Package ‘BioIndex’

May 29, 2026

Type Package

Title Biological Indicators and Indices for MEDITS Survey Data

Version 0.6.4

Date 2026-04-29

Author Walter Zupa [aut, cre],
Loredana Casciaro [rev],
Cosmidano Neglia [rev],
Isabella Bitetto [rev],
Maria Teresa Spedicato [aut]

Maintainer Walter Zupa <`zupa@fondazionecoispa.org`>

Description Supports the standardized analysis of Mediterranean International Bottom Trawl Survey (MEDITS) data and the calculation of biological indicators for selected species and population components. The package provides functions to estimate abundance and biomass indices, analyse size structure and length frequency distributions, derive sex ratio and maturity related metrics, explore spatial patterns, and assess temporal trends across surveys. Developed for integration within the Regional Database for Fisheries (RDBFIS) framework, it is intended to work on quality checked input data and to produce reproducible outputs that can support monitoring, comparative analyses among Geographical Sub-Areas (GSAs) and countries, and fishery management.

License GPL-3

Encoding UTF-8

LazyData TRUE

LazyDataCompression xz

RoxygenNote 7.3.3

Depends R (>= 4.1)

Imports dplyr, ggplot2, gridExtra, hms, magrittr, marmap, methods, mgcv, reshape2, shiny, shinyjs, stringr, terra, tidyterra, zip

Suggests knitr, mapproj, maps (>= 3.4.1), rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2026-05-29 10:30:02 UTC

Contents

`aggregate_gsas` *Aggregate MEDITS data across multiple GSAs*

Description

Aggregate TA, TB, and TC MEDITS tables across a user-defined set of Geographical Sub-Areas (GSAs), and update the associated stratification objects accordingly.

Usage

```
aggregate_gsas(
  ta,
  tb,
  tc,
  gsas,
  strata_scheme = BioIndex::strata_scheme,
  stratification = BioIndex::stratification
)
```

Arguments

`ta`	A data frame containing the MEDITS `TA` table.
`tb`	A data frame containing the MEDITS `TB` table.
`tc`	A data frame containing the MEDITS `TC` table.
`gsas`	A numeric or character vector of GSA codes to be aggregated.
`strata_scheme`	A data frame describing the depth-strata scheme for the selected GSAs. Defaults to `BioIndex::strata_scheme`.

`stratification`	A data frame containing the stratification information for the selected GSAs. Defaults to `BioIndex::stratification`.

Details

This function filters the input TA, TB, and TC tables to retain only the selected GSAs, concatenates the original `AREA` and `HAUL_NUMBER` values to create unique haul identifiers across GSAs, and replaces the `AREA` field with a single aggregated GSA code obtained by collapsing the selected GSA values into one string. The same aggregation is applied to the `stratification` and `strata_scheme` tables.

The function performs three main operations:

1. Checks that all requested GSAs are present in the `AREA` column of the input `ta`, `tb`, and `tc` tables.
2. Filters the three MEDITS tables to the selected GSAs and updates `HAUL_NUMBER` by prefixing it with the original `AREA` value, ensuring uniqueness after aggregation.
3. Filters and updates the `stratification` and `strata_scheme` tables so that all selected GSAs are represented under a single aggregated GSA code.

The aggregated GSA code is created using `paste(gsas, collapse = "")`. For example, `c(17, 18)` becomes `"1718"`.

Value

A list with five elements:

[[1]] The filtered and aggregated `TA` table.

[[2]] The filtered and aggregated `TB` table.

[[3]] The filtered and aggregated `TC` table.

[[4]] The filtered and updated `stratification` table.

[[5]] The filtered and updated `strata_scheme` table.

See Also

`strata_scheme`, `stratification`

Examples

```
# Use internal data
data("strata_scheme", package = "BioIndex")
data("stratification", package = "BioIndex")
data("TA", package = "BioIndex")
data("TB", package = "BioIndex")
data("TC", package = "BioIndex")

d <- aggregate_gsas(
  ta = TA,
  tb = TB,
  tc = TC,
```

```
  gsas = 10,
  strata_scheme = BioIndex::strata_scheme,
  stratification = BioIndex::stratification
)

ta_agg <- d[[1]]
tb_agg <- d[[2]]
tc_agg <- d[[3]]
stratification_agg <- d[[4]]
strata_scheme_agg <- d[[5]]
```

`ALK` *ALK (Age Length Key)*

Description

Computes the Age-Length Key (ALK) from biological samples, a crucial parameter for converting length distributions into age classes and analyzing stock growth dynamics.

Usage

```
ALK(
  ta,
  te,
  sp,
  GSA,
  country = "all",
  nyears = NA,
  wd = NA,
  save = TRUE,
  verbose = FALSE
)
```

Arguments

`ta`	MEDITS or MEDITS-like TA table
`te`	MEDITS or MEDITS-like TE table
`sp`	species RUBIN code (MEDITS format, e.g. "MERLMER")
`GSA`	reference GSA for the analysis
`country`	reference country
`nyears`	number of years of the time series to be considered in the analysis
`wd`	path of the working directory
`save`	boolean. If TRUE the outputs are saved in the local folder
`verbose`	boolean. If TRUE messages are promted in the console

Value

A `data.frame` representing the Age-Length Key.

`ALKf` *Estimation of ALK*

Description

Estimation of ALK

Usage

```
ALKf(te, sp, GEAR, GSA, country = NA, years = 5, wd = NA, save = TRUE)
```

Arguments

`te`	MEDITS or MEDITS-like TE table
`sp`	species RUBIN code (MEDITS format, e.g. "MERLMER")
`GEAR`	type of gear reported in the corresponding TA file
`GSA`	reference GSA for the analysis
`country`	reference country
`years`	number of years to be considered in the analysis
`wd`	path of the working directory
`save`	boolean. If TRUE the outputs are saved in the local folder

Value

A `data.frame` representing the Age-Length Key for females.

`BioIndex` *Main function to perform BioIndex analysis*

Description

BioIndex is an R package designed to support the standardized analysis of MEDITS trawl survey data and the calculation of biological indicators for selected species and population components.

Usage

```
BioIndex(
  ta,
  tb,
  tc,
  sspp,
  rec_threshold,
  spaw_threshold,
  haul_threshold = 30,
  sexes = "all",
  depth,
  GSA,
  country = "all",
  map_lim,
  depth_lines = c(10, 200, 800),
  strata = BioIndex::strata_scheme,
  stratification_tab = BioIndex::stratification,
  resolution = NA,
  buffer = 0.1,
  wd = NA,
  zip = TRUE,
  save = TRUE,
  verbose = TRUE
)
```

Arguments

`ta`	data frame of the TA table in the MEDITS format
`tb`	data frame of the TB table in the MEDITS format
`tc`	data frame of the TC table in the MEDITS format
`sspp`	reference species for the analysis
`rec_threshold`	cutoff threshold for recruits (reported in mm)
`spaw_threshold`	cutoff threshold for spawners (reported in mm)
`haul_threshold`	minimum number of individuals to be used in estimation of the spatial indicators
`sexes`	reference sex for the analysis
`depth`	reference depth range
`GSA`	reference GSA for the analysis
`country`	reference country
`map_lim`	coordinates limits for the maps
`depth_lines`	depth contours to be plotted in the maps (3 values allowed, e.g c(50,200,800))
`strata`	data frame of the reference strata for the study area
`stratification_tab`	data frame of the stratification scheme
`resolution`	resolution of the depth line

`buffer`	buffer around the map
`wd`	path of the working directory
`zip`	boolean. If TRUE the results are stored in a zip file into the working directory
`save`	boolean. If TRUE the results are stored in the working directory
`verbose`	boolean. If TRUE messages are promted in the console

Value

A `list` containing the results of the BioIndex workflow, including data frames and plot objects.

