Package: ambiR (via r-universe)

October 17, 2026

Type Package

Title Calculate AZTI's Marine Biotic Index

Version 0.2.0.9000

Maintainer Ciarán J. Murray <cjm@niva-dk.dk>

Description Calculate AZTI's Marine Biotic Index - AMBI. The included list of benthic fauna species according to their sensitivity to pollution. Matching species in sample data to the list allows the calculation of fractions of individuals in the different sensitivity categories and thereafter the AMBI index. The Shannon Diversity Index H' and the Danish benthic fauna quality index DKI (Dansk Kvalitetsindeks) can also be calculated, as well as the multivariate M-AMBI index. Borja, A., Franco, J. ,Pérez, V. (2000) ` ` A marine biotic index to establish the ecological quality of soft bottom benthos within European estuarine and coastal environments" <doi:10.1016/S0025-326X(00)00061-8>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests spelling, testthat (>= 3.0.0), usethis, devtools, knitr, rmarkdown, scales, ggplot2

Config/testthat/edition 3

Depends R (>= 3.5)

Imports dplyr, tidyr, cli, magrittr, utils, stats, lifecycle

Roxygen list(markdown = TRUE)

URL https://niva-denmark.github.io/ambiR/, https://github.com/niva-denmark/ambiR/, https://github.com/NIVA-Denmark/ambiR

BugReports https://github.com/NIVA-Denmark/ambiR/issues

Config/Needs/website rmarkdown

VignetteBuilder knitr

Language en-GB

Config/roxygen2/version 8.1.0

Repository https://niva-denmark.r-universe.dev

Date/Publication 2026-09-17 09:36:05 UTC

RemoteUrl https://github.com/niva-denmark/ambir

RemoteRef HEAD

RemoteSha 8b6ef7e0c334e8f4ed3efd0e61a0c51ae995d321

Contents

AMBI 2
AMBI_sal 7
AMBI_species 8
DKI 10
DKI2 11
ES 13
H_sal 15
Hdash 16
ISI 17
MAMBI 19
NO_species 21
NQI1 23
NSI 24
test_data 26
test_data_DK 26
test_data_NO 27

Index **28**

`AMBI` *Calculates AMBI, the AZTI Marine Biotic Index*

Description

`AMBI()` matches a list of species counts with the official AMBI species list and calculates the AMBI index.

Usage

```
AMBI(
  df,
  by = NULL,
  var_rep = NA_character_,
  var_species = "species",
  var_count = "count",
  df_species = NULL,
  var_group_AMBI = "group",
  groups_strict = TRUE,
  quiet = FALSE,
  interactive = FALSE,
  format_pct = NA,
  show_class = TRUE,
  exact_species_match = FALSE,
  version = ""
)
```

Arguments

df	a dataframe of species observations
by	a vector of column names found in df by which calculations should be grouped *e.g. c("station","date")*
var_rep	*optional* column name in df which contains the name of the column identifying replicates. If replicates are used, the AMBI index will be calculated for each replicate before an average is calculated for each combination of by variables. If the Shannon diversity index H is calculated this will be done for species counts collected within by groups without any consideration of replicates.
var_species	name of the column in df containing species names
var_count	name of the column in df containing count/density/abundance
df_species	*optional* dataframe of user-specified species groups. By default, the function matches species in df with the official species list from AZTI. If a dataframe with a user-defined list of species is provided, then a search for species groups will also be made in this list. *see Details*.
var_group_AMBI	*optional* name of the column in df_species containing the groups for the AMBI index calculations. These should be specified as integer values from 1 to 7. Any other values will be ignored. If df_species is not specified then var_group_AMBI will be ignored.
groups_strict	By default, any user-assigned species group which conflicts with an original AMBI group assignment will be ignored and the original group remains unchanged. If the argument groups_strict = FALSE is used then user-assigned groups will always override AMBI groups in case of conflict. *DO NOT use this option unless you are sure you know what you are doing! It could invalidate your results.*
quiet	warnings about low numbers of species and/or individuals are contained in the warnings dataframe. By default (quiet = FALSE) these warnings are also shown

in the console. If the function is called with the parameter `quiet = TRUE` then warnings will not be displayed in the console.

`interactive` (default `FALSE`) if a species name in the input data is not found in the AMBI species list, then this will be seen in the output dataframe `matched`. If *interactive* mode is selected, the user will be given the opportunity to assign *manually* a species group (*I, II, III, IV, V*) or to mark the species as *not assigned* to a species group (see details).

`format_pct` (*optional*) By default, frequency results including the fraction of total numbers within each species group are expressed as real numbers . If this is argument is given a positive integer value (e.g. `format_pct = 2`) then the fractions are expressed as percentages with the number of digits shown after the decimal point equal to the number specified. *NOTE* by formatting as percentages, values are converted to text and may lose precision.

`show_class` (default `TRUE`). If `TRUE` then the `AMBI` results will include a column showing the AMBI disturbance classification *Undisturbed*, *Slightly disturbed*, *Moderately disturbed*, or *Heavily disturbed*.

`exact_species_match`
by default, a family name without *sp.* will be matched with a family name on the AMBI (or user-specified) species list which includes *sp.*. If the option `exact_species_match = TRUE` is used, species names will be matched only with identical names.

`version` (default `""`) This argument is passed to `AMBI_species()` when retrieving the list of species and AMBI groups. By default, species names will be matched with the most recent AMBI species list.

Details

The theory behind the AMBI index calculations and details of the method, as developed by Borja et al. (2000),

AMBI method:

Species can be matched to one of five groups, the distribution of individuals between the groups reflecting different levels of stress on the ecosystem.

