Package 'RFLPtools'

January 20, 2025

Type Package

Title Tools to Analyse RFLP Data

Version 2.0

Date 2022-02-07

Author Fabienne Flessa [aut],
Alexandra Kehl [aut] (<https://orcid.org/0000-0002-1547-0713>),
Mohammed Aslam Imtiaz [aut],
Matthias Kohl [aut, cre] (<https://orcid.org/0000-0001-9514-8910>)

Maintainer Matthias Kohl <Matthias.Kohl@stamats.de>

Description Provides functions to analyse DNA fragment samples (i.e. derived from RFLP-analysis) and standalone BLAST report files (i.e. DNA sequence analysis).

Depends R(>= 4.0.0), RColorBrewer

Imports stats, utils, graphics, grDevices

Suggests knitr, rmarkdown, lattice, MKomics

VignetteBuilder knitr

License LGPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2022-02-08 09:40:02 UTC

Contents

RFLPtools-package *Tools To Analyse RFLP-Data*

Description

RFLPtools provides functions to analyse DNA fragment samples (i.e. derived from RFLP-analysis) and standalone BLAST report files (i.e. DNA sequence analysis).

Details

Package:	RFLPtools
Version:	2.0
Date:	2022-02-07
Depends:	R(>= 4.0.0)
Imports:	stats, utils, graphics, grDevices, RColorBrewer
Suggests:	knitr, rmarkdown, lattice, MKomics
License:	LGPL-3

Author(s)

Fabienne Flessa <`Fabienne.Flessa@uni-bayreuth.de`>,
Alexandra Kehl <`Alexandra.Kehl@uni-tuebingen.de`>,
Mohammed Aslam Imtiaz,
Matthias Kohl <`Matthias.Kohl@stamats.de`>

Maintainer: Matthias Kohl <`Matthias.Kohl@stamats.de`>

References

Local Blast download: https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=Download

Blast News: https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastNews

Ian A. Dickie, Peter G. Avis, David J. McLaughlin, Peter B. Reich. Good-Enough RFLP Matcher (GERM) program. Mycorrhiza 2003, 13:171-172.

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

Matsumoto, Masaru; Furuya, Naruto; Takanami, Yoichi; Matsuyama, Nobuaki. RFLP analysis of the PCR-amplified 28S rDNA in Rhizoctonia solani. Mycoscience 1996 37:351-356.

Persoh, D., Melcher, M., Flessa, F., Rambold, G.: First fungal community analyses of endophytic ascomycetes associated with Viscum album ssp. austriacum and itshost Pinus sylvestris. Fungal Biology 2010 Jul;114(7):585-96.

Poussier, Stephane; Trigalet-Demery, Danielle; Vandewalle, Peggy; Goffinet, Bruno; Luisetti, Jacques; Trigalet, Andre. Genetic diversity of Ralstonia solanacearum as assessed by PCR-RFLP of the hrp gene region, AFLP and 16S rRNA sequence analysis, and identification of an African subdivision. Microbiology 2000 146:1679-1692.

T. A. Saari, S. K. Saari, C. D. Campbell, I. J Alexander, I. C. Anderson. FragMatch - a program for the analysis of DNA fragment data. Mycorrhiza 2007, 17:133-136

Examples

```
data(RFLPdata)
res <- RFLPdist(RFLPdata)
plot(hclust(res[[1]]), main = "Euclidean distance")

par(mfrow = c(1,2))
plot(hclust(RFLPdist(RFLPdata, nrBands = 3)), cex = 0.7)
RFLPplot(RFLPdata, nrBands = 3, mar.bottom = 6, cex.axis = 0.8)

data(RFLPref)
RFLPrefplot(RFLPdata, RFLPref, nrBands = 6, cex.axis = 0.8)


library(MKomics)
data(BLASTdata)
res <- simMatrix(BLASTdata, sequence.range = TRUE, Min = 500)
myCol <- colorRampPalette(brewer.pal(8, "RdYlGn"))(128)
simPlot(res, col = myCol, minVal = 0,
        labels = colnames(res), title = "(Dis-)Similarity Plot")
```

`BLASTdata` *Example data set for BLAST data*

Description

This is an example data set for BLAST data generated with standalone BLAST from NCBI.

Usage

```
data(RFLPdata)
```

Format

A data frame with 737 observations on the following four variables

`query.id` character: sequence identifier.

`subject.id` character: subject identifier.

`identity` numeric: identity between sequences (in percent).

`alignment.length` integer: number of nucleotides.

`mismatches` integer: number of mismatches.

`gap.opens` integer: number of gaps.

`q.start` integer: query sequence start.

`q.end` integer: query sequence end.

`s.start` integer: subject sequence start.

`s.end` integer: subject sequence end.

`evalue` numeric: evalue.

`bit.score` numeric: score value.

Details

The data was generated with standalone BLAST from NCBI. Pairwise similarities of DNA sequences are calculated among all sequences to analyse applying Standalone Blast with the parameters -m 8 -r 2 -G 5 -E 2.

Alternatively data can be generated with "local BLAST" implemented in BioEdit v7.0.9 using the additional parameters -m 8 -r 2 -G 5 -E 2 and by selecting "open output" and "tabular output".

Source

The data set was generated by F. Flessa.

References

Standalone Blast download: `https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/`

Blast News: `https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastNews`

BioEdit: `https://bioedit.software.informer.com/`

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

Examples

```
data(BLASTdata)
str(BLASTdata)
```

---

diffDist *Distance Matrix Computation*

Description

This function computes and returns the distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix. Instead of the row values as in the case of `dist`, the successive differences of the row values are used.

Usage

```
diffDist(x, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
```

Arguments

`x`	a numeric matrix, data frame or `"dist"` object.
`method`	the distance measure to be used. This must be one of `"euclidean"`, `"maximum"`, `"manhattan"`, `"canberra"`, `"binary"` or `"minkowski"`. Any unambiguous substring can be given.
`diag`	logical value indicating whether the diagonal of the distance matrix should be printed by `print.dist`.
`upper`	logical value indicating whether the upper triangle of the distance matrix should be printed by `print.dist`.
`p`	The power of the Minkowski distance.

