Package ‘RAC’

October 12, 2022

Version 1.5
Date 2018-04-24
Title R Package for Aqua Culture
Maintainer Baldan D. <damiano.baldan91@gmail.com>
Depends R (>= 3.1.0), matrixStats, rstudioapi, plotrix
Suggests MASS, knitr, rmarkdown
Imports maps, maptools, ncdf4, rgdal, rgeos, sp, stats, utils, raster
Description Solves the individual bioenergetic balance for different aquaculture sea fish (Sea Bream and Sea Bass; Brigolin et al., 2014 <doi:10.3354/aei00093>) and shellfish (Mussel and Clam; Brigolin et al., 2009 <doi:10.1016/j.ecss.2009.01.029>; Solidoro et al., 2000 <doi:10.3354/meps199137>). Allows for spatialized model runs and population simulations.
License MIT + file LICENSE
RoxygenNote 5.0.1
VignetteBuilder knitr
NeedsCompilation no
Author Baldan D. [aut, cre],
Palazzo D. [ctb],
Porporato E.M.D [ctb],
Brigolin D. [ctb]
Repository CRAN
Date/Publication 2018-05-02 07:38:09 UTC

R topics documented:

  Bass_ind_dataloader ........................................ 4
  Bass_ind_equations ........................................ 4
  Bass_ind_main ............................................... 5
  Bass_ind_post ............................................... 5
  Bass_ind_pre ............................................... 6
  Bass_ind_RKsolver ......................................... 6
<table>
<thead>
<tr>
<th>R topics documented:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bass_indSkeleton</td>
</tr>
<tr>
<td>Bass_popDataloader</td>
</tr>
<tr>
<td>Bass_popEquations</td>
</tr>
<tr>
<td>Bass_popLoop</td>
</tr>
<tr>
<td>Bass_popMain</td>
</tr>
<tr>
<td>Bass_popPost</td>
</tr>
<tr>
<td>Bass_popPre</td>
</tr>
<tr>
<td>Bass_popRKSolver</td>
</tr>
<tr>
<td>Bass_popSkeleton</td>
</tr>
<tr>
<td>Bass_spatialDataloader</td>
</tr>
<tr>
<td>Bass_spatialLoop</td>
</tr>
<tr>
<td>Bass_spatialMain</td>
</tr>
<tr>
<td>Bass_spatialPost</td>
</tr>
<tr>
<td>Bass_spatialPre</td>
</tr>
<tr>
<td>Bass_spatialPreInt</td>
</tr>
<tr>
<td>Bass_spatialRKSolver</td>
</tr>
<tr>
<td>Bream_indDataloader</td>
</tr>
<tr>
<td>Bream_indEquations</td>
</tr>
<tr>
<td>Bream_indMain</td>
</tr>
<tr>
<td>Bream_indPost</td>
</tr>
<tr>
<td>Bream_indPre</td>
</tr>
<tr>
<td>Bream_indRKSolver</td>
</tr>
<tr>
<td>Bream_indSkeleton</td>
</tr>
<tr>
<td>Bream_popDataloader</td>
</tr>
<tr>
<td>Bream_popEquations</td>
</tr>
<tr>
<td>Bream_popLoop</td>
</tr>
<tr>
<td>Bream_popMain</td>
</tr>
<tr>
<td>Bream_popPost</td>
</tr>
<tr>
<td>Bream_popPre</td>
</tr>
<tr>
<td>Bream_popRKSolver</td>
</tr>
<tr>
<td>Bream_popSkeleton</td>
</tr>
<tr>
<td>Bream_spatialDataloader</td>
</tr>
<tr>
<td>Bream_spatialLoop</td>
</tr>
<tr>
<td>Bream_spatialMain</td>
</tr>
<tr>
<td>Bream_spatialPost</td>
</tr>
<tr>
<td>Bream_spatialPre</td>
</tr>
<tr>
<td>Bream_spatialPreInt</td>
</tr>
<tr>
<td>Bream_spatialRKSolver</td>
</tr>
<tr>
<td>ClamF_indDataloader</td>
</tr>
<tr>
<td>ClamF_indEquations</td>
</tr>
<tr>
<td>ClamF_indMain</td>
</tr>
<tr>
<td>ClamF_indPost</td>
</tr>
<tr>
<td>ClamF_indPre</td>
</tr>
<tr>
<td>ClamF_indRKSolver</td>
</tr>
<tr>
<td>ClamF_indSkeleton</td>
</tr>
<tr>
<td>ClamF_popDataloader</td>
</tr>
<tr>
<td>ClamF_pop_equations</td>
</tr>
<tr>
<td>ClamF_pop_loop</td>
</tr>
<tr>
<td>ClamF_pop_main</td>
</tr>
<tr>
<td>ClamF_pop_post</td>
</tr>
<tr>
<td>ClamF_pop_pre</td>
</tr>
<tr>
<td>ClamF_pop_RKsolver</td>
</tr>
<tr>
<td>ClamF_pop_skeleton</td>
</tr>
<tr>
<td>Clam_ind_dataloader</td>
</tr>
<tr>
<td>Clam_ind_equations</td>
</tr>
<tr>
<td>Clam_ind_main</td>
</tr>
<tr>
<td>Clam_ind_post</td>
</tr>
<tr>
<td>Clam_ind_pre</td>
</tr>
<tr>
<td>clam_ind_RKsolver</td>
</tr>
<tr>
<td>Clam_ind_skeleton</td>
</tr>
<tr>
<td>Clam_pop_dataloader</td>
</tr>
<tr>
<td>Clam_pop_equations</td>
</tr>
<tr>
<td>Clam_pop_loop</td>
</tr>
<tr>
<td>Clam_pop_main</td>
</tr>
<tr>
<td>Clam_pop_post</td>
</tr>
<tr>
<td>Clam_pop_pre</td>
</tr>
<tr>
<td>Clam_pop_RKsolver</td>
</tr>
<tr>
<td>Clam_pop_skeleton</td>
</tr>
<tr>
<td>Mussel_ind_dataloader</td>
</tr>
<tr>
<td>Mussel_ind_equations</td>
</tr>
<tr>
<td>Mussel_ind_main</td>
</tr>
<tr>
<td>Mussel_ind_post</td>
</tr>
<tr>
<td>Mussel_ind_pre</td>
</tr>
<tr>
<td>Mussel_ind_RKsolver</td>
</tr>
<tr>
<td>Mussel_ind_skeleton</td>
</tr>
<tr>
<td>Mussel_pop_dataloader</td>
</tr>
<tr>
<td>Mussel_pop_equations</td>
</tr>
<tr>
<td>Mussel_pop_loop</td>
</tr>
<tr>
<td>Mussel_pop_main</td>
</tr>
<tr>
<td>Mussel_pop_post</td>
</tr>
<tr>
<td>Mussel_pop_pre</td>
</tr>
<tr>
<td>Mussel_pop_RKsolver</td>
</tr>
<tr>
<td>Mussel_pop_skeleton</td>
</tr>
<tr>
<td>Mussel_spatial_dataloader</td>
</tr>
<tr>
<td>Mussel_spatial_loop</td>
</tr>
<tr>
<td>Mussel_spatial_main</td>
</tr>
<tr>
<td>Mussel_spatial_post</td>
</tr>
<tr>
<td>Mussel_spatial_pre</td>
</tr>
<tr>
<td>Mussel_spatial_pre_int</td>
</tr>
<tr>
<td>Mussel_spatial_RKsolver</td>
</tr>
<tr>
<td>Mussel_spatial_skeleton</td>
</tr>
<tr>
<td>Pop_fun</td>
</tr>
</tbody>
</table>

Index 58
**Bass_ind_dataloader**

*Function that loads forcings data for Seabass individual bioenergetic model and performs the interpolation*

**Description**

Function that loads forcings data for Seabass individual bioenergetic model and performs the interpolation.

