

Package ‘algaeClassify’

May 29, 2019

Title Determine phytoplankton functional groups based on functional traits

Version 0.1.4

Date 2018-10-11

Author Vijay Patil [aut,cre], Torsten Seltmann [aut], Nico Salmaso [aut], Orlane Anneville [aut], Marc Lajeunesse [aut], Dietmar Straile [aut]

Maintainer Vijay Patil <vpatil@usgs.gov>

URL http://github.com/vppatil/GEISHA_phytoplankton

BugReports https://github.com/vppatil/GEISHA_phytoplankton/issues

Description Functions that facilitate the use of accepted taxonomic nomenclature, collection of functional trait data, and assignment of functional group classifications to phytoplankton species. Possible classifications include Morpho-functional group (MFG; Salmaso et al. 2015 <doi>) and CSR (Reynolds 1988 <doi>) Classifications can be based on user-supplied trait data, a published trait database for 1222 freshwater European taxa, or a library of literature-derived MFG classifications for common freshwater species. The algaeClassify package is a product of the GEISHA (Global Evaluation of the Impacts of Storms on freshwater Habitat and Structure of phytoplankton Assemblages), funded by CESAB (Centre for Synthesis and Analysis of Biodiversity) and the USGS John Wesley Powell Center for Synthesis and Analysis, with data and other support provided by members of GLEON (Global Lake Ecology Observation Network). This software is preliminary or provisional and is subject to revision. It is being provided to meet the need for timely best science. The software has not received final approval by the U.S. Geological Survey (USGS). No warranty, expressed or implied, is made by the USGS or the U.S. Government as to the functionality of the software and related material nor shall the fact of release constitute any such warranty. The software is provided on the condition that neither the USGS nor the U.S. Government shall be held liable for any damages resulting from the authorized or unauthorized use of the software.

Depends R (>= 3.4.0)

Imports htr,
lubridate,
plyr,
RCurl,
rvest,
taxize,
XML,
xml2

License GPL-2 | GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1

R topics documented:

accum	2
algae_search	3
bestmatch	4
date_mat	4
genus_species_extract	5
lakegeneva	6
phyto_ts_aggregate	6
sampeff	7
species.mfg.library	8
species_to_mfg	8
species_to_mfg_df	9
spp_list_algaebase	10
traitranges	11
traits_to_csr	11
traits_to_mfg	12
traits_to_mfg_df	13
Index	14

accum	<i>Split a dataframe column with binomial name into genus and species columns. Plots change in species richness over time, generates species accumulation curve, and compares SAC against simulated idealized curve assuming all unique taxa have equal probability of being sampled at any point in the time series. (author Dietmar Straile)</i>
-------	--

Description

Split a dataframe column with binomial name into genus and species columns. Plots change in species richness over time, generates species accumulation curve, and compares SAC against simulated idealized curve assuming all unique taxa have equal probability of being sampled at any point in the time series. (author Dietmar Straile)

Usage

```
accum(b_data, phyto_name = "phyto_name", column, n = 50,  
      save.pdf = FALSE, lakename = "", datename = "date_dd_mm_yy",  
      dateformat = "%d-%m-%y")
```

Arguments

b_data	Name of data.frame object
phyto_name	Character string: field containing phytoplankton id (species, genus, etc.)
column	column name or number for field containing abundance (biomass,biovol, etc.) can be NA for presence absence
n	number of simulations for randomized ideal species accumulation curve
save.pdf	TRUE/FALSE- should plots be displayed or saved to a pdf?
lakename	optional character string for adding lake name to pdf output
datename	character string name of b_data field containing date
dateformat	character string: posix format for datename column

Value

a two panel plot with trends in richness on top, and cumulative richness vs. simulated accumulation curve on bottom

Examples

```
data(lakegeneva)
#example dataset with 50 rows

accum(b_data=lakegeneva,column=6,n=10,save.pdf=FALSE)
```

algae_search	<i>Compare a genus and species name against the algaebase online database</i>
--------------	---

Description

Compare a genus and species name against the algaebase online database

Usage

```
algae_search(genus, species = "", long = F)
```

Arguments

genus	Character string
species	Character string
long	if TRUE, returns higher taxonomy (Kingdom through Family)

Value

A data.frame with the original name, binary flags indicating an exact match and an currently accepted name, potential accepted synonyms, and the best accepted match as a single string and as separate genus and species strings. Higher taxonomy may also be appended if long=T

See Also

<http://www.algaebase.org> for phytoplankton taxonomy database, <https://powellcenter.usgs.gov/geisha> for project information