- *Group I*. Species very sensitive to organic enrichment and present under unpolluted conditions (initial state). They include the specialist carnivores and some deposit- feeding *tubicolous polychaetes*.
- *Group II*. Species indifferent to enrichment, always present in low densities with non-significant variations with time (from initial state, to slight unbalance). These include suspension feeders, less selective carnivores and scavengers.
- *Group III*. Species tolerant to excess organic matter enrichment. These species may occur under normal conditions, but their populations are stimulated by organic enrichment (slight unbalance situations). They are surface deposit-feeding species, as *tubicolous spionids*.
- *Group IV*. Second-order opportunistic species (slight to pronounced unbalanced situations). Mainly small sized *polychaetes*: subsurface deposit-feeders, such as *cirratulids*.
- *Group V*. First-order opportunistic species (pronounced unbalanced situations). These are deposit- feeders, which proliferate in reduced sediments.

The distribution of individuals between these ecological groups, according to their sensitivity to pollution stress, gives a biotic index ranging from 0.0 to 6.0.

$Biotic\ Index = 0.0 * f_I + 1.5 * f_{II} + 3.0 * f_{III} + 4.5 * f_{IV} + 6.0 * f_V$

where:

f_i = fraction of individuals in Group $i \in \{I, II, III, IV, V\}$

Under certain circumstances, the AMBI index should not be used:

- The percentage of individuals not assigned to a group is higher than 20%
- The (not null) number of species is less than 3
- The (not null) number of individuals is less than 6

In these cases the function will still perform the calculations but will also return a warning.(see below)

Results:

The output of the function consists of a list of at least three dataframes:

- `AMBI` containing the calculated `AMBI` index, as well as other information.
- (`AMBI_rep`) generated only if replicates are used, showing the `AMBI` index for each replicate.
- `matched` showing the species matches used.
- `warnings` containing any warnings generated regarding numbers of of species or numbers of individuals.

Species matching and interactive mode:

The function will check for a species list supplied in the function call using the argument `df_species`, if this is specified. The function will also search for names in the AMBI standard list. After this, if no match is found in either, then the species will be recorded with a an `NA`value for species group and will be ignored in calculations.

By calling the function once and then checking the output from this first function call, the user can identify species names which were not matched. Then, if necessary, they can provide or update a dataframe with a list of user-defined species group assignments, before running the function a second time.

Conflicts:

If there is a conflict between a user-provided group assignment for a species and the group specified in the AMBI species group information, only one of them will be selected. The outcome depends on a number of things:

- some species in the AMBI list are considered *reallocatable* (RA) - that is, there can be disagreement about which species group they should belong to. For these species, any user-specified groups will replace the default group.
- if a species is not *reallocatable*, then any user-specified groups will *by default* be ignored. However, if the function is called with the argument `groups_strict = FALSE` then the user-specified groups will override AMBI species groups.

Any conflicts and their outcomes will be recorded in the `matched` output.

interactive *mode:*

If the function is called using the argument `interactive = TRUE` then the user has an opportunity to *manually* assign species groups (*I, II, III, IV, V*) for any species names which were not identified. The user does this by typing `1`, `2`, `3`, `4` or `5` and pressing *Enter*. Alternatively, the user can type `0` to mark the species as recognised but not assigned to a group. By typing *Enter*

without any number the species will be recorded as unidentified (NA). This is the same result which would have been returned when calling the function in non-interactive mode. There are two other options: typing s will display a list of 10 species names which occur close to the unrecognised name when names are sorted in alphabetical order. Entering s a second time will display the next 10 names, and so on. Finally, entering x will abort the interactive species assignment process. Any species groups assigned manually at this point will be discarded and the calculations will process as in the non-interactive mode.
Any user-provided group information will be recorded in the matched results.
See vignette("interactive") for an example.

Value

a list of dataframes:

- AMBI : results of the AMBI index calculations. For each unique combination of by variables, the following values are calculated:
  - AMBI : the AMBI index value
  - AMBI_SD : sample standard deviation of AMBI *included only when replicates are used* has specified var_rep.
  - N : number of individuals
  - S : number of species
  - H : Shannon diversity index *H'*
  - fNA : fraction of individuals *not assigned*, that is, matched to a species in the AMBI species list with *Group 0*. Note that this is different from the number of rows where no match was found. Species not matched are excluded from the totals.
- AMBI_rep : results of the AMBI index calculations *per replicate*. This dataframe is present only if the observation data includes replicates and the user has specified var_rep. Similar to the main AMBI result but does not include results for H (Shannon diversity index) or for AMBI_SD (sample standard deviation of AMBI) which are not estimated at replicate level.
- matched : the original dataframe with columns added from the species list. Contains the following columns:
  - group : showing the species group. Any species/taxa in df which were not matched will have an NA value in this column.
  - RA : a value of 1 indicates that the species is *reallocatable* according to the AMBI list. That is, it could be re-assigned to a different species group.
  - source : this column is included only if a user-specified list was provided df_species, or if species groups were assigned interactively. An "I" in this column indicates that the group was assigned interactively. A "U" shows that the group information came from a user-provided species list. An NA value indicates that no interactive or user-provided changes were applied.
- warnings : a dataframe showing warnings for any combination of by variables a warning where
  - The percentage of individuals not assigned to a group is higher than 20%
  - The (not null) number of species is less than 3
  - The (not null) number of individuals is less than 6

References

Borja, Á., Franco, J., Pérez, V. (2000). "A Marine Biotic Index to Establish the Ecological Quality of Soft-Bottom Benthos Within European Estuarine and Coastal Environments." *Marine Pollution Bulletin* 40 (12) 1100–1114. doi:10.1016/S0025326X(00)000618.