Author(s)

Walter Zupa <`zupa@fondazionecoispa.org`>

Examples

```
BioIndex(ta=TA[TA$YEAR %in% c(2015, 2016), ], tb=TB[TB$YEAR %in% c(2015, 2016), ],
tc=TC[TC$YEAR %in% c(2015, 2016), ], sspp=”MERLMER”,rec_threshold=200,
spaw_threshold=210,sexes=”all”, depth=c(10,800), GSA=10, country=”all”,
map_lim=c(13.3,15.2,39.9,41.3),depth_lines=c(50,200,800),
strata=BioIndex::strata_scheme, stratification_tab =
BioIndex::stratification, resolution=NA, buffer=0.1, wd=tempdir(),
zip=FALSE, save=TRUE, verbose=TRUE)
```

`bubbleplot_RS_by_hauls` *Bubbleplot of abundance indices for recruits and spawners*

Description

The function generates bubbleplots of abundance indices for recruits and spawners If no resolution is specified (res = NA), the function works offline and uses an internal bathymetry dataset (med_bathy) covering the Mediterranean and Black Sea, reducing the computational time.

Usage

```
bubbleplot_RS_by_hauls(
  mTATC,
  map_range,
  thresh_rec,
  thresh_spaw,
  depths = c(50, 200, 800),
  res = NA,
  buffer = 0.1,
  wd = NA,
```

```
  save = TRUE,
  verbose = FALSE
)
```

Arguments

`mTATC`	mTATC table
`map_range`	range of coordinates for the map
`thresh_rec`	threshold value to select recruits data from mTATC table (reported in mm)
`thresh_spaw`	threshold value to select spawners data from mTATC table (reported in mm)
`depths`	three reference bathymetric lines to be plotted in the maps
`res`	resolution of the depth lines
`buffer`	buffer around the map
`wd`	working directory
`save`	boolean. If TRUE the outputs are saved in the local folder
`verbose`	boolean. If TRUE messages are prompted in the console

Value

A `list` containing two `ggplot` objects: the bubble plot of recruits (`pr`) and spawners (`ps`) abundance indices.

Author(s)

Walter Zupa `<zupa@fondazionecoispa.org>`

Examples

```
# Create a merged dataset for GSA 10
mTATC <- merge_TATC(TA[TA$AREA==10,], TC[TC$AREA==10,], "MERLMER")
bubbleplot_RS_by_hauls(mTATC, map_range=c(15,21,39,43), thresh_rec=200, thresh_spaw=250)
```

`bubble_plot_by_haul_indexes` *Bubble plot of abundance and biomass indices by haul*

Description

The function generates bubble plot of abundance and biomass indices by haul. If no resolution is specified (res = NA), the function works offline and uses an internal bathymetry dataset (med_bathy) covering the Mediterranean and Black Sea, reducing the computational time.

Usage

```
bubble_plot_by_haul_indexes(
  mTATB,
  map_lim,
  depth_lines,
  buffer = 0,
  res = NA,
  wd = NA,
  save = TRUE,
  verbose = TRUE
)
```

Arguments

`mTATB`	data frame
`map_lim`	coordinates limits for the plotted map
`depth_lines`	vector of three depth bathymetrical lines to be plotted
`buffer`	buffer to the coordinate limits in map units
`res`	resolution of the bathymetrical lines
`wd`	working directory
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)
`verbose`	boolean. If TRUE a message is printed

Value

A `ggplot` object representing the bubble plot of abundance or biomass by hauls.

Author(s)

Walter Zupa <`zupa@fondazionecoispa.org`>

Examples

```
# Use internal data
data(TA)
data(TB)
# Create a merged dataset for GSA 10 and year 2016
mTATB <- merge_TATB(TA[TA$AREA == 10 & TA$YEAR == 2016, ],
                    TB[TB$AREA == 10 & TB$YEAR == 2016, ],
                    "MERLMER")
bubble_plot_by_haul_indexes(mTATB, map_lim = NA, depth_lines = c(200, 500, 800), save = FALSE)
```

`centroidi` *centroidi*

Description

centroidi

Usage

```
centroidi
```

Format

An object of class `PackedSpatVector` of length 1.

`cgpmgrid` *cgpmgrid*

Description

cgpmgrid

Usage

```
cgpmgrid
```

Format

An object of class `PackedSpatVector` of length 1.

`check_date_haul` *Check date haul (RoME)*

Description

Check if in TB, TC or TE the date by haul is the same of the one reported in TA.

Usage

```
check_date_haul(DataTA, Data, year, wd = NA, suffix = NA)
```

Arguments

`DataTA`	MEDITS TA table
`Data`	MEDITS TB, TC or TE table
`year`	reference year for the analysis
`wd`	working directory
`suffix`	name of the log file

Value

`TRUE` if the validation passes, `FALSE` otherwise.

Note

This function is an internal routine based on **RoME version 0.2.3**. It is provided within `BioIndex` to ensure the package remains functional and self-sufficient for data validation.

Examples

```
data(TA)
data(TB)
check_date_haul(TA, TB, year = 2007)
```

`check_dictionary` *Check dictionary (RoME)*

Description

The function checks whether the values contained in specific fields are consistent with the allowed values of the dictionaries.

Usage

```
check_dictionary(
  ResultData,
  Field,
  Values,
  year,
  wd = NA,
  suffix = NA,
  verbose = FALSE
)
```

Arguments

`ResultData`	data frame in MEDITS tables
`Field`	field of the table to be checked
`Values`	vector of the allowed values
`year`	reference year for the analysis
`wd`	working directory
`suffix`	name of the log file
`verbose`	boolean. If TRUE messages are prompted in the console

Value

`TRUE` if the validation passes, `FALSE` otherwise.

Note

This function is an internal routine based on **RoME version 0.2.3**. It is provided within `BioIndex` to ensure the package remains functional and self-sufficient for data validation.

Examples

```
data(TA)
check_dictionary(TA, Field = "COUNTRY", Values = c("ITA"), year = 2007)
```

`check_hauls_TBTA` *Check hauls TB TA (RoME)*

Description

Check if all the hauls in TB are in TA.

Usage

```
check_hauls_TBTA(DataTA, DataTB, year, wd = NA, suffix = NA)
```

Arguments

`DataTA`	MEDITS TA table
`DataTB`	MEDITS TB table
`year`	reference year for the analysis
`wd`	working directory
`suffix`	name of the log file

Value

`TRUE` if the validation passes, `FALSE` otherwise.