**Usage**

`Bass_ind_dataloader(userpath)`

**Arguments**

- `userpath`: the path where folder containing model inputs and outputs is located.

**Value**

A list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d].

---

**Bass_ind_equations**

*Seabass bioenergetic individual model differential equations*

**Description**

Seabass bioenergetic individual model differential equations.

**Usage**

`Bass_ind_equations(Param, Temp, G, Food, weight)`

**Arguments**

- `Param`: vector containing all metabolic parameters.
- `Temp`: water temperature forcing at time t.
- `G`: food entering the cage at time t.
- `Food`: food characterization (Proteins, Lipids, Carbohydrates).
- `weight`: individual weight at time t.

**Value**

Model output at time t.
**Bass_ind_main**  

*Bass ind_main*  

**Seabass bioenergetic individual model**

**Description**

Solves the bioenergetic balance for Seabass

**Usage**

`Bass_ind_main(userpath, forcings)`

**Arguments**

- **userpath**  
  the path where forcing are located
- **forcings**  
  a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

**Value**

A list containing model outputs: weight, excreted quantities and quantities to waste, actual and potential ingestion, temperature limitation functions and metabolic rates

---

**Bass_ind_post**  

*Bass ind_post*  

**Seabass bioenergetic individual model postprocessor**

**Description**

Seabass bioenergetic individual model postprocessor

**Usage**

`Bass_ind_post(userpath, output, times, Dates, CS)`

**Arguments**

- **userpath**  
  the path where the working folder is located
- **output**  
  output list containing the output of the RK solver
- **times**  
  the vector containing informations on integration extremes
- **Dates**  
  the vector containing the date
- **CS**  
  the commercial size of Seabass

**Value**

A list containing the fish weight, proteins, lipids and carbohydrates wasted or produced with excretions, potential and actual ingestion rates, temperature limitation functions and metabolic rates
**Bass_ind_pre**  
*Seabass bioenergetic individual model preprocessor*

**Description**
Seabass bioenergetic individual model preprocessor

**Usage**
```python
Bass_ind_pre(userpath, forcings)
```

**Arguments**
- **userpath** the path where folder containing model inputs and outputs is located
- **forcings** a list containing model forcings

**Value**
a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

---

**Bass_ind_RKsolver**  
*Solves the Seabass bioenergetic balance with a 4th order Runge Kutta method*

**Description**
Solves the Seabass bioenergetic balance with a 4th order Runge Kutta method

**Usage**
```python
Bass_ind_RKsolver(Param, Temperature, G, Food, IC, times)
```

**Arguments**
- **Param** vector containing all metabolic parameters
- **Temperature** water temperature forcing time series
- **G** food entering the cage time series
- **Food** food characterization (Proteins, Lipids, Carbohydrates)
- **IC** initial conditions
- **times** vector containing integration extremes and integration timestep

**Value**
a list containing the fish weight, proteins, lipids and carbohydrates wasted or produced with excretions, potential and actual ingestion rates, temperature limitation functions and metabolic rates
Bass_ind_skeleton

Creates the folders structure for Seabass individual bioenergetic model

Description

Creates the folders structure for Seabass individual bioenergetic model

Usage

Bass_ind_skeleton(userpath)

Arguments

userpath the path where forcing are located

Bass_pop_dataloader

Function that loads forcings data for Seabass population model and performs the interpolation

Description

Function that loads forcings data for Seabass population model and performs the interpolation

Usage

Bass_pop_dataloader(userpath)

Arguments

userpath the path where folder containing model inputs and outputs is located

Value

a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]
Bass_pop_equations  

**Description**

Seabass bioenergetic population model differential equations

**Usage**

Bass_pop_equations(Param, N, Temp, G, Food, weight)

**Arguments**

- **Param**  
  vector containing all metabolic parameters
- **N**  
  the number of individuals at time t
- **Temp**  
  water temperature forcing at time t
- **G**  
  food entering the cage at time t
- **Food**  
  food characterization (Proteins, Lipids, Carbohydrates)
- **weight**  
  individual weight at time t

**Value**

model output at time t

---

Bass_pop_loop  

**Description**

Function that runs the Monte Carlo simulation for the Seabass population model

**Usage**

Bass_pop_loop(Param, Tint, Gint, Food, IC, times, N, userpath)

**Arguments**

- **Param**  
  a vector containing model parameters
- **Tint**  
  the interpolated water temperature time series
- **Gint**  
  the interpolated feeding rate time series
- **Food**  
  the food characterization
- **IC**  
  initial condition
- **times**  
  integration extremes and integration timestep
- **N**  
  time series with number of individuals
- **userpath**  
  the path where the working folder is located
**Bass_pop_main**

**Value**

A list with RK solver outputs

---

**Bass_pop_main**  
*Seabass bioenergetic population model*

**Description**

Seabass bioenergetic population model

**Usage**

`Bass_pop_main(userpath, forcings)`

**Arguments**

- **userpath**: the path where the working folder is located
- **forcings**: a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

**Value**

A list containing model outputs: weight, excreted quantities and quantities to waste, actual and potential ingestion, temperature limitation functions and metabolic rates

---

**Bass_pop_post**  
*Postprocess the Seabass population bioenergetic model results*

**Description**

Postprocess the Seabass population bioenergetic model results

**Usage**

`Bass_pop_post(userpath, output, times, Dates, N, CS)`

**Arguments**

- **userpath**: the path where the working folder is located
- **output**: list containing the output of the RK solver
- **times**: the vector containing informations on integration extremes
- **Dates**: the vector containing the date
- **N**: the number of individuals time series
- **CS**: the commercial size of Seabass
Value
output: a list containing the fish weight, proteins, lipids and carbohydrates wasted or produced with excretions, potential and actual ingestion rates, temperature limitation functions and metabolic rates

Bass_pop_pre  Seabass bioenergetic population model preprocessor

Description
Seabass bioenergetic population model preprocessor

Usage
Bass_pop_pre(userpath, forcings)

Arguments
userpath  the path where folder containing model inputs and outputs is located
forcings  a list containing model forcings

Value
a list containing the time series in the odd positions and related forcings in the even positions.
Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

Bass_pop_RKsolver  Solves the Seabass population bioenergetic balance with a 4th order Runge Kutta method

Description
Solves the Seabass population bioenergetic balance with a 4th order Runge Kutta method

Usage
Bass_pop_RKsolver(Param, Temperature, G, Food, IC, times, N)

Arguments
Param  vector containing all metabolic parameters
Temperature  water temperature forcing time series
G  food entering the cage at time series
Food  food characterization (Proteins, Lipids, Carbohydrates)
IC  initial condition on weight
times  integration times
N  number of individuals time series
**Value**

a list containing the fish weight, proteins, lipids and carbohydrates wasted or produced with excretions, potential and actual ingestion rates, temperature limitation functions and metabolic rates

---

**Bass_pop_skeleton**

*Creates the folders structure for Seabass population model*

---

**Description**

Creates the folders structure for Seabass population model

**Usage**

`Bass_pop_skeleton(userpath)`

**Arguments**

- `userpath` the path where forcing are located

---

**Bass_spatial_dataloader**

*Function that loads forcings data for Bass spatialized model and performs the interpolation*

---

**Description**

Function that loads forcings data for Bass spatialized model and performs the interpolation

**Usage**

`Bass_spatial_dataloader(userpath)`

**Arguments**

- `userpath` the path where folder containing model inputs and outputs is located

**Value**

a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]
Bass_spatial_loop  
"Bass bioenergetic spatialized model - spatialization loop"

**Description**

Solves the bioenergetic balance for Bass

**Usage**

Bass_spatial_loop(userpath, forcings)

**Arguments**

- **userpath**: the path where the working folder is located
- **forcings**: a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

**Value**

a list containing the outputs that main script saves to .nc, .csv and .asc files

Bass_spatial_main  
"Bass bioenergetic spatialized model - spatialization loop"

