Package 'algaeClassify'

September 12, 2019

Title Determine Phytoplankton Functional Groups Based on Functional Traits

Version 1.2.0

Date 2019-9-04

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>

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:10.1111/fwb.12520>) and CSR (Reynolds 1988; Functional morphology and the
adaptive strategies of phytoplankton. In C.D. Sandgren (ed). Growth and reproductive
strategies of freshwater phytoplankton, 388-433. Cambridge University Press, New York).
Version 1.2.0 also includes functions to query names using the algaebase online taxonomic
database (www.algaebase.org; <doi:10.7872/crya.v35.iss2.2014.105>).
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 httr,
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	*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)*

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 = NA, n = 100,
  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 if the dataset only contains a species list for each sampling date.
`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
head(lakegeneva)

accum(b_data=lakegeneva,column='biovol_um3_ml',n=10,save.pdf=FALSE)
```

---

algae_search	*Compare a genus and species name against the algaebase online database*

Description

Compare a genus and species name against the algaebase online database

Usage

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

Arguments

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

Value

A data.frame with the following fields: orig.name (submitted name), match.name (best match), genus and species (from the match.name), exact.match (1 indicates whether there was a perfect match for orig.name. 0 otherwise), accepted (1 if the orig.name currently accepted, 0 otherwise), synonyms (currently accepted synonyms, if any. For genus-only search, returns genera associated with species formerly classified with the orig.name genus) Empire,Kingdom,Phylum,Class,Order,Family: character strings with current higher taxonomy associated with match.name.

See Also

http://www.algaebase.org for phytoplankton taxonomy database, https://powellcenter.usgs.gov/geisha for project information. Algaebase should be cited separately in any publications using this function: <doi:10.7872/crya.v35.iss2.2014.105>

Examples

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

---

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

Description

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

Usage

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

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
trunc	TRUE/FALSE. if true and no match, retry with last three letters truncated

Value

a character string with the best match, or 'multiplePartialMatches'

Examples

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

---

date_mat	*Transform a phytoplankton timeseries into a matrix of abundances for ordination*

Description

Transform a phytoplankton timeseries into a matrix of abundances for ordination

Usage

```
date_mat(phyto.df, abundance.var = "biovol_um3_ml",
  summary.type = "abundance", taxa.name = "phyto_name",
  date.name = "date_dd_mm_yy", format = "%d-%m-%y",
  time.agg = c("day", "month", "year", "monthyear"), fun = mean_naomit)
```

Arguments

phyto.df	Name of data.frame object
abundance.var	Character string: field containing abundance data. Can be NA if the dataset only contains a species list for each sampling date.
summary.type	'abundance' for a matrix of aggregated abundance,'presence.absence' for 1 (present) and 0 (absent).
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 = 'monthyear', 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.mat1<-date_mat(lakegeneva,time.agg='month',summary.type='presence.absence')
geneva.mat2<-date_mat(lakegeneva,time.agg='month',summary.type='abundance')

geneva.mat1
geneva.mat2
```

---

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   *example dataset from lake Geneva, Switzerland*

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

---

mean_naomit *Compute mean value while ignoring NA's*

Description

Compute mean value while ignoring NA's

Usage

```
mean_naomit(x)
```

Arguments

x   A numeric vector that may contain NA's

Value

the mean value

Examples

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

mean_naomit(lakegeneva$biovol_um3_ml)
```

---

phyto_ts_aggregate *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.*

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",
  SummaryType = c("abundance", "presence.absence"),
  AbundanceVar = "biovol_um3_ml", GroupingVar1 = "phyto_name",
  GroupingVar2 = NA, GroupingVar3 = NA, remove.rare = FALSE,
  fun = sum, format = "%d-%m-%y")
```

Arguments

`phyto.data`	data.frame
`DateVar`	character string: field name for date variable. character or POSIX data.
`SummaryType`	'abundance' for a matrix of aggregated abundance,'presence.absence' for 1 (present) and 0 (absent).
`AbundanceVar`	character string with field name containing abundance data Can be NA if data is only a species list and aggregated presence/absence is desired.
`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

Examples

```
data(lakegeneva)
lakegeneva<-genus_species_extract(lakegeneva,'phyto_name')
lg.genera=phyto_ts_aggregate(lakegeneva,SummaryType='presence.absence',
                             GroupingVar1='genus')
head(lg.genera)
```

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

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 labeling 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
```

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

Description

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

Usage

```
spp_list_algaebase(phyto.df, phyto.name = 1, lakename = "",
  long = FALSE, write = FALSE)
```

Arguments

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

Value

A data.frame with the following fields: orig.name (submitted name), match.name (best match), genus and species (from the match.name), exact.match (1 indicates whether there was a perfect match for orig.name. 0 otherwise), accepted (1 if the orig.name currently accepted, 0 otherwise), synonyms (currently accepted synonyms, if any. For genus-only search, returns genera associated with species formerly classified with the orig.name genus) Empire,Kingdom,Phylum,Class,Order,Family: character strings with current higher taxonomy associated with match.name

See Also

http://www.algaebase.org for up-to-date phytoplankton taxonomy, https://powellcenter.usgs.gov/geisha for project information. Algaebase should be cited separately in any publications using this function: <doi:10.7872/crya.v35.iss2.2014.105>

Examples

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

---

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

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   *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*

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, msv.source = "Reynolds 2006",
  traitrange = traitranges)
```

Arguments

sav	numeric estimate of cell or colony surface area /volume ratio
msv	numeric product of surface area/volume ratio and maximum linear dimension
msv.source	character string with reference source for distinguishing criteria
traitrange	data frame with trait criteria for c,s,r groups. The included table can be replaced with user-defined criteria if desired. Measurements are: Surface area/volume ratio (sav), maximum linear dimension (mld) and mld*sav (msv).

Value

a character string with one of 5 return values: C,CR,S,R, or SR. CR and SR groups reflect overlap between criteria for the 3 main groups.

See Also

/urlhttps://powellcenter.usgs.gov/geisha for project information

Examples

```
traits_to_csr(sav=0.2,msv=10,msv.source='Reynolds 2006',traitrange=traitranges)
```

---

traits_to_csr_df	*Add CSR functional group classifications to a dataframe of phytoplankton species, based on surface to volume ratio and maximum linear dimension ranges proposed by Reynolds et al. 1988;2006*

Description

Add CSR functional group classifications to a dataframe of phytoplankton species, based on surface to volume ratio and maximum linear dimension ranges proposed by Reynolds et al. 1988;2006

Usage

```
traits_to_csr_df(df, sav, msv, msv.source = "Reynolds 2006",
  traitrange = traitranges)
```

Arguments

df	name of dataframe
sav	character string with name of column that contains surface to volume ratio values
msv	character string with name of column that contains maximum linear dimension * surface to volume ratio values
msv.source	character string with reference source for distinguishing criteria
traitrange	data frame with trait criteria for c,s,r groups. The included table can be replaced with user-defined criteria if desired. Measurements are: Surface area/volume ratio (sav), maximum linear dimension (mld) and mld*sav (msv).

Value

a character string with one of 5 return values: C,CR,S,SR, or R

Examples

```
csr.df<-data.frame(msv=10,sav=1)

csr.df$CSR<-traits_to_csr_df(csr.df,'msv','sav')

print(csr.df)
```

---

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

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

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

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
func.dframe$MFG<-traits_to_mfg_df(func.dframe,c("flagella","size","colonial",
                              "filament","centric","gelatinous",
                              "aerotopes","class","order"))

print(func.dframe)
```

Index