See Also

`MAMBI()` which calculates *M-AMBI* the multivariate AMBI index using results of `AMBI()`.

Examples

```
# example (1) - using test data included with package

  AMBI(test_data, by = c("station"), var_rep = "replicate")


# example (2)

  df <- data.frame(station = c("1", "1", "2", "2", "2"),
  species = c("Acidostoma neglectum",
            "Acrocirrus validus",
            "Acteocina bullata",
            "Austrohelice crassa",
            "Capitella nonatoi"),
            count = c(2, 4, 5, 3, 7))

   AMBI(df, by = c("station"))


# example (3) - conflict with AZTI species group

  df_user <- data.frame(
              species = c("Cumopsis fagei"),
              group = c(1))

  AMBI(test_data, by = c("station"), var_rep = "replicate", df_species = df_user)
```

AMBI_sal *Minimum AMBI as a linear function of salinity*

Description

Used by `DKI2()`, adjusting the `AMBI` index to account for decreasing species diversity with decreasing salinity.

Usage

```
AMBI_sal(psal, intercept = 3.083, slope = -0.111)
```

Arguments

psal	numeric, salinity
intercept	numeric, default 3.083
slope	numeric, default -0.111

Details

AMBI_sal() and H_sal() are named, respectively, *AMBI_min* and *H_max* in the DKI documentation (Carstensen et al., 2014). They are renamed in ambiR to reflect the fact that they are functions of salinity and *not* minimum or maximum values from data being used.

Value

a numeric value AMBI_min

Examples

```
AMBI_sal(20.1)
```

AMBI_species *Returns species list for AMBI calculations*

Description

AMBI_species() returns a dataframe with list of species and AMBI group. Called by the function AMBI() and then used to match species in observed data and find species groups.

latest version 8th October 2024

Usage

```
AMBI_species(version = "")
```

Arguments

version	*string*, version of the species list to return. The default value is the empty string ("") which returns the latest version of the list *(8th October 2024)*. Other valid values for version are "2022" *(31st May 2022)*, "2017" *(5th June 2017)* or "2014" *(13th November 2014)*. Using version = "NO" will return the latest version used by Norwegian authorities (see NO_species())

Details

The species groups, as described by Borja et al. (2000):

- *Group I*
  Species very sensitive to organic enrichment and present under unpolluted conditions (initial state). They include the specialist carnivores and some deposit-feeding *tubicolous polychaetes*.

- *Group II*
  Species indifferent to enrichment, always present in low densities with non-significant variations with time (from initial state, to slight unbalance). These include suspension feeders, less selective carnivores and scavengers.

- *Group III*
  Species tolerant to excess organic matter enrichment. These species may occur under normal conditions, but their populations are stimulated by organic enrichment (slight unbalance situations). They are surface deposit-feeding species, such as *tubicolous spionids*.

- *Group IV*
  Second-order opportunistic species (slight to pronounced unbalanced situations). Mainly small sized *polychaetes*: subsurface deposit-feeders, such as *cirratulids*.

- *Group V*
  First-order opportunistic species (pronounced unbalanced situations). These are deposit-feeders, which proliferate in reduced sediments.

Value

A data frame with 11,952 rows* and 3 columns:

species Species name or genus (spp.)

group Species group for AMBI index calculation: `1`, `2`, `3`, `4` or `5`. A value of `0` indicates that the species is not assigned to a species group.

RA reallocatable (`0` or `1`), a `1` indicates that a species could be re-assigned to a different species group.

References

Borja, Á., Franco, J., Pérez, V. (2000). "A Marine Biotic Index to Establish the Ecological Quality of Soft-Bottom Benthos Within European Estuarine and Coastal Environments." *Marine Pollution Bulletin* 40 (12): 1100–1114. doi:10.1016/S0025326X(00)000618.

See Also

`AMBI()` which uses the species list to calculate the AMBI index.

Examples

```
AMBI_species() %>% head()

AMBI_species() %>% tail()
```

DKI *Calculates DKI* (v1)*, the Danish Quality Index*

Description

DKI() calculates the original version of the Danish quality index DKI (Carstensen et al., 2014)

The DKI is based on AMBI and can only be calculated after first calculating *AMBI*, the AZTI Marine Biotic Index, and *H'*, the Shannon diversity index. Both indices are included in output from the function AMBI().

The function uses an estimated maximum possible value of H' in Danish waters (H_max) as a reference value to normalise DKI. If this value is not specified as an argument, the default value is used (5.0) (Borja et al., 2007)

Usage

```
DKI(AMBI, H, N, S, H_max = 5)
```

Arguments

AMBI	AMBI, the AZTI Marine Biotic Index, calculated using AMBI()
H	H', the Shannon diversity index, calculated using Hdash()
N	number of individuals - returned by both AMBI() and Hdash()
S	number of species - returned by both AMBI() and Hdash()
H_max	maximum H' used to normalise AMBI, *default 5*

Details

The AMBI() and Hdash() functions take a dataframe of observations as an argument. The DKI functions, DKI2() and DKI(), do *not* take a dataframe as an argument. Instead they take values of the input parameters, either single values or as vectors.

To calculate DKI for a dataframe of AMBI values, it could be called from e.g. within a dplyr::mutate() function call. See the examples below.