Details

This function computes and returns the distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix. Instead of the row values as in the case of `dist`, the successive differences of the row values are used.

It's a simple wrapper function arround `dist`. For more details about the distances we refer to `dist`.

The function may be helpful, if there is a shift w.r.t.\ the measured bands; e.g.\ `c(550, 500, 300, 250)` vs.\ `c(510, 460, 260, 210)`.

Value

`diffDist` returns an object of class `"dist"`; cf. `dist`.

Author(s)

Matthias Kohl <`Matthias.Kohl@stamats.de`>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

Examples

```
## assume a shift in the measured bands
M <- rbind(c(550, 500, 300, 250), c(510, 460, 260, 210),
           c(550, 500, 300, 200))
dist(M)
diffDist(M)
```

FragMatch — *Compute matches for RFLP data via FragMatch.*

Description

Compute matches for RFLP data using FragMatch - a program for the analysis of DNA fragment data.

Usage

```
FragMatch(newData, refData, maxValue = 1000, errorBound = 25,
          weight = 1, na.rm = TRUE)
```

Arguments

`newData` data.frame with new RFLP data; see `newDataGerm`.

`refData` data.frame with reference RFLP data; see `refDataGerm`.

`maxValue` numeric: maximum value for which the error bound is applied. Can be a vector of length larger than 1.

`errorBound` numeric: error bound corresponding to `maxValue`. Can be a vector of length larger than 1.

`weight` numeric: weight for weighting partial matches; see details section.

`na.rm` logical: indicating whether NA values should be stripped before the computation proceeds.

Details

A rather simple algorithm which consists of counting the number of matches where it is considered a match if the value is inside a range of +/- `errorBound`.

If there is more than one enzyme, one can use weights to give the partial perfect matches for a certain enzyme a higher (or also smaller) weight.

Value

A character matrix with entries of the form "a_b" which means that there were a out of b possible matches.

Author(s)

Mohammed Aslam Imtiaz, Matthias Kohl <Matthias.Kohl@stamats.de>

References

T. A. Saari, S. K. Saari, C. D. Campbell, I. J Alexander, I. C. Anderson. FragMatch - a program for the analysis of DNA fragment data. Mycorrhiza 2007, 17:133-136

See Also

newDataGerm, refDataGerm

Examples

```
data(refDataGerm)
data(newDataGerm)

res <- FragMatch(newDataGerm, refDataGerm)
```

---

germ *Compute matches for RFLP data via GERM.*

Description

Compute matches for RFLP data using the Good-Enough RFLP Matcher (GERM) program.

Usage

```
germ(newData, refData, parameters = list("Max forward error" = 25,
                                         "Max backward error" = 25,
                                         "Max sum error" = 100,
                                         "Lower measurement limit" = 100),
     method = "joint", na.rm = TRUE)
```

Arguments

`newData` data.frame with new RFLP data; see newDataGerm.

`refData` data.frame with reference RFLP data; see refDataGerm.

`parameters` list of the four program parameters of GERM; see details section.

`method` matching and ranking method used for computation; see details section.

`na.rm` logical: indicating whether NA values should be stripped before the computation proceeds.

Details

There are four matching and ranking methods which are "joint", "forward", "backward", and "sum". For more details see Dickie et al. (2003).

The parameters of the GERM software are: "Max forward error": Used if "matching and ranking method" is set to "forward" or "joint". "Max backward error": Used if "matching and ranking method" is set to "backward" or "joint". "Max sum error": Used for matching if "matching and ranking method" is set to "sum". "Lower measurement limit": The lower bound of measurements (often 100 or 50, depending on ladder used).

Value

A named list with the results.

Author(s)

Mohammed Aslam Imtiaz, Matthias Kohl <Matthias.Kohl@stamats.de>

References

Ian A. Dickie, Peter G. Avis, David J. McLaughlin, Peter B. Reich. Good-Enough RFLP Matcher (GERM) program. Mycorrhiza 2003, 13:171-172.

See Also

newDataGerm, refDataGerm

Examples

```
data(refDataGerm)
data(newDataGerm)

## Example 1
res1 <- germ(newDataGerm[1:7,], refDataGerm)

## Example 2
res2 <- germ(newDataGerm[8:15,], refDataGerm)

## Example 3
res3 <- germ(newDataGerm[16:20,], refDataGerm)

## all three examples in one step
res.all <- germ(newDataGerm, refDataGerm)
```

linCombDist *Linear Combination of Distances*

Description

This function computes linear combinations of distances.

Usage

```
linCombDist(x, distfun1, w1, distfun2, w2, diag = FALSE, upper = FALSE)
```

Arguments

x	object which is passed to distfun1 and distfun2.
distfun1	function used to compute an object of class "dist".
w1	weight for result of distfun1.
distfun2	function used to compute an object of class "dist".
w2	weight for result of distfun2.
diag	see dist
upper	see dist

Details

This function computes and returns the distance matrix computed by a linear combination of two distance matrices.

Value

linCombDist returns an object of class "dist"; cf. dist.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

Examples

```
## assume a shift in the measured bands
M <- rbind(c(550, 500, 300, 250), c(510, 460, 260, 210),
           c(700, 650, 450, 400), c(550, 490, 310, 250))
dist(M)
diffDist(M)

## convex combination of dist and diffDist
linCombDist(M, distfun1 = dist, w1 = 0.5, distfun2 = diffDist, w2 = 0.5)

## linear combination
linCombDist(M, distfun1 = dist, w1 = 2, distfun2 = diffDist, w2 = 5)

## maximum distance
linCombDist(M, distfun1 = function(x) dist(x, method = "maximum"), w1 = 0.5,
            distfun2 = function(x) diffDist(x, method = "maximum"), w2 = 0.5)

data(RFLPdata)
distfun <- function(x) linCombDist(x, distfun1 = dist, w1 = 0.1, distfun2 = diffDist, w2 = 0.9)
par(mfrow = c(2, 2))
plot(hclust(RFLPdist(RFLPdata, nrBands = 3, distfun = distfun)), cex = 0.7, cex.lab = 0.7)
RFLPplot(RFLPdata, nrBands = 3, distfun = distfun, mar.bottom = 6, cex.axis = 0.8)
plot(hclust(RFLPdist(RFLPdata, nrBands = 3)), cex = 0.7, cex.lab = 0.7)
RFLPplot(RFLPdata, nrBands = 3, mar.bottom = 6, cex.axis = 0.8)
```

`newDataGerm` *Example data set from GERM software*

Description

This is the reference data taken from the GERM software.

