Package ‘rfishbase’

March 1, 2019

Title  R Interface to 'FishBase'

Description  A programmatic interface to <http://www.fishbase.org>, re-written based on an accompanying 'RESTful' API. Access tables describing over 30,000 species of fish, their biology, ecology, morphology, and more. This package also supports experimental access to <http://www.sealifebase.org> data, which contains nearly 200,000 species records for all types of aquatic life not covered by 'FishBase.'

Version  3.0.0.1

Encoding  UTF-8

License  CC0

URL  https://github.com/ropensci/rfishbase

BugReports  https://github.com/ropensci/rfishbase/issues

LazyData  true

Depends  R (>= 3.0)

Imports  dplyr, purrr, methods, utils, memoise, readr, rlang, magrittr, stringr

Suggests  devtools, testthat, rmarkdown, knitr

VignetteBuilder  knitr

RoxygenNote  6.1.0

NeedsCompilation  no

Author  Carl Boettiger [cre, aut] (<https://orcid.org/0000-0002-1642-628X>),
        Scott Chamberlain [aut] (<https://orcid.org/0000-0003-1444-9135>),
        Duncan Temple Lang [aut],
        Peter Wainwright [aut]

Maintainer  Carl Boettiger <cboettig@gmail.com>

StagedInstall  no

Repository  CRAN

Date/Publication  2019-03-01 07:07:16 UTC
### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>rfishbase-package</td>
<td>3</td>
</tr>
<tr>
<td>brains</td>
<td>3</td>
</tr>
<tr>
<td>common_names</td>
<td>4</td>
</tr>
<tr>
<td>common_to_sci</td>
<td>5</td>
</tr>
<tr>
<td>country</td>
<td>6</td>
</tr>
<tr>
<td>countrysub</td>
<td>6</td>
</tr>
<tr>
<td>countrysubref</td>
<td>7</td>
</tr>
<tr>
<td>c_code</td>
<td>8</td>
</tr>
<tr>
<td>diet</td>
<td>8</td>
</tr>
<tr>
<td>distribution</td>
<td>9</td>
</tr>
<tr>
<td>docs</td>
<td>10</td>
</tr>
<tr>
<td>ecology</td>
<td>11</td>
</tr>
<tr>
<td>ecosystem</td>
<td>12</td>
</tr>
<tr>
<td>estimate</td>
<td>13</td>
</tr>
<tr>
<td>faoareas</td>
<td>14</td>
</tr>
<tr>
<td>fecundity</td>
<td>15</td>
</tr>
<tr>
<td>fishbase</td>
<td>15</td>
</tr>
<tr>
<td>fooditems</td>
<td>16</td>
</tr>
<tr>
<td>genetics</td>
<td>17</td>
</tr>
<tr>
<td>introductions</td>
<td>17</td>
</tr>
<tr>
<td>larvae</td>
<td>18</td>
</tr>
<tr>
<td>length_freq</td>
<td>19</td>
</tr>
<tr>
<td>length_length</td>
<td>20</td>
</tr>
<tr>
<td>length_weight</td>
<td>21</td>
</tr>
<tr>
<td>list_fields</td>
<td>22</td>
</tr>
<tr>
<td>load_taxa</td>
<td>23</td>
</tr>
<tr>
<td>maturity</td>
<td>23</td>
</tr>
<tr>
<td>morphology</td>
<td>24</td>
</tr>
<tr>
<td>morphometrics</td>
<td>25</td>
</tr>
<tr>
<td>occurrence</td>
<td>26</td>
</tr>
<tr>
<td>oxygen</td>
<td>26</td>
</tr>
<tr>
<td>popchar</td>
<td>27</td>
</tr>
<tr>
<td>popgrowth</td>
<td>28</td>
</tr>
<tr>
<td>popqb</td>
<td>29</td>
</tr>
<tr>
<td>predators</td>
<td>30</td>
</tr>
<tr>
<td>ration</td>
<td>31</td>
</tr>
<tr>
<td>references</td>
<td>32</td>
</tr>
<tr>
<td>reproduction</td>
<td>32</td>
</tr>
<tr>
<td>sealifebase</td>
<td>33</td>
</tr>
<tr>
<td>spawning</td>
<td>34</td>
</tr>
<tr>
<td>species</td>
<td>34</td>
</tr>
<tr>
<td>species_fields</td>
<td>35</td>
</tr>
<tr>
<td>species_list</td>
<td>36</td>
</tr>
<tr>
<td>species_names</td>
<td>36</td>
</tr>
<tr>
<td>speed</td>
<td>37</td>
</tr>
<tr>
<td>stocks</td>
<td>38</td>
</tr>
</tbody>
</table>
Description

