

# Package ‘BEQI2’

January 27, 2015

**Type** Package

**Title** Benthic Ecosystem Quality Index 2

**Description** Tool for analysing benthos data. It estimates several quality indices like the total abundance of species, species richness, Margalef's d, AZTI Marine Biotic Index (AMBI), and the BEQI-2 index. Furthermore, additional (optional) features are provided that enhance data preprocessing: (1) genus to species conversion, i.e., taxa counts at the taxonomic genus level can optionally be converted to the species level and (2) pooling: small samples are combined to bigger samples with a standardized size to (a) meet the data requirements of the AMBI, (b) generate comparable species richness values and (c) give a higher benthos signal to noise ratio.

**Version** 2.0-0

**Date** 2015-01-06

**Depends** R (>= 3.0.2), methods

**Imports** knitr, markdown, jsonlite, xtable, plyr, reshape2

**Suggests** testthat

**VignetteBuilder** knitr

**License** GPL (>= 3)

**Author** Willem van Loon [aut, cph],  
Dennis Walvoort [aut, cre]

**Maintainer** Dennis Walvoort <dennis.Walvoort@wur.nl>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-01-07 06:33:26

## R topics documented:

abundance . . . . .	2
beqi2 . . . . .	3
BEQI2-pkg . . . . .	3

BEQI2dir . . . . .	3
entropy . . . . .	4
eqr . . . . .	4
genusToSpecies . . . . .	5
harmonize . . . . .	6
isAzoic . . . . .	6
margalef . . . . .	7
pool . . . . .	7
readAMBI . . . . .	8
readBEQI . . . . .	8
readERF . . . . .	9
readFIBI . . . . .	10
readITI . . . . .	11
readSettings . . . . .	11
readTWN . . . . .	12
rename . . . . .	13
speciesRichness . . . . .	13
stripSpaces . . . . .	14
%inInterval% . . . . .	14
<b>Index</b>	<b>15</b>

---

abundance

*Abundance*

---

### Description

Compute abundance for each taxon.

### Usage

abundance(taxon, count)

### Arguments

taxon	character vector with taxa
count	integer vector with counts

### Value

integer vector with abundance per taxon.

---

beqi2	<i>Perform BEQI-2 Analysis</i>
-------	--------------------------------

---

**Description**

This function performs a complete BEQI-2 analysis following the settings provided in filename.  
Perform BEQI-2 Analysis

**Usage**

```
beqi2(filename = NULL, tmpdir = tempfile(pattern = "BEQI2"),
       browse = TRUE)
```

```
BEQI2(filename = NULL, tmpdir = tmpdir(), browse = TRUE)
```

**Arguments**

filename	name of the JSON file defining all analysis steps.
tmpdir	directory to store temporary files (for debugging only)
browse	load resulting report in a browser? TRUE or FALSE

---

BEQI2-pkg	<i>Marine Benthic Ecosystem Quality Index</i>
-----------	---

---

**Description**

Tools for analysing marine benthic ecosystems.

---

BEQI2dir	<i>Create BEQI-2 Directory Structure</i>
----------	--

---

**Description**

Creates a BEQI2-directory structure and populates it with some relevant BEQI2-files. Users may wish to modify this directory structure and add their own data.

**Usage**

```
BEQI2dir(path = NULL)
```

**Arguments**

path	name of an existing directory. This directory should be empty to prevent loss of data. If missing, a dialogue will appear.
------	--

---

entropy                      *Shannon's Entropy*

---

### Description

Compute entropy according to Shannon (1948)

### Usage

entropy(taxon, count)

### Arguments

taxon	character vector with taxa
count	integer vector with counts

### Value

Shannon's entropy

### References

Shannon, C. E., 1948. A Mathematical Theory of Communication. Bell System Technical Journal 27: 379-423.

---

eqr                              *Ecological Quality Ratio (EQR)*

---

### Description

The ecological quality ratio is the ratio between a parameter value and its reference value:

$$EQR = \frac{x - bad}{good - bad}$$

Depending on bad and good, the EQR usually (but not necessarily!) varies between 0 (bad ecological quality) and 1 (good ecological quality).

### Usage

eqr(x, bad, good)

### Arguments

x	numeric vector containing benthic indices
bad	the reference value for a bad status
good	the reference value for a good status

**Value**

numeric vector with EQR-values: low values indicate bad ecological quality and high values indicate good ecological quality.

---

genusToSpecies	<i>Genus to Species Conversion</i>
----------------	------------------------------------

---

**Description**

For each sample, the algorithm tries to convert taxa on the genus level to the species level. Counts at the genus level will be distributed over the species level proportional to the available number of species.