Usage

```
data(newDataGerm)
```

Format

A data frame with 20 observations on the following six variables

`Sample` character: sample identifier.

`Enzyme` character: enzyme used.

`Band` integer: band number.

`MW` integer: molecular weight.

`Genus` character: genus of sample.

`Species` character: species of sample.

Details

See GERM software.

Source

The data set was taken from the GERM software (table 'Example Unknowns').

References

Ian A. Dickie, Peter G. Avis, David J. McLaughlin, Peter B. Reich. Good-Enough RFLP Matcher (GERM) program. Mycorrhiza 2003, 13:171-172.

Examples

```
data(newDataGerm)
str(newDataGerm)
```

`nrBands` *Function to compute number of bands.*

Description

Computes groups based on the number of bands per sample in a RFLP data set. Each group comprises RFLP-samples with equal number of bands.

Usage

```
nrBands(x)
```

Arguments

`x`	data.frame with RFLP data; see `RFLPdata`.

Details

The function computes groups based on the number of bands per sample in a RFLP data set. Each group comprises RFLP-samples with equal number of bands.

Value

Number of bands per RFLP-samples.

Author(s)

Fabienne Flessa <`Fabienne.Flessa@uni-bayreuth.de`>,
Alexandra Kehl <`Alexandra.Kehl@uni-tuebingen.de`>,
Matthias Kohl <`Matthias.Kohl@stamats.de`>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

RFLPdata, RFLPdist2, dist

Examples

```
data(RFLPdata)
nrBands(RFLPdata)
```

read.blast *Read BLAST data*

Description

Function to read BLAST data generated with standalone BLAST from NCBI.

Usage

```
read.blast(file, sep = "\t")
```

Arguments

file	character: BLAST file to read in.
sep	the field separator character. Values on each line of the file are separated by this character. Default "\t".

Details

The function reads data which was generated with standalone BLAST from NCBI; see ftp://ftp.ncbi.nih.gov/blast/executables/

Possible steps:
1) Install NCBI BLAST
2) Generate and import database(s)
3) Apply BLAST with options outfmt and out; e.g.
`blastn -query Testquery -db Testdatabase -outfmt 6 -out out.txt`
or
`blastn -query Testquery -db Testdatabase -outfmt 10 -out out.csv`
One can also call BLAST from inside R by using function system
system("blastn -query Testquery -db Testdatabase -outfmt 6 -out out.txt")
4) Read in the results
`test.res <- read.blast(file = "out.txt")`
or
`test.res <- read.blast(file = "out.csv", sep = ",")`

Value

A data.frame with variables

query.id character: sequence identifier.

subject.id character: subject identifier.

identity numeric: identity between sequences (in percent).

alignment.length integer: number of nucleotides.

mismatches integer: number of mismatches.

gap.opens integer: number of gaps.

q.start integer: query sequence start.

q.end integer: query sequence end.

s.start integer: subject sequence start.

s.end integer: subject sequence end.

evalue numeric: evalue.

bit.score numeric: score value.

Author(s)

Fabienne Flessa <Fabienne.Flessa@uni-bayreuth.de>,
Alexandra Kehl <Alexandra.Kehl@uni-tuebingen.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

References

Standalone Blast download: https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/

Blast News: https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastNews

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

BLASTdata, simMatrix

Examples

```
Dir <- system.file("extdata", package = "RFLPtools") # input directory
filename <- file.path(Dir, "BLASTexample.txt")
BLAST1 <- read.blast(file = filename)
str(BLAST1)
```

read.rflp *Read RFLP data*

Description

Function to read RFLP data (e.g. generated with software package Gene Profiler 4.05 (Scanalytics Inc.)) for DNA fragment analysis and genotyping, and exported to a text file.

Usage

```
read.rflp(file)
```

Arguments

`file`	character: RFLP file to read in.

Details

The function reads data from a text file which was generated e.g. with the software package Gene Profiler 4.05 (Scanalytics Inc.) for DNA fragment analysis and genotyping. The data file contains sample identifier (`Sample`), band number (`Band`), molecular weight (`MW`) and gel identifier (`Gel`) (see `RFLPdata`).

If gel identifier `Gel` is missing it is extracted from the sample identifier `Sample`.

Value

A `data.frame` with variables

`Sample` character: sample identifier.

`Band` integer: band number.

`MW` integer: molecular weight.

`Gel` character: gel identifier.

Author(s)

Fabienne Flessa <`Fabienne.Flessa@uni-bayreuth.de`>,
Alexandra Kehl <`Alexandra.Kehl@uni-tuebingen.de`>,
Matthias Kohl <`Matthias.Kohl@stamats.de`>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

`RFLPdata`, `RFLPdist`

Examples

```
Dir <- system.file("extdata", package = "RFLPtools") # input directory
filename <- file.path(Dir, "RFLPexample.txt")
RFLP1 <- read.rflp(file = filename)
str(RFLP1)

filename <- file.path(Dir, "AZ091016_report.txt")
RFLP2 <- read.rflp(file = filename)
str(RFLP2)
```

`refDataGerm` *Example data set from GERM software*

Description

This is the reference data taken from the GERM software.

Usage

```
data(refDataGerm)
```

Format

A data frame with 250 observations on the following six variables

`Sample` character: sample identifier.

`Enzyme` character: enzyme used.

`Band` integer: band number.

`MW` integer: molecular weight.

`Genus` character: genus of sample.

`Species` character: species of sample.