A programmatic interface to http://www.fishbase.org, re-written based on an accompanying 'RESTful' API. Access tables describing over 30,000 species of fish, their biology, ecology, morphology, and more. This package also supports experimental access to http://www.sealifebase.org data, which contains nearly 200,000 species records for all types of aquatic life not covered by FishBase.’

Author(s)

Carl Boettiger <carl@ropensci.org>
Scott Chamberlain <scott@ropensci.org>

Usage

brains(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.
... unused; for backwards compatibility only
common_names

Value

a table of species brains

Examples

```r
## Not run:
brains("Oreochromis niloticus")

## End(Not run)
```

<table>
<thead>
<tr>
<th>common_names</th>
<th>common names</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

Return a table of common names

Usage

```r
common_names(species_list, server = NULL, Language = NULL,
fields = NULL)
```

Arguments

- `species_list`: A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- `server`: can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.
- `Language`: a string specifying the language for the common name, e.g. "English"
- `fields`: a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

Details

Note that there are many common names for a given sci name, so sci_to_common doesn’t make sense

Value

a data.frame of common names by species queried. If multiple species are queried, The resulting data.frames are concatenated.
**common_to_sci**

**Examples**

```r
common_names(c("Labroides bicolor", "Bolbometopon muricatum"))
# subset by English language names
fish <- common_names("Bolbometopon muricatum")
```

---

**Description**

Return a list of scientific names corresponding to given the common name(s).

**Usage**

```r
common_to_sci(x, Language = NULL, ..., server = NULL)
```

**Arguments**

- `x` a common name or list of common names
- `Language` a string specifying the language for the common name, e.g. "English"
- `...` unused; for backwards compatibility only
- `server` can be set to either "fishbase" or "sealifebase" to switch between databases.

**Details**

If more than one scientific name matches the common name (e.g. "trout"), the function will simply return a list of all matching scientific names. If given more than one common name, the resulting strings of matching scientific names are simply concatenated.

**Value**

a character vector of scientific names

**See Also**

*species_list*, *synonyms*

**Examples**

```r
common_to_sci(c("Bicolor cleaner wrasse", "humphead parrotfish"), Language="English")
common_to_sci(c("Coho Salmon", "trout"))
```
### countrysub

**Description**

return a table of countrysub for the requested species

**Usage**

countrysub(species_list = NULL, fields = NULL, server = NULL, ...)

**Arguments**

- **species_list**: A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- **fields**: a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
- **server**: can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

**Details**

e.g. http://www.fishbase.us/Country

**Examples**

```r
## Not run:
countrysub(species_list(Genera='Labroides'))
## End(Not run)
```

---

### country

**Description**

return a table of country for the requested species, as reported in FishBASE.org

**Usage**

country(species_list = NULL, fields = NULL, server = NULL, ...)

**Arguments**

- **species_list**: A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- **fields**: a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
- **server**: can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

**Details**

e.g. http://www.fishbase.us/Country

**Examples**

```r
## Not run:
country(species_list(Genera='Labroides'))
## End(Not run)
```
countrysubref

Arguments

- **species_list**: A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- **fields**: A character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later.
- **server**: Can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. `Sys.setenv(FISHBASE_API="sealifebase")`.