**Description**

Solves the bioenergetic balance for Bass

**Usage**

Bass_spatial_main(userpath, forcings)

**Arguments**

- **userpath**: the path where the working folder is located
- **forcings**: list containing the time series in the odd positions and related forcings in the even positions. Forcings inputted are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

**Value**

saves .nc, .csv and .asc outputs in the 'Outputs' folder
Bass_spatial_post

Postprocess the Mussel spatialized model results

Description

Postprocess the Mussel spatialized model results

Usage

Bass_spatial_post(userpath, output, times, Dates, CS)

Arguments

userpath the path where the working folder is located
output output list containing the output of the RK solver
times the vector containing informations on integration extremes
Dates the vector containing the date
CS the commercial size of Bass

Value

a list containing the fish weight, proteins, lipids and carbohydrates wasted or produced with excretions, potential and actual ingestion rates, temperature limitation functions and metabolic rates

Bass_spatial_pre

Bass bioenergetic spatialized model preprocessor

Description

Bass bioenergetic spatialized model preprocessor

Usage

Bass_spatial_pre(userpath, forcings)

Arguments

userpath the path where folder containing model inputs and outputs is located
forcings a list containing forcings used by the model

Value

a list containing the data used by the main script
**Bass_spatial_pre_int**  
Bass bioenergetic spatialized model preprocessor - used inside spatialization loop

**Description**
Bass bioenergetic spatialized model preprocessor - used inside spatialization loop

**Usage**
```python
Bass_spatial_pre_int(userpath, forcings)
```

**Arguments**
- `userpath` the path where folder containing model inputs and outputs is located
- `forcings` a list containing forcings used by the model

**Value**
a list containing data used by the main script

---

**Bass_spatial_RKsolver**  
Solves the Seabass bioenergetic balance with a 4th order Runge Kutta method - used in spatialized model

**Description**
Solves the Seabass bioenergetic balance with a 4th order Runge Kutta method - used in spatialized model

**Usage**
```python
Bass_spatial_RKsolver(Param, Temperature, G, Food, IC, times)
```

**Arguments**
- `Param` vector containing all metabolic parameters
- `Temperature` water temperature forcing time series
- `G` food entering the cage time series
- `Food` food characterization (Proteins, Lipids, Carbohydrates)
- `IC` initial conditions
- `times` vector containing integration extremes and integration timestep

**Value**
a list containing the fish weight, proteins, lipids and carbohydrates wasted or produced with excretions, potential and actual ingestion rates, temperature limitation functions and metabolic rates
**Bass_spatial_skeleton**

*Creates the folders structure for Bass spatialized model*

**Description**

Creates the folders structure for Bass spatialized model

**Usage**

```
Bass_spatial_skeleton(userpath)
```

**Arguments**

- `userpath`: the path where forcing are located

**Bream_ind_dataloader**

*Function that loads forcings data for Seabream individual bioenergetic model and performs the interpolation*

**Description**

Function that loads forcings data for Seabream individual bioenergetic model and performs the interpolation

**Usage**

```
Bream_ind_dataloader(userpath)
```

**Arguments**

- `userpath`: the path where folder containing model inputs and outputs is located

**Value**

A list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]
**Bream_ind_equations**  
Seabream bioenergetic individual model differential equations

**Description**  
Seabream bioenergetic individual model differential equations

**Usage**  
\[ \text{Bream\_ind\_equations}(\text{Param}, \text{Temp}, \text{G}, \text{Food}, \text{weight}) \]

**Arguments**
- **Param**: vector containing all metabolic parameters
- **Temp**: water temperature forcing at time t
- **G**: food entering the cage at time t
- **Food**: food characterization (Proteins, Lipids, Carbohydrates)
- **weight**: individual weight at time t

**Value**  
model output at time t

**Bream_ind_main**  
Seabream bioenergetic individual model

**Description**  
Seabream bioenergetic individual model

**Usage**  
\[ \text{Bream\_ind\_main}(\text{userpath}, \text{forcings}) \]

**Arguments**
- **userpath**: the path where the working folder is located
- **forcings**: a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

**Value**  
A list containing model outputs: weight, excreted quantities and quantities to waste, actual and potential ingestion, temperature limitation functions and metabolic rates
**Bream_ind_post**

*Postprocess the Seabream individual bioenergetic model results*

**Description**

Postprocess the Seabream individual bioenergetic model results

**Usage**

Bream_ind_post(userpath, output, times, Dates, CS)

**Arguments**

- **userpath**: the path where the working folder is located
- **output**: output list containing the output of the RK solver
- **times**: the vector containing informations on integration extremes
- **Dates**: the vector containing the date
- **CS**: the commercial size of Seabream

**Value**

a list containing the fish weight, proteins, lipids and carbohydrates wasted or produced with excretions, potential and actual ingestion rates, temperature limitation functions and metabolic rates

---

**Bream_ind_pre**

*Seabream bioenergetic individual model preprocessor*

**Description**

Preprocesses the data for the bioenergetic balance for Sea Bream

**Usage**

Bream_ind_pre(userpath, forcings)

**Arguments**

- **userpath**: the path where folder containing model inputs and outputs is located
- **forcings**: a list containing model forcings

**Value**

a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]
Bream_ind_RKsolver  \textit{Solves the Seabream individual bioenergetic balance with a 4th order Runge Kutta method}

\begin{description}
\item[Description] Solves the Seabream individual bioenergetic balance with a 4th order Runge Kutta method
\item[Usage] \texttt{Bream\_ind\_RKsolver(Param, Temperature, G, Food, IC, times)}
\item[Arguments]
  \begin{itemize}
  \item \texttt{Param} vector containing all metabolic parameters
  \item \texttt{Temperature} water temperature forcing time series
  \item \texttt{G} food entering the cage time series
  \item \texttt{Food} food characterization (Proteins, Lipids, Carbohydrates)
  \item \texttt{IC} vector containing initial conditions on weight
  \item \texttt{times} vector containing integration extremes and timestep
  \end{itemize}
\item[Value] a list containing the fish weight, proteins, lipids and carbohydrates wasted or produced with excretions, potential and actual ingestion rates, temperature limitation functions and metabolic rates
\end{description}

Bream_ind_skeleton  \textit{Creates the folders structure for Seabream individual bioenergetic model}

\begin{description}
\item[Description] Creates the folders structure for Seabream individual bioenergetic model
\item[Usage] \texttt{Bream\_ind\_skeleton(userpath)}
\item[Arguments]
  \begin{itemize}
  \item \texttt{userpath} the path where forcing are located
  \end{itemize}
\end{description}
Function that loads forcings data for Seabream population model and performs the interpolation

Description
Function that loads forcings data for Seabream population model and performs the interpolation

Usage
Bream_pop_dataloader(userpath)

Arguments
- userpath: the path where folder containing model inputs and outputs is located

Value
a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

Seabream bioenergetic population model differential equations

Description
Seabream bioenergetic population model differential equations

Usage
Bream_pop_equations(Param, N, Temp, G, Food, weight)

Arguments
- Param: vector containing all metabolic parameters
- N: the number of individuals at time t
- Temp: water temperature forcing at time t
- G: food entering the cage at time t
- Food: food characterization (Proteins, Lipids, Carbohydrates)
- weight: individual weight at time t

Value
model output at time t
Bream_pop_loop

*Function that runs the Monte Carlo simulation for the Seabream population model*

**Description**

Function that runs the Monte Carlo simulation for the Seabream population model

**Usage**

\[ \text{Bream\_pop\_loop(Param, Tint, Gint, Food, IC, times, N, userpath)} \]

**Arguments**

- **Param**: a vector containing model parameters
- **Tint**: the interpolated water temperature time series
- **Gint**: the interpolated feeding rate time series
- **Food**: the food characterization
- **IC**: initial condition
- **times**: integration extremes and integration timestep
- **N**: time series with number of individuals
- **userpath**: the path where the working folder is located