Details

See GERM software.

Source

The data set was taken from the GERM software (table 'Example Data').

References

Ian A. Dickie, Peter G. Avis, David J. McLaughlin, Peter B. Reich. Good-Enough RFLP Matcher (GERM) program. Mycorrhiza 2003, 13:171-172.

Examples

```
data(refDataGerm)
str(refDataGerm)
```

RFLPcombine *Combine RFLP data sets*

Description

Function to combine an arbitrary number of RFLP data sets.

Usage

```
RFLPcombine(...)
```

Arguments

`...`	two or more data.frames with RFLP data.

Details

The data sets are combined using `rbind`.

If data sets with identical sample identifiers are given, the identifiers are made unique using `make.unique`.

Value

A `data.frame` with variables

`Sample` character: sample identifier.

`Band` integer: band number.

`MW` integer: molecular weight.

`Gel` character: gel identifier.

Author(s)

Fabienne Flessa <Fabienne.Flessa@uni-bayreuth.de>,
Alexandra Kehl <Alexandra.Kehl@uni-tuebingen.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

`RFLPdata`

Examples

```
data(RFLPdata)
res <- RFLPcombine(RFLPdata, RFLPdata, RFLPdata)
RFLPplot(res, nrBands = 4)
```

RFLPdata *Example data set for RFLP data*

Description

This is an example data set for RFLP data.

Usage

```
data(RFLPdata)
```

Format

A data frame with 737 observations on the following four variables

- `Sample` character: sample identifier.
- `Band` integer: band number.
- `MW` integer: molecular weight.
- `Gel` character: gel identifier.

Details

The molecular weight was determined using the software package Gene Profiler 4.05 (Scanalytics Inc.) for DNA fragment analysis and genotyping, and exported to a text file.

Source

The data set was generated by F. Flessa.

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

Examples

```
data(RFLPdata)
str(RFLPdata)
```

RFLPdist *Compute distances for RFLP data.*

Description

Within each group containing RFLP-samples exhibiting a equal number of bands, the distance between the molecular weights is computed.

Usage

```
RFLPdist(x, distfun = dist, nrBands, LOD = 0)
```

Arguments

`x`	data.frame with RFLP data; see `RFLPdata`.
`distfun`	function computing the distance with default `dist`; cf. `dist`.
`nrBands`	if not missing, then only samples with the specified number of bands are considered.
`LOD`	threshold for low-bp bands.

Details

For each number of bands the given distance between the molecular weights is computed. The result is a named list of distances where the names correspond to the number of bands which occur in each group.

If `nrBands` is specified only samples with this number of bands are considered.

If `LOD` > `0` is specified, all values below `LOD` are removed before the distances are calculated.

Value

A named list with the distances; see `dist`.

In case `nrBands` is not missing, an object of S3 class `dist`.

Author(s)

Fabienne Flessa <`Fabienne.Flessa@uni-bayreuth.de`>,
Alexandra Kehl <`Alexandra.Kehl@uni-tuebingen.de`>,
Matthias Kohl <`Matthias.Kohl@stamats.de`>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

Poussier, Stephane; Trigalet-Demery, Danielle; Vandewalle, Peggy; Goffinet, Bruno; Luisetti, Jacques; Trigalet, Andre. Genetic diversity of Ralstonia solanacearum as assessed by PCR-RFLP of the hrp gene region, AFLP and 16S rRNA sequence analysis, and identification of an African subdivision. Microbiology 2000 146:1679-1692

Matsumoto, Masaru; Furuya, Naruto; Takanami, Yoichi; Matsuyama, Nobuaki. RFLP analysis of the PCR-amplified 28S rDNA in Rhizoctonia solani. Mycoscience 1996 37:351 - 356

See Also

RFLPdata, dist

Examples

```
## Euclidean distance
data(RFLPdata)
res <- RFLPdist(RFLPdata)
names(res) ## number of bands
res$"6"

RFLPdist(RFLPdata, nrBands = 6)

## Other distances
res1 <- RFLPdist(RFLPdata, distfun = function(x) dist(x, method = "manhattan"))
res2 <- RFLPdist(RFLPdata, distfun = function(x) dist(x, method = "maximum"))
res[[1]]
res1[[1]]
res2[[1]]

## cut dendrogram at height 50
clust4bd <- hclust(res[[2]])
cgroups50 <- cutree(clust4bd, h=50)
cgroups50

## or
library(MKomics)
res3 <- RFLPdist(RFLPdata, distfun = corDist)
res3$"9"

## hierarchical clustering
par(mfrow = c(2,2))
plot(hclust(res[[1]]), main = "Euclidean distance")
plot(hclust(res1[[1]]), main = "Manhattan distance")
plot(hclust(res2[[1]]), main = "Maximum distance")
plot(hclust(res3[[1]]), main = "Pearson correlation distance")


## Similarity matrix
library(MKomics)
myCol <- colorRampPalette(brewer.pal(8, "RdYlGn"))(128)
ord <- order.dendrogram(as.dendrogram(hclust(res[[1]])))
temp <- as.matrix(res[[1]])
simPlot(temp[ord,ord], col = rev(myCol), minVal = 0,
        labels = colnames(temp), title = "(Dis-)Similarity Plot")
```

```
## or
library(lattice)
levelplot(temp[ord,ord], col.regions = rev(myCol),
          at = do.breaks(c(0, max(temp)), 128),
          xlab = "", ylab = "",
          ## Rotate label of x axis
          scales = list(x = list(rot = 90)),
          main = "(Dis-)Similarity Plot")

## multidimensional scaling
loc <- cmdscale(res[[5]])
x <- loc[,1]
y <- -loc[,2]
plot(x, y, type="n", xlab="", ylab="", xlim = 1.05*range(x), main="Multidemsional scaling")
text(x, y, rownames(loc), cex=0.8)
```

RFLPdist2 *Compute distances for RFLP data.*

Description

If gel image quality is low, faint bands may be disregarded and may lead to wrong conclusions. This function computes the distance between the molecular weights of RFLP samples, including samples containing one or more additional bands. Thus, failures during band detection could be identified. Visualisation of band patterns using this method can be done by `RFLPplot` using the argument `nrMissing`.