Examples

```r
## Not run:
countrysub(species_list(Genus='Labroides'))

## End(Not run)
```

countrysubref countrysubref

description

return a table of countrysubref

Usage

```r
countrysubref(server = NULL)
```

Arguments

- **server**: Can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. `Sys.setenv(FISHBASE_API="sealifebase")`.

Examples

```r
## Not run:
countrysubref()

## End(Not run)
```
Description

return a table of country information for the requested c_code, as reported in FishBASE.org

Usage

c_code(c_code = NULL, server = NULL, ...)

Arguments

c_code  a C_Code or list of C_Codes (FishBase country code)
server  can be set to either "fishbase" or "sealifebase" to switch between databases.

NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

...  unused; for backwards compatibility only

Details

e.g. http://www.fishbase.us/Country

Examples

## Not run:
c_code(440)

## End(Not run)

diet  diet

Description

diet

Usage

diet(species_list = NULL, fields = NULL, server = NULL, ...)

```
distribution

Arguments

species_list A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

... unused; for backwards compatibility only

Value

a table of species diet

References


Examples

## Not run:
diet("Oreochromis niloticus")

## End(Not run)

distribution  distribution

description

return a table of species locations as reported in FishBASE.org FAO location data

Usage

distribution(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

... unused; for backwards compatibility only
Details

Currently this is ~ FAO areas table (minus "note" field) e.g. http://www.fishbase.us/Country/FaoAreaList.php?ID=5537

Examples

```r
## Not run:
distribution(species_list(Genus='Labroides'))

## End(Not run)
```

Description

documentation of tables and fields

Usage

docs(table = NULL, server = NULL, ...)

Arguments

- `table` the table for which the documentation should be displayed. If no table is given, documentation summarizing all available tables is shown.
- `server` can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.
- `...` unused; for backwards compatibility only

Value

A data.frame which lists the name of each table (if no table argument is given), along with a description of the table and a URL linking to further information about the table. If a specific table is named in the table argument, then the function will return a data.frame listing all the fields (columns) found in that table, a description of what the field label means, and the units in which the field is measured. These descriptions of the columns are not made available by FishBase.org and must be manually generated and curated by FishBase.org users. At this time, many fields are still missing. Please take a moment to fill in any fields you use in the source table here: https://github.com/ropensci/fishbaseapi/tree/master/docs/docs-sources
Examples

```r
tables <- docs()
# Describe the fecundity table
dplyr::filter(tables, table == "fecundity")$description
## See fields in fecundity table
docs("fecundity")
## Note: only
```

Description

ecology

Usage

```r
ecology(species_list = NULL, fields = NULL, server = NULL)
```

Arguments

- `species_list` A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- `fields` a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
- `server` can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

Details

By default, will only return one entry (row) per species. Increase limit to get multiple returns for different stocks of the same species, though often data is either identical to the first or simply missing in the additional stocks.

Value

a table of species ecology data

References

Examples

## Not run:
ecosystem("Oreochromis niloticus")

## trophic levels and standard errors for a list of species
ecology(c("Oreochromis niloticus", "Salmo trutta"),
       fields=c("SpecCode", "FoodTroph", "FoodSeTrohp", "DietTroph", "DietSeTrohp"))

## End(Not run)

ecosystem ecosystem

Description
ecosystem

Usage
ecosystem(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

- **species_list**: A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- **fields**: a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
- **server**: can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

... unused; for backwards compatibility only

Value

a table of species ecosystems data

Examples

## Not run:
ecosystem("Oreochromis niloticus")

## End(Not run)
**Description**

estimate

**Usage**

```r
estimate(species_list = NULL, fields = NULL, server = NULL, ...)```

**Arguments**

- `species_list` A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- `fields` a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
- `server` can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. `Sys.setenv(FISHBASE_API="sealifebase")`.