**Value**

a list with RK solver outputs

Bream_pop_main

*Seabream bioenergetic population model*

**Description**

Seabream bioenergetic population model

**Usage**

\[ \text{Bream\_pop\_main(userpath, forcings)} \]

**Arguments**

- **userpath**: the path where the working folder is located
- **forcings**: a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]
Bream_pop_post

Value
A list containing model outputs: weight, excreted quantities and quantities to waste, actual and potential ingestion, temperature limitation functions and metabolic rates

Bream_pop_post  Postprocess the Bream population bioenergetic model results

Description
Postprocess the Bream population bioenergetic model results

Usage
Bream_pop_post(userpath, output, times, Dates, N, CS)

Arguments
- userpath: the path where the working folder is located
- output: output list containing the output of the RK solver
- times: the vector containing informations on integration extremes
- Dates: the vector containing the date
- N: the number of individuals
- CS: the commercial size of Seabream

Value
A list containing the fish weight, proteines, lipids and carbohydrates wasted or produced with excretions, potential and actual ingestion rates, temperature limitation functions and metabolic rates

Bream_pop_pre  Seabream bioenergetic population model preprocessor

Description
Preprocesses the data for the bioenergetic balance for Sea Bream

Usage
Bream_pop_pre(userpath, forcings)

Arguments
- userpath: the path where folder containing model inputs and outputs is located
- forcings: a list containing model forcings
**Bream_pop_skeleton**

**Value**

a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

---

**Bream_pop_RKsolver**

_Solves the Seabream population bioenergetic balance with a 4th order Runge Kutta method_

---

**Description**

Solves the Seabream population bioenergetic balance with a 4th order Runge Kutta method

**Usage**

\[\text{Bream\_pop\_RKsolver}(\text{Param, Temperature, G, Food, IC, times, N})\]

**Arguments**

- **Param** vector containing all metabolic parameters
- **Temperature** water temperature forcing time series
- **G** food entering the cage time series
- **Food** food characterization (Proteins, Lipids, Carbohydrates)
- **IC** initial condition on weight
- **times** integration times
- **N** number of individuals time series

**Value**

a list containing the fish weight, proteins, lipids and carbohydrates wasted or produced with excretions, potential and actual ingestion rates, temperature limitation functions and metabolic rates

---

**Bream_pop_skeleton**

_Creates the folders structure for Seabream population model_

---

**Description**

Creates the folders structure for Seabream population model

**Usage**

\[\text{Bream\_pop\_skeleton}(\text{userpath})\]

**Arguments**

- **userpath** the path where forcing are located
Bream_spatial_dataloader

Function that loads forcings data for Bream spatialized model and performs the interpolation

Description
Function that loads forcings data for Bream spatialized model and performs the interpolation

Usage
Bream_spatial_dataloader(userpath)

Arguments
userpath the path where folder containing model inputs and outputs is located

Value
a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

Bream_spatial_loop Bream bioenergetic spatialized model - spatialization loop

Description
Solves the bioenergetic balance for Bream

Usage
Bream_spatial_loop(userpath, forcings)

Arguments
userpath the path where the working folder is located
forcings a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

Value
A list containing model outputs: weight, excreted quantities and quantities to waste, actual and potential ingestion, temperature limitation functions and metabolic rates
**Bream_spatial_main**  
*Bream bioenergetic spatialized model - spatialization loop*

**Description**

Solves the bioenergetic balance for Bream

**Usage**

Bream_spatial_main(userpath, forcings)

**Arguments**

- **userpath**  
  the path where the working folder is located

- **forcings**  
  list containing the time series in the odd positions and related forcings in the even positions. Forcings inputted are: Water temperature [Celsius degrees] and feeding rate [g/individual x d]

**Value**

saves .nc; .csv and .asc outputs in the 'Outputs' folder

---

**Bream_spatial_post**  
*Postprocess the Bream spatialized model results*

**Description**

Postprocess the Bream spatialized model results

**Usage**

Bream_spatial_post(userpath, output, times, Dates, CS)

**Arguments**

- **userpath**  
  the path where the working folder is located

- **output**  
  output list containing the output of the RK solver

- **times**  
  the vector containing informations on integration extremes

- **Dates**  
  the vector containing the date

- **CS**  
  the commercial size of Bream

**Value**

a list containing the model outputs saved by the main script to .nc; .csv and .asc files
**Bream\_spatial\_pre**

*Bream bioenergetic spatialized model preprocessor*

**Description**

Bream bioenergetic spatialized model preprocessor

**Usage**

Bream\_spatial\_pre(userpath, forcings)

**Arguments**

- **userpath** the path where folder containing model inputs and outputs is located
- **forcings** a list containing forcings used by the model

**Value**

a list containing the data used in the main script

**Bream\_spatial\_pre\_int**

*Bream bioenergetic spatialized model preprocessor - used inside spatialization loop*

**Description**

Bream bioenergetic spatialized model preprocessor - used inside spatialization loop

**Usage**

Bream\_spatial\_pre\_int(userpath, forcings)

**Arguments**

- **userpath** the path where folder containing model inputs and outputs is located
- **forcings** a list containing forcings used by the model

**Value**

a list containing the data used by the main script
Bream_spatial_RKsolver

Solves the Seabream bioenergetic balance with a 4th order Runge Kutta method - used in spatialized model

Description
Solves the Seabream bioenergetic balance with a 4th order Runge Kutta method - used in spatialized model

Usage
Bream_spatial_RKsolver(Param, Temperature, G, Food, IC, times)

Arguments
Param vector containing all metabolic parameters
Temperature water temperature forcing time series
G food entering the cage time series
Food food characterization (Proteins, Lipids, Carbohydrates)
IC initial conditions
times vector containing integration extremes and integration timestep

Value
a list containing the fish weight, proteines, lipids and carbohydrates wasted or produced with excre-
tions, potential and actual ingestion rates, temperature limitation functions and metabolic rates

Bream_spatial_skeleton

Creates the folders structure for Bream spatialized model

Description
Creates the folders structure for Bream spatialized model

Usage
Bream_spatial_skeleton(userpath)

Arguments
userpath the path where forcing are located
ClamF_ind_dataloader  
*Function that loads forcings data for Clam individual bioenergetic model (alternative version) and performs the interpolation*

**Description**

Function that loads forcings data for Clam individual bioenergetic model (alternative version) and performs the interpolation.

**Usage**

ClamF_ind_dataloader(userpath)

**Arguments**

- **userpath** the path where folder containing model inputs and outputs is located.

**Value**

A list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3].

---

ClamF_ind_equations  
*Clam bioenergetic individual model differential equations (alternative version)*

**Description**

Clam bioenergetic individual model differential equations (alternative version).

**Usage**

ClamF_ind_equations(Param, Tint, Chlint, Ww)

**Arguments**

- **Param** a vector containing model parameters.
- **Tint** the interpolated water temperature at time t.
- **Chlint** the interpolated chlorophyll at time t.
- **Ww** clam wet weight at time t.