Note

This function is an internal routine based on **RoME version 0.2.3**. It is provided within `BioIndex` to ensure the package remains functional and self-sufficient for data validation.

Examples

```
data(TA)
data(TB)
check_hauls_TBTA(TA, TB, year = 2007)
```

`check_numeric_range` *Check numeric range (RoME)*

Description

The function checks whether the values contained in specific fields are consistent with the allowed ranges.

Usage

```
check_numeric_range(
  ResultData,
  Field,
  Values,
  year,
  wd = NA,
  suffix = NA,
  verbose = FALSE
)
```

Arguments

`ResultData`	data frame in MEDITS tables
`Field`	field of the table to be checked
`Values`	vector of the allowed values
`year`	reference year for the analysis
`wd`	working directory
`suffix`	name of the log file
`verbose`	boolean. If TRUE messages are prompted in the console

Value

`TRUE` if the validation passes, `FALSE` otherwise.

Note

This function is an internal routine based on **RoME version 0.2.3**. It is provided within `BioIndex` to ensure the package remains functional and self-sufficient for data validation.

Examples

```
data(TA)
check_numeric_range(TA, Field = "HAUL_DURATION", Values = c(10, 120), year = 2007)
```

`continent` *continent*

Description

continent

Usage

```
continent
```

Format

An object of class `PackedSpatVector` of length 1.

`convert_coordinates` *MEDITS coordinates in decimal degrees*

Description

The function returns the data frame of the TA table with the coordinates expressed as decimal degrees.

Usage

```
convert_coordinates(Data)
```

Arguments

`Data`	data frame of TA table

Value

the function return the same data frame with the coordinates converted in the decimal degrees format

Examples

```
data(TA)
convert_coordinates(TA)
```

`dd.distance` *Estimate hauls distances (decimal degrees)*

Description

Function to estimate the hauls length using TA (table A, hauls data) with coordinates in the decimal degrees format (dd.ddd). The distances could be returned expressed in meters, kilometers and nautical miles.

Usage

```
dd.distance(data, unit = "m", verbose = TRUE)
```

Arguments

`data`	data frame of the hauls data (TA, table A) with coordinates reported as decimal degrees
`unit`	string value indicating the measure unit of the distance. Allowed values: "m" for meters, "km" for kilometers and "NM" for nautical miles.
`verbose`	give verbose output reporting in the output the selected measure unit of the distance.

Value

The function returns the vector of the distances expressed in the selected measure unit.

Examples

```
data(TA)
ta_dd <- MEDITS.to.dd(TA)
dd.distance(ta_dd, unit = "km", verbose = FALSE)
```

`dd.to.MEDITS` *Conversion of decimal degrees coordinates in MEDITS format*

Description

Conversion of decimal degrees coordinates in MEDITS format

Usage

```
dd.to.MEDITS(data)
```

Arguments

`data`	data frame of the hauls data (TA, table A) in MEDITS format

Value

The function returns the data frame of the TA (table A) reporting the coordinates in MEDITS format.

Examples

```
data(TA)
ta_dd <- MEDITS.to.dd(TA)
dd.to.MEDITS(ta_dd)
```

`hauls_position` *Plot of hauls time series*

Description

This function generates a spatial plot of haul positions over time, displaying haul labels and bathymetric lines over a customizable map extent. If no resolution is specified (res = NA), the function works offline and uses an internal bathymetry dataset (med_bathy) covering the Mediterranean and Black Sea, reducing the computational time.

Usage

```
hauls_position(
  mTATB,
  country = "all",
  map_lim,
  depth_lines,
  buffer = 0,
  res = NA,
  wd = NA,
  save = TRUE,
  verbose = TRUE
)
```

Arguments

`mTATB`	data frame
`country`	country code as reported in MEDITS format. "all" code to perform the analysis on all the countries of the same GSA
`map_lim`	coordinates limits for the plotted map
`depth_lines`	vactor of three depth bathymetrical lines to be plotted
`buffer`	buffer to the coordinate limits in map units
`res`	resolution of the bathymetrical lines
`wd`	working directory
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)
`verbose`	boolean. If TRUE messages are reported in the console

Value

A `ggplot` object representing the spatial distribution of hauls.

Author(s)

Walter Zupa `<zupa@fondazionecoispa.org>`

Examples

```
# Use internal data
data(TA)
data(TB)
# Create a merged dataset for GSA 10 and year 2016
mTATB <- merge_TATB(TA[TA$AREA == 10 & TA$YEAR == 2016, ],
                    TB[TB$AREA == 10 & TB$YEAR == 2016, ],
                    "MERLMER")
hauls_position(mTATB, country = "all", map_lim = NA, depth_lines = c(200, 500, 800), save = FALSE)
```

`index_on_grid` *Generating maps of indexes*

Description

Generates density and biomass maps based on the spatial grid, allowing for the visual identification of resource concentration areas or potential biological hotspots.

Usage

```
index_on_grid(
  mTATBsp,
  stratum,
  wd = NA,
  map_range,
  threshold = 30,
  verbose = FALSE,
  save = TRUE
)
```

Arguments

`mTATBsp`	spatial mTATB
`stratum`	reference stratum range (allowed values: "10,200","10,800","200,800","5,35","5,45")
`wd`	working directory
`map_range`	range of coordinates for the map
`threshold`	minimum number of individuals per haul
`verbose`	boolean. If TRUE messages are promted in the console
`save`	boolean. If TRUE the results are stored in the working directory

Value

A `ggplot` object displaying the generated map of indices on the spatial grid.

Examples

```
data(TA)
data(TB)
data(TC)
m <- merge_TATBTC(TA[TA$AREA == 10, ], TB[TB$AREA == 10, ], TC[TC$AREA == 10, ],
                  species = "MERLMER", country = "all", verbose = FALSE)
mTATBsp <- overlayGrid(m[[1]], m[[2]], GSA = 10, save = FALSE, verbose = FALSE)[[1]]
map_range <- c(9, 15, 39, 42)
index_on_grid(mTATBsp, stratum = "10,800", map_range = map_range, threshold = 5,
              save = FALSE, verbose = FALSE)
```

`index_recr` *Estimation of abundance indices for recruits*

Description

Estimates specific abundance indices for the recruitment phase (using the user-defined cutoff value), allowing for the assessment of reproductive success and the prediction of future stock productivity.

Usage

```
index_recr(
  mTATB,
  mTATC,
  GSA,
  country,
  depth_range,
  cutoff,
  stratification,
  wd = NA,
  save = TRUE
)
```

Arguments

`mTATB`	data frame
`mTATC`	data frame
`GSA`	reference GSA for the analysis
`country`	vector of reference countries for the analysis
`depth_range`	range of depth strata to perform the analysis (min, max)
`cutoff`	cutoff value for splitting recruits portion of population (reported in mm)
`stratification`	data frame of strata surface area
`wd`	working directory
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)

Value

A `data.frame` containing the abundance indices and density for the recruitment fraction.