... unused; for backwards compatibility only

**Value**

a table of estimates from some models on trophic levels

**References**

http://www.fishbase.us/manual/English/FishbaseThe_FOOD_ITEMS_table.htm

**Examples**

```r
## Not run:
estimate("Oreochromis niloticus")

## End(Not run)```
**Description**

return a table of species locations as reported in FishBASE.org FAO location data

**Usage**

```r
faoareas(species_list = NULL, fields = NULL, server = NULL, ...)
```

**Arguments**

- `species_list` A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- `fields` a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
- `server` can be set to either "fishbase" or "sealifebase" to switch between databases.
  NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.
- `...` unused; for backwards compatibility only

**Details**

currently this is ~ FAO areas table (minus "note" field) e.g. http://www.fishbase.us/Country/FaoAreaList.php?ID=5537

**Value**

a tibble, empty tibble if no results found

**Examples**

```r
## Not run:
faoareas()

## End(Not run)
```
### Description

fecundity

### Usage

fecundity(species_list = NULL, fields = NULL, server = NULL, ...)

### Arguments

- **species_list**: A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- **fields**: a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later.
- **server**: can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.
- ... unused; for backwards compatibility only

### Value

A table of species fecundity

### Examples

```r
## Not run:
fecundity("Oreochromis niloticus")

## End(Not run)
```

### fishbase

* A table of all the the species found in FishBase, including taxonomic classification and the Species Code (SpecCode) by which the species is identified in FishBase.

### Description

A table of all the the species found in FishBase, including taxonomic classification and the Species Code (SpecCode) by which the species is identified in FishBase.
Author(s)
Carl Boettiger <carl@ropensci.org>

References
FishBase.org

---

### Description

**Description**

**fooditems**

### Usage

**Usage**

`fooditems(species_list = NULL, fields = NULL, server = NULL, ...)`

### Arguments

- **species_list**: A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- **fields**: A character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later.
- **server**: Can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

### Value

A table of species fooditems

### References

http://www.fishbase.org/manual/english/fishbase_the_food_items_table.htm

### Examples

```r
## Not run:
fooditems("Oreochromis niloticus")

## End(Not run)```
Description

genetics

Usage

genetics(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields       a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server       can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’. unused; for backwards compatibility only

Value

a table of species genetics data

Examples

## Not run:
genetics("Oreochromis niloticus")
genetics("Labroides dimidiatus")

## End(Not run)

introductions

Description

introductions

Usage

introductions(species_list = NULL, fields = NULL, server = NULL, ...)

Bitte beachten Sie, dass der Text in der Datei einige Symbole wie # verwendet wird, die in R-Code interpretiert werden. Der Code wird normalerweise nicht ausgeführt, sondern verwendet zur Erläuterung des Verhaltens des `genetics()`-Funktionen.
Arguments

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields  a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API=“sealifebase”)’.

... unused; for backwards compatibility only

Value

a table of species introductions data

Examples

```r
## Not run:
introductions("Oreochromis niloticus")

## End(Not run)
```

---

**Description**

larvae

**Usage**

`larvae(species_list = NULL, fields = NULL, server = NULL, ...)`

**Arguments**

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields  a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API=“sealifebase”)’.

... unused; for backwards compatibility only
Value

a table of larval data

Examples

```r
## Not run:
larvae("Oreochromis niloticus")

## End(Not run)
```

---

**Description**

return a table of species fooditems

Usage

```r
length_freq(species_list = NULL, fields = NULL, server = NULL, ...)
```

Arguments

- `species_list` A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- `fields` a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
- `server` can be set to either "fishbase" or "sealifebase" to switch between databases.
  NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’. e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.
  ... unused; for backwards compatibility only

Value

a table of length_freq information by species; see details

References

http://www.fishbase.org/manual/english/lengthfrequency.htm

Examples

```r
## Not run:
length_freq("Oreochromis niloticus")

## End(Not run)
```
Description

return a table of lengths

Usage

length_length(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields  a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server  can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