Value

DKI index value

References

Borja, A., Josefson, A., Miles, A., Muxika, I., Olsgard, F., Phillips, G., Rodriguez, J., Rygg, B. (2007). An Approach to the Intercalibration of Benthic Ecological Status Assessment in the North Atlantic Ecoregion, According to the European Water Framework Directive. *Marine Pollution Bulletin*, 55(1-6), 42-52. doi:10.1016/j.marpolbul.2006.08.018.

Carstensen, J., Krause-Jensen, D., Josefson, A. (2014). "Development and testing of tools for intercalibration of phytoplankton, macrovegetation and benthic fauna in Danish coastal areas." Aarhus University, DCE – Danish Centre for Environment and Energy, 85 pp. *Scientific Report from DCE – Danish Centre for Environment and Energy* No. 93. https://dce2.au.dk/pub/SR93.pdf

See Also

DKI v1 has been superseded by DKI2() a salinity-normalised version of DKI. For more details, see‘vignette("other-indices").

Examples

```
# Simple example

DKI(AMBI = 1.61, H = 2.36, N = 25, S = 6)


# ------ Example workflow for calculating DKI from species counts ----

# calculate AMBI index
df <- AMBI(test_data_DK, by = c("station"), var_rep="sample")[["AMBI"]]

# modify names which were not recognized by AMBI() and recalculate
df_obs <- dplyr::mutate(test_data_DK, species = gsub(" indet\\.", "", species))
df <- ambiR::AMBI(df_obs, by = c("station"), var_rep="sample")[["AMBI"]]

# show AMBI results
df

# calculate DKI from AMBI results
df <- dplyr::mutate(df, DKI = DKI(AMBI, H, N, S))

dplyr::select(df, station, AMBI, H, N, S, DKI)
```

DKI2 *Calculates DKI (v2), the revised Danish Quality Index*

Description

DKI2() calculates a salinity-normalised version of the Danish quality index (DKI) (Carstensen et al., 2014)

The *DKI* index is based on AMBI and can only be calculated after first calculating *AMBI*, the AZTI Marine Biotic Index, and *H'*, the Shannon diversity index. Both indices are included in output from the function `AMBI()`.

This function uses linear relationships between salinity and limits for `AMBI` and `Hdash` to normalise the index. This is done to account for expected lower species diversity in regions with lower salinity.

Since the index is normalised to salinity, the function also requires measured or estimated salinity `psal` as an argument.

Usage

```
DKI2(AMBI, H, N, psal)
```

Arguments

`AMBI`	AMBI, the AZTI Marine Biotic Index, calculated using `AMBI()`
`H`	H', the Shannon diversity index, calculated using `Hdash()`
`N`	number of individuals - generated by both `AMBI()` and `Hdash()`
`psal`	salinity

Details

The `AMBI()` and `Hdash()` functions take a dataframe of observations as an argument. The *DKI* functions, `DKI2()` and `DKI()`, do *not* take a dataframe as an argument. Instead they take values of the input parameters, either single values or as vectors.

To calculate DKI for a dataframe of `AMBI` values, it could be called from e.g. within a `dplyr::mutate()` function call. See the examples below.

Value

`DKI` index value

References

Carstensen, J., Krause-Jensen, D., Josefson, A. (2014). "Development and testing of tools for inter-calibration of phytoplankton, macrovegetation and benthic fauna in Danish coastal areas." Aarhus University, DCE – Danish Centre for Environment and Energy, 85 pp. *Scientific Report from DCE – Danish Centre for Environment and Energy* No. 93. https://dce2.au.dk/pub/SR93.pdf

See Also

For more details, see'vignette("other-indices").

- `DKI()` calculate DKI using the original method
- `AMBI_sal()` minimum AMBI for normalisation = *f(salinity)*
- `H_sal()` maximum H' for normalisation = *f(salinity)*

Examples

```
# Simple example

DKI2(AMBI = 1.61, H = 2.36, N = 25, psal = 21.4)


# ------ Example workflow for calculating DKI (v2) from species counts ----

# calculate AMBI index
df <- AMBI(test_data_DK, by = c("station"), var_rep = "sample")[["AMBI"]]

# modify names which were not recognized by AMBI() and recalculate
df_obs <- dplyr::mutate(test_data_DK, species = gsub(" indet\\.", "", species))
df <- ambiR::AMBI(df_obs, by = c("station"), var_rep="sample")[["AMBI"]]

# show AMBI results
df

# add salinity values - these are realistic but invented values
df$psal <- c(21.3, 26.5)

# calculate DKI from AMBI results
df <- dplyr::mutate(df, DKI=DKI2(AMBI, H, N, psal))

dplyr::select(df, station, AMBI, H, N, psal, DKI)
```

ES *Calculates ES the Hurlbert diversity index*

Description

ES() matches a list of species counts with the AMBI species list and calculates *ES* the Hurlbert diversity index, which gives the expected sample species richness for a subsample of the population. (Hurlbert, 1971)

Usage

```
ES(
  df,
  by = NULL,
  var_species = "species",
  var_count = "count",
  check_species = TRUE,
  df_species = NULL,
  n = 100
)
```

Arguments

df	a dataframe of species observations
by	a vector of column names found in df by which calculations should be grouped *e.g. c("station","date")*
var_species	name of the column in df containing species names
var_count	name of the column in df containing count/density/abundance
check_species	boolean, default = TRUE. If TRUE, then only species found in the species list are included in *H'* index. By default, the AZTI species list is used.
df_species	*optional* dataframe with user-specified species list.
n	subsample size, usually indicated in the abbreviated name for the index. e.g. ES100 is ES calculated with a subsample of 100 individuals (the default) The sampling is done for each *by* group.

