Package ‘phytoclass’

September 29, 2023

Title Estimate Chla Biomass of Phytoplankton Groups

Version 1.0.0

Description Determine the chlorophyll a (Chl a) biomass of different phytoplankton groups based on their pigment biomarkers. The method uses non-negative matrix factorisation and simulated annealing to minimise error between the observed and estimated values of pigment concentrations (Hayward et al. (2023) <doi:10.1002/lom3.10541>).
The approach is similar to the widely used 'CHEMTAX' program (Mackey et al. 1996) <doi:10.3354/meps144265>, but is more straightforward, accurate, and not reliant on initial guesses for the pigment to Chl a ratios for each phytoplankton group.

Imports bestNormalize, dplyr, dynamicTreeCut, ggplot2, Metrics, RcppML, stats, tidyr

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 4.2)

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Alexander Hayward [aut, cre, cph],
Andy McKenzie [aut]

Maintainer Alexander Hayward <phytoclass@outlook.com>

Repository CRAN

Date/Publication 2023-09-29 12:50:02 UTC

R topics documented:

`Bounded_weights` *Add weights to the data, bound at a maximum.*

Description

Add weights to the data, bound at a maximum.

Usage

```
Bounded_weights(S, weight.upper.bound = 30)
```

Arguments

`S`	Sample data matrix – a matrix of pigment samples
`weight.upper.bound`	Upper bound for weights (default is 30)

Value

A vector with upper bounds for weights

Examples

```
Bounded_weights(Sm, weight.upper.bound = 30)
```

`Cluster` *Cluster things*

Description

Cluster things

Usage

```
Cluster(Data, min_cluster_size)
```

Arguments

`Data`	S (sample) matrix
`min_cluster_size`	the minimum size required for a cluster

Value

A named list of length two. The first element "cluster.list" is a list of clusters, and the second element "cluster.plot" the cluster analysis object (dendogram) that can be plotted.

Examples

```
Cluster.result <- Cluster(Sm, 14)
Cluster.result$cluster.list
plot(Cluster.result$cluster.plot)
```

`Fm` *Fm data*

Description

Fm data

Usage

```
Fm
```

Format

`Fm`**:**

A data frame with 9 rows and 15 columns:

chl.c1 XX

Per XX

X19but XX ...

Source

XX

Matrix_checks *Remove any column values that average 0. Further to this, also remove phytoplankton groups from the F matrix if their diagnostic pigment isn't present.*

Description

Remove any column values that average 0. Further to this, also remove phytoplankton groups from the F matrix if their diagnostic pigment isn't present.

Usage

```
Matrix_checks(S, Fmat)
```

Arguments

`S`	Sample data matrix – a matrix of pigment samples
`Fmat`	Pigment to Chl a matrix

Value

Named list with new S and Fmat matrices

Examples

```
MC <- Matrix_checks(Sm, Fm)
Snew <- MC$Snew
```

min_max *min_max data*

Description

min_max data

Usage

```
min_max
```

Format

`min_max`**:**

A data frame with 76 rows and 4 columns:

class XX

Pig_Abbrev XX

min XX

max max ...

Source

XX

NNLS_MF

Performs the non-negative matrix factorisation for given phytoplankton pigments and pigment ratios, to attain an estimate of phytoplankton class abundances.

Description

Performs the non-negative matrix factorisation for given phytoplankton pigments and pigment ratios, to attain an estimate of phytoplankton class abundances.

Usage

```
NNLS_MF(Fn, S, cm = NULL)
```

Arguments

`Fn`	Pigment to Chl a matrix
`S`	Sample data matrix – a matrix of pigment samples
`cm`	Weights for each column

Value

A list containing

1. The F matrix (pigment: Chl *a*) ratios
2. The root mean square error (RMSE)
3. The C matrix (class abundances for each group)

Examples

```
MC <- Matrix_checks(Sm,Fm)
Snew <- MC$Snew
Fnew <- MC$Fnew
cm <- Bounded_weights(Snew, weight.upper.bound = 30)
NNLS_MF(Fnew, Snew, cm)
```

simulated_annealing *Perform simulated annealing algorithm for S and F matrices*

Description

Perform simulated annealing algorithm for S and F matrices

Usage

```
simulated_annealing(
  S,
  Fmat = NULL,
  user_defined_min_max = NULL,
  do_matrix_checks = TRUE,
  niter = 500,
  step = 0.009,
  weight.upper.bound = 30,
  verbose = TRUE
)
```

Arguments

`S` Sample data matrix – a matrix of pigment samples

`Fmat` Pigment to Chl a matrix

`user_defined_min_max`
data frame with some format as min_max built-in data

`do_matrix_checks`
This should only be set to TRUE when using the default values. This will remove pigment columns that have column sums of 0. Set to FALSE if using customised names for pigments and phytoplankton groups

`niter` Number of iterations (default is 500)

`step` Step ratio used (default is 0.009)

`weight.upper.bound`
Upper limit of the weights applied (default value is 30).

`verbose` Logical value. Output error and temperature at each iteration. Default value of TRUE

Value

A list containing

1. Fmat matrix
2. RMSE (Root Mean Square Error)
3. condition number
4. Class abundances

5. Figure (plot of results)
6. MAE (Mean Absolute Error)
7. Error

Examples

```
# Using the built-in matrices Sm and Fm
set.seed(5326)
sa.example <- simulated_annealing(Sm, Fm, niter = 5)
sa.example$Figure
```

`Sm` *Sm data*

Description

Sm data

Usage

```
Sm
```

Format

`Sm`**:**

A data frame with 29 rows and 15 columns:

chl.c1 XX

Per XX

X19but XX ...

Source

XX

Steepest_Desc *Stand-alone version of steepest descent algorithm*

Description

Stand-alone version of steepest descent algorithm

Usage

```
Steepest_Desc(Fmat, S, num.loops)
```

Arguments

Fmat	Pigment to Chl a matrix
S	Sample data matrix – a matrix of pigment samples
num.loops	Number of loops/iterations to perform (no default)

Value

A list containing

1. The F matrix (pigment: Chl *a*) ratios
2. RMSE (Root Mean Square Error)
3. Condition number
4. class abundances
5. Figure (plot of results)
6. MAE (Mean Absolute Error)

Examples

```
MC <- Matrix_checks(Sm,Fm)
Snew <- MC$Snew
Fnew <- MC$Fnew
SDRes <- Steepest_Desc(Fnew,Snew, num.loops = 20)
```

Index