Details

This table contains relationships for the conversion of one length type to another for over 8,000 species of fish, derived from different publications, e.g. Moutopoulos and Stergiou (2002) and Gaygusuz et al (2006), or from fish pictures, e.g. Collette and Nauen (1983), Compagno (1984) and Randall (1997). The relationships, which always refer to centimeters, may consist either of a regression linking two length types, of the form: Length type (2) = a + b x Length type (1) Length type (2) = b' x Length type (1) The available length types are, as elsewhere in FishBase, TL = total length; FL = fork length; SL = standard length; WD = width (in rays); OT = other type (to be specified in the Comment field). When a version of equation (1) is presented, the length range, the number of fish used in the regression, the sex and the correlation coefficient are presented, if available. When a version of equation (2) is presented, the range and the correlation coefficient are omitted, as the ratio in (2) will usually be estimated from a single specimen, or a few fish covering a narrow range of lengths.

Value

a table of lengths

References

### Description

The LENGTH-WEIGHT table presents the $a$ and $b$ values of over 5,000 length-weight relationships of the form $W = a \times L^b$, pertaining to about 2,000 fish species.

### Usage

```r
length_weight(species_list = NULL, fields = NULL, server = NULL, ...)
```

### Arguments

- `species_list` A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- `fields` A character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later.
- `server` Can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

### Details

See references for official documentation. From FishBase.org: Length-weight relationships are important in fisheries science, notably to raise length-frequency samples to total catch, or to estimate biomass from underwater length observations. The units of length and weight in FishBase are centimeter and gram, respectively. Thus when length-weight relationships are not in cm-g, the intercept 'a' is transformed as follows:

$$a'(cm, g) = a(mm, g) \times 10^b$$

However, published length-weight relationships are sometimes difficult to use, as they may be based on a length measurement type (e.g., fork length) different from ones length measurements (expressed e.g., as total length). Therefore, to facilitate conversion between length types, an additional LENGTH-LENGTH table, # presented below, was devised which presents linear regressions or ratios linking length types (e.g., FL vs. TL). We included a calculated field with the weight of a 10 cm fish (which should be in the order of 10 g for normal, fusiform shaped fish), to allow identification of gross errors, given knowledge of the body form of a species.
Value

a table of length_weight information by species; see details

References


Examples

```r
## Not run:
length_weight("Oreochromis niloticus")
## End(Not run)
```

---

```r
table
```

<table>
<thead>
<tr>
<th>list_fields</th>
<th>list_fields</th>
</tr>
</thead>
</table>

Description

list fields

Usage

`list_fields(fields = NULL, server = NULL, implemented_only = TRUE)`

Arguments

- `fields` field (column name) to search for
- `server` base URL to the FishBase API (by default). For SeaLifeBase, use `https://fishbase.ropensci.org/sealifebase`
- `implemented_only` by default, only return those tables that have been implemented.

Details

Calling `list_fields()` with no arguments will return the full table of all known fields. Then users can employ standard filter techniques like grep for partial name matching; see examples.

Value

a data frame listing the table names (matching function names in rfishbase) and the matching column names those tables have implemented.
load_taxa

Examples

list_fields("Temp")

## Regex matching on full table
library(dplyr)
list_fields() %>% filter(grepl("length", columns, ignore.case = TRUE))

load_taxa  load_taxa

Description

load_taxa

Usage

load_taxa(server = NULL, ...)

Arguments

server  API for Fishbase or Sealifebase?
...  for compatibility with previous versions

Value

the taxa list

maturity  maturity

Description

maturity

Usage

maturity(species_list = NULL, fields = NULL, server = NULL, ...)
Arguments

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields       a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server       can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

... unused; for backwards compatibility only

Value

a table of species maturity

Examples

## Not run:
maturity("Oreochromis niloticus")

## End(Not run)

---

Description

morphology

Usage

morphology(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields       a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server       can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

... unused; for backwards compatibility only
morphometrics

Value

A table of species morphology data

Examples

## Not run:
morphometrics("Oreochromis niloticus")

## End(Not run)

Description

morphometrics

Usage

morphometrics(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields A character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later.

server Can be set to either "fishbase" or "sealifebase" to switch between databases.