Examples

```
algae_search(genus='Anabaena', species='flos-aquae', long=TRUE)
```

bestmatch	<i>fuzzy partial matching between a scientific name and a list of possible matches</i>
-----------	--

Description

fuzzy partial matching between a scientific name and a list of possible matches

Usage

```
bestmatch(enteredName, possibleNames, maxErr = 3)
```

Arguments

enteredName	Character string with name to check
possibleNames	Character vector of possible matches
maxErr	maximum number of different bits allowed for a partial match

Value

a character string with the best match, or 'multiplePartialMatches' if no best match

Examples

```
possibleMatches=c('Viburnum edule','Viburnum acerifolia')
bestmatch(enteredName='Viburnum edulus',possibleNames=possibleMatches)
```

date_mat	<i>Transform a phytoplankton timeseries into a matrix of abundances for ordination</i>
----------	--

Description

Transform a phytoplankton timeseries into a matrix of abundances for ordination

Usage

```
date_mat(phyto.df, abundance.var = "biovol_um3_ml",
  taxa.name = "phyto_name", date.name = "date_dd_mm_yy",
  format = "%d-%m-%y", time.agg = c("day", "month", "year",
  "monthyear"), fun = function(x) mean(x[!base::is.na(x)]))
```

Arguments

phyto.df	Name of data.frame object
abundance.var	Character string: field containing abundance data. NA for presence/absence
taxa.name	Character string: field containing taxonomic identifiers
date.name	Character string: field containing date.
format	Character string: POSIX format string for formatting date column
time.agg	Character string: time interval for aggregating abundance. default is day.
fun	function for aggregation. default is mean, excluding NA's

Value

A matrix of phytoplankton abundance, with taxa in rows and time in columns. If time.agg = 'mon-thyear', returns a 3dimensional matrix (taxa,month,year). If abundance.var = NA, matrix cells will be 1 for present, 0 for absent

Examples

```
data(lakegeneva)
#example dataset with 50 rows

geneva.mat<-date_mat(lakegeneva,abundance.var=NA)

geneva.mat
```

`genus_species_extract` *Split a dataframe column with binomial name into genus and species columns.*

Description

Split a dataframe column with binomial name into genus and species columns.

Usage

```
genus_species_extract(phyto.df, phyto.name)
```

Arguments

phyto.df	Name of data.frame object
phyto.name	Character string: field in phyto.df containing species name.

Value

A data.frame with new character fields 'genus' and 'species'

Examples

```
data(lakegeneva)
#example dataset with 50 rows

head(lakegeneva) #need to split the phyto_name column
new.lakegeneva=genus_species_extract(lakegeneva,'phyto_name')

head(new.lakegeneva)
```

lakegeneva	<i>example dataset from lake Geneva, Switzerland</i>
------------	--

Description

example dataset from lake Geneva, Switzerland

Usage

```
data(lakegeneva)
```

Format

A data frame with columns:

lake lake name
phyto_name phytoplankton species name
month month of sampling
year year of sampling
date_dd_mm_yy date of sampling
biovol_um3_ml biovolume

phyto_ts_aggregate	<i>Aggregate phytoplankton timeseries based on abundance. Up to 3 grouping variables can be given: e.g. genus, species, stationid, depth range. If no abundance var is given, will aggregate to presence/absence of grouping vars.</i>
--------------------	--

Description

Aggregate phytoplankton timeseries based on abundance. Up to 3 grouping variables can be given: e.g. genus, species, stationid, depth range. If no abundance var is given, will aggregate to presence/absence of grouping vars.

Usage

```
phyto_ts_aggregate(phyto.data, DateVar = "date_dd_mm_yy",
  AbundanceVar = "biovol_um3_ml", GroupingVar1 = "phyto_name",
  GroupingVar2 = NA, GroupingVar3 = NA, remove.rare = F, fun = sum,
  format = "%d-%m-%y")
```

Arguments

phyto.data	data.frame
DateVar	character string: field name for date variable. character or POSIX data.
AbundanceVar	character string with field name containing abundance data
GroupingVar1	character string: field name for first grouping variable. defaults to spp.
GroupingVar2	character string: name of additional grouping var field
GroupingVar3	character string: name of additional grouping var field
remove.rare	TRUE/FALSE. If TRUE, removes all instances of GroupingVar1 that occur < 5 of time periods.
fun	function used to aggregate abundance based on grouping variables
format	character string: format for DateVar POSIXct conversion

Value

a data.frame with grouping vars, date_dd_mm_yy, and abundance or presence/absence