Examples

```
data(TA)
data(TB)
data(TC)
data(stratification)
m <- merge_TATBTC(TA[TA$AREA == 10, ], TB[TB$AREA == 10, ], TC[TC$AREA == 10, ],
                  species = "MERLMER", country = "all", verbose = FALSE)
index_recr(m[[1]], m[[2]], GSA = 10, country = "all", depth_range = c(10, 800),
           cutoff = 200, stratification = stratification, save = FALSE)
```

`index_spawn` *Estimation of abundance indices for spawners (females)*

Description

Calculates abundance and biomass indices focused on the adult fraction (spawners, using the user-defined cutoff value), an essential indicator for evaluating the self-renewal capacity of the population.

Usage

```
index_spawn(
  mTATB,
  mTATC,
  GSA,
  country,
  depth_range,
  cutoff,
  stratification,
  wd = NA,
  save = TRUE
)
```

Arguments

`mTATB`	data frame
`mTATC`	data frame
`GSA`	reference GSA for the analysis
`country`	vector of reference countries for the analysis
`depth_range`	range of depth strata to perform the analysis (min, max)
`cutoff`	cutoff value for splitting spawner portion of population (reported in mm)
`stratification`	data frame of strata surface area
`wd`	working directory
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)

Value

A `data.frame` containing the abundance and biomass indices for the spawner fraction.

Examples

```
data(TA)
data(TB)
data(TC)
data(stratification)
m <- merge_TATBTC(TA[TA$AREA == 10, ], TB[TB$AREA == 10, ], TC[TC$AREA == 10, ],
                  species = "MERLMER", country = "all", verbose = FALSE)
index_spawn(m[[1]], m[[2]], GSA = 10, country = "all", depth_range = c(10, 800),
            cutoff = 200, stratification = stratification, save = FALSE)
```

`index_ts_F` *Estimation of abundance indices for females*

Description

Estimation of abundance indices for females

Usage

```
index_ts_F(
  mTATB,
  GSA,
  country_analysis,
  depth_range,
  strata_scheme,
  stratification,
  wd = NA,
  save = TRUE
)
```

Arguments

`mTATB`	data frame
`GSA`	reference GSA for the analysis
`country_analysis`	vector of reference countries for the analysis
`depth_range`	range of depth strata to perform the analysis (min, max)
`strata_scheme`	data frame of the stratification scheme
`stratification`	data frame of strata surface area
`wd`	working directory
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)

Value

A `data.frame` containing the time series of indices specifically for females.

Examples

```
data(TA)
data(TB)
data(TC)
data(strata_scheme)
data(stratification)
m <- merge_TATBTC(TA[TA$AREA == 10, ], TB[TB$AREA == 10, ], TC[TC$AREA == 10, ],
                  species = "MERLMER", country = "all", verbose = FALSE)
strata_scheme_g10 <- strata_scheme[strata_scheme$GSA == 10 & strata_scheme$COUNTRY == "ITA", ]
index_ts_F(m[[1]], GSA = 10, country_analysis = "all", depth_range = c(10, 800),
        strata_scheme = strata_scheme_g10, stratification = stratification, save = FALSE)
```

`index_ts_M` *Estimation of abundance indices for males*

Description

Estimation of abundance indices for males

Usage

```
index_ts_M(
  mTATB,
  GSA,
  country_analysis,
  depth_range,
  strata_scheme,
  stratification,
  wd = NA,
  save = TRUE
)
```

Arguments

`mTATB`	data frame
`GSA`	reference GSA for the analysis
`country_analysis`	vector of reference countries for the analysis
`depth_range`	range of depth strata to perform the analysis (min, max)
`strata_scheme`	data frame of the stratification scheme
`stratification`	data frame of strata surface area
`wd`	working directory
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)

Value

A `data.frame` containing the time series of indices specifically for males.

Examples

```
data(TA)
data(TB)
data(TC)
data(strata_scheme)
data(stratification)
m <- merge_TATBTC(TA[TA$AREA == 10, ], TB[TB$AREA == 10, ], TC[TC$AREA == 10, ],
                 species = "MERLMER", country = "all", verbose = FALSE)
strata_scheme_g10 <- strata_scheme[strata_scheme$GSA == 10 & strata_scheme$COUNTRY == "ITA", ]
index_ts_M(m[[1]], GSA = 10, country_analysis = "all", depth_range = c(10, 800),
        strata_scheme = strata_scheme_g10, stratification = stratification, save = FALSE)
```

`indices_ts` *Estimation of abundance and biomass indices*

Description

Computes time series of stratified abundance and biomass indices, serving as the primary tool for monitoring temporal trends in stock status.

Usage

```
indices_ts(
  mTATB,
  GSA,
  country = "all",
  depth_range,
  strata_scheme,
  stratification,
  wd = NA,
  save = TRUE
)
```

Arguments

`mTATB`	data frame
`GSA`	reference GSA for the analysis
`country`	reference countries in the GSA for the analysis
`depth_range`	range of depth strata to perform the analysis (min, max)
`strata_scheme`	data frame of the stratification scheme
`stratification`	data frame of strata surface area
`wd`	working directory
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)

Value

A `list` of `ggplot` objects representing the time series of calculated indices.

Author(s)

Walter Zupa <`zupa@fondazionecoispa.org`>

Examples

```
data(TA)
data(TB)
# Create a merged dataset for GSA 10
mTATB <- merge_TATB(TA[TA$AREA==10,], TB[TB$AREA==10,], "MERLMER")
# Run indices_ts
indices_ts(mTATB, GSA=10, country="all", depth_range=c(10,800),
           strata_scheme=BioIndex::strata_scheme,
           stratification=BioIndex::stratification, wd=tempdir(), save=FALSE)
```

`IUT` *Interception Union Tets*

Description

Interception Union Tets

Usage

```
IUT(abundance, biomass, species, lastn = 5, GSA = 10, wd = NA, save = TRUE)
```

Arguments

`abundance`	dataframe of abundance time series as produced by indices_ts function
`biomass`	dataframe of biomass time series as produced by indices_ts function
`species`	reference species for the analysis (MEDITS code)
`lastn`	number of recent years for diagnosis of change
`GSA`	reference GSA for the analysis (default is 10)
`wd`	working directory (default is NA, uses tempdir())
`save`	boolean. If TRUE results are saved in the output folder

Value

A `data.frame` containing the Indicator of Unfished Trends (IUT) results.

LFD *Length Frequency Distribution (LFD) Estimation and Plotting*

Description

Input parameters include biological and haul data, stratification schemes, reference filters (e.g., sex, GSA, country, depth), and output control options (working directory, saving, verbosity).