**Value**

A list containing the clam weights, temperature limitation functions and metabolic rates at time t.
ClamF_ind_main  
*Clam bioenergetic individual model (alternative version)*

**Description**

Clam bioenergetic individual model (alternative version)

**Usage**

```r
ClamF_ind_main(userpath, forcings)
```

**Arguments**

- `userpath`  
  the path where the working folder is located
- `forcings`  
  a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3]

**Value**

A list containing model outputs: weights, temperature limitation functions and metabolic rates

ClamF_ind_post  
*Postprocess the Clam indivual bioenergetic model (alternative version) results*

**Description**

Postprocess the Clam indivual bioenergetic model (alternative version) results

**Usage**

```r
ClamF_ind_post(userpath, output, times, Dates, CS)
```

**Arguments**

- `userpath`  
  the path where the working folder is located
- `output`  
  output list containing the output of the RK solver
- `times`  
  the vector containing informations on integration extremes
- `Dates`  
  the vector containing the date
- `CS`  
  the commercial size of Clam

**Value**

A list containing the clam weights, temperature limitation functions and metabolic rates
### ClamF_ind_pre

**Clam bioenergetic individual model preprocessor (alternative version)**

**Description**

Clam bioenergetic individual model preprocessor (alternative version)

**Usage**

```
ClamF_ind_pre(userpath, forcings)
```

**Arguments**

- **userpath**  
  the path where folder containing model inputs and outputs is located

- **forcings**  
  a list containing model forcings

**Value**

a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3]

### ClamF_ind_RKsolver

**Solves the Clam bioenergetic balance (alternative version) with a 4th order Runge Kutta method**

**Description**

Solves the Clam bioenergetic balance (alternative version) with a 4th order Runge Kutta method

**Usage**

```
ClamF_ind_RKsolver(Param, times, IC, Tint, Chlint)
```

**Arguments**

- **Param**  
  a vector containing model parameters

- **times**  
  integration extremes and integration timestep

- **IC**  
  initial condition

- **Tint**  
  the interpolated water temperature time series

- **Chlint**  
  the interpolated chlorophyll a time series

**Value**

a list containing the clam weights, temperature limitation functions and metabolic rates
### ClamF_ind_skeleton

*Description*

Creates the folders structure for Clam individual bioenergetic model (alternative version)

**Usage**

ClamF_ind_skeleton(userpath)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userpath</td>
<td>the path where forcing are located</td>
</tr>
</tbody>
</table>

### ClamF_pop_dataloader

*Description*

Function that loads forcings data for Clam population model (alternative version) and performs the interpolation

**Usage**

ClamF_pop_dataloader(userpath)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userpath</td>
<td>the path where folder containing model inputs and outputs is located</td>
</tr>
</tbody>
</table>

**Value**

a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3]
ClamF_pop_equations  

Clam bioenergetic population model differential equations (alternative version)

Description

Clam bioenergetic population model differential equations (alternative version)

Usage

ClamF_pop_equations(Param, Tint, Chlint, Ww)

Arguments

<table>
<thead>
<tr>
<th>Param</th>
<th>a vector containing model parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tint</td>
<td>the interpolated water temperature at time t</td>
</tr>
<tr>
<td>Chlint</td>
<td>the interpolated chlorophyll at time t</td>
</tr>
<tr>
<td>Ww</td>
<td>clam wet weight at time t</td>
</tr>
</tbody>
</table>

Value

a list containing the clam weights, temperature limitation functions and metabolic rates at time t

ClamF_pop_loop  

Function that runs the Monte Carlo simulation for the Clam population model (alternative version)

Description

Function that runs the Monte Carlo simulation for the Clam population model (alternative version)

Usage

ClamF_pop_loop(Param, times, IC, Tint, Chlint, N, userpath)

Arguments

<table>
<thead>
<tr>
<th>Param</th>
<th>a vector containing model parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>times</td>
<td>integration extremes and integration timestep</td>
</tr>
<tr>
<td>IC</td>
<td>initial condition</td>
</tr>
<tr>
<td>Tint</td>
<td>the interpolated water temperature time series</td>
</tr>
<tr>
<td>Chlint</td>
<td>the interpolated chlorophyll a time series</td>
</tr>
<tr>
<td>N</td>
<td>time series with number of individuals</td>
</tr>
<tr>
<td>userpath</td>
<td>the path where the working folder is located</td>
</tr>
</tbody>
</table>
ClamF_pop_main  
Clam bioenergetic population model (alternative version)

Description
Clam bioenergetic population model (alternative version)

Usage
ClamF_pop_main(userpath, forcings)

Arguments
userpath  the path where the working folder is located
forcings  a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3]

Value
A list containing model outputs: weights, temperature limitation functions and metabolic rates

ClamF_pop_post  
Postprocess the Clam population bioenergetic model results (alternative model)

Description
Postprocess the Clam population bioenergetic model results (alternative model)

Usage
ClamF_pop_post(userpath, output, times, Dates, N, CS)

Arguments
userpath  the path where the working folder is located
output  output list containing the output of the RK solver
times  the vector containing informations on integration extremes
Dates  the vector containing the date
N  the number of individuals
CS  the commercial size of Clam
**ClamF_pop_pre**

**Value**

a list containing the clam weights, temperature limitation functions and metabolic rates

**Description**

Clam bioenergetic population model (alternative version) preprocessor

**Usage**

ClamF_pop_pre(userpath, forcings)

**Arguments**

- **userpath**
  the path where folder containing model inputs and outputs is located
- **forcings**
  a list containing model forcings

**Value**

a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m³]

**ClamF_pop_RKsolver**

Solves the Clam bioenergetic balance (alternative version) with a 4th order Runge Kutta method

**Description**

Solves the Clam bioenergetic balance (alternative version) with a 4th order Runge Kutta method

**Usage**

ClamF_pop_RKsolver(Param, times, IC, Tint, Chlint)

**Arguments**

- **Param**
  a vector containing model parameters
- **times**
  integration extremes and integration timestep
- **IC**
  initial condition on weight
- **Tint**
  the interpolated water temperature time series
- **Chlint**
  the interpolated chlorophyll a time series
ClamF_pop_skeleton  

*Description*

Creates the folders structure for Clam population model (alternative version)

*Usage*

ClamF_pop_skeleton(userpath)

*Arguments*

userpath  
the path where forcing are located

---

Clam_ind_dataloader  

*Description*

Function that loads forcings data for Clam individual bioenergetic model and performs the interpolation

*Usage*

Clam_ind_dataloader(userpath)

*Arguments*

userpath  
the path where folder containing model inputs and outputs is located

*Value*

a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l], particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]
**Clam_ind_equations**

**Description**

Clam bioenergetic individual model differential equations

**Usage**

```r
Clam_ind_equations(Param, Tint, Phy, DT, POCint, POMint, TSSint, Wd)
```

**Arguments**

- **Param**
  - a vector containing model parameters
- **Tint**
  - the interpolated water temperature at time t
- **Phy**
  - the interpolated phytoplankton at time t
- **DT**
  - the interpolated detritus at time t
- **POCint**
  - the interpolated POC at time t
- **POMint**
  - the interpolated POM at time t
- **TSSint**
  - the interpolated TSS at time t
- **Wd**
  - the weight of the clam at time t

**Value**

a list containing the clam weights, temperature limitation functions and metabolic rates at time t

**Clam_ind_main**

**Description**

Clam bioenergetic individual model

**Usage**

```r
Clam_ind_main(userpath, forcings)
```

**Arguments**

- **userpath**
  - the path where the working folder is located
- **forcings**
  - a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l], particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]
Clam_ind_post

Postprocess the Clam individual bioenergetic model results

Description

Postprocess the Clam individual bioenergetic model results

Usage

Clam_ind_post(userpath, output, times, Dates, CS)

Arguments

- userpath: the path where the working folder is located
- output: output list containing the output of the RK solver
- times: the vector containing informations on integration extremes
- Dates: the vector containing the date
- CS: the commercial size of Clam

Value

A list containing model outputs: weights, temperature limitation functions and metabolic rates

Clam_ind_pre

Clam bioenergetic individual model preprocessor

Description

Clam bioenergetic individual model preprocessor

Usage

Clam_ind_pre(userpath, forcings)

Arguments

- userpath: the path where folder containing model inputs and outputs is located
- forcings: a list containing model forcings
Value

a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l], particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]

clam_ind_RKsolver

Solves the Clam individual bioenergetic balance with a 4th order Runge Kutta method

Description

Solves the Clam individual bioenergetic balance with a 4th order Runge Kutta method

Usage

clam_ind_RKsolver(Param, times, IC, Tint, Phyint, DTint, POCint, POMint, TSSint)