Details

If the function is called with the argument check_species = TRUE then only species which are successfully matched with the specified species list are included in the calculations. This is the default. If the function is called with check_species = FALSEthen all rows are counted.

Value

a list of two dataframes:

- ES : results of the ES index calculations. For each unique combination of byvariables the following values are calculated:
  - ES<n> : the expected sample species richness, *ES* for a sample size n of 100, the variable name is ES100
  - S : the number of species
  - N : the number of individuals
- match : the original dataframe with columns added from the species list. For a user-specified list provided df_species, all columns will be included. If the user-specified species list contains only a single column with species names, then a new column match will be created, with a value of 1 indicating a match and an NA value where no match was found.

For the default AZTI species list the following additional columns will be included:

- group : showing the AMBI species group
- RA : indicating that the species is *reallocatable* according to the AZTI list. That is, it could be re-assigned to a different species group.

References

Hurlbert, S.H. (1971), The Nonconcept of Species Diversity: A Critique and Alternative Parameters. Ecology, 52: 577-586. doi:10.2307/1934145

See Also

For more details, see‘vignette("other-indices").

Examples

```
ES(test_data)
```

`H_sal` *Maximum H' as a linear function of salinity*

Description

Used by `DKI2()`, adjusting the Shannon diversity index `H'` to account for decreasing species diversity with decreasing salinity.

Usage

```
H_sal(psal, intercept = 2.117, slope = 0.086)
```

Arguments

`psal`	numeric salinity
`intercept`	numeric, default 2.117
`slope`	numeric default 0.086

Details

`AMBI_sal()` and `H_sal()` are named, respectively, *AMBI_min* and *H_max* in the DKI documentation (Carstensen et al., 2014). They are renamed in ambiR to reflect the fact that they are functions of salinity and *not* minimum or maximum values from data being used.

Value

a numeric value H_max

Examples

```
H_sal(20.1)
```

Hdash *Calculates H' the Shannon diversity index*

Description

`Hdash()` matches a list of species counts with the AMBI species list and calculates *H'* the Shannon diversity index. (Shannon, 1948)

Usage

```
Hdash(
  df,
  by = NULL,
  var_species = "species",
  var_count = "count",
  check_species = TRUE,
  df_species = NULL
)
```

Arguments

`df`	a dataframe of species observations
`by`	a vector of column names found in `df` by which calculations should be grouped *e.g. c("station","date")*
`var_species`	name of the column in `df` containing species names
`var_count`	name of the column in `df` containing count/density/abundance
`check_species`	boolean, default = TRUE. If TRUE, then only species found in the species list are included in *H'* index. By default, the AZTI species list is used.
`df_species`	*optional* dataframe with user-specified species list.

Details

If the function is called with the argument `check_species = TRUE` then only species which are successfully matched with the specified species list are included in the calculations. This is the default. If the function is called with `check_species = FALSE`then all rows are counted.

Value

a list of two dataframes:

- `H` : results of the AMBI index calculations. For each unique combination of `by`variables the following values are calculated:
  - `H` : the Shannon diversity Index, *H'*
  - `S` : the number of species
  - `N` : the number of individuals

- `match` : the original dataframe with columns added from the species list. For a user-specified list provided `df_species`, all columns will be included. If the user-specified species list contains only a single column with species names, then a new column `match` will be created, with a value of `1` indicating a match and an `NA` value where no match was found.

For the default AZTI species list the following additional columns will be included:

- `group` : showing the AMBI species group
- `RA` : indicating that the species is *reallocatable* according to the AZTI list. That is, it could be re-assigned to a different species group.

References

Shannon, C. E. (1948) "A mathematical theory of communication," in *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379-423. doi:10.1002/j.15387305.1948.tb01338.x

Examples

```
Hdash(test_data, by=c("station"))
```

`ISI` *Calculates ISI, the Indicator Species Index*

Description

`ISI()` matches a list of species with the Norwegian species list and calculates *ISI* the Indicator Species Index. (Borgersen et al, 2019)

Usage

```
ISI(
  df,
  by = NULL,
  var_species = "species",
  df_species = NULL,
  version = "2018"
)
```

Arguments

`df`	a dataframe of species observations
`by`	a vector of column names found in `df` by which calculations should be grouped *e.g. c("station","date")*
`var_species`	name of the column in `df` containing species names
`df_species`	*optional* dataframe with user-specified species list.
`version`	*string*, version of the index to be calculated. The default value is `"2018"`. The the only other valid value for `version` is `"2012"`.

Details

If the function is called with the argument `check_species = TRUE` then only species which are successfully matched with the specified species list are included in the calculations. This is the default. If the function is called with `check_species = FALSE`then all rows are counted.

Value

a list of two dataframes:

- `ISI` : results of the ISI index calculations. For each unique combination of `by`variables the following values are calculated:
  - `ISI<version>` : the sensitivity index `ISI2018` or `ISI2012`
  - `S` : the number of species
  - `N` : the number of individuals
- `match` : the original dataframe with columns added from the species list. For a user-specified list provided `df_species`, all columns will be included. If the user-specified species list contains only a single column with species names, then a new column `match` will be created, with a value of `1` indicating a match and an `NA` value where no match was found. One of the following columns will be included, depending on the ISI version:
  - `ES100min5_2018` : species sensitivity values used to calculate `ISI2018`
  - `ES100min5_2012` : species sensitivity values used to calculate `ISI2012`

References

Borgersen, G., Trannum, H.C., Gundersen, H., Vedal, J. (2019). Oppdatering av bløtbunnsartenes sensitivitetsverdier (Report No. 7366). Norsk institutt for vannforskning (NIVA). https://hdl.handle.net/11250/2600903

Rygg, B., Norling, K. (2013). Norwegian Sensitivity Index (NSI) for marine macroinvertebrates, and an update of Indicator Species Index (ISI). (Report No. 6475). Norsk institutt for vannforskning (NIVA). 46 pp.

