Package ‘phytoclass’

May 11, 2024

Title Estimate Chla Concentrations of Phytoplankton Groups

Version 1.2.0

Description Determine the chlorophyll a (Chl a) concentrations 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 phytoplankton groups.

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

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 3.8)

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://github.com/AndyMcKenzieFromNZ/phytoclass

BugReports https://github.com/AndyMcKenzieFromNZ/phytoclass/issues

NeedsCompilation no

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

Maintainer Alexander Hayward <phytoclass@outlook.com>

Repository CRAN

Date/Publication 2024-05-11 09:30:02 UTC
Bounded_weights

**R topics documented:**

- Bounded_weights .................................................. 2
- Cluster ..................................................................... 3
- Fm ......................................................................... 3
- Fp .......................................................................... 4
- Matrix_checks .......................................................... 4
- min_max .................................................................... 5
- NNLS_MF .................................................................... 5
- simulated_annealing ................................................. 6
- Sm ............................................................................ 7
- Sp ............................................................................ 8
- Steepest_Desc .......................................................... 9

**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 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 data

Description
Fm data

Usage
Fm

Format
Fm:
A data frame with 9 rows and 15 columns:
chl_c1 XX
Per XX
X19but XX ...

Matrix_checks

Source
XX

Fp
Fp data

Description
Fp data

Usage
Fp

Format
Fp:
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

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

---

**min_max**

---

**Description**

*min_max data*

**Usage**

*min_max*

**Format**

```r
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**

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

This is the main phytoclass algorithm. It performs simulated annealing algorithm for S and F matrices. See the examples (Fm, Sm) for how to set up matrices, and the vignette for more detailed instructions. Different pigments and phytoplankton groups may be used.

**Description**

This is the main phytoclass algorithm. It performs simulated annealing algorithm for S and F matrices. See the examples (Fm, Sm) for how to set up matrices, and the vignette for more detailed instructions. Different pigments and phytoplankton groups may be used.

**Usage**

```r
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

Sp:
A data frame with 29 rows and 15 columns:

chl_c1  XX
Per   XX
X19but XX ...

Source

XX

Description

Sp data

Usage

Sp

Format

Sp:
A data frame with 29 rows and 15 columns:

chl_c1  XX
Per   XX
X19but XX ...

Source

XX
Stand-alone version of steepest descent algorithm. This is similar to the CHEMTAX steepest descent algorithm. It is not required to use this function, and as results are not bound by minimum and maximum, results may be unrealistic.

Usage

Steepest_Desc(Fmat, S, num.loops)

Arguments

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fmat</td>
<td>Pigment to Chl a matrix</td>
</tr>
<tr>
<td>S</td>
<td>Sample data matrix – a matrix of pigment samples</td>
</tr>
<tr>
<td>num.loops</td>
<td>Number of loops/iterations to perform (no default)</td>
</tr>
</tbody>
</table>

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

* datasets
  Fm, 3
  Fp, 4
  min_max, 5
  Sm, 7
  Sp, 8

Bounded_weights, 2

Cluster, 3

Fm, 3
Fp, 4

Matrix_checks, 4
min_max, 5

NNLS_MF, 5

simulated_annealing, 6
Sm, 7
Sp, 8
Steepest_Desc, 9