Arguments

Param a vector containing model parameters
times integration extremes and integration timestep
IC initial condition
Tint the interpolated water temperature time series
Phyint the interpolated phytoplankton time series
DTint the interpolated detritus time series
POCint the interpolated POC time series
POMint the interpolated POM time series
TSSint the interpolated TSS time series

Value

a list containing the clam weights, temperature limitation functions and metabolic rates
Clam_ind_skeleton

Description
Creates the folders structure for Clam individual bioenergetic model

Usage
Clam_ind_skeleton(userpath)

Arguments
userpath the path where forcing are located

Clam_pop_dataloader

Description
Function that loads forcings data for Clam population model and performs the interpolation

Usage
Clam_pop_dataloader(userpath)

Arguments
userpath the path where folder containing model inputs and outputs is located

Value
a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChla/m^3], particulated organic carbon (POC) concentration [mgC/l], particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]
**Clam_pop_equations**

*Clam bioenergetic population model differential equations*

**Description**

Clam bioenergetic population model differential equations

**Usage**

```r
Clam_pop_equations(Param, Tint, Phy, DT, POCint, POMint, TSSint, Wd)
```

**Arguments**

- `Param`: a vector containing model parameters
- `Tint`: the interpolated water temperature at time t
- `Phy`: the interpolated phytoplankton at time t
- `DT`: the interpolated detritus at time t
- `POCint`: the interpolated POC at time t
- `POMint`: the interpolated POM at time t
- `TSSint`: the interpolated TSS at time t
- `Wd`: the weight of the clam at time t

**Value**

a list containing the clam weights, temperature limitation functions and metabolic rates at time t

---

**Clam_pop_loop**

*Function that runs the Monte Carlo simulation for the Clam population model*

**Description**

Function that runs the Monte Carlo simulation for the Clam population model

**Usage**

```r
Clam_pop_loop(Param, times, IC, Tint, Phyint, DTint, POCint, POMint, TSSint, N, userpath)
```
Arguments

Param a vector containing model parameters
times integration extremes and integration timestep
IC initial condition
Tint the interpolated water temperature time series
Phyint the interpolated phytoplankton time series
DTint the interpolated detritus time series
POCint the interpolated POC time series
POMint the interpolated POM time series
TSSint the interpolated TSS time series
N time series with number of individuals
userpath the path where the working folder is located

Value

a list with RK solver outputs

---

Clam_pop_main Clam bioenergetic population model

Description

Clam bioenergetic population model

Usage

Clam_pop_main(userpath, forcings)

Arguments

userpath the path where the working folder is located
forcings a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l], particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]

Value

A list containing model outputs: weights, temperature limitation functions and metabolic rates
Clam_pop_post

Postprocess the Clam population bioenergetic model results

Description

Postprocess the Clam population bioenergetic model results

Usage

Clam_pop_post(userpath, output, times, Dates, N, CS)

Arguments

- userpath: the path where the working folder is located
- output: output list containing the output of the RK solver
- times: the vector containing informations on integration extremes
- Dates: the vector containing the date
- N: the number of individuals
- CS: the commercial size of Clam

Value

a list containing the clam weights, temperature limitation functions and metabolic rates

Clam_pop_pre

Clam bioenergetic population model preprocessor

Description

Clam bioenergetic population model preprocessor

Usage

Clam_pop_pre(userpath, forcings)

Arguments

- userpath: the path where folder containing model inputs and outputs is located
- forcings: a list containing model forcings

Value

a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m³], particulated organic carbon (POC) concentration [mgC/l], particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]
Clam_pop_RKsolver  
*Solves the Clam bioenergetic balance for population with a 4th order Runge Kutta method*

Description

Solves the Clam bioenergetic balance for population with a 4th order Runge Kutta method

Usage

Clam_pop_RKsolver(Param, times, IC, Tint, Phyint, DTint, POCint, POMint, TSSint)

Arguments

- **Param**  
  a vector containing model parameters

- **times**  
  integration extremes and integration timestep

- **IC**  
  initial condition

- **Tint**  
  the interpolated water temperature time series

- **Phyint**  
  the interpolated phytoplankton time series

- **DTint**  
  the interpolated detritus time series

- **POCint**  
  the interpolated POC time series

- **POMint**  
  the interpolated POM time series

- **TSSint**  
  the interpolated TSS time series

Value

a list containing the clam weights, temperature limitation functions and metabolic rates

Clam_pop_skeleton  
*Creates the folders structure for Clam population model*

Description

Creates the folders structure for Clam population model

Usage

Clam_pop_skeleton(userpath)

Arguments

- **userpath**  
  the path where forcing are located
**Mussel_ind_dataloader**

*Function that loads forcings data for Mussel individual bioenergetic model and performs the interpolation*

**Description**

Function that loads forcings data for Mussel individual bioenergetic model and performs the interpolation

**Usage**

```
Mussel_ind_dataloader(userpath)
```

**Arguments**

- **userpath**

  the path where folder containing model inputs and outputs is located

**Value**

a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m³], particulated organic carbon (POC) concentration [mgC/l] and its characterization in terms of C/P and N/P molar ratios, particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]

---

**Mussel_ind_equations**

*Mussel bioenergetic individual model differential equations*

**Description**

Mussel bioenergetic individual model differential equations

**Usage**

```
Mussel_ind_equations(Param, Tint, Phyint, DTint, POCint, Ccont, Ncont, Pcont, POMint, TSSint, Wb, R, t, trip)
```

**Arguments**

- **Param**

  a vector containing model parameters

- **Tint**

  the interpolated water temperature at time t

- **Phyint**

  the interpolated phytoplankton at time t

- **DTint**

  the interpolated detritus at time t

- **POCint**

  the interpolated POC at time t
Mussel_ind_main

\begin{itemize}
\item \texttt{Ccont} \hspace{1cm} \text{the C/C content of the POC at time t}
\item \texttt{Ncont} \hspace{1cm} \text{the N/C content of POC at time t}
\item \texttt{Pcont} \hspace{1cm} \text{the P/C content of POC at time t}
\item \texttt{POMint} \hspace{1cm} \text{the interpolated POM at time t}
\item \texttt{TSSint} \hspace{1cm} \text{the interpolated TSS at time t}
\item \texttt{Wb} \hspace{1cm} \text{the somatic tissue dry weight at time t}
\item \texttt{R} \hspace{1cm} \text{the gonadatic tissue dry weight at time t}
\item \texttt{t} \hspace{1cm} \text{the time}
\item \texttt{trip} \hspace{1cm} \text{vector containing the flags with resting periods}
\end{itemize}

**Value**

\text{the outputs at time t}

---

**Mussel_ind_main** \hspace{1cm} **Mussel bioenergetic individual model**

---

**Description**

Solves the bioenergetic balance for Mussel

**Usage**

\texttt{Mussel\_ind\_main(userpath, forcings)}

**Arguments**

\begin{itemize}
\item \texttt{userpath} \hspace{1cm} \text{the path where the working folder is located}
\item \texttt{forcings} \hspace{1cm} \text{a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l] and its characterization in terms of C/P and N/P molar ratios, particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]}
\end{itemize}

**Value**

\text{A list containing model outputs: weight, length mussel CNP, pseudofaeces CNP production, faeces CNP production, temperature limitation functions, metabolic rates and oxygen consumption}
Mussel_ind_post

Postprocess the Mussel individual bioenergetic model results

Description

Postprocess the Mussel individual bioenergetic model results

Usage

Mussel_ind_post(userpath, output, times, Dates, CS)

Arguments

userpath  the path where the working folder is located
output  output list containing the output of the RK solver
times  the vector containing informations on integration extremes
Dates  the vector containing the date
CS  the commercial size of Mussel

Value

a list containing the weights of the mussel, the excreted CNP, the mussel CNP, temperature limitation functions, metabolic rates, oxygen consumption

Mussel_ind_pre

Mussel bioenergetic individual model preprocessor

Description

Mussel bioenergetic individual model preprocessor

Usage

Mussel_ind_pre(userpath, forcings)