See Also

`NSI()` which calculates the Norwegian Sensitivity Index. For more details, see‘vignette("other-indices").

Examples

```
ISI(test_data_NO, by="station", var_species ="species")
```

MAMBI *Calculates M-AMBI, the multivariate AZTI Marine Biotic Index*

Description

Calculates M-AMBI the multivariate AMBI index, based on the three separate species diversity metrics:

- AMBI index `AMBI`.
- Shannon diversity index `H'`
- Species richness `S`.

"AMBI, richness and diversity, combined with the use, in a further development, of factor analysis together with discriminant analysis, is presented as an objective tool (named here M-AMBI) in assessing ecological quality status" (Muxika et al., 2007)

Usage

```
MAMBI(
  df,
  by = NULL,
  var_H = "H",
  var_S = "S",
  var_AMBI = "AMBI",
  limits_AMBI = c(bad = 6, high = 0),
  limits_H = c(bad = 0, high = NA),
  limits_S = c(bad = 0, high = NA),
  bounds = c(PB = 0.2, MP = 0.39, GM = 0.53, HG = 0.77)
)
```

Arguments

`df`	a dataframe of diversity metrics.
`by`	a vector of column names found in `df` by which calculations should be grouped *e.g.* `c("station")`. If grouping columns are specified, then the mean values of the 3 metrics will be calculated within each group before calculating `M-AMBI` (default `NULL`).
`var_H`	name of the column in `df` containing `H'` Shannon species diversity (default `"H"`).
`var_S`	name of the column in `df` containing `S` species richness (default `"S"`).
`var_AMBI`	name of the column in `df` containing `AMBI` index (default `"AMBI"`).
`limits_AMBI`	named vector with length 2, specifying the values of `AMBI` corresponding to *(i)* worst possible condition (`"bad"`) where `M-AMBI` and `EQR` are equal to 0.0 and *(ii)* the best possible condition (`"high"`) where `M-AMBI` and `EQR` are equal to 1.0. Default `c("bad" = 6, "high" = 0)`.

limits_H — named vector with length 2, specifying the values of H' corresponding to *(i)* worst possible condition ("bad") where M-AMBI and EQR are equal to 0.0 and *(ii)* the best possible condition ("high") where M-AMBI and EQR are equal to 1.0. Default c("bad" = 0, "high" = NA). If the "bad" value is NA then the lowest value occurring in df and if "high" is NA then the highest value will be used.

limits_S — named vector with length 2, specifying the values of S corresponding to *(i)* worst possible condition ("bad") where M-AMBI and EQR are equal to 0.0 and *(ii)* the best possible condition ("high") where M-AMBI and EQR are equal to 1.0. Default c("bad" = 0, "high" = NA). If the "bad" value is NA then the lowest value occurring in df and if "high" is NA then the highest value will be used.

bounds — A named vector (*length 4*) of EQR boundary values used to normalise M-AMBI to EQR values where the boundary between *Good* and *Moderate* ecological status is 0.6. They specify the values of M-AMBI corresponding to the boundaries between *(i) Poor* and *Bad* status ("PB"), *(ii) Moderate* and *Poor* status ("MP"), *(iii) Good* and *Moderate* status ("GM"), and *(iv) High* and *Good* status ("HG"). Default c("PB" = 0.2, "MP" = 0.39, "GM" = 0.53, "HG" = 0.77).

Details

The input dataframe df should contain the three metrics AMBI, H' and S, identified by the column names var_AMBI (default "AMBI"), var_H (default "H") and var_S (default "S").

If any of these three metrics is not found in the input data, then the function will return an error.

AMBI is calculated using the AMBI() function. H' can be calculated using the Hdash() function but it is also included as additional output from AMBI() when called with the non-default argument H = TRUE. S is an output from both functions AMBI() and Hdash().

This means that the input to MAMBI() can be generated from species count data using only using the AMBI() function.

Value

a dataframe containing results of the M-AMBI index calculations. For each unique combination of by variables, the following values are calculated:

- M-AMBI : the M-AMBI index value.
- x,y,z : factor scores giving coordinates in the new factor space.

If no by variables are specified (by = NULL), then M-AMBI will be calculated for each row in df.

In addition, the dataframe returned contains 2 *extra* rows. These contain the limits applied for each of the metrics, corresponding to "bad" (M-AMBI = 0.0) and "high" (M-AMBI = 1.0), as specified in the arguments limits_AMBI, limits_H, limits_S or taken from data.

References

Muxika, I., Borja, A., Bald, J. (2007) "Using historical data, expert judgement and multivariate analysis in assessing reference conditions and benthic ecological status, according to the European Water Framework Directive", Marine Pollution Bulletin, 55, 1–6, doi:10.1016/j.marpolbul.2006.05.025.

See Also

AMBI() which calculates the indices required as input for MAMBI().