Usage

```
LFD(
  mTATC,
  sex = "all",
  GSA,
  country = "all",
  depth_range,
  strata_scheme,
  stratification,
  wd = NA,
  save = TRUE,
  verbose = TRUE
)
```

Arguments

`mTATC`	Data frame resulting from the merge of TA and TC datasets using ‘merge_TATBTC()‘. It must include raised numbers.
`sex`	Character. Target sex for the analysis. Allowed values: ‘"F"‘, ‘"M"‘, ‘"I"‘, ‘"N"‘, or ‘"all"‘ (default) for combined sexes.
`GSA`	Numeric. The GSA (Geographical Sub-Area) code of reference for the analysis.
`country`	Character vector. Reference country or countries for the analysis. Use ‘"all"‘ (default) to include all countries available in the data.
`depth_range`	Numeric vector of length 2, specifying the minimum and maximum depths (in meters) to filter the hauls used in the analysis.
`strata_scheme`	Data frame containing the stratification scheme. Must include ‘CODE‘, ‘MIN_DEPTH‘, ‘MAX_DEPTH‘, ‘GSA‘, ‘COUNTRY‘.
`stratification`	Data frame with surface areas per stratum. Must include columns: ‘GSA‘, ‘CODE‘, ‘COUNTRY‘, ‘SURF‘.
`wd`	Character. Working directory used to save output plots and tables. Required if ‘save = TRUE‘.
`save`	Logical. If ‘TRUE‘ (default), the function saves output tables and plots to ‘wd/output/‘. If ‘FALSE‘, no files are saved.
`verbose`	Logical. If ‘TRUE‘ (default), informative messages are printed to the console to trace the function steps.

Details

This function estimates and plots the length frequency distribution (LFD) by year and by stratum, based on merged biological and haul data from MEDITS surveys. It computes raised numbers, applies stratification weights, and generates publication-ready plots and CSV outputs. Stratification weighting is based on swept area and surface area of the strata.

The function:

- Filters hauls by depth and country.
- Computes raised numbers per haul.
- Applies stratified weights based on the area per stratum.
- Outputs and/or plots the length frequency distribution by year and stratum.
- Handles sex-specific or combined-sex analyses.

The plot outputs include:

- Combined LFD by year across all strata.
- LFD by stratum (faceted), stratified by year.

Output files are saved as CSV and high-resolution JPEGs if ‘save = TRUE‘ and ‘wd‘ is defined.

Value

A named list with two elements:

‘LFD‘ A data frame of total LFD by year across all strata.

‘LFD by stratum‘ A data frame of LFD by year and by stratum.

A `list` of `ggplot` objects representing the length-frequency distributions.

Examples

```
data(TA)
data(TC)
# Create a merged dataset for GSA 10
mTATC <- merge_TATC(TA[TA$AREA==10,], TC[TC$AREA==10,], "MERLMER")
LFD(mTATC, sex="all", GSA=10, country="all", depth_range=c(10,800),
    strata_scheme=BioIndex::strata_scheme,
    stratification=BioIndex::stratification, wd=tempdir(), save=FALSE)
```

Lquant *Estimation of L50 and L95*

Description

Estimation of L50 and L95

Usage

```
Lquant(lfd, wd = NA, sspp, GSA, save = TRUE, verbose = TRUE)
```

Arguments

`lfd`	data frame of combined LFD
`wd`	working directory
`sspp`	MEDITS code for the selected species
`GSA`	reference area for the analysis
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)
`verbose`	boolean. If TRUE messages are reported in the console

Value

A `data.frame` containing the length quantiles (e.g., L95) time series.

LW *LW (Length-Weight relationship)*

Description

Estimates the length-weight relationship for the selected species, a fundamental biological indicator to assess fish condition and convert numerical length estimates into biomass equivalents.

Usage

```
LW(
  ta,
  te,
  sp,
  GSA,
  country = "all",
  nyears = NA,
  wd = NA,
  save = TRUE,
  verbose = FALSE
)
```

Arguments

`ta`	MEDITS or MEDITS-like TA table
`te`	MEDITS or MEDITS-like TE table
`sp`	species RUBIN code (MEDITS format, e.g. "MERLMER")
`GSA`	reference GSA for the analysis
`country`	reference country
`nyears`	number of years of the time series to be considered in the analysis
`wd`	path of the working directory
`save`	boolean. If TRUE the outputs are saved in the local folder
`verbose`	boolean. If TRUE messages are promted in the console

Value

A `data.frame` containing the length-weight relationship parameters and statistics.

`LWf` *Estimation of LW relationship*

Description

Estimation of LW relationship

Usage

```
LWf(TE, sp, GEAR, GSA, country = NA, n_records = 10, wd = NA, save = TRUE)
```

Arguments

`TE`	MEDITS or MEDITS-like TE table
`sp`	species RUBIN code (MEDITS format, e.g. "MERLMER")
`GEAR`	type of gear reported in the corresponding TA file
`GSA`	reference GSA for the analysis
`country`	reference country
`n_records`	minimum number of records to perform the analysis
`wd`	path of the working directory
`save`	boolean. If TRUE the outputs are saved in the local folder

Value

A `data.frame` containing the length-weight relationship parameters specifically for females.

Author(s)

Walter Zupa <`zupa@fondazionecoispa.org`>

`MEDITS.distance` *Estimation of haul distance*

Description

Estimation of haul distance

Usage

```
MEDITS.distance(data, unit = "m", verbose = TRUE)
```

Arguments

`data`	data frame containing the hauls data (TA, table A).
`unit`	string value indicating the measure unit of the distance. Allowed values: "m" for meters, "km" for kilometers and "NM" for nautical miles.
`verbose`	give verbose output reporting in the output the selected measure unit of the distance.

Value

The function returns the vector of the distances expressed in the selected measure unit.

Examples

```
data(TA)
MEDITS.distance(TA, unit = "m", verbose = FALSE)
```

`MEDITS.to.dd` *Conversion of MEDITS format coordinates in decimal degrees format*

Description

Conversion of MEDITS format coordinates in decimal degrees format

Usage

```
MEDITS.to.dd(data)
```

Arguments

`data`	data frame of the hauls data (TA, table A) in MEDITS format

Value

The function returns the data frame of the TA table with the coordinates expressed as decimal degrees

Examples

```
data(TA)
MEDITS.to.dd(TA)
```

`med_bathy` *Mediterranean and Black Sea bathymetry (0â€“1000 m, bathy object)*

Description

A precomputed `bathy` object containing bathymetric data for the Mediterranean Sea and the Black Sea. Depth values are restricted between 0 and -1000 meters. This dataset was downloaded using `getNOAA.bathy` with a resolution of 1 arc-minute, and filtered to remove deeper areas.

Usage

```
data(med_bathy)
```

Format

An object of class `bathy` (a matrix with longitude and latitude as axes, and depth in meters)

Details

The object can be used directly with functions from the **marmap** package, such as `plot.bathy()` and `get.depth()`.

The spatial extent includes:

- Longitude: from -6Â° to 42Â°
- Latitude: from 30Â° to 47Â°
- Depth: from 0 to -1000 meters

Only marine cells within this depth range are retained. Land and deeper areas are set to `NA`.