Arguments

userpath  the path where folder containing model inputs and outputs is located
forcings  a list containing model forcings

Value

a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l] and its characterization in terms of C/P and N/P molar ratios, particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]
Mussel_ind_RKsolver

Solves the Mussel individual bioenergetic balance with a 4th order Runge Kutta method

Description

Solves the Mussel individual bioenergetic balance with a 4th order Runge Kutta method

Usage

Mussel_ind_RKsolver(Param, times, IC, Tint, Phyint, DTint, POCint, Ccont, Ncont, Pcont, POMint, TSSint)

Arguments

<table>
<thead>
<tr>
<th>Param</th>
<th>a vector containing model parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>times</td>
<td>integration extremes and integration timestep</td>
</tr>
<tr>
<td>IC</td>
<td>initial condition</td>
</tr>
<tr>
<td>Tint</td>
<td>the interpolated water temperature time series</td>
</tr>
<tr>
<td>Phyint</td>
<td>the interpolated phytoplankton time series</td>
</tr>
<tr>
<td>DTint</td>
<td>the interpolated detritus time series</td>
</tr>
<tr>
<td>POCint</td>
<td>the interpolated POC time series</td>
</tr>
<tr>
<td>Ccont</td>
<td>the C/C content of the POC</td>
</tr>
<tr>
<td>Ncont</td>
<td>the N/C content of POC</td>
</tr>
<tr>
<td>Pcont</td>
<td>the P/C content of POC</td>
</tr>
<tr>
<td>POMint</td>
<td>the interpolated POM time series</td>
</tr>
<tr>
<td>TSSint</td>
<td>the interpolated TSS time series</td>
</tr>
</tbody>
</table>

Value

A list containing the weights of the mussel, the excreted CNP, the mussel CNP, temperature limitation functions, metabolic rates, oxygen consumption
**Mussel_ind_skeleton**  
*Creates the folders structure for Mussel individual bioenergetic model*

**Description**

Creates the folders structure for Mussel individual bioenergetic model

**Usage**

```python
Mussel_ind_skeleton(userpath)
```

**Arguments**

- `userpath`  
  the path where forcing are located

---

**Mussel_pop_dataloader**  
*Function that loads forcings data for Mussel population model and performs the interpolation*

**Description**

Function that loads forcings data for Mussel population model and performs the interpolation

**Usage**

```python
Mussel_pop_dataloader(userpath)
```

**Arguments**

- `userpath`  
  the path where folder containing model inputs and outputs is located

**Value**

a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l] and its characterization in terms of C/P and N/P molar ratios, particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]
Mussel_pop_equations    Mussel bioenergetic population model differential equations

Description

Mussel bioenergetic population model differential equations

Usage

Mussel_pop_equations(Param, N, Tint, Phyint, DTint, POCint, Ccont, Ncont, Pcont, POMint, TSSint, Wb, R, t, trip)

Arguments

Param  a vector containing model parameters
N  the number of individuals at time t
Tint  the interpolated water temperature at time t
Phyint  the interpolated phytoplankton at time t
DTint  the interpolated detritus at time t
POCint  the interpolated POC at time t
Ccont  the C/C content of the POC at time t
Ncont  the N/C content of POC at time t
Pcont  the P/C content of POC at time t
POMint  the interpolated POM at time t
TSSint  the interpolated TSS at time t
Wb  the somatic tissue dry weight at time t
R  the gondadic tissue dry weight at time t
t  the time
trip  vector containing the flags with resting periods

Value

the outputs at time t
**Mussel_pop_loop**

*Function that runs the Monte Carlo simulation for the Mussel population model*

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function that runs the Monte Carlo simulation for the Mussel population model</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Mussel_pop_loop(Param, times, IC, Tint, Phyint, DTint, POCint, Ccont, Ncont, Pcont, POMint, TSSint, N, userpath)</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Param</code></td>
</tr>
<tr>
<td><code>times</code></td>
</tr>
<tr>
<td><code>IC</code></td>
</tr>
<tr>
<td><code>Tint</code></td>
</tr>
<tr>
<td><code>Phyint</code></td>
</tr>
<tr>
<td><code>DTint</code></td>
</tr>
<tr>
<td><code>POCint</code></td>
</tr>
<tr>
<td><code>Ccont</code></td>
</tr>
<tr>
<td><code>Ncont</code></td>
</tr>
<tr>
<td><code>Pcont</code></td>
</tr>
<tr>
<td><code>POMint</code></td>
</tr>
<tr>
<td><code>TSSint</code></td>
</tr>
<tr>
<td><code>N</code></td>
</tr>
<tr>
<td><code>userpath</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a list with RK solver outputs</td>
</tr>
</tbody>
</table>
**Mussel_pop_main**  *Mussel bioenergetic population model*

**Description**
Solves the bioenergetic balance for Mussel and simulates a population

**Usage**

```python
Mussel_pop_main(userpath, forcings)
```

**Arguments**
- `userpath`: the path where the working folder is located
- `forcings`: a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l] and its characterization in terms of C/P and N/P molar ratios, particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]

**Value**
A list containing model outputs: weight, length mussel CNP, pseudofecies CNP production, temperature limitation functions, metabolic rates and oxygen consumption

---

**Mussel_pop_post**  *Postprocess the Mussel population bioenergetic model results*

**Description**
Postprocess the Mussel population bioenergetic model results

**Usage**

```python
Mussel_pop_post(userpath, output, times, Dates, N, CS)
```

**Arguments**
- `userpath`: the path where the working folder is located
- `output`: output list containing the output of the RK solver
- `times`: the vector containing informations on integration extremes
- `Dates`: the vector containing the date
- `N`: the number of individuals
- `CS`: the commercial size of Seabass
Mussel_pop_pre

Value
a list containing the weights of the mussel, the excreted CNP, the mussel CNP, temperature limitation functions, metabolic rates, oxygen consumption

Mussel bioenergetic population model preprocessor

Description
Mussel bioenergetic population model preprocessor

Usage
Mussel_pop_pre(userpath, forcings)

Arguments
userpath the path where folder containing model inputs and outputs is located
forcings a list containing forcings used by the model

Value
a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l] and its characterization in terms of C/P and N/P molar ratios, particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]

Mussel_pop_RKsolver
Solves the Mussel population bioenergetic balance with a 4th order Runge Kutta method

Description
Solves the Mussel population bioenergetic balance with a 4th order Runge Kutta method

Usage
Mussel_pop_RKsolver(Param, times, IC, Tint, Phyint, DTint, POCint, Ccont, Ncont, Pcont, POMint, TSSint, N)
Arguments

Param  a vector containing model parameters
times  integration extremes and integration timestep
IC    initial condition
Tint  the interpolated water temperature time series
Phyint  the interpolated phytoplankton time series
DTint  the interpolated detritus time series
POCint  the interpolated POC time series
Ccont  the C/C content of the POC
Ncont  the N/C content of POC
Pcont  the P/C content of POC
POMint  the interpolated POM time series
TSSint  the interpolated TSS time series
N    the number of individuals time series

Value

a list containing the weights of the mussel, the excreted CNP, the mussel CNP, temperature limitation functions, metabolic rates, oxygen consumption

Mussel_pop_skeleton  Creates the folders structure for Mussel population model

Description

Creates the folders structure for Mussel population model

Usage

Mussel_pop_skeleton(userpath)

Arguments

userpath  the path where forcing are located
**Mussel_spatial_dataloader**

*Function that loads forcings data for Mussel spatialized model and performs the interpolation*

**Description**

Function that loads forcings data for Mussel spatialized model and performs the interpolation

**Usage**

```python
Mussel_spatial_dataloader(userpath)
```

**Arguments**

- `userpath` the path where folder containing model inputs and outputs is located

**Value**

A list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l] and its characterization in terms of C/P and N/P molar ratios, particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]

