

# Package

October 12, 2018

**Title** Determine phytoplankton functional groups based on functional traits

**Version** 0.1.1

**Date** 2018-10-11

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

**Maintainer** Vijay Patil <vpatil@usgs.gov>

**URL** [http://github.com/vppatil/GEISHA\\_phytoplankton](http://github.com/vppatil/GEISHA_phytoplankton)

**BugReports** [https://github.com/vppatil/GEISHA\\_phytoplankton/issues](https://github.com/vppatil/GEISHA_phytoplankton/issues)

**Description** The algaeClassify package contains functions designed to facilitate the assignment of morpho-functional group (MFG) classifications to phytoplankton species based on a combination of taxonomy (Class, Order) and a suite of 7 binomial functional traits. Classifications can also be made using only a species list and a database of trait-derived classifications included in the package. MFG classifications are derived from Salmaso, Nico, Luigi Naselli-Flores, and Judit Padisak. "Functional classifications and their application in phytoplankton ecology." Freshwater Biology 60.4 (2015): 603-619, and this reference should be cited when using the package. 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 (the Centre for Synthesis and Analysis of Biodiversity) and the USGS John Wesley Powell Center, with data and other support provided by members of GLEON (the 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)

**License** GPL (>=3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

R topics documented:

genus_species_extract . . . . .	2
lakegeneva . . . . .	3
mfg.csr . . . . .	3
phyto_convert_df . . . . .	4
species.mfg.library . . . . .	4
species_phyto_convert . . . . .	5
traits_to_mfg . . . . .	5
traits_to_mfg_df . . . . .	6
traits_to_mfg_nosize . . . . .	7
<b>Index</b>	<b>8</b>

---

genus_species_extract	<i>Split a dataframe column with binomial name into genus and species columns.</i>
-----------------------	--

---

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 containing binomial 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 (50 lines)</i>
------------	---

---

**Description**

example dataset from lake Geneva, Switzerland (50 lines)

**Usage**

```
data(lakegeneva)
```

**Format**

A data frame with columns:

**lake** lake name

**date\_dd\_mm\_yy** date dd-mm-yy format

**phyto\_name** phytoplankton species binomial name

---

mfg.csr	<i>MFG-CSR correspondence based on CSR-trait relationships in Reynolds et al. 1988 and MFG-trait relationships in Salmaso et al. 2015</i>
---------	---

---

**Description**

MFG-CSR correspondence based on CSR-trait relationships in Reynolds et al. 1988 and MFG-trait relationships in Salmaso et al. 2015

**Usage**

```
data(mfg.csr)
```

**Format**

A data frame with columns:

**MFG.number** shortened MFG designation

**MFG** full MFG name from Salmaso et al. 2015

**CSR** CSR classification including intermediate classes

---

phyto_convert_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

```
phyto_convert_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

a single MFG classification as character string

### Examples

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

new.lakegeneva=genus_species_extract(lakegeneva, 'phyto_name')
new.lakegeneva=phyto_convert_df(new.lakegeneva)
head(new.lakegeneva)
```

---

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

---

species\_phyto\_convert    *Conversion of a single genus and species name to a single MFG*

---

### Description

Conversion of a single genus and species name to a single MFG

### Usage

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

---

traits_to_mfg	<i>Assign a morphofunctional group based on binary functional traits and higher taxonomy</i>
---------------	--

---

### Description

Assign a morphofunctional group based on binary functional traits and higher taxonomy

### 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(1,"large",1,0,NA,0,0,"Euglenophyceae","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(flag=1,size=c("large","small"),col=0,fil=0,cent=NA,gel=0,
  aer=0,cl="Euglenophyceae",or="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("flag","size","col","fil","cent","gel","aer","cl","or"))
```

---

traits_to_mfg_nosize	<i>Assign a morphofunctional group based on binary functional traits and higher taxonomy</i>
----------------------	--

---

**Description**

Assign a morphofunctional group based on binary functional traits and higher taxonomy

**Usage**

```
traits_to_mfg_nosize(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(1,"large",1,0,NA,0,0,"Euglenophyceae","Euglenales")
```

# Index

## \*Topic **datasets**

lakegeneva, [3](#)

mfg.csr, [3](#)

species.mfg.library, [4](#)

genus\_species\_extract, [2](#)

lakegeneva, [3](#)

mfg.csr, [3](#)

phyto\_convert\_df, [4](#)

species.mfg.library, [4](#)

species\_phyto\_convert, [5](#)

traits\_to\_mfg, [5](#)

traits\_to\_mfg\_df, [6](#)

traits\_to\_mfg\_nosize, [7](#)