Value

A table of species morphometrics data

Examples

## Not run:
morphometrics("Oreochromis niloticus")

## End(Not run)
Description

occurrence

Usage

occurrence()

Details

THE OCCURRENCE TABLE HAS BEEN DROPPED BY FISHBASE - THIS FUNCTION NOW RETURNS A STOP MESSAGE.

Description

oxygen

Usage

oxygen(species_list = NULL, fields = NULL, server = NULL, ...) 

Arguments

species_list A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

Value

a table of species oxygen data
**Description**

Table of maximum length (Lmax), weight (Wmax) and age (tmax)

**Usage**

```r
popchar(species_list = NULL, fields = NULL, server = NULL, ...)
```

**Arguments**

- `species_list`: A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- `fields`: A character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later.
- `server`: Can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

... unused; for backwards compatibility only

**Details**

See references for official documentation. From FishBase.org: This table presents information on maximum length (Lmax), weight (Wmax) and age (tmax) from various localities where a species occurs. The largest values from this table are also entered in the SPECIES table. The POPCHAR table also indicates whether the Lmax, Wmax and tmax values or various combinations thereof refer to the same individual fish.

**References**


**Examples**

```r
## Not run:
opchar("Oreochromis niloticus")

## End(Not run)
```
Description

This table contains information on growth, natural mortality and length at first maturity, which serve as inputs to many fish stock assessment models. The data can also be used to generate empirical relationships between growth parameters or natural mortality estimates, and their correlates (e.g., body shape, temperature, etc.), a line of research that is useful both for stock assessment and for increasing understanding of the evolution of life-history strategies.

Usage

```r
popgrowth(species_list = NULL, fields = NULL, server = NULL, ...)
```

Arguments

- `species_list`: A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- `fields`: A character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later.
- `server`: Can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.
- `...`: unused; for backwards compatibility only

Value

A table of population growth information by species; see details

References


Examples

```r
## Not run:
popgrowth("Oreochromis niloticus")

## End(Not run)
```
Description

popqb

Usage

popqb(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields  a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server  can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’. Unused; for backwards compatibility only

Value

a table of species popqb

References


Examples

## Not run:
popqb("Orechromis niloticus")

## End(Not run)
predators

Description

predators

Usage

predators(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’. ... unused; for backwards compatibility only

Value

a table of predators

References


Examples

## Not run:
predators("Oreochromis niloticus")

## End(Not run)
ration

Description

ration

Usage

ration(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’. ... unused; for backwards compatibility only

Value

a table of species ration

References


Examples

## Not run:
ration("Oreochromis niloticus")

## End(Not run)
reproduction

Description
references

Usage
references(codes = NULL, fields = NULL, server = NULL, ...)

Arguments
codes One or more Fishbase reference numbers, matching the RefNo field
fields a character vector specifying which fields (columns) should be returned. By
default, all available columns recognized by the parser are returned. Mostly for
backwards compatibility as users can subset by column later
server can be set to either "fishbase" or "sealifebase" to switch between databases.
NOTE: it is usually easier to leave this as NULL and set the source instead using
the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API=”sealifebase”).

Value
a tibble (data.frame) of reference data

Examples
## Not run:
 references(codes = 1)
 references(codes = 1:6)
 references(codes = 1:6, fields = c('Author', 'Year', 'Title'))
 references() # all references

## End(Not run)

reproduction

Description
reproduction

Usage
reproduction(species_list = NULL, fields = NULL, server = NULL, ...)

reproduction

Description
references

Usage
references(codes = NULL, fields = NULL, server = NULL, ...)

