Package 'BDAlgo'

October 30, 2023

Type Package

Title Bloom Detecting Algorithm

Version 0.1.0

Date 2023-10-30

Description The Bloom Detecting Algorithm enables the detection of blooms within a time series of species abundance and extracts 22 phenological variables. For details, see Karasiewicz et al. (2022) <doi:10.3390/jmse10020174>.

Depends inflection

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Stephane Karasiewicz [aut, cre]
(<https://orcid.org/0000-0003-1652-9921>)

Maintainer Stephane Karasiewicz <skaraz.science@gmail.com>

Repository CRAN

Date/Publication 2023-10-30 13:20:05 UTC

R topics documented:

Abundance   *Abundance data sample*

Description

The Abundance data are a sample of the entire data set used in Karasiewicz et al.(2022).

Usage

```
data(Abundance)
```

Format

The Abundance is a matrix covering two stations and three species.

Source

Karasiewicz S., and Lefebvre A. (2022). Environmental Impact on Harmful Species Pseudo-nitzschia spp. and Phaeocystis globosa Phenology and Niche. *JMSE* 10(2), 174. doi:10.3390/jmse10020174.

References

Karasiewicz S., and Lefebvre A. (2022). Environmental Impact on Harmful Species Pseudo-nitzschia spp. and Phaeocystis globosa Phenology and Niche. *JMSE* 10(2), 174. doi:10.3390/jmse10020174.

BDAlgo   *The Bloom Detecting Algorithm*

Description

The Bloom Detecting Algorithm enables the detection of blooms within a time series of species abundance and extracts 22 phenological variables. See details for more information.

Usage

```
BDAlgo(data, threshold=c(4,4), perc_of_peak=0.85, nbr_days=300, date_col= 2,
station_col=1, Sp = c("Sp1", "Sp2"), min_Log = 2, SP_label = c("Species1", "Species2"),
Log=TRUE, PDF=FALSE, saving_path=NULL)
```

Arguments

`data,`	a data frame table containing the species abundance time series with the station and date (dd/mm/yyyy).
`threshold`	a numeric vector, the abundance threshold from which a detected peak can be considered a bloom.
`perc_of_peak`	a numeric value between 0 and 1, that creates an upper lower threshold under which the lower point after the peak could be considered the end of the bloom. See the details for more information.
`nbr_days`	a numeric value corresponding to the maximum of days length of a bloom.
`date_col`	a numeric value corresponding to the column of the 'date' variable.
`station_col`	a numeric value corresponding to the column of the 'station' variable.
`Sp`	a character vector corresponding to the species column in the data to which you wish to apply the function.
`min_Log`	a numeric value that is used as a lower threshold for the abundance of the data. See details for more information.
`SP_label`	a character vector corresponding to the species label to use for the output graph.
`Log`	TRUE or FALSE if you want to log(x+1) transform the abundance data.
`PDF`	TRUE or FALSE if you wish to save each graph in a single PDF.
`saving_path`	a character vector used as a directory path to save the output graphs and rda files.

Details

The data format required is a simple table with samples as rows and species as columns, with date (dd/mm/yyyy) and station in character. The dates are converted into date format in the algorithm.

The Bloom Detecting Algorithm detects the bloom of a species within a time series of abundance according to three conditions. But first, the algorithm locate the high and low points of the curves. For each high point, the closest was considered to be a bloom if:

1. The high points were above the value of `threshold` parameter value, which by default is 4, corresponding to the log10 of 10,000 cells/L. The threshold of 10,000 cells/L was used here as the algorithm was created to fit phytoplankton species. See the reference for more information.

2. The low points before and after the high points were inferior to the `perc_of_peak` of 0.85 (85%) of the high point value. In this case, some humps can be merged, as blooms can sometimes be bimodal.

3. The merging of two humps would occur when the value of one of the lowest points did not fit the second condition. The merging of two humps cannot occur if the merging causes the increasing or decreasing phase of the bloom to be greater than `nbr_days`, by default 300 days.

These three conditions were necessary as they enabled the extraction of the phenological bloom, which in our case corresponded to HABs. The HAB case study helped us define the hump minimum abundance `threshold` as well as the amplitude `nbr_days` and shape `perc_of_peak`.

`Log` parameter simply transforms the abundance in log(x+1) as it helps with the large variation in the data value. The `min_Log` parameter (by default, 2 corresponding to 100 cells/L) was the minimum we fixed for our study.

The output graphs are time series of your species abundance (grey dots) for each station with the fitted smooth spline (grey line) and confidence interval (grey shaded area). The colored dots correspond to different timing phenological variables, which need to be in the following order for each bloom detected to confirm the validity of the algorithm fit. Yellow (DBS) > Turquoise (DMF) > Red (DMA) > Purple (DMM) > Pink (DBE)

Value

The BD_Algo function returns:

1. a graph per species and per station.

2. a list containing the species (Sp) and for each station the following data:

- smooth_spline: the results of the smooth spline. See smooth.spline for more information on the return values.

- conf_intervall: the data of the confidence intervals.

- all_date: the character vector with all the dates used in the smooth spline.

- all_bloom: the phenological data frame of each bloom with the timing variables (DBS, DMF, DMA, DMM, DBE) corrected.

- all_bloom_date: the raw phenological data frame of each bloom.

Warnings may occur during the smooth spline applications.

Author(s)

Stephane Karasiewicz, <skaraz.science@gmail.fr>

References

Karasiewicz S., and Lefebvre A. (2022). Environmental Impact on Harmful Species Pseudo-nitzschia spp. and Phaeocystis globosa Phenology and Niche. *JMSE* 10(2), 174. doi:10.3390/jmse10020174.

Examples

```
library(BDAlgo)
data(Abundance)
algo <- BDAlgo(Abundance, threshold=c(4,4), perc_of_peak=0.85, nbr_days=300, date_col= 2,
station_col=1, Sp = c("Pseunitz", "Phaeocy"), min_Log = 2,SP_label = c("Pseudo", "Phae"),
Log=TRUE,PDF=FALSE, saving_path= NULL)
```

Index