**Usage**

```
genusToSpecies(id, taxon, count)
```

**Arguments**

id	sample identifier
taxon	taxon name on either the genus or species level
count	total number of individuals of a specific taxon in a sample

**Value**

data.frame with three columns: id the sample identifier, taxon the taxon name, and count the count after genus to species conversion.

**Note**

The updated counts are not necessarily integers.

**Examples**

```
genusToSpecies(id = c(1, 1, 1),
               taxon = c("Genus1", "Genus1 s1", "Genus1 s2"), count = c(4, 2, 1))
```

---

`harmonize`*Harmonize Case*

---

**Description**

Convert text to the most occurring case. In case of ties, the first occurrence in sorted order will be taken.

**Usage**

```
harmonize(x)
```

**Arguments**

`x` character vector

**Value**

character vector with harmonized names (i.e., same case)

**Examples**

```
x <- c("F00", "Foo", "bar", "F00", "bar", "F00", "Bar")
y <- BEQI2:::harmonize(x)
stopifnot(all.equal(y, c("F00", "F00", "bar", "F00", "bar", "F00", "bar")))
```

---

`isAzoic`*Test for Azoic Samples*

---

**Description**

Case-insensitive test for taxa starting with 'azoi'

**Usage**

```
isAzoic(x)
```

**Arguments**

`x` character vector containing taxa

**Value**

logical vector, with elements TRUE for azoic samples, and FALSE otherwise.

---

margalef	<i>Margalef Index of Diversity</i>
----------	------------------------------------

---

**Description**

Margalef Index of Diversity is given by

$$D = \frac{S - 1}{\ln(N)}$$

**Usage**

```
margalef(taxon, count)
```

**Arguments**

taxon	character vector with taxa
count	integer vector with counts

**Details**

For  $N = 1$ , the index is set to 0.

**Value**

Margalef diversity index (numeric vector of length 1)

---

pool	<i>Pooling</i>
------	----------------

---

**Description**

This function randomly assigns samples to pools of approximately equal area

**Usage**

```
pool(sampleId = 1:length(area), area, targetArea, maxTry = 100L)
```

**Arguments**

sampleId	identifier
area	sample corresponding to sampleId
targetArea	vector of length 2 containing the lower and upper bound of the pooled area
maxTry	maximum number of unsuccessful pooling tries before the algorithm gives up.

**Value**

vector with identifiers (integers) indicating the pool to which each sample belongs (NA for samples that could not be pooled)

---

readAMBI	<i>Read AMBI Sensitivity Data</i>
----------	-----------------------------------

---

**Description**

This function reads and checks files with AMBI sensitivity data. If `filename = NULL` Borja's data will be read

**Usage**

```
readAMBI(filename = NULL)
```

**Arguments**

`filename` name of the AMBI sensitivity file (character)

**Details**

The function performs the following tasks:

- checks the existence of `filename`;
- checks availability of required columns (case insensitive);
- removes redundant spaces;
- removes duplicated records.

---

readBEQI	<i>Read BEQI input files</i>
----------	------------------------------

---

**Description**

This function reads and checks BEQI input files. The format has been specified in Van Loon (2013).

**Usage**

```
readBEQI(filename)
```

**Arguments**

`filename` name of BEQI input file (character)



## Details

The function performs the following tasks:

- checks the existence of filename;
- checks availability of required columns (case insensitive);
- make column names with aggregation data case-insensitive;
- removes redundant spaces;
- checks if DATE-field adheres to ISO 8601 (YYYY-mm-dd);
- constructs a unique identifier ID by concatenating columns OBJECTID and DATE;
- aggregate (by summation) VALUE-fields of records that only differ in VALUE-field value;
- checks that each ID has a unique AREA;
- checks azoic samples for VALUE=0;
- removes records with VALUE=0, not belonging to azoic samples;
- checks VALUE-field on missing values;
- checks if VALUE-field is an integer;

## References

Willem van Loon, 2013. BEQI2 INPUT FORMAT

---

readERF                      *Read Ecotopes References File*

---

## Description

This function reads and checks files with reference values

## Usage

```
readERF(filename, extra = NULL)
```

## Arguments

filename	name of the ecotopes reference file (character)
extra	additional user-defined indices to be checked (character, see details)

## Details

The function performs the following tasks:

- checks the existence of `filename`;
- checks availability of required columns (case insensitive);
- removes redundant spaces
- removes duplicated records

Argument `extra` is a character vector of additional benthic indices to be checked for. For example, if `extra = "ITI"`, then the ecotope reference file should also contain columns `ITIREF` and `ITIBAD`.

The format of the ecotopes reference file is documented in the BEQI2-package vignette.