Arguments
codes One or more Fishbase reference numbers, matching the RefNo field
fields a character vector specifying which fields (columns) should be returned. By
default, all available columns recognized by the parser are returned. Mostly for
backwards compatibility as users can subset by column later
server can be set to either "fishbase" or "sealifebase" to switch between databases.
NOTE: it is usually easier to leave this as NULL and set the source instead using
the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API=”sealifebase”).

Value
a tibble (data.frame) of reference data

Examples
## Not run:
 references(codes = 1)
 references(codes = 1:6)
 references(codes = 1:6, fields = c('Author', 'Year', 'Title'))
 references() # all references

## End(Not run)
Arguments

species_list A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

... unused; for backwards compatibility only

Value

a table of species reproduction

Examples

```r
## Not run:
reproduction("Oreochromis niloticus")

## End(Not run)
```

sealifebase A table of all the species found in SeaLifeBase, including taxonomic classification and the Species Code (SpecCode) by which the species is identified in SeaLifeBase

Description

A table of all the species found in SeaLifeBase, including taxonomic classification and the Species Code (SpecCode) by which the species is identified in SeaLifeBase

Author(s)

Carl Boettiger <carl@ropensci.org>

References

www.sealifebase.org
spawning

Description

spawning

Usage

spawning(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

Value

a table of species spawning

Examples

## Not run:
spawning("Oreochromis niloticus")

## End(Not run)

species

Description

Provide wrapper to work with species lists.

Usage

species(species_list = NULL, fields = NULL, server = NULL, ...)


species_fields

Arguments

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fIELDS a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server  can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

... unused; for backwards compatibility only

Details

The Species table is the heart of FishBase. This function provides a convenient way to query, tidy, and assemble data from that table given an entire list of species. For details, see: http://www.fishbase.org/manual/english/fishbasethe_species_table.htm

Species scientific names are defined according to fishbase taxonomy and nomenclature.

Value

a data.frame with rows for species and columns for the fields returned by the query (FishBase 'species' table)

Examples

## Not run:

species(c("Labroides bicolor", "Bolbometopon muricatum"))
species(c("Labroides bicolor", "Bolbometopon muricatum"), fields = species_fields$habitat)

## End(Not run)

---

species_fields  A list of the species_fields available

Description

A list of the species_fields available

Author(s)

Carl Boettiger <carl@ropensci.org>

References

FishBase.org
species_list

Description

Return the a species list given a taxonomic group

Usage

species_list(Class = NULL, Order = NULL, Family = NULL,
Subfamily = NULL, Genus = NULL, Species = NULL, Speccode = NULL,
SuperClass = NULL, server = getOption("FISHBASE_API", FISHBASE_API))

Arguments

Class Request all species in this taxonomic Class
Order Request all species in this taxonomic Order
Family Request all species in this taxonomic Family
Subfamily Request all species in this taxonomic SubFamily
Genus Request all species in this taxonomic Genus
Species Request all species in this taxonomic Species
SpecCode Request species name of species matching this SpecCode
SuperClass Request all species of this Superclass
server fishbase or

Examples

## All species in the Family
species_list(Family = 'Scaridae')
## All species in the Genus
species_list(Genus = 'Labroides')

species_names

Description

returns species names given FishBase’s SpecCodes

Usage

species_names(codes, server = NULL)
speed

Arguments

- **codes**: a vector of speccodes (e.g. column from a table)
- **server**: can be set to either "fishbase" or "sealifebase" to switch between databases.
  
  **NOTE**: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

Value

A character vector of species names for the SpecCodes

<table>
<thead>
<tr>
<th>speed</th>
<th>speed</th>
</tr>
</thead>
</table>

Description

speed

Usage

speed(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

- **species_list**: A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
- **fields**: a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
- **server**: can be set to either "fishbase" or "sealifebase" to switch between databases.
  
  **NOTE**: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’.

Value

a table of species speed data

Examples

```r
## Not run:
speed("Oreochromis niloticus")

## End(Not run)
```
Description

stocks

Usage

stocks(species_list = NULL, fields = NULL, server = NULL, ...)

