mat_parF, mat_parM, xcutoff)
```

## Arguments

nfemale	Numeric scalar for number of female observations.
nmale	Numeric scalar for number of male observations.
shapes	Numeric vector for beta-distribution shape parameters - used to generate x values
mat_parF	Named ("x50", "xr") numeric vector with female maturity parameters ("x50" is the x value at which 50% are mature; "xr" is the x range between 25% and 75% mature)
mat_parM	Named ("x50", "xr") numeric vector with male maturity parameters
xcutoff	Named ("female", "male") numeric vector with x cutoffs below which immature animals are unclassified

## Value

data.frame with columns "x", "true.sex", "obs.sex" (observed sex including unclassified), "maturity" (binary: 1 if mature; 0 if immature).

## Examples

```
sim.dat<-sim_mat_data(nfemale = 30, nmale = 30, shapes = c(3, 3),
  mat_parF = c(x50 = 0.5, xr = 0.1),
  mat_parM = c(x50 = 0.4, xr = 0.1),
  xcutoff = c(female = 0.5, male = 0.4))
```

---

sim_vb_data	<i>Simulate sex-specific von Bertalanffy data with missing classifications.</i>
-------------	---------------------------------------------------------------------------------

---

## Description

sim\_vb\_data simulates sex-specific growth data according to the von Bertalanffy growth model and a logistic model governing maturity.

## Usage

```
sim_vb_data(nfemale, nmale, mean_ageF, mean_ageM, growth_parF, growth_parM,
            mat_parF, mat_parM, distribution)
```

## Arguments

nfemale	Numeric scalar for number of female observations.
nmale	Numeric scalar for number of male observations.
mean_ageF	Numeric scalar for female mean age - used to generate ages from <code>rnbinom(mu = mean_ageF)</code>
mean_ageM	Numeric scalar for male mean age - used to generate ages from <code>rnbinom(mu = mean_ageM)</code>
growth_parF	Named ("linf", "k", "t0", "sigma") numeric vector with female growth parameters
growth_parM	Named ("linf", "k", "t0", "sigma") numeric vector with male growth parameters
mat_parF	Named ("A50", "MR") numeric vector with female maturation parameters A50 is the age at 50% maturity, MR is age range between 25% and 75% maturity.
mat_parM	Named ("A50", "MR") numeric vector with male maturation parameters.
distribution	Character with options: "normal" or "lognormal".

## Value

data.frame with columns "age", "length", "true.sex", "obs.sex" (observed sex assuming immature animals unclassified), "maturity" (binary: 1 if mature; 0 if immature).

## Examples

```
sim.dat<-sim_vb_data(nfemale = 30, nmale = 30, mean_ageF = 3, mean_ageM = 3,
                    growth_parF = c(linf = 30, k = 0.2, t0 = -1, sigma = 0.1),
                    growth_parM = c(linf = 25, k = 0.2, t0 = -1, sigma = 0.1),
                    mat_parF = c(A50 = 3, MR = 1), mat_parM = c(A50 = 2, MR = 1),
                    distribution = "lognormal")

plot(jitter(sim.dat$age), sim.dat$length,
     xlim=c(0, max(sim.dat$age)), ylim=c(0, max(sim.dat$length)),
     col = c("red", "blue", "grey")[match(sim.dat$obs.sex,c("female", "male", "immature"))],
     pch = 19, xlab="age", ylab="Length")
```

---

vb_bind_gr	<i>Gradient of the negative log-likelihood for potentially constrained von Bertalanffy growth model (typically used internally).</i>
------------	--------------------------------------------------------------------------------------------------------------------------------------

---

## Description

vb\_bind\_gr returns the parameter gradients of negative log-likelihood for the von Bertalanffy model. Equality constraints across sexes can be implemented for any combination of parameters using the binding argument.

## Usage

```
vb_bind_gr(theta, binding, data, distribution)
```

## Arguments

theta	A parameter vector of the same length as the maximum of binding. Unconstrained parameters take the order: lnlinfF, lnlinfM, lnkF, lnkM, lnnt0F, lnnt0M, lnsigmaF, lnsigmaM.
binding	A (4x2) parameter index matrix with rows named (in order): "lnlinf", "lnk", "lnnt0", "lnsigma" and the left column for the female parameter index and right column for male parameter index. Used to impose arbitrary equality constraints across the sexes (see Examples).
data	A data.frame with columns: "age", "length" and "weights". "weights" are set to 1 or 0 for known females or males, respectively; proportions otherwise.
distribution	Character with options: "normal" or "lognormal".

## Value

Vector of parameter gradients:

## Examples

```
## Unconstrained model
binding <- matrix(c(1:8), ncol = 2, byrow = TRUE)
rownames(binding) <- c("lnlinf", "lnk", "lnnt0", "lnsigma")
colnames(binding) <- c("female", "male")
## starting values
start.par <- c(rep(log(25), 2), rep(log(0.2), 2), rep(log(3), 2), rep(log(1), 2))
vb_bind_gr(theta = start.par, binding = binding,
            data = data.frame(age = rep(1, 2), length = rep(10, 2), weights = c(1, 0)),
            distribution = "lognormal")
```



---

vb_bind_nll	<i>Negative log-likelihood for potentially constrained von Bertalanffy growth model (typically used internally).</i>
-------------	----------------------------------------------------------------------------------------------------------------------

---

## Description

vb\_bind\_nll returns the negative log-likelihood for the von Bertalanffy model. Equality constraints across sexes can be implemented for any combination of parameters using the binding argument.

## Usage