See Also

<http://www.algaebase.org> for up-to-date phytoplankton taxonomy, <https://powellcenter.usgs.gov/geisha> for project information

Examples

```
data(lakegeneva)
lakegeneva<-genus_species_extract(lakegeneva,'phyto_name')
lakegeneva.common.genera=phyto_ts_aggregate(lakegeneva,AbundanceVar=NA,GroupingVar1='genus')
head(lakegeneva.common.genera)
```

sampeff	<i>Visually assess change in sampling effort over time (author: Dietmar Straile)</i>
---------	--

Description

Visually assess change in sampling effort over time (author: Dietmar Straile)

Usage

```
sampeff(b_data, column, save.pdf = F, lakename = "",
        datecolumn = "date_dd_mm_yy", dateformat = "%d-%m-%y")
```

Arguments

b_data	Name of data.frame object
column	column name or number for field containing abundance (biomass,biovol, etc.) can be NA for presence absence
save.pdf	TRUE/FALSE Should the output plot be saved to a file? defaults to FALSE
lakename	Character string for labelling output plot
datecolumn	Character String or number specifying dataframe field with date information
dateformat	Character string specifying POSIX data format

Value

a time-series plot of minimum relative abundance over time. This should change systematically with counting effort.

Examples

```
data(lakegeneva)
#example dataset with 50 rows

sampeff(lakegeneva,column=6) #column 6 contains biovolume
```

species.mfg.library	<i>Trait-based MFG classifications for common Eurasion/North American phytoplankton species. See accompanying manuscript for sources</i>
---------------------	--

Description

Trait-based MFG classifications for common Eurasion/North American phytoplankton species. See accompanying manuscript for sources

Usage

```
data(species.mfg.library)
```

Format

A data frame with columns:

genus genus name

species species name

MFG corresponding MFG classification based on Salmaso et al. 2015

MFG2 corresponding MFG classification based on Salmaso et al. 2015

Habitat corresponding MFG classification based on Salmaso et al. 2015

Note corresponding MFG classification based on Salmaso et al. 2015

species_to_mfg	<i>Conversion of a single genus and species name to a single MFG. Uses species.mfg.library</i>
----------------	--

Description

Conversion of a single genus and species name to a single MFG. Uses species.mfg.library

Usage

```
species_to_mfg(genus, species, flag = 1)
```


Arguments

genus	Character string: genus name
species	Character string: species name
flag	Resolve ambiguous mfg: 1 = return(NA), 2 = manual selection

Value

a single MFG classification as character string

Examples

```
species_to_mfg('Scenedesmus', 'bijuga')
#returns "11a-NakeChlor"
```

species_to_mfg_df	<i>Wrapper function to apply species_phyto_convert() across a data.frame</i>
-------------------	--

Description

Wrapper function to apply species_phyto_convert() across a data.frame

Usage

```
species_to_mfg_df(phyto.df, flag = 1)
```

Arguments

phyto.df	Name of data.frame. Must have character fields named 'genus' and 'species'
flag	Resolve ambiguous MFG: 1 = return(NA), 2 = manual selection

Value

input data.frame with a new character column of MFG classifications

Examples

```
data(lakegeneva)
#example dataset with 50 rows

new.lakegeneva <- genus_species_extract(lakegeneva, 'phyto_name')
new.lakegeneva <- species_to_mfg_df(new.lakegeneva)
head(new.lakegeneva)
```

spp_list_algaebase	<i>Wrapper for applying algae_search function to a data.frame that contains phytoplankton species</i>
--------------------	---

Description

Wrapper for applying algae_search function to a data.frame that contains phytoplankton species

Usage

```
spp_list_algaebase(phyto.df, lakename = "", phyto.name = "phyto_name",
  long = F, write = T)
```

Arguments

phyto.df	data.frame containing genus and species columns or a column with binomial names
lakename	Character string with waterbody name for naming output files
phyto.name	Name or number of column that contains binomial names
long	TRUE/FALSE: should higher taxonomy (Kingdom:Family) be included in output?
write	TRUE/FALSE: should output be written as .csv file?