Arguments

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields  a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server  can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.

...  unused; for backwards compatibility only

Value

a table of species stocks data

Examples

## Not run:
stocks("Oreochromis niloticus")

## End(Not run)

Description

swimming

Usage

swimming(species_list = NULL, fields = NULL, server = NULL, ...)

swimming
Arguments

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

fields  a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

server  can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’. ... unused; for backwards compatibility only

Value

a table of species swimming data

Examples

```r
## Not run:
swimming("Oreochromis niloticus")

## End(Not run)
```

Description

Check for alternate versions of a scientific name

Usage

```r
synonyms(species_list = NULL, server = NULL, ...)
```

Arguments

species_list  A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.

server  can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")’. ... unused; for backwards compatibility only

Details

For further information on fields returned, see: http://www.fishbase.org/manual/english/fishbasethe_synonyms_table.htm
Value

A table with information about the synonym. Will generally be only a single row if a species name
is given. If a FishBase SpecCode is given, all synonyms matching that SpecCode are shown, and
the table indicates which one is Valid for FishBase. This may or may not match the valid name for
Catalog of Life (Col), also shown in the table. See examples for details.

Examples

# Query using a synonym:
synonyms("Callyodon muricatus")

# Check for misspellings or alternate names
synonyms("Labroides dimidatus") # Species name misspelled

# See all synonyms
species("Bolbometopon muricatum")

taxonomy  Taxonomy

Description

Taxonomy

Usage

taxonomy(genus = NULL, species = NULL, ...)

Arguments

genus, species (character) genus or species name. pass in either one or both per species). Func-
tion will warn if this needs to be increased, otherwise can be left as is.

... unused; for backwards compatibility only

Examples

taxonomy(genus = "Oreochromis", species = "amphimelas")
taxonomy(genus = "Oreochromis")
taxonomy(species = "amphimelas")
validate_names

validate_names validate_names

Description

Check for alternate versions of a scientific name and return the scientific names FishBase recognizes as valid

Usage

validate_names(species_list, server = NULL, ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>species_list</td>
<td>A vector of scientific names (each element as &quot;genus species&quot;). If empty, a table for all fish will be returned.</td>
</tr>
<tr>
<td>server</td>
<td>can be set to either &quot;fishbase&quot; or &quot;sealifebase&quot; to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API’, e.g. ‘Sys.setenv(FISHBASE_API=&quot;sealifebase&quot;)’.</td>
</tr>
</tbody>
</table>

... unused; for backwards compatibility only

Value

a string of the validated names

Examples

validate_names("Abramites ternetzi")
Index

*Topic data
  fishbase, 15
  sealifebase, 33
  species_fields, 35

*Topic package
  rfishbase-package, 3

brains, 3

brains, 3

c.code, 8
common_names, 4
common_to_sci, 5
country, 6
countrysub, 6
countrysubref, 7
diet, 8
distribution, 9
docs, 10

ecology, 11
ecosystem, 12
estimate, 13

faoareas, 14
fecundity, 15
fishbase, 15
fooditems, 16

genetics, 17

introductions, 17

larvae, 18
length_freq, 19
length_length, 20
length_weight, 21
list_fields, 22
load_taxa, 23

maturity, 23

morphology, 24
morphometrics, 25

occurrence, 26
oxygen, 26

popchar, 27
popgrowth, 28
poplf(length_freq), 19
popll(length_length), 20
poplw(length_weight), 21
popqb, 29
predators, 30

ration, 31
references, 32
reproduction, 32

rfishbase(rfishbase-package), 3
rfishbase-package, 3

sci_to_common(common_names), 4
sealifebase, 33
spawning, 34
species, 34
species_fields, 35
species_info(species), 34
species_list, 36
species_names, 36
speed, 37
stocks, 38
swimming, 38
synonyms, 39

taxonomy, 40

validate_names, 41