Source

NOAA ETOPO1 via `marmap::getNOAA.bathy()`

See Also

`getNOAA.bathy`, `plot.bathy`, `bubbleplot_RS_by_hauls`

`merge_TATB` *Merge TA and TB tables (haul-level catches)*

Description

Combines haul-level operational data (TA) with species catch data (TB), providing the essential foundation for calculating stratified abundance (N) and biomass (kg) indices. The function keeps only valid hauls, performs quality checks (times, positions, wing opening, haul/date consistency) using internal validation functions (based on `RoME` version 0.2.3), computes spatial means, swept area, depth stratum, biomass/density indicators, andâ€"optionallyâ€"writes the merged data to disk.

Usage

```
merge_TATB(
  ta,
  tb,
  species,
  country = "all",
  strata = BioIndex::strata_scheme,
  wd = NA,
  save = FALSE,
  verbose = TRUE
)
```

Arguments

`ta`	A ‘data.frame‘/‘data.table‘ containing a full TA dataset (columns listed in ‘BioIndex::TA_cols‘).
`tb`	A ‘data.frame‘/‘data.table‘ containing a full TB dataset (columns listed in ‘BioIndex::TB_cols‘).
`species`	Character, MEDITS 7-letter code (e.g. ‘"ARISFOL"‘). The first 4 characters are interpreted as *GENUS* and the last 3 as *SPECIES*.
`country`	Character vector of ISO-3 country codes to keep; use ‘"all"‘ (default) to keep every country present in ‘ta‘.
`strata`	Depth-stratum reference table; default is ‘BioIndex::strata_scheme‘.
`wd`	Working directory. If ‘NA‘ (the default when ‘save = FALSE‘) no files are written.
`save`	Logical. If ‘TRUE‘, writes ‘"mergeTATB_<GENUS><SPECIES>.csv"‘ to ‘file.path(wd, "output")‘.
`verbose`	Logical; if ‘TRUE‘ (default) prints progress messages.

Details

The implementation mirrors the original BioIndex routine.

Data Validation: The function performs syntactic data validation using internal implementations of the validation routines (based on `RoME` v0.2.3).

Optimisations: Includes two speed-ups: (1) vectorised replacement of missing ‘"NA NA"‘ records, (2) a single loop over depth strata instead of a nested haul Ã— stratum loop. Results are identical to the reference routine.

Value

A ‘data.frame‘ with one row per haul and the following groups of variables: * **TA** metadata (haul position, times, depths, etc.) * **TB** catch totals (numbers and weight) * Calculated fields: mean lat/lon, swept area, depth stratum, density (‘N_h‘, ‘N_km2‘) and biomass (‘kg_h‘, ‘kg_km2‘)

Note

This version of `BioIndex` uses internal validation functions based on **RoME version 0.2.3**. The package is fully self-sufficient and does not require external validation packages to be installed.

See Also

[BioIndex] package documentation.

Examples

```
# Use internal data
data("TA", package = "BioIndex")
data("TB", package = "BioIndex")
m_tb <- merge_TATB(
  ta       = TA,
  tb       = TB,
  species  = "ARISFOL",
  country  = "ITA",
  wd       = tempdir(),
  save     = FALSE
)
head(m_tb)
```

`merge_TATBTC` *Merge TAâ€“TB and TAâ€“TC tables (MEDITS protocol)*

Description

This function is the primary entry point for integrating MEDITS survey data (e.g. TA, TB, and TC tables), producing unified datasets essential for population analysis and the estimation of demographic indicators. The routine:

- filters hauls by validity and by country;

- performs a full set of quality checks (times, positions, wing opening, haul/date consistency) using internal validation functions (based on `RoME` version 0.2.3);
- merges TA with TB and TA with TC, respectively;
- computes swept area, mean positions, depth stratum, raising factors, densities and biomasses;
- (optionally) writes the two merged tables to `file.path(wd, "output")`.

Usage

```
merge_TATBTC(
  ta,
  tb,
  tc,
  species,
  country = "all",
  strata = BioIndex::strata_scheme,
  wd = NA,
  save = TRUE,
  verbose = TRUE
)
```

Arguments

`ta`	A MEDITS or MEDITS-like **TA** table (columns listed in `BioIndex::TA_cols`).
`tb`	A MEDITS or MEDITS-like **TB** table (columns listed in `BioIndex::TB_cols`).
`tc`	A MEDITS or MEDITS-like **TC** table (columns listed in `BioIndex::TC_cols`).
`species`	Character. MEDITS 7-letter RUBIN code (e.g. `"MERLMER"`): the first 4 letters are used as *GENUS*, the last 3 as *SPECIES*.
`country`	Character vector of MEDITS country codes to keep. Use `"all"` (default) to include every country present in `ta`.
`strata`	A data-frame with the depth-stratification scheme adopted by the MEDITS survey. Defaults to `BioIndex::strata_scheme`.
`wd`	Working directory. When `save = TRUE`, the merged tables are written to `file.path(wd, "output")`.
`save`	Logical. If `TRUE` (default) the function writes `"mergeTATB_<GENUS><SPECIES>.csv"` and `"mergeTATC_<GENUS><SPECIES>.csv"`.
`verbose`	Logical (default `TRUE`); prints progress messages.

Details

The implementation reproduces the official BioIndex workflow.

Data Validation: The function automatically performs syntactic data validation using internal routines based on `RoME` version 0.2.3 logic to ensure data integrity and standardisation.

Optimisations: Two micro-optimisations are included:

1. vectorised replacement of missing `"NA NA"` entries in both merges;
2. a single loop over depth strata (instead of haul Ã— stratum) to assign `STRATUM_CODE`.

Numerical outputs are identical to the reference routine.

Value

A list of two `data.frames`:

`merge_TA_TB` One row per haul with TA metadata, TB catch totals, depth stratum, densities and biomasses.

`merge_TA_TC` One row per haul/length-class/sex/maturity with TA metadata, TC counts, depth stratum, raising factor, densities and biomasses.

A `list` containing two data frames: the first with merged TA-TB-TC data at the haul level, and the second with the length-frequency distribution.

Note

This version of `BioIndex` uses internal validation functions based on **RoME version 0.2.3**. The package is fully self-sufficient and does not require external validation packages to be installed.

Author(s)

Walter Zupa `<zupa@fondazionecoispa.org>`

Examples

```
# Use internal data

data("TA", package = "BioIndex")
data("TB", package = "BioIndex")
data("TC", package = "BioIndex")
m <- merge_TATBTC(
  ta      = TA,
  tb      = TB,
  tc      = TC,
  species = "MERLMER",
  country = "all",
  wd      = tempdir(),
  verbose = TRUE
)
mTATB <- m[[1]]   # TAâ€“TB merged table
mTATC <- m[[2]]   # TAâ€“TC merged table
```

`merge_TATC` *Merge TA and TC tables (haul-level lengthâ€“frequency data)*

Description

Links haul metadata (TA) with length-frequency samples (TC), enabling the analysis of population demographic structure and the reconstruction of length-frequency distributions (LFD). The function keeps only valid hauls, runs all relevant quality checks (times, positions, wing opening, haul/date consistency) using internal validation functions (based on `RoME` version 0.2.3), computes swept area, depth stratum, raising factors, abundance/biomass indicators, andâ€"optionallyâ€"writes the merged data.