Value

A character string of the species' morphofunctional group

See Also

<http://www.algaebase.org> for up-to-date phytoplankton taxonomy, <https://powellcenter.usgs.gov/geisha> for project information

Examples

```
data(lakegeneva)
lakegeneva=lakegeneva[1:3,] ##use 3 rows for testing
lakegeneva<-genus_species_extract(lakegeneva,phyto.name='phyto_name')
lakegeneva.algaebase<-spp_list_algaebase(lakegeneva,write=FALSE)
```

traitranges	<i>surface/volume ratio and max linear dimension criteria for CSR From Reynolds 1988 and Reynolds 2006</i>
-------------	--

Description

surface/volume ratio and max linear dimension criteria for CSR From Reynolds 1988 and Reynolds 2006

Usage

```
data(traitranges)
```

Format

A data frame with columns:

Measurement measurement type

C.min minimum value for C

S.min minimum value for S

R.min minimum value for R

C.max maximum value for C

S.max maximum value for S

R.max maximum value for R

units units of measurement

source source for criteria

traits_to_csr	<i>Assign phytoplankton species to CSR functional groups, based on surface to volume ratio and maximum linear dimension ranges proposed by Reynolds et al. 1988;2006</i>
---------------	--

Description

Assign phytoplankton species to CSR functional groups, based on surface to volume ratio and maximum linear dimension ranges proposed by Reynolds et al. 1988;2006

Usage

```
traits_to_csr(sav, msv)
```

Arguments

sav numeric estimate of cell or colony surface/volume ratio in um^2/um^3

msv numeric product of surface/volume ratio multiplied by maximum linear dimension (msv; um)

Value

a character string with one of 3 return values: C,S, or R

See Also

[/urlhttps://powellcenter.usgs.gov/geisha](https://powellcenter.usgs.gov/geisha) for project information

Examples

```
traits_to_csr(sav=0.2,msv=10)
```

traits_to_mfg	<i>Assign MFG based on binary functional traits and taxonomy (Class and Order)</i>
---------------	--

Description

Assign MFG based on binary functional traits and taxonomy (Class and Order)

Usage

```
traits_to_mfg(flagella = NA, size = NA, colonial = NA,
  filament = NA, centric = NA, gelatinous = NA, aerotopes = NA,
  class = NA, order = NA)
```

Arguments

flagella	1 if flagella are present, 0 if they are absent.
size	Character string: 'large' or 'small'. Classification criteria is left to the user.
colonial	1 if typically colonial growth form, 0 if typically unicellular.
filament	1 if dominant growth form is filamentous, 0 if not.
centric	1 if diatom with centric growth form, 0 if not. NA for non-diatoms.
gelatinous	1 mucilagenous sheath is typically present, 0 if not.
aerotopes	1 if aerotopes allowing buoyancy regulation are typically present, 0 if not.
class	Character string: The taxonomic class of the species
order	Character string: The taxonomic order of the species

Value

A character string of the species' morphofunctional group

See Also

<http://www.algaebase.org> for up-to-date phytoplankton taxonomy, <https://powellcenter.usgs.gov/geisha> for project information

Examples

```
traits_to_mfg(flagella = 1, size = "large", colonial = 1, filament = 0, centric = NA, gelatinous = 0,
aerotopes = 0, class = "Euglenophyceae", order = "Euglenales")
```

traits_to_mfg_df	<i>Assign morphofunctional groups to a dataframe of functional traits and higher taxonomy</i>
------------------	---

Description

Assign morphofunctional groups to a dataframe of functional traits and higher taxonomy

Usage

```
traits_to_mfg_df(dframe, arg.names = c("flagella", "size", "colonial",
"filament", "centric", "gelatinous", "aerotopes", "class", "order"))
```

Arguments

dframe	An R dataframe containing functional trait information and higher taxonomy
arg.names	Character string of column names corresponding to arguments for traits_to_mfg()

Value

A character vector containing morpho-functional group (MFG) designations

Examples

```
#create a two-row example dataframe of functional traits
func.dframe=data.frame(flagella=1,size=c("large","small"),colonial=0,filament=0,centric=NA,
                        gelatinous=0,aerotopes=0,class="Euglenophyceae",order="Euglenales",
                        stringsAsFactors=FALSE)

#check the dataframe
print(func.dframe)

#run the function to produce a two-element character vector
traits_to_mfg_df(func.dframe,c("flagella","size","colonial","filament","centric","gelatinous",
"aerotopes","class","order"))
```

Index

*Topic **datasets**

- lakegeneva, [6](#)
- species.mfg.library, [8](#)
- traitranges, [11](#)

accum, [2](#)

algae_search, [3](#)

bestmatch, [4](#)

date_mat, [4](#)

genus_species_extract, [5](#)

lakegeneva, [6](#)

phyto_ts_aggregate, [6](#)

sampeff, [7](#)

species.mfg.library, [8](#)

species_to_mfg, [8](#)

species_to_mfg_df, [9](#)

spp_list_algaebase, [10](#)

traitranges, [11](#)

traits_to_csr, [11](#)

traits_to_mfg, [12](#)

traits_to_mfg_df, [13](#)