Usage

```
RFLPdist2(x, distfun = dist, nrBands, nrMissing, LOD = 0,
          diag = FALSE, upper = FALSE)
```

Arguments

`x`	data.frame with RFLP data; see `RFLPdata`.
`distfun`	function computing the distance with default `dist`; cf. `dist`.
`nrBands`	samples with number of bands equal to `nrBands` are to be considered.
`nrMissing`	number of bands that might be missing.
`LOD`	threshold for low-bp bands.
`diag`	see `dist`
`upper`	see `dist`

Details

For a given number of bands the given distance between the molecular weights is computed. It is assumed that a number of bands might be missing. Hence all samples with number of bands in nrBands, nrBands+1, ..., nrBands+nrMissing are compared.

If `LOD > 0` is specified, it is assumed that missing bands can only occur for molecular weights smaller than `LOD`. As a consequence only samples which have `nrBands` bands with molecular weight larger or equal to `LOD` are selected.

For computing the distance between the molecular weight of a sample S1 with x bands and a Sample S2 with x+y bands the distances between the molecular weight of sample S1 and the molecular weight of all possible subsets of S2 with x bands are computed. The distance between S1 and S2 is then defined as the minimum of all these distances.

If `LOD > 0` is specified, only all combinations of values below `LOD` are considered.

This option may be useful, if gel image quality is low, and the detection of bands is doubtful.

Value

An object of class `"dist"` returned; cf. `dist`.

Author(s)

Fabienne Flessa `<Fabienne.Flessa@uni-bayreuth.de>`,
Alexandra Kehl `<Alexandra.Kehl@uni-tuebingen.de>`,
Matthias Kohl `<Matthias.Kohl@stamats.de>`

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

Ian A. Dickie, Peter G. Avis, David J. McLaughlin, Peter B. Reich. Good-Enough RFLP Matcher (GERM) program. Mycorrhiza 2003, 13:171-172.

See Also

`RFLPdata`, `nrBands`, `RFLPdist`, `dist`

Examples

```
## Euclidean distance
data(RFLPdata)
nrBands(RFLPdata)
res0 <- RFLPdist(RFLPdata, nrBands = 4)
res1 <- RFLPdist2(RFLPdata, nrBands = 4, nrMissing = 1)
res2 <- RFLPdist2(RFLPdata, nrBands = 4, nrMissing = 2)
res3 <- RFLPdist2(RFLPdata, nrBands = 4, nrMissing = 3)

## assume missing bands only below LOD
res1.lod <- RFLPdist2(RFLPdata, nrBands = 4, nrMissing = 1, LOD = 60)
```

```
## hierarchical clustering
par(mfrow = c(2,2))
plot(hclust(res0), main = "0 bands missing")
plot(hclust(res1), main = "1 band missing")
plot(hclust(res2), main = "2 bands missing")
plot(hclust(res3), main = "3 bands missing")

## missing bands only below LOD
par(mfrow = c(1,2))
plot(hclust(res0), main = "0 bands missing")
plot(hclust(res1.lod), main = "1 band missing below LOD")

## Similarity matrix
library(MKomics)
myCol <- colorRampPalette(brewer.pal(8, "RdYlGn"))(128)
ord <- order.dendrogram(as.dendrogram(hclust(res1)))
temp <- as.matrix(res1)
simPlot(temp[ord,ord], col = rev(myCol), minVal = 0,
        labels = colnames(temp), title = "(Dis-)Similarity Plot")

## missing bands only below LOD
ord <- order.dendrogram(as.dendrogram(hclust(res1.lod)))
temp <- as.matrix(res1.lod)
simPlot(temp[ord,ord], col = rev(myCol), minVal = 0,
      labels = colnames(temp), title = "(Dis-)Similarity Plot\n1 band missing below LOD")


## or
library(lattice)
levelplot(temp[ord,ord], col.regions = rev(myCol),
          at = do.breaks(c(0, max(temp)), 128),
          xlab = "", ylab = "",
          ## Rotate label of x axis
          scales = list(x = list(rot = 90)),
          main = "(Dis-)Similarity Plot")


## Other distances
res11 <- RFLPdist2(RFLPdata, distfun = function(x) dist(x, method = "manhattan"),
                 nrBands = 4, nrMissing = 1)
res12 <- RFLPdist2(RFLPdata, distfun = corDist, nrBands = 4, nrMissing = 1)
res13 <- RFLPdist2(RFLPdata, distfun = corDist, nrBands = 4, nrMissing = 1, LOD = 60)
par(mfrow = c(2,2))
plot(hclust(res1), main = "Euclidean distance\n1 band missing")
plot(hclust(res11), main = "Manhattan distance\n1 band missing")
plot(hclust(res12), main = "Pearson correlation distance\n1 band missing")
plot(hclust(res13), main = "Pearson correlation distance\n1 band missing below LOD")
```

RFLPdist2ref *Compute distance between RFLP data and RFLP reference data.*

Description

Function to compute distance between RFLP data and RFLP reference data.

Usage

```
RFLPdist2ref(x, ref, distfun = dist, nrBands, LOD = 0)
```

Arguments

`x`	data.frame with RFLP data; e.g. `RFLPdata`.
`ref`	data.frame with RFLP reference data; e.g. `RFLPref`.
`distfun`	function computing the distance with default `dist`; cf. `dist`.
`nrBands`	only samples and reference samples with this number of bands are considered.
`LOD`	threshold for low-bp bands.

Details

For each sample with `nrBands` bands the distance to each reference sample with `nrBands` bands is computed. The result is a matrix with the corresponding distances where rows represent the samples and columns the reference samples.

If `LOD` > `0` is specified, all values below `LOD` are removed before the distances are calculated. This applies to `x` and `ref`.

Value

A matrix with distances.