Examples

```
df <- data.frame(station = c(1, 1, 1, 2, 2, 2, 3, 3),
                 replicates = c("a", "b", "c", "a", "b", "c", "a", "b"),
                 AMBI = c(1.8, 1.5, 1.125, 1.875, 2.133, 1.655, 3.5, 4.75),
                 H = c(1.055, 0.796, 0.562, 2.072, 2.333, 1.789, 1.561, 1.303),
                 S = c(3, 3, 2, 12, 12, 10, 5, 6))

MAMBI(df, by = c("station"))
```

NO_species *Returns species list used by Norwegian index calculations*

Description

NO_species() returns a dataframe with list of species and sensitivities for use in functions calculating Norwegian indices ISI() and NSI().

latest version 13th November 2024

Usage

```
NO_species()
```

Details

This table contains the species-specific sensitivities used to calculate the Norwegian indices ISI() and NSI().

Similarly to the AMBI index, the principle of the Norwegian quality indices is that high species diversity indicates good environmental status and species primarily found at sites with high diversity and considered sensitive, while sites with low species diversity are considered as stressed environments and species more frequently found at such sites are considered more tolerant. Species diversity thus serves as a proxy for the degree of disturbance at a site.

The calculation of species sensitivity values for ISI and NSI is based on analysis of ES100 values calculated for a large number of grab samples from the Norwegian monitoring data. The sensitivity values for a species are the weighted averages of ES100 values across samples where the species was present.

- *ES100avg_2018*
  The weighted average of ES100 values across samples where the species was present. Theses are the updated 2018 values.

- *ES100min5_2018*
  The weighted average of `ES100` values across samples where the species was present but based on the 5 samples having the lowest `ES100` value. This value was determined only for species found in at least six samples. Theses are the updated 2018 values.

- *ES100avg_2012*
  The weighted average of `ES100` values across samples where the species was present. These are the values determined by Rygg & Norling (2013)

- *ES100min5_2012*
  The weighted average of `ES100` values across samples where the species was present but based on the 5 samples having the lowest `ES100` value. This value was determined only for species found in at least six samples. These are the values determined by Rygg & Norling (2013)

, For more details, please see Borgersen et al. (2019):

Value

A data frame with 760 rows* and 10 columns:

species Species name or genus (spp.)

ES100avg_2018 Species sensitivity value for ES100 used to calculate `NSI()` version 2018

ES100min5_2018 Species sensitivity value used to calculate `ISI()` version 2018

ES100avg_2012 Species sensitivity value for ES100 used to calculate `NSI()` version 2012

ES100min5_2012 Species sensitivity value used to calculate `ISI()` version 2012

group_AMBI AMBI species group. This information is included by Borgersen et al. (2019). For the authoritative list see `AMBI_species()`

group_NSI NSI species group. See Borgersen et al. (2019)

group_ISI ISI species group

Revidert Indicates that information has been revised.

Kommentar Comment in Norwegian describing revision.

References

"Sensitivitetsverdier for NSI2018 og ISI 2018_13.11.2024.xlsx" retrieved 01-09-2026 https://www.vannportalen.no/sharepoi
Borgersen, G., Trannum, H.C., Gundersen, H., Vedal, J. (2019). Oppdatering av bløtbunnsartenes sensitivitetsverdier (Report No. 7366). Norsk institutt for vannforskning (NIVA). https://hdl.handle.net/11250/2600903 Rygg, B., Norling, K. (2013). Norwegian Sensitivity Index (NSI) for marine macroinvertebrates, and an update of Indicator Species Index (ISI). (Report No. 6475). Norsk institutt for vannforskning (NIVA). 46 pp.

See Also

`NSI()` and `ISI()` use this species list to calculate these indices.

Examples

```
NO_species() %>% head()

NO_species() %>% tail()
```

NQI1 *Calculates NQI1, the Norwegian Quality Index*

Description

`NQI1()` calculates the Norwegian Quality Index, NQI1 (Rygg, 2006)

The NQI1 is based on AMBI and can only be calculated after first calculating *AMBI*, the AZTI Marine Biotic Index, using output from the function `AMBI()`.

NQI1 is the only Norwegian benthic fauna index which is inter-calibrated with all countries belonging to the North East Atlantic Geographical Intercalibration Group (NEAGIG)(Borgersen et al, 2019) .

Usage

```
NQI1(AMBI, N, S)
```

Arguments

`AMBI`	AMBI, the AZTI Marine Biotic Index, calculated using `AMBI()`
`N`	number of individuals - returned by both `AMBI()` and `Hdash()`
`S`	number of species - returned by both `AMBI()` and `Hdash()`

Details

While `AMBI()` takes a dataframe of observations as an argument, `NQI1()` does *not*. Instead, similarly to the DKI functions, it takes values of the input parameters, either single values or as vectors.

To calculate NQI1 for a dataframe of `AMBI` values, it could be called from e.g. within a `dplyr::mutate()` function call. See the examples below.

Value

`NQI1` index value

References

Rygg, B. (2006) Developing indices for quality-status classification of marine soft-bottom fauna in Norway. (Report No. 5208) Norwegian Institute for Water Research (NIVA). 33 pp. https://hdl.handle.net/11250/213219 Borgersen, G., Trannum, H.C., Gundersen, H., Vedal, J. (2019). Oppdatering av bløtbunnsartenes sensitivitetsverdier (Report No. 7366). Norsk institutt for vannforskning (NIVA). https://hdl.handle.net/11250/2600903

See Also

For more details, see‘vignette("other-indices").