```
vb_bind_nll(theta, binding, data, distribution)
```

## Arguments

theta	A parameter vector of the same length as the maximum of binding. Unconstrained parameters take the order: lnlinfF, lnlinfM, lnkF, lnkM, lnnt0F, lnnt0M, lnsigmaF, lnsigmaM.
binding	A (4x2) parameter index matrix with rows named (in order): "lnlinf", "lnk", "lnnt0", "lnsigma" and the left column for the female parameter index and right column for male parameter index. Used to impose arbitrary equality constraints across the sexes (see Examples).
data	data.frame with columns: "age", "length" and "weights". "weights" are set to 1 or 0 for known females or males, respectively; proportions otherwise.
distribution	Character with options: "normal" or "lognormal"

## Value

Complete data negative log-likelihood:

## Examples

```
## Unconstrained model
binding <- matrix(c(1:8), ncol = 2, byrow = TRUE)
rownames(binding) <- c("lnlinf", "lnk", "lnnt0", "lnsigma")
colnames(binding) <- c("female", "male")
## starting values
start.par <- c(rep(log(25), 2), rep(log(0.2), 2), rep(log(3), 2), rep(log(1), 2))
vb_bind_nll(theta = start.par, binding = binding,
             data = data.frame(age = rep(1, 2), length = rep(10, 2), weights = c(1, 0)),
             distribution = "normal")
```

vb\_growth\_mix

*Fit finite mixture von Bertalanffy growth model.***Description**

vb\_growth\_mix fits sex-specific growth models where some of the animals are of unknown sex. Optimization is via the Expectation-Maximisation algorithm. Equality constraints across sexes can be implemented for any combination of parameters using the binding argument. Assumes a normal distribution currently.

**Usage**

```
vb_growth_mix(start.list, data, binding, maxiter.em = 1000, reltol = 1e-08,
  plot.fit = FALSE, verbose = TRUE, optim.method = "BFGS",
  estimate.mixprop = TRUE, distribution)
```

**Arguments**

start.list	A list with a list called par containing starting values for: "mixprop", "growth.par" (see Examples).
data	A data.frame with columns: "age", "length" and "obs.sex". "obs.sex" must have values "female", "male", "immature".
binding	A (4x2) parameter index matrix with rows named (in order): "lnlrf", "lnk", "lnnt0", "lnsigma" and the left column for the female parameter index and right column for mal parameter index. Used to impose arbitrary equality constraints across the sexes (see Examples).
maxiter.em	Integer for maximum number of EM iterations (1e3 default).
reltol	Relative tolerance for EM observed data log likelihood convergence (1e-10 default).
plot.fit	Logical, if TRUE fit plotted per iteration. Red and blue circles are used for known females and males, respectively. Immature / unsexed animals are plotted as triangle with the colour indicating the expected probability of being female or male (FALSE default).
verbose	Logical, if TRUE iteration and observed data log-likelihood printed.
optim.method	Character, complete data optimisation method to use in optim.
estimate.mixprop	Logical, if TRUE the mixing proportion is estimated, otherwise fixed at the starting value.
distribution	Character with options: "normal" or "lognormal".

**Value**

List containing the components:

logLik.vec      Observed data log-likelihood at each iteration.

logLik	Observed data log-likelihood on the last EM iteration.
complete_data	Data frame of the data (re-ordered) with component probabilities (tau).
coefficients	Parameter estimates (on the real line) and associated standard errors on the real line.
vcov	Estimated variance covariance matrix of the parameters estimated on the real line. Can be used to obtain parameter standard errors on the natural scale.
convergence	Binary with a "0" denoting convergence of the EM algorithm.

### Examples

```
set.seed(1010)
sim.dat <- sim_vb_data(nfemale = 50, nmale = 50, mean_ageF = 4, mean_ageM = 4,
  growth_parF = c(linf = 30, k = 0.5, t0 = -1, sigma = 0.1),
  growth_parM = c(linf = 25, k = 0.5, t0 = -1, sigma = 0.1),
  mat_parF = c(A50 = 5, MR = 2), mat_parM = c(A50 = 3, MR = 2),
  distribution = "lognormal")

## Model fit with constrained Brody's growth coefficient
## Set up the constraint
binding <- matrix(c(1:2, rep(3, 2), 4:7), ncol = 2, byrow = TRUE)
rownames(binding) <- c("lnlinf", "lnk", "lnnt0", "lnsigma")
colnames(binding) <- c("female", "male")
## starting values
start.par <- c(c(log(30), log(25)), rep(log(0.3), 1), rep(log(1), 2), rep(log(.1), 2))
start.list <- list(par = list(mixprop = 0.5, growth.par = start.par))
vb.bind.fit <- vb_growth_mix(data = sim.dat, start.list = start.list,
  binding = binding, distribution = "lognormal",
  reltol = 1e-6)
```

---

vb_lengths	<i>von Bertalanffy growth function.</i>
------------	-----------------------------------------

---

### Description

vb\_lengths returns the predicted length-at-age for given named set of parameters for the von Bertalanffy growth function:

$$l = L_{\infty}(1 - e^{-k(a-t_0)})$$

### Usage

```
vb_lengths(theta, age)
```

### Arguments

theta	A numeric vector with named values "linf", "k", "t0".
age	A numeric vector of ages.

**Value**

Predicted length-at-age.

**Examples**

```
vb_lengths(theta = c("linf" = 30, "k" = 0.2, "t0" = -1), age = 0:10)
```

# Index

## \*Topic **datasets**

Epusillus, [2](#)

Espinax, [2](#)

Epusillus, [2](#)

Espinax, [2](#)

get\_growth\_post\_prob, [3](#)

get\_maturity\_post\_prob, [4](#)

maturity\_mix, [4](#)

sim\_mat\_data, [6](#)

sim\_vb\_data, [7](#)

vb\_bind\_gr, [8](#)

vb\_bind\_nll, [9](#)

vb\_growth\_mix, [10](#)

vb\_lengths, [11](#)