Usage

```
merge_TATC(
  ta,
  tc,
  species,
  country = "all",
  strata = BioIndex::strata_scheme,
  wd = NA,
  save = TRUE,
  verbose = TRUE
)
```

Arguments

`ta`	A ‘data.frame‘/‘data.table‘ containing a full TA dataset (columns listed in ‘BioIndex::TA_cols‘).
`tc`	A ‘data.frame‘/‘data.table‘ containing a full TC dataset (columns listed in ‘BioIndex::TC_cols‘).
`species`	Character, MEDITS 7-letter code (e.g. ‘"ARISFOL"‘). The first 4 characters are interpreted as *GENUS* and the last 3 as *SPECIES*.
`country`	Character vector of ISO-3 country codes to keep; use ‘"all"‘ (default) to keep every country present in ‘ta‘.
`strata`	Depth-stratum reference table; default is ‘BioIndex::strata_scheme‘.
`wd`	Working directory. If ‘NA‘ (the default when ‘save=FALSE‘) no files are written.
`save`	Logical. If ‘TRUE‘, writes ‘"mergeTATC_<GENUS><SPECIES>.csv"‘ to ‘file.path(wd, "output")‘.
`verbose`	Logical; if ‘TRUE‘ (default) prints progress messages.

Details

The code is identical to the official BioIndex routine.

Data Validation: The function performs syntactic data validation using internal implementations of the validation routines (based on `RoME` v0.2.3).

Optimisations: Includes two micro-optimisations: vectorised handling of missing ‘"NA NA"‘ entries and a single depth-stratum loop. Numerical results remain unchanged.

Value

A ‘data.frame‘ in which each row represents one length-class (and sex/maturity) sampled in a haul, including: * **TA** metadata * **TC** length-frequency counts and weights * Calculated fields: mean lat/lon, swept area, depth stratum, raising factor, density (‘N_h‘, ‘N_km2‘) and biomass (‘kg_h‘, ‘kg_km2‘)

Note

This version of `BioIndex` uses internal validation functions based on **RoME version 0.2.3**. The package is fully self-sufficient and does not require external validation packages to be installed.

See Also

[BioIndex] package documentation.

Examples

```
# Use internal data
data("TA", package = "BioIndex")
data("TC", package = "BioIndex")
m_tc <- merge_TATC(
  ta       = TA,
  tc       = TC,
  species  = "ARISFOL",
  country  = "ESP",
  wd       = tempdir(),
  save=FALSE
)
head(m_tc)
```

`MIW` *Estimation of Mean Individual Weight (MIW) time series*

Description

Calculates the Mean Individual Weight (MIW) time series, providing a summary indicator of the average size within the captured population and potential shifts due to fishing pressure.

Usage

```
MIW(
  mTATB,
  GSA,
  country = "all",
  depth_range,
  strata_scheme,
  stratification,
  wd = NA,
```

```
  save = TRUE,
  verbose = TRUE
)
```

Arguments

`mTATB`	data frame of the merged TA and TB
`GSA`	reference GSA for the analysis
`country`	reference countries in the GSA for the analysis
`depth_range`	range of depth strata to perform the analysis (min, max)
`strata_scheme`	data frame of the stratification scheme
`stratification`	data frame of strata surface area
`wd`	working directory
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)
`verbose`	boolean. If TRUE messages are reported in the console

Value

A `data.frame` containing the Mean Individual Weight (MIW) time series.

Examples

```
data(TA)
data(TB)
# Create a merged dataset for GSA 10
mTATB <- merge_TATB(TA[TA$AREA==10,], TB[TB$AREA==10,], "MERLMER")
MIW(mTATB, GSA=10, country="all", depth_range=c(10,800),
    strata_scheme=BioIndex::strata_scheme,
    stratification=BioIndex::stratification, wd=tempdir(), save=FALSE)
```

`overlayGrid` *Overlay mTATB and mTATC on GFCM spatial grid*

Description

Overlays survey haul data onto a standard spatial grid (e.g., GFCM grid), harmonizing biological information with a consistent geographic structure for standardized spatial analysis.

Usage

```
overlayGrid(
  mTATB,
  mTATC,
  GSA = NA,
  country = "all",
  wd = NA,
```

```
  save = TRUE,
  verbose = FALSE
)
```

Arguments

`mTATB`	data frame of the merged TA and TB
`mTATC`	data frame of the merged TA and TC
`GSA`	reference GSA for the analysis
`country`	reference countries for the analysis
`wd`	working directory used to save results
`save`	boolean. If TRUE the outputs are saved in the local folder
`verbose`	boolean. If TRUE messages are prompted in the console

Value

A `list` or `data.frame` containing the merged biological data mapped to the spatial grid.

Examples

```
# Use internal data
data(TA)
data(TB)
data(TC)
# Create a merged dataset for GSA 10
m <- merge_TATBTC(TA[TA$AREA==10,], TB[TB$AREA==10,], TC[TC$AREA==10,], species="MERLMER")
mTATB <- m[[1]]
mTATC <- m[[2]]
overlayGrid(mTATB, mTATC, GSA=10, save=FALSE)
```

`quant` *Quantile estimation*

Description

Quantile estimation

Usage

```
quant(weighted, qlin = 0.95)
```

Arguments

`weighted`	LFD data.frame
`qlin`	reference quantile for the analysis

Value

A `numeric` value or vector containing the requested quantiles.

`run_BioIndex_app` *launches the embedded Shiny application included in the package.*

Description

launches the embedded Shiny application included in the package.

Usage

```
run_BioIndex_app()
```

Value

`No return value, called for side effects` (launches the Shiny application).

Author(s)

Walter Zupa <`zupa@fondazionecoispa.org`>

Examples

```
if (interactive()) {
  run_BioIndex_app()
}
```

`sex_ratio` *Sex ratio*

Description

Assesses the proportion of sexes within the population, providing critical information on demographic structure, reproductive potential, and stock resilience.

Usage

```
sex_ratio(
  mTATB,
  GSA,
  country,
  depth_range,
  stratas,
  stratification,
  wd = NA,
  save = TRUE,
  verbose = FALSE
)
```

Arguments

`mTATB`	data frame of the merged TA and TB
`GSA`	reference GSA for the analysis
`country`	vector of reference countries for the analysis
`depth_range`	range of depth strata to perform the analysis (min, max)
`stratas`	data frame of the reference strata for the study area
`stratification`	data frame of strata surface area
`wd`	working directory
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)
`verbose`	boolean. If TRUE a message is printed

Value

A `data.frame` containing the sex ratio statistics by year and stratum.