Author(s)

Fabienne Flessa <`Fabienne.Flessa@uni-bayreuth.de`>,
Alexandra Kehl <`Alexandra.Kehl@uni-tuebingen.de`>,
Matthias Kohl <`Matthias.Kohl@stamats.de`>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

`RFLPdata`, `dist`

Examples

```
## Euclidean distance
data(RFLPdata)
data(RFLPref)
nrBands(RFLPref)
RFLPdist2ref(RFLPdata, RFLPref, nrBands = 4)
RFLPdist2ref(RFLPdata, RFLPref, nrBands = 6)

Dir <- system.file("extdata", package = "RFLPtools") # input directory
filename <- file.path(Dir, "AZ091016_report.txt")
RFLP1 <- read.rflp(file = filename)
RFLP2 <- RFLPqc(RFLP1)
nrBands(RFLP2)
RFLPdist2ref(RFLP1, RFLPref, nrBands = 4)
RFLPdist2ref(RFLP1, RFLPref, nrBands = 5)
```

RFLPlod *Remove bands below LOD*

Description

Function to exclude bands below a given LOD.

Usage

```
RFLPlod(x, LOD)
```

Arguments

x	data.frame with RFLP data.
LOD	threshold for low-bp bands.

Details

Low-bp bands may be regarded as unreliable. Function RFLPlod can be used to exclude such bands, which are likely to be absent in some other samples, before further analyses.

Value

A data.frame with variables

Sample character: sample identifier.

Band integer: band number.

MW integer: molecular weight.

Gel character: gel identifier.

Author(s)

Fabienne Flessa <Fabienne.Flessa@uni-bayreuth.de>,
Alexandra Kehl <Alexandra.Kehl@uni-tuebingen.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

RFLPdata

Examples

```
data(RFLPdata)
## remove bands with MW smaller than 60
RFLPdata.lod <- RFLPlod(RFLPdata, LOD = 60)
par(mfrow = c(1, 2))
RFLPplot(RFLPdata, nrBands = 4, ylim = c(40, 670))
RFLPplot(RFLPdata.lod, nrBands = 4, ylim = c(40, 670))
title(sub = "After applying RFLPlod")
```

RFLPplot *Function to plot RFLP data.*

Description

Given RFLP data is plotted where the samples are sorted according to the corresponding dendrogram.

Usage

```
RFLPplot(x, nrBands, nrMissing, distfun = dist,
         hclust.method = "complete", mar.bottom = 5,
         cex.axis = 0.5, colBands, xlab = "",
         ylab = "molecular weight", ylim, ...)
```

Arguments

x	data.frame with RFLP data; see RFLPdata.
nrBands	if not missing, then only samples with the specified number of bands are considered.
nrMissing	if not missing, then it is assumed that some bands may be missing. That is, all samples with number of bands in nrBands, nrBands+1, ..., nrBands+nrMissing are considered.

distfun	function computing the distance with default dist; see dist.
hclust.method	method used for hierarchical clustering; see hclust.
mar.bottom	bottom margin of the plot; see par.
cex.axis	size of the x-axis annotation.
colBands	color for the bands. Has to be of length 1 or number of samples. If missing, "Set1" of **RColorBrewer** is used; see ColorBrewer.
xlab	passed to function plot.
ylab	passed to function plot.
ylim	passed to function plot. If missing an appropriate range of y-values is computed.
...	additional arguments passed to function plot except xlim which is defined inside of RFLPplot.

Details

RFLP data is plotted. The samples are sorted according to the corresponding dendrogram which is computed via function hclust.

The option to specify nrMissing may be useful, if gel image quality is low, and the detection of bands is doubtful.

Value

invisible

Author(s)

Fabienne Flessa <Fabienne.Flessa@uni-bayreuth.de>,
Alexandra Kehl <Alexandra.Kehl@uni-tuebingen.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

RFLPdata, dist

Examples

```
data(RFLPdata)
par(mfrow = c(1,2))
plot(hclust(RFLPdist(RFLPdata, nrBands = 3)), cex = 0.7)
RFLPplot(RFLPdata, nrBands = 3, mar.bottom = 6, cex.axis = 0.8)

par(mfrow = c(1,2))
plot(hclust(RFLPdist2(RFLPdata, nrBands = 9, nrMissing = 1)), cex = 0.7)
```

```
RFLPplot(RFLPdata, nrBands = 9, nrMissing = 1, mar.bottom = 6, cex.axis = 0.8)


distfun <- function(x) dist(x, method = "maximum")
par(mfrow = c(1,2))
plot(hclust(RFLPdist(RFLPdata, nrBands = 3, distfun = distfun),
            method = "average"), cex = 0.7, cex.lab = 0.7)
RFLPplot(RFLPdata, nrBands = 3, distfun = distfun, hclust.method = "average",
         mar.bottom = 6, cex.axis = 0.8)
```

RFLPqc *Quality control for RFLP data*

Description

Function to perform quality control for RFLP data based on a comparison between the total length of the digested PCR amplification product and the sum of the fragment lengths. If the sum is smaller or larger than the PCR amplification product (within a certain range to define), the samples can be excluded from further analyses. This function is helpful for data sets containig faint or uncertain bands. It is necessary to include the total length of the PCR amplification product for each sample as largest fragment in the data set, see RFLPdata.

Usage

```
RFLPqc(x, rm.band1 = TRUE, QC.lo = 0.8, QC.up = 1.07, QC.rm = FALSE)
```

Arguments

x	data.frame with RFLP data.
rm.band1	logical: remove first band.
QC.lo	numeric: a real number in (0,1).
QC.up	numeric: a real number larger than 1.
QC.rm	logical: remove samples with unsufficient quality.

Details

In case the first band corresponds to the total length of the fragment one can perform a quality control comparing the length of the first band with the sum of the lengths of the remaining bands for each sample. If the sum is smaller than QC.lo times the length of the first band or larger than QC.up times the length of the first band, respectively, a text message is printed.

If rm.band1 = TRUE band 1 of all samples is removed and the remaining band numbers are reduced by 1.

If QC.rm = TRUE samples of insufficient quality are entirely removed from the given data and the resulting data.frame is returned.

Value

A data.frame with variables

Sample character: sample identifier.