Examples

```
# Simple example

NQI1(AMBI = 1.61, N = 25, S = 6)


# ------ Example workflow for calculating NQI1 from species counts ----

# calculate AMBI index using Norwegian example data

df <- AMBI(test_data_NO,
               by = "station",
               var_rep = "sample",
               var_species = "species",
               var_count = "count",
               quiet = TRUE)[["AMBI"]]

# show AMBI results
df

# calculate NQI1 from AMBI results
df <- dplyr::mutate(df, NQI1 = NQI1(AMBI, N, S))
dplyr::select(df, AMBI, N, S, NQI1)
```

NSI *Calculates NSI, the Norwegian Sensitivity Index*

Description

`NSI()` matches a list of species counts with the Norwegian species list and calculates *NSI* the Norwegian Sensitivity Index. (Borgersen et al, 2019)

Usage

```
NSI(
  df,
  by = NULL,
  var_species = "species",
  var_count = "count",
  df_species = NULL,
  version = "2018"
)
```

Arguments

df	a dataframe of species observations
by	a vector of column names found in df by which calculations should be grouped *e.g. c("station","date")*
var_species	name of the column in df containing species names
var_count	name of the column in df containing count/density/abundance
df_species	*optional* dataframe with user-specified species list.
version	*string*, version of the index to be calculated. The default value is "2018". The the only other valid value for version is "2012".

Details

If the function is called with the argument check_species = TRUE then only species which are successfully matched with the specified species list are included in the calculations. This is the default. If the function is called with check_species = FALSEthen all rows are counted.

Value

a list of two dataframes:

- NSI : results of the NSI index calculations. For each unique combination of byvariables the following values are calculated:
  - NSI<version> : the sensitivity index NSI2018 or NSI2012
  - S : the number of species
  - N : the number of individuals
- match : the original dataframe with columns added from the species list. For a user-specified list provided df_species, all columns will be included. If the user-specified species list contains only a single column with species names, then a new column match will be created, with a value of 1 indicating a match and an NA value where no match was found. One of the following columns will be included, depending on the NSI version:
  - ES100avg_2018 : species sensitivity values used to calculate NSI2018
  - ES100avg_2012 : species sensitivity values used to calculate NSI2012

References

Borgersen, G., Trannum, H.C., Gundersen, H., Vedal, J. (2019). Oppdatering av bløtbunnsartenes sensitivitetsverdier (Report No. 7366). Norsk institutt for vannforskning (NIVA). https://hdl.handle.net/11250/2600903

Rygg, B., Norling, K. (2013). Norwegian Sensitivity Index (NSI) for marine macroinvertebrates, and an update of Indicator Species Index (ISI). (Report No. 6475). Norsk institutt for vannforskning (NIVA). 46 pp.

See Also

ISI() which calculates the Indicator Species Index, another Norwegian benthic diversity index. For more details, see‘vignette("other-indices").

Examples

```
NSI(test_data_NO, by="station", var_species ="species", var_count="count")
```

`test_data` *AMBI test dataset*

Description

Example data included with the AMBI tool from AZTI (*example_BDheader.xls*).

Usage

```
test_data
```

Format

The test dataset `test_data` has 53 rows and 4 variables:

station 3 sampling sites 1, 2, 3

replicate unique samples taken at each site, identified *a*, *b*, *c*

species Name of observed species/taxon

count Number of individuals

Source

AZTI

Examples

```
head(test_data)
```

`test_data_DK` *Danish test dataset*

Description

Example data from Danish monitoring data. Used for testing DKI calculations.

Usage

```
test_data_DK
```

Format

The test dataset `test_data_DK` has 112 rows and 4 variables:

station 2 sampling sites "42", "49" *(ObservationsStedNavn)*

sample unique samples taken at each site, identified *1*, *2*, *3*, *4*, *5 (Prøvetagningsnummer)*

species Name of observed species/taxon *(Artsnavn)*

count Number of individuals *(Antal)*

Source

Danish National Aquatic Monitoring and Assessment Programme.

Examples

```
head(test_data_DK)
```

`test_data_NO` *Norwegian test dataset*

Description

Example data from Norwegian monitoring data. Used for testing Norwegian index calculations.

Usage

```
test_data_NO
```

Format

The test dataset `test_data_NO` has 396 rows and 4 variables:

station 2 sampling sites "Ærøy (U10)", "Ærøydypet (U12)" *(Vannlokalitetsnavn)*

sample unique samples taken at each site, identified *G1*, *G2 (Provenr)*

species Name of observed species/taxon *(VitenskapligNavn)*

count Number of individuals *(Verdi)*

Source

Vann-Nett

Examples

```
head(test_data_NO)
```

Index

∗ **datasets**
test_data, 26
test_data_DK, 26
test_data_NO, 27

AMBI, 2
AMBI(), *2*, *8–10*, *12*, *20, 21*, *23*
AMBI_sal, 7
AMBI_sal(), *8*, *12*, *15*
AMBI_species, 8
AMBI_species(), *8*, *22*

DKI, 10
DKI(), *10*, *12*
DKI2, 11
DKI2(), *7*, *10–12*, *15*
dplyr::mutate(), *10*, *12*, *23*

ES, 13
ES(), *13*

H_sal, 15
H_sal(), *8*, *12*, *15*
Hdash, 16
Hdash(), *10*, *12*, *16*, *20*, *23*

ISI, 17
ISI(), *17*, *21, 22*, *25*

MAMBI, 19
MAMBI(), *7*

NO_species, 21
NO_species(), *8*, *21*
NQI1, 23
NQI1(), *23*
NSI, 24
NSI(), *18*, *21, 22*, *24*

test_data, 26
test_data_DK, 26
test_data_NO, 27