Author(s)

Walter Zupa <`zupa@fondazionecoispa.org`>

Examples

```
data(TA)
data(TB)
# Create a merged dataset for GSA 10
mTATB <- merge_TATB(TA[TA$AREA==10,], TB[TB$AREA==10,], "MERLMER")
sex_ratio(mTATB, GSA=10, country="all", depth_range=c(10,800),
          stratas=BioIndex::strata_scheme,
          stratification=BioIndex::stratification, wd=tempdir(), save=FALSE)
```

`sex_ratio_on_grid` *Plot sex ratio spatial distribution*

Description

This function calculates and plots the spatial distribution of sex ratio (F / (F + M)) over the GFCM grid for a given species. It uses data from merged MEDITS datasets (TA, TB, TC), previously spatialized with ‘overlayGrid()‘. The function filters hauls by total abundance threshold and depth range, calculates sex ratio per grid cell, and plots the output using color-coded categories based on quantiles. The result is a choropleth map of sex ratio patterns, useful to highlight spatial differences in sex structure. Raw sex ratio estimates per grid cell can also be exported.

Usage

```
sex_ratio_on_grid(
  mTATBsp,
  depth,
  wd = NA,
  map_range,
  threshold = 30,
  verbose = FALSE,
  save = TRUE
)
```

Arguments

`mTATBsp`	A spatial version of the merged TA-TB dataset, processed through ‘overlayGrid()‘, containing sex information.
`depth`	Character string specifying the depth range used for filtering (e.g., ‘"10,800"‘). Must correspond to available internal GFCM grid masks.
`wd`	Working directory. Used to save output files if ‘save = TRUE‘.
`map_range`	numeric vector with coordinates defining the map extent (‘xmin‘, ‘xmax‘, ‘ymin‘, ‘ymax‘).
`threshold`	Minimum number of individuals required per haul to be included in the sex ratio estimation. Default is 30.
`verbose`	Logical. If ‘TRUE‘, informative messages are printed to the console.
`save`	Logical. If ‘TRUE‘, the plot and sex ratio table are saved to the working directory.

Value

A ‘data.frame‘ containing sex ratio (‘ratio‘), standard deviation (‘sd‘), and coefficient of variation (‘CV‘) for each GFCM grid cell.

Examples

```
data(TA)
data(TB)
data(TC)
m <- merge_TATBTC(TA[TA$AREA == 10, ], TB[TB$AREA == 10, ], TC[TC$AREA == 10, ],
                  species = "MERLMER", country = "all", verbose = FALSE)
mTATBsp <- overlayGrid(m[[1]], m[[2]], GSA = 10, save = FALSE, verbose = FALSE)[[1]]
map_range <- c(9, 15, 39, 42)
sex_ratio_on_grid(mTATBsp, depth = "10,800", map_range = map_range, threshold = 5,
                  save = FALSE, verbose = FALSE)
```

`spear` *Spearman test for timeseries*

Description

Spearman test for timeseries

Usage

```
spear(x)
```

Arguments

`x`	time series

Value

A `data.frame` representing the Spearman correlation coefficient and related statistics.

Examples

```
x <- runif(10, 1, 100)
spear(x)
```

`spearman` *Spearman test*

Description

Spearman test

Usage

```
spearman(
  abundance = NA,
  biomass = NA,
  years,
  sspp = NA,
  wd = NA,
  save = TRUE,
  verbose = FALSE
)
```

Arguments

`abundance`	data frame of abundance indices
`biomass`	data frame of biomass indices
`years`	reference years for the analysis
`sspp`	reference species for the analysis
`wd`	path of working directory
`save`	boolean. If TRUE the plot is saved in the user defined working directory (wd)
`verbose`	boolean. If TRUE messages are prompted in the console

Value

A `data.frame` or `list` with the results of the Spearman rank correlation test.

Examples

```
abundance <- data.frame(year = 2010:2020, index = runif(11, 1, 10))
biomass <- data.frame(year = 2010:2020, index = runif(11, 5, 20))
spearman(abundance=abundance, biomass=biomass, years=c(2010,2020),
         sspp="MERLMER", wd=tempdir(), save=FALSE)
```

`strata_scheme` *stratification scheme*

Description

Stratification scheme adopted in the bottom trawl demersal survey (e.g. MEDITS survey).

Usage

```
strata_scheme
```

Format

An object of class `data.frame` with 126 rows and 5 columns.

Author(s)

Walter Zupa <`zupa@fondazionecoispa.org`>

`strata_scheme_rapana` *stratification scheme (rapa whelk)*

Description

Stratification scheme adopted in the rapa whelk survey (e.g. Black Sea beam trawl survey).

Usage

```
strata_scheme_rapana
```

Format

An object of class `data.frame` with 7 rows and 5 columns.

Author(s)

Walter Zupa <`zupa@fondazionecoispa.org`>

`stratification` *stratification*

Description

Data frame containing the surface area relative to the depth strata adopted in the stratification scheme (strata_scheme) of the demersal surveys (e.g. MEDITS survey).

Usage

```
stratification
```

Format

An object of class `data.frame` with 276 rows and 6 columns.

`stratification_rapana` *stratification (rapa whelk)*

Description

Data frame containing the surface area relative to the depth strata adopted in the stratification scheme (strata_scheme) of the rapa whelk surveys (e.g. Black Sea beam trawl survey).

Usage

```
stratification_rapana
```

Format

An object of class `data.frame` with 7 rows and 6 columns.

`stratum_0_125` *stratum_0_125*

Description

stratum_0_125

Usage

```
stratum_0_125
```

Format

An object of class `PackedSpatVector` of length 1.

`stratum_0_200` *stratum_0_200*

Description

stratum_0_200

Usage

```
stratum_0_200
```

Format

An object of class `PackedSpatVector` of length 1.

`stratum_0_35` *stratum_0_35*

Description

stratum_0_35

Usage

```
stratum_0_35
```

Format

An object of class `PackedSpatVector` of length 1.

`stratum_0_45` *stratum_0_45*

Description

stratum_0_45

Usage

```
stratum_0_45
```

Format

An object of class `PackedSpatVector` of length 1.

`stratum_0_800` *stratum_0_800*

Description

stratum_0_800

Usage

```
stratum_0_800
```

Format

An object of class `PackedSpatVector` of length 1.

`stratum_200_800` *stratum_200_800*

Description

stratum_200_800

Usage

`stratum_200_800`

Format

An object of class `PackedSpatVector` of length 1.

`TA` *TA table example*

Description

TA table example

Usage

`TA`

Format

An object of class `data.frame` with 100 rows and 43 columns.

`TA_cols` *TA table headings*

Description

TA table headings

Usage

`TA_cols`

Format

An object of class `character` of length 22.

Author(s)

Walter Zupa <`zupa@fondazionecoispa.org`>

`TB` *TB table example*

Description

TB table example

Usage

```
TB
```

Format

An object of class `data.frame` with 3059 rows and 19 columns.

`TB_cols` *TB table headings*

Description

TB table headings

Usage

```
TB_cols
```

Format

An object of class `character` of length 8.

Author(s)

Walter Zupa <`zupa@fondazionecoispa.org`>

`TC` *TC table example*

Description

TC table example

Usage

```
TC
```

Format

An object of class `data.frame` with 11185 rows and 22 columns.

`TC_cols` *TC table headings*

Description

TC table headings

Usage

```
TC_cols
```

Format

An object of class `character` of length 11.

Author(s)

Walter Zupa `<zupa@fondazionecoispa.org>`

Index