

Package: ambiR (via r-universe)

May 19, 2026

Type Package

Title Calculate AZTI's Marine Biotic Index

Version 0.1.1

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, ggplot2, scales

Config/testthat/edition 3

RoxygenNote 7.3.3

Depends R (>= 3.5)

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

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
Repository https://niva-denmark.r-universe.dev
Date/Publication 2025-12-20 11:54:28 UTC
RemoteUrl https://github.com/niva-denmark/ambir
RemoteRef HEAD
RemoteSha 4e75b9ce5143fd8ea8b0108d9eba206b75ce0a95

Contents

AMBI	2
AMBI_sal	7
AMBI_species	8
DKI	9
DKI2	11
H_sal	12
Hdash	13
MAMBI	14
test_data	17

Index **18**

AMBI	<i>Calculates AMBI, the AZTI Marine Biotic Index</i>
------	--

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
  )

```

Arguments

<code>df</code>	a dataframe of species observations
<code>by</code>	a vector of column names found in <code>df</code> by which calculations should be grouped <i>e.g.</i> <code>c("station","date")</code>
<code>var_rep</code>	<i>optional</i> column name in <code>df</code> 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 <code>by</code> variables. If the Shannon diversity index H is calculated this will be done for species counts collected within <code>by</code> groups without any consideration of replicates.
<code>var_species</code>	name of the column in <code>df</code> containing species names
<code>var_count</code>	name of the column in <code>df</code> containing count/density/abundance
<code>df_species</code>	<i>optional</i> dataframe of user-specified species groups. By default, the function matches species in <code>df</code> 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. <i>see</i> Details .
<code>var_group_AMBI</code>	<i>optional</i> name of the column in <code>df_species</code> 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 <code>df_species</code> is not specified then <code>var_group_AMBI</code> will be ignored.
<code>groups_strict</code>	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 <code>groups_strict = FALSE</code> is used then user-assigned groups will always override AMBI groups in case of conflict. <i>DO NOT use this option unless you are sure you know what you are doing! It could invalidate your results.</i>
<code>quiet</code>	warnings about low numbers of species and/or individuals are contained in the warnings dataframe. By default (<code>quiet = FALSE</code>) these warnings are also shown in the console. If the function is called with the parameter <code>quiet = TRUE</code> then warnings will not be displayed in the console.
<code>interactive</code>	(default <code>FALSE</code>) 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 <i>interactive</i> mode is selected, the user will be given the opportunity to assign <i>manually</i> a species group (<i>I, II, III, IV, V</i>) or to mark the species as <i>not assigned</i> to a species group (see details).
<code>format_pct</code>	(<i>optional</i>) By default, frequency results including the fraction of total numbers within each species group are expressed as real numbers. If this argument is given a positive integer value (e.g. <code>format_pct = 2</code>) then the fractions are expressed as percentages with the number of digits shown after the decimal point equal to the number specified. <i>NOTE</i> 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.

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.

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

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 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 (*8. October 2024*). Currently, the only other valid value for version is "2022" (*31. May 2022*).

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)

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'* *H_max* in Danish waters as a reference value to normalise DKI. If this value is not specified as an argument, the default value is used 5.0

"However, in the present exercise, the Danish method used H_{max} (~5) as a kind of reference" ([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 - generated by both AMBI() and Hdash()
S	number of species - generated by both AMBI() and Hdash()
H_max	maximum H' used to normalise AMBI, <i>default 5</i>

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](https://doi.org/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.

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
```

```
dfAMBI <- AMBI(test_data, by = c("station"), var_rep="replicate")["AMBI"]

# show AMBI results
dfAMBI

# calculate DKI from AMBI results
dplyr::mutate(dfAMBI, DKI = DKI(AMBI, H, N, S))
```

DKI2

*Calculates DKI (v2)***Description**

`DKI2()` calculate 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.

@references 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>

Usage

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

Arguments

AMBI	AMBI, the AZTI Marine Biotic Index, calculated using <code>AMBI()</code>
H	H' , the Shannon diversity index, calculated using <code>Hdash()</code>
N	number of individuals - generated by both <code>AMBI()</code> and <code>Hdash()</code>
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

See Also

- [DKI\(\)](#) calculate DKI using the original method
- [AMBI_sal\(\)](#) minimum AMBI for normalisation = $f(\text{salinity})$
- [H_sal\(\)](#) maximum H' for normalisation = $f(\text{salinity})$

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
dfAMBI <- AMBI(test_data, by = c("station"), var_rep = "replicate")["AMBI"]

# show AMBI results
dfAMBI

# add salinity values - these are realistic but invented values
dfAMBI <- dplyr::mutate(dfAMBI, psal=ifelse(station == 1, 21.3, 26.5))

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

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	<i>Calculates H' the Shannon diversity index</i>
-------	---

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

<code>df</code>	a dataframe of species observations
<code>by</code>	a vector of column names found in <code>df</code> by which calculations should be grouped <i>e.g.</i> <code>c("station", "date")</code>
<code>var_species</code>	name of the column in <code>df</code> containing species names
<code>var_count</code>	name of the column in <code>df</code> containing count/density/abundance
<code>check_species</code>	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.
<code>df_species</code>	<i>optional</i> 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 byvariables 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"))
```

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 <i>e.g.</i> 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 (<i>length 4</i>) of EQR boundary values used to normalise M-AMBI to EQR values where the boundary between <i>Good</i> and <i>Moderate</i> ecological status is 0.6. They specify the values of M-AMBI corresponding to the boundaries between (i) <i>Poor</i> and <i>Bad</i> status ("PB"), (ii) <i>Moderate</i> and <i>Poor</i> status ("MP"), (iii) <i>Good</i> and <i>Moderate</i> status ("GM"), and (iv) <i>High</i> and <i>Good</i> 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](https://doi.org/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"))
```

test_data	<i>AMBI test dataset</i>
-----------	--------------------------

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)
```

Index

* datasets

test_data, 17

AMBI, 2

AMBI(), 2, 8–11, 16

AMBI_sal, 7

AMBI_sal(), 7, 12, 13

AMBI_species, 8

AMBI_species(), 8

DKI, 9

DKI(), 9–12

DKI2, 11

DKI2(), 7, 10–12

dplyr::mutate(), 10, 11

H_sal, 12

H_sal(), 7, 12, 13

Hdash, 13

Hdash(), 10, 11, 13, 16

MAMBI, 14

MAMBI(), 6

test_data, 17