Band integer: band number.

MW integer: molecular weight.

Gel character: gel identifier.

Author(s)

Fabienne Flessa <Fabienne.Flessa@uni-bayreuth.de>,
Alexandra Kehl <Alexandra.Kehl@uni-tuebingen.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

RFLPdata, RFLPdist

Examples

```
Dir <- system.file("extdata", package = "RFLPtools") # input directory
filename <- file.path(Dir, "AZ091016_report.txt")
RFLP1 <- read.rflp(file = filename)
str(RFLP1)

RFLP2 <- RFLPqc(RFLP1, rm.band1 = FALSE) # identical to RFLP1
identical(RFLP1, RFLP2)

RFLP3 <- RFLPqc(RFLP1)
str(RFLP3)

RFLP4 <- RFLPqc(RFLP1, rm.band1 = TRUE, QC.rm = TRUE)
str(RFLP4)
```

RFLPref *Example data set for RFLP reference*

Description

This is an example data set for RFLP reference.

Usage

```
data(RFLPref)
```

Format

A data frame with 35 observations on the following five variables

`Sample` character: sample identifier.

`Band` integer: band number.

`MW` integer: molecular weight.

`Taxonname` character: taxon name.

`Accession` character: accession number.

Details

This example data set for RFLP reference consists of seven RFLP reference samples. Taxon names are assigned by sequence comparison with GenBank database (https://www.ncbi.nlm.nih.gov/BLAST/), and supplemented with imaginary accession numbers.

Source

The data set was generated by F. Flessa.

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

Examples

```
data(RFLPref)
str(RFLPref)
```

RFLPrefplot *Function for a visual comparison of RFLP samples with reference samples.*

Description

Given RFLP samples are plotted together with reference samples and sorted by their distance to the reference sample.

Usage

```
RFLPrefplot(x, ref, distfun = dist, nrBands, mar.bottom = 5,
            cex.main = 1.2, cex.axis = 0.5, devNew = FALSE,
            colBands, xlab = "", ylab = "molecular weight",
            ylim, ...)
```

Arguments

`x` data.frame with RFLP data; e.g. `RFLPdata`.

`ref` data.frame with RFLP reference data; e.g. `RFLPref`.

`distfun` function computing the distance with default `dist`; see `dist`.

`nrBands` if not missing, then only samples with the specified number of bands are considered.

`mar.bottom` bottom margin of the plot; see `par`.

`cex.main` size of the plot title.

`cex.axis` size of the x-axis annotation.

`devNew` logical. Open new graphics device for each plot.

`colBands` color for the bands. Has to be of length 1 or number of samples. If missing, `"Set1"` of **RColorBrewer** is used; see `ColorBrewer`.

`xlab` passed to function `plot`.

`ylab` passed to function `plot`.

`ylim` passed to function `plot`. If missing an appropriate range of y-values is computed.

`...` additional arguments passed to function `plot` except `xlim` which is defined inside of `RFLPplot`.

Details

Given RFLP samples are plotted together with reference samples and sorted by their distance to the reference sample.

Value

invisible

Author(s)

Fabienne Flessa <`Fabienne.Flessa@uni-bayreuth.de`>,
Alexandra Kehl <`Alexandra.Kehl@uni-tuebingen.de`>,
Matthias Kohl <`Matthias.Kohl@stamats.de`>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

`RFLPplot`

Examples

```
data(RFLPdata)
data(RFLPref)
dev.new(width = 12)
RFLPrefplot(RFLPdata, RFLPref, nrBands = 4, cex.axis = 0.5)

dev.new()
RFLPrefplot(RFLPdata, RFLPref, nrBands = 6, cex.axis = 0.8)
RFLPrefplot(RFLPdata, RFLPref, nrBands = 9, cex.axis = 0.8)

RFLPrefplot(RFLPdata, RFLPref[RFLPref$Sample == "Ni_29_A3",], nrBands = 4, cex.axis = 0.7)

Dir <- system.file("extdata", package = "RFLPtools") # input directory
filename <- file.path(Dir, "AZ091016_report.txt")
RFLP1 <- read.rflp(file = filename)
RFLP2 <- RFLPqc(RFLP1)

dev.new(width = 12)
RFLPrefplot(RFLP1, RFLPref, nrBands = 4, cex.axis = 0.8)

dev.new()
RFLPrefplot(RFLP1, RFLPref, nrBands = 5, cex.axis = 0.8)
```

sim2dist *Convert similarity matrix to dist object.*

Description

Function to convert similarity matrix to object of S3 class "dist".

Usage

```
sim2dist(x, maxSim = 1)
```

Arguments

x	symmetric matrix: similarity matrix.
maxSim	maximum similarity possible.

Details

Similarity is converted to distance by maxSim - x. The resulting matrix is converted to an object of S3 class "dist" by as.dist

Value

Object of S3 class "dist" is returned; see dist.

Author(s)

Fabienne Flessa <Fabienne.Flessa@uni-bayreuth.de>,
Alexandra Kehl <Alexandra.Kehl@uni-tuebingen.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

BLASTdata, simMatrix

Examples

```
data(BLASTdata)

## without sequence range
## Not run:
res <- simMatrix(BLASTdata)

## End(Not run)

## with sequence range
range(BLASTdata$alignment.length)
res1 <- simMatrix(BLASTdata, sequence.range = TRUE, Min = 100, Max = 450)
res2 <- simMatrix(BLASTdata, sequence.range = TRUE, Min = 500)

## visualize similarity matrix
library(MKomics)
simPlot(res2, minVal = 0,
        labels = colnames(res2), title = "(Dis-)Similarity Plot")


## or
library(lattice)
myCol <- colorRampPalette(brewer.pal(8, "RdYlGn"))(128)
levelplot(res2, col.regions = myCol,
          at = do.breaks(c(0, max(res2)), 128),
          xlab = "", ylab = "",
          ## Rotate label of x axis
          scales = list(x = list(rot = 90)),
          main = "(Dis-)Similarity Plot")

## convert to distance
res.d <- sim2dist(res2)

## hierarchical clustering
plot(hclust(res.d))
```

`simMatrix` *Similarity matrix for BLAST data.*

Description

Function to compute similarity matrix for all-vs-all BLAST results of rDNA sequences generated with standalone BLAST from NCBI or local BLAST implemented in BioEdit.