**Mussel_spatial_loop**

*Mussel bioenergetic spatialized model - spatialization loop*

**Description**

Solves the bioenergetic balance for Mussel

**Usage**

```python
Mussel_spatial_loop(userpath, forcings)
```

**Arguments**

- `userpath` the path where the working folder is located
- `forcings` a list containing the time series in the odd positions and related forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l] and its characterization in terms of C/P and N/P molar ratios, particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]
Mussel_spatial_main  

*Mussel bioenergetic spatialized model - spatialization loop*

**Description**

Solves the bioenergetic balance for Mussel

**Usage**

Mussel_spatial_main(userpath, forcings)

**Arguments**

- **userpath**: the path where the working folder is located
- **forcings**: a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l] and its characterization in terms of C/P and N/P molar ratios, particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]

**Value**

saves .nc; .csv and .asc outputs in the 'Outputs' folder

Mussel_spatial_post  

*Postprocess the Mussel spatialized model results*

**Description**

Postprocess the Mussel spatialized model results

**Usage**

Mussel_spatial_post(userpath, output, times, Dates, CS)

**Arguments**

- **userpath**: the path where the working folder is located
- **output**: output list containing the output of the RK solver
- **times**: the vector containing informations on integration extremes
- **Dates**: the vector containing the date
- **CS**: the commercial size of Mussel
Mussel_spatial_pre

**Value**

a list containing the weights of the mussel, the excreted CNP, the mussel CNP, temperature limitation functions, metabolic rates, oxygen consumption

---

Mussel_spatial_pre

*Mussel bioenergetic spatialized model preprocessor*

**Description**

Mussel bioenergetic spatialized model preprocessor

**Usage**

Mussel_spatial_pre(userpath, forcings)

**Arguments**

userpath the path where folder containing model inputs and outputs is located
forcings a list containing forcings used by the model

**Value**

a list containing the time series in the odd positions and realted forcings in the even positions. Forcings returned are: Water temperature [Celsius degrees], Chlorophyll a concentration [mgChl-a/m^3], particulated organic carbon (POC) concentration [mgC/l] and its characterization in terms of C/P and N/P molar ratios, particulated organic matter (POM) concentration [mgC/l], total suspended solids (TSS) concentration [mg/l]

---

Mussel_spatial_pre_int

*Mussel bioenergetic spatialized model preprocessor - used inside spatialization loop*

**Description**

Mussel bioenergetic spatialized model preprocessor - used inside spatialization loop

**Usage**

Mussel_spatial_pre_int(userpath, forcings)

**Arguments**

userpath the path where folder containing model inputs and outputs is located
forcings a list containing forcings used by the model
Mussel_spatial_RKsolver

Solves the Mussel individual bioenergetic balance with a 4th order Runge Kutta method for spatialized model

Description

Solves the Mussel individual bioenergetic balance with a 4th order Runge Kutta method for spatialized model

Usage

Mussel_spatial_RKsolver(Param, times, IC, Tint, Phyint, DTint, POCint, Ccont, Ncont, Pcont, POMint, TSSint)

Arguments

- Param: a vector containing model parameters
- times: integration extremes and integration timestep
- IC: initial condition
- Tint: the interpolated water temperature time series
- Phyint: the interpolated phytoplankton time series
- DTint: the interpolated detritus time series
- POCint: the interpolated POC time series
- Ccont: the C/C content of the POC
- Ncont: the N/C content of POC
- Pcont: the P/C content of POC
- POMint: the interpolated POM time series
- TSSint: the interpolated TSS time series

Value

a list containing the weights of the mussel, the excreted CNP, the mussel CNP, temperature limitation functions, metabolic rates, oxygen consumption
**Mussel\_spatial\_skeleton**

*Creates the folders structure for Mussel spatialized model*

---

**Description**

Creates the folders structure for Mussel spatialized model

**Usage**

```r
Mussel\_spatial\_skeleton(userpath)
```

**Arguments**

- **userpath**: the path where forcing are located

---

**Pop\_fun**

*Function that solves the population dynamics equations including discontinuities*

---

**Description**

Function that solves the population dynamics equations including discontinuities

**Usage**

```r
Pop\_fun(Nseed, mort, manag, times)
```

**Arguments**

- **Nseed**: number of seeded individuals
- **mort**: mortality rate
- **manag**: list of management actions (seeded/harvested individuals)
- **times**: vector containing informations on integration times

**Value**

a time series with the number of individuals
Index

Bass_ind_dataloader, 4
Bass_ind_equations, 4
Bass_ind_main, 5
Bass_ind_post, 5
Bass_ind_pre, 6
Bass_ind_RKsolver, 6
Bass_ind_skeleton, 7
Bass_pop_dataloader, 7
Bass_pop_equations, 8
Bass_pop_loop, 8
Bass_pop_main, 9
Bass_pop_post, 9
Bass_pop_pre, 10
Bass_pop_RKsolver, 10
Bass_pop_skeleton, 11
Bass_spatial_dataloader, 11
Bass_spatial_loop, 12
Bass_spatial_main, 12
Bass_spatial_post, 13
Bass_spatial_pre, 13
Bass_spatial_pre_int, 14
Bass_spatial_RKsolver, 14
Bass_spatial_skeleton, 15
Bream_ind_dataloader, 15
Bream_ind_equations, 16
Bream_ind_main, 16
Bream_ind_post, 17
Bream_ind_pre, 17
Bream_ind_RKsolver, 18
Bream_ind_skeleton, 18
Bream_pop_dataloader, 19
Bream_pop_equations, 19
Bream_pop_loop, 20
Bream_pop_main, 20
Bream_pop_post, 21
Bream_pop_pre, 21
Bream_pop_RKsolver, 22
Bream_pop_skeleton, 22
Bream_spatial_dataloader, 23
Bream_spatial_loop, 23
Bream_spatial_main, 24
Bream_spatial_post, 24
Bream_spatial_pre, 25
Bream_spatial_pre_int, 25
Bream_spatial_RKsolver, 26
Bream_spatial_skeleton, 26
Clam_ind_dataloader, 34
Clam_ind_equations, 35
Clam_ind_main, 35
Clam_ind_post, 36
Clam_ind_pre, 36
Clam_ind_RKsolver, 37
Clam_ind_skeleton, 38
Clam_pop_dataloader, 38
Clam_pop_equations, 39
Clam_pop_loop, 39
Clam_pop_main, 40
Clam_pop_post, 41
Clam_pop_pre, 41
Clam_pop_RKsolver, 42
Clam_pop_skeleton, 42
ClamF_ind_dataloader, 27
ClamF_ind_equations, 27
ClamF_ind_main, 28
ClamF_ind_post, 28
ClamF_ind_pre, 29
ClamF_ind_RKsolver, 29
ClamF_ind_skeleton, 30
ClamF_pop_dataloader, 30
ClamF_pop_equations, 31
ClamF_pop_loop, 31
ClamF_pop_main, 32
ClamF_pop_post, 32
ClamF_pop_pre, 33
ClamF_pop_RKsolver, 33
ClamF_pop_skeleton, 34
Mussel_ind_dataloader, 43
>> Mussel_ind_equations, 43
Mussel_ind_main, 44
Mussel_ind_post, 45
Mussel_ind_pre, 45
Mussel_ind_RKsolver, 46
Mussel_ind_skeleton, 47
Mussel_pop_dataloader, 47
Mussel_pop_equations, 48
Mussel_pop_loop, 49
Mussel_pop_main, 50
Mussel_pop_post, 50
Mussel_pop_pre, 51
Mussel_pop_RKsolver, 51
Mussel_pop_skeleton, 52
Mussel_spatial_dataloader, 53
Mussel_spatial_loop, 53
Mussel_spatial_main, 54
Mussel_spatial_post, 54
Mussel_spatial_pre, 55
Mussel_spatial_pre_int, 55
Mussel_spatial_RKsolver, 56
Mussel_spatial_skeleton, 57

Pop_fun, 57