Package 'phytoclass'

May 9, 2026

Title Estimate Chla Concentrations of Phytoplankton Groups

Version 2.3.1

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, dynamicTreeCut, ggplot2, RcppML, stats, tidyr, progress

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.8)

LazyData true

Suggests knitr, rmarkdown, testthat (>= 3.0.0), quarto, here

VignetteBuilder knitr

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

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

Config/testthat/edition 3

NeedsCompilation no

Author Alexander Hayward [aut, cre, cph],
Tylar Murray [aut],
Sebastian Di Geronimo [aut],
Mohd Aasim Maqsood Khan [aut],
Andy McKenzie [aut]

Maintainer Alexander Hayward <phytoclass@outlook.com>

Repository CRAN

Date/Publication 2026-01-30 15:20:03 UTC

Contents

`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,
  minSamplesPerCluster,
  row_ids = NULL,
  dist_method = "euclidean",
  hclust_method = "ward.D2"
)
```

Arguments

`Data`	S (sample) matrix
`minSamplesPerCluster`	the minimum number of samples required for a cluster
`row_ids`	A vector of custom row names to be added to dendrogram
`dist_method`	Distance metric to be used in `stats::dist`. This should be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski".
`hclust_method`	Cluster method to be used in `stats::hclust`. This should be one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC).

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)
```

convergence_figure *Convergence Figure*

Description

A figure to show the pigment ratios for each phytoplankton group for each iteration.

Usage

```
convergence_figure(fm_iter, niter = NULL)
```

Arguments

fm_iter	A data.frame with columns of iter, phyto, pigment and ratio
niter	Optional: the number of iterations on the x axis. If NULL, will extract from the iter column of fm_iter.

Value

A figure with each pigment ratio per iteration per group

Examples

```
# ADD_EXAMPLES_HERE
```

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

`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

```
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 51 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, S_weights = NULL)
```

Arguments

Fn	Pigment to Chl a matrix
S	Sample data matrix – a matrix of pigment samples
S_weights	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

```
Fmat <- as.matrix(phytoclass::Fm)
S <- as.matrix(phytoclass::Sm)
S_weights <- as.numeric(phytoclass:::Bounded_weights(S))
place <- which(Fmat[, seq(ncol(Fmat) - 2)] > 0)
num.loops <- 2
# Run Steepest_Descent
result <- phytoclass:::Steepest_Descent(Fmat, place, S, S_weights, num.loops)
```

---

phyto_figure *Phytoplankton Class Abundance Figure*

Description

This function plots the class abundances as output by `simulated_annealing`.

Usage

```
phyto_figure(c_matrix)
```

Arguments

c_matrix	C matrix of class abundance concentrations

Value

A stacked line plot with sample number on x axis, chl a concentrations on y axis, and phytoplankton groups as colors

Examples

```
# ADD_EXAMPLES_HERE
```

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

```
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,
  seed = NULL,
  check_converge = 100,
  alt_pro_name = NULL
)
```

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
seed	Set number to reproduce the same results

check_converge	TRUE/FALSE/integer; set the number of F matrices to for convergence checking
alt_pro_name	Optional: additional alternate versions of divinyl-chlorophyll-a spellings used to detect prochlorococcus (Default: "dvchl", "dvchla", "dv_chla")

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
8. F_mat_iter
9. converge_plot

Examples

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

`simulated_annealing_Prochloro`

Perform simulated annealing algorithm for samples with divinyl chlorophyll and prochlorococcus. Chlorophyll must be the final column of both S and F matrices, with Divinyl Chlorophyll a the 2nd to last column. See how the example Sp and Fp matrices are organised.

Description

Perform simulated annealing algorithm for samples with divinyl chlorophyll and prochlorococcus. Chlorophyll must be the final column of both S and F matrices, with Divinyl Chlorophyll a the 2nd to last column. See how the example Sp and Fp matrices are organised.

Usage

```
simulated_annealing_Prochloro(
  S,
  Fmat = NULL,
  user_defined_min_max = NULL,
  do_matrix_checks = TRUE,
```

```
  niter = 500,
  step = 0.009,
  weight.upper.bound = 30,
  verbose = TRUE,
  seed = NULL,
  check_converge = 100
)
```

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
seed	Set seed number to reproduce the same results
check_converge	TRUE/FALSE/integer; set the number of F matrices to for convergence checking

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 Sp and Fp.
set.seed(5326)
sa.example <- simulated_annealing_Prochloro(Sp, Fp, niter = 1)
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

`Sp` *Sp data*

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

Steepest_Desc	*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.*

Description

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

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