Usage

```
simMatrix(x, sequence.range = FALSE, Min, Max)
```

Arguments

`x`	data.frame with BLAST data; see `BLASTdata`.
`sequence.range`	logical: use sequence range.
`Min`	minimum sequence length.
`Max`	maximum sequence length.

Details

The given BLAST data is used to compute a similarity matrix using the following algorithm: First, the length of each sequence (LS) comprised in the input data file is extracted. If there is more than one comparison for one sequence including different parts of the respective sequence, that one with maximum base length is chosen. Subsequently, the number of matching bases (mB) is calculated by multiplying two variables comprised in the BLAST output: the identity between sequences (%) and the number of nucleotides divided by 100. The, resulting value is rounded to integer. Furthermore, the similarity is calculated by dividing mB by LS. Finally, the similarity matrix including all sequences is built. If the similarity of a combination is not shown in the BLAST report file (because the similarity was lower than 70%), this comparison is included in the similarity matrix with the result zero.

Value

Similarity matrix.

Author(s)

Fabienne Flessa <`Fabienne.Flessa@uni-bayreuth.de`>,
Alexandra Kehl <`Alexandra.Kehl@uni-tuebingen.de`>,
Matthias Kohl <`Matthias.Kohl@stamats.de`>

References

Standalone Blast download: https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/

Blast News: https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastNews

BioEdit: https://bioedit.software.informer.com/

Persoh, D., Melcher, M., Flessa, F., Rambold, G.: First fungal community analyses of endophytic ascomycetes associated with Viscum album ssp. austriacum and itshost Pinus sylvestris. Fungal Biology 2010 Jul;114(7):585-96.

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

BLASTdata, sim2dist

Examples

```
data(BLASTdata)

## without sequence range
## code takes some time
## Not run:
res <- simMatrix(BLASTdata)

## End(Not run)

## with sequence range
range(BLASTdata$alignment.length)
res1 <- simMatrix(BLASTdata, sequence.range = TRUE, Min = 100, Max = 450)
res2 <- simMatrix(BLASTdata, sequence.range = TRUE, Min = 500)
```

simulateRFLPdata *Simulate RFLP data.*

Description

Simulates RFLP data for comparions of algorithms.

Usage

```
simulateRFLPdata(N = 10, nrBands = 3:12, bandCenters = seq(100, 800, by = 100),
                 delta = 50, refData = FALSE)
```

Arguments

`N` integer: number samples which shall be simulated per number of bands.

`nrBands` integer: vector of number of bands.

`bandCenters` numeric: vector of band centers.

`delta` numeric: uniform distribution with `min = bandCenter - delta` and `max = bandCenter + delta` is used.

`refData` logical: if TRUE, additonal columns `Taxonname` and `Accesion` are generated.

Details

The function can be used to simulate RFLP data. For every number of band specified in `nrBands` a total number of `N` samples are generated.

First the band centers are randomly selected (with replacement) from `bandCenter` which form the centers of intervals of length `2*delta`. From these intervals uniform random numbers are drawn leading to randomly generated RFLP data.

Value

A data frame with `N*length(nrBands)` observations on the following four variables

`Sample` character: sample identifier.

`Band` integer: band number.

`MW` integer: molecular weight.

`Enzyme` character: enzyme name.

is generated. If `refData = TRUE` then the following two additional variables are added.

`Taxonname` character: taxon name.

`Accession` character: accession number.

Author(s)

Mohammed Aslam Imtiaz, Matthias Kohl <`Matthias.Kohl@stamats.de`>

See Also

`RFLPdata`, `RFLPref`

Examples

```
simData <- simulateRFLPdata()
```

write.hclust *Cut a hierarchical cluster tree and write cluster identifiers to a text file.*

Description

The tree obtained by a hierarchical cluster analysis is cut into groups by using `cutree` and the results are exported to a text file.

Usage

```
write.hclust(x, file, prefix, h = NULL, k = NULL, append = FALSE, dec = ",")
```

Arguments

`x`	object of class `hclust`: result of hierarchical cluster analysis computed via function `hclust`.
`file`	either a character string naming a file or a connection open for writing. `""` indicates output to the console.
`prefix`	character. Information about the cluster analysis.
`h`	numeric scalar or vector with heights where the tree should be cut.
`k`	an integer scalar or vector with the desired number of groups.
`append`	logical. Only relevant if `file` is a character string. If `TRUE`, the output is appended to the file. If `FALSE`, any existing file of the name is destroyed.
`dec`	the string to use for decimal points in numeric or complex columns: must be a single character.

Details

The results are written to file by a call to `write.table` where the columns in the resulting file are seperated by tabulators (i.e. `sep="\t"`) and no row names are exported (i.e. `row.names = FALSE`).

Author(s)

Fabienne Flessa <`Fabienne.Flessa@uni-bayreuth.de`>,
Alexandra Kehl <`Alexandra.Kehl@uni-tuebingen.de`>,
Matthias Kohl <`Matthias.Kohl@stamats.de`>

References

Flessa, F., Kehl, A., Kohl, M. Analysing diversity and community structures using PCR-RFLP: a new software application. Molecular Ecology Resources 2013 Jul; 13(4):726-33.

See Also

`write.table`, `cutree`

Examples

```
data(RFLPdata)
res <- RFLPdist(RFLPdata, nrBands = 4)
cl <- hclust(res)
## Not run:
write.hclust(cl, file = "Test.txt", prefix = "Bd4", h = 50)

## End(Not run)

res <- RFLPdist2(RFLPdata, nrBands = 4, nrMissing = 1)
cl <- hclust(res)
## Not run:
write.hclust(cl, file = "Test.txt", append = TRUE, prefix = "Bd4_Mis1", h = 60)

## End(Not run)
```

Index