## References

Van Loon, W, 2013. Loon2013-BEQI2-Specs-Ecotopes-27nov.doc

---

readFIBI

*Read Freshwater Inflow Biotic Index File*

---

## Description

Read Freshwater Inflow Biotic Index File

## Usage

```
readFIBI(filename)
```

## Arguments

`filename`            name of the FIBI file (character)

## Details

The function performs the following tasks:

- checks the existence of `filename`;
- checks availability of required columns (case insensitive);
- removes redundant spaces;
- removes duplicated records.
- checks if all FIBI classes are I, II, III, or IV

---

readITI	<i>Read Infaunal Trophic Index File</i>
---------	---

---

**Description**

This function reads and checks files containing Infaunal trophic index data (Gittenberger et al., 2011)

**Usage**

```
readITI(filename)
```

**Arguments**

filename            name of the ITI file (character)

**Details**

The function performs the following tasks:

- checks the existence of filename;
- checks availability of required columns (case insensitive);
- removes redundant spaces;
- removes duplicated records.
- checks if all ITI classes are I, II, III, or IV

---

readSettings	<i>Read BEQI Settings File</i>
--------------	--------------------------------

---

**Description**

This function reads BEQI settings files (JSON)

**Usage**

```
readSettings(filename)
```

**Arguments**

filename            name of BEQI input file (character)

## Details

The function performs the following tasks:

- checks the existence of filename;
- reads JSON file while ignoring C-style comments;
- checks availability of required keys in the JSON-file
- checks values in JSON-file

---

readTWN

*Read TWN Data*

---

## Description

This function reads files in the Taxa Waterbeheer Nederland (TWN) format.

## Usage

```
readTWN(filename)
```

## Arguments

filename            name of TWN listBEQI input file (character)

## Details

The function adds a new column `taxon`. Its contents depending on TWN-status:

- status = 10 taxonname
- status = 20 prefername
- status = 80 parentname

## Value

a `data.frame` with four columns: `taxonname`, `taxongroup`, `taxonlevel`, `taxon`

## References

[sofus.ecosys.nl/taxabase.htm](http://sofus.ecosys.nl/taxabase.htm)

[www.aquo.nl/faq/faq-twn](http://www.aquo.nl/faq/faq-twn)

---

rename	<i>Renaming Taxon Names</i>
--------	-----------------------------

---

**Description**

Convert taxon name *x* to taxon name *to* by looking it up in *from*. Look-up is case insensitive.

**Usage**

```
rename(x, from, to)
```

**Arguments**

<i>x</i>	character vector with names
<i>from</i>	character vector of old names
<i>to</i>	character vector of new names

**Value**

character vector of `length(x)` with converted names

---

speciesRichness	<i>Species Richness</i>
-----------------	-------------------------

---

**Description**

Species richness (*S*) is defined as the number of taxa (lowest identification level possible) per sampling unit (data pool or box core sample).

**Usage**

```
speciesRichness(taxon, count)
```

**Arguments**

<i>taxon</i>	character vector with taxa
<i>count</i>	integer vector with counts

**Value**

species richness (integer vector of length 1)

---

<code>stripSpaces</code>	<i>Remove Redundant Spaces</i>
--------------------------	--------------------------------

---

**Description**

This function removes redundant spaces from character vectors

**Usage**

```
stripSpaces(x)
```

**Arguments**

<code>x</code>	character vector
----------------	------------------

**Value**

character vector without trailing or multiple spaces

**Examples**

```
stopifnot(BEQI2:::stripSpaces(" Hello World ") == "Hello World")
```

---

<code>%inInterval%</code>	<i>Test if a Value is in an Interval</i>
---------------------------	--

---

**Description**

This function tests if values are part of a closed interval.

**Usage**

```
e1 %inInterval% e2
```

**Arguments**

<code>e1</code>	numeric value
<code>e2</code>	numeric interval

**Value**

TRUE if the interval includes the value, FALSE otherwise

**Examples**

```
## Not run: 3 %inInterval% c(1, 4)
```

# Index

## \*Topic **package**

BEQI2-pkg, [3](#)

%inInterval%, [14](#)

abundance, [2](#)

BEQI (BEQI2-pkg), [3](#)

BEQI2 (beqi2), [3](#)

beqi2, [3](#)

BEQI2-pkg, [3](#)

BEQI2-pkg-package (BEQI2-pkg), [3](#)

BEQI2dir, [3](#)

entropy, [4](#)

eqr, [4](#)

genusToSpecies, [5](#)

harmonize, [6](#)

isAzoic, [6](#)

margalef, [7](#)

pool, [7](#)

readAMBI, [8](#)

readBEQI, [8](#)

readERF, [9](#)

readFIBI, [10](#)

readITI, [11](#)

readSettings, [11](#)

readTWN, [12](#)

rename, [13](#)

speciesRichness, [13](#)

stripSpaces, [14](#)