

R documentation

of 'CONOP-class.Rd' etc.

July 3, 2017

CONOP-class

Class "CONOP"

Description

A class that contains all output from a CONOP run.

Objects from the Class

Objects can be created by calls of the form `new("CONOP", ...)`.

Slots

Summary: Object of class "vector". It contains information on the CONOP run such as the name of the run, the misfit of the best solution and some metrics to estimate the quality of the run.

Config: Object of class "list". It contains the configuration file of the run.

Sectfile: Object of class "data.frame". It corresponds exactly to the SECTFILE file.

Eventfile: Object of class "data.frame". It corresponds exactly to the EVENTFILE file.

Obsdfile: Object of class "matrix". It corresponds exactly to the OBSDFILE file.

Plcdfile: Object of class "matrix". It corresponds exactly to the PLCDFILE file.

Composfile: Object of class "matrix". It corresponds exactly to the COMPOSFILE file.

Extnfile: Object of class "matrix". It corresponds exactly to the EXTNFILE file.

Events: Object of class "matrix". It contains, for each event, its misfit, weighted misfit and the number of sections in which it is observed.

Sections: Object of class "list". It is a list of events with their depth (observed and placed) and their misfit for each separate section studied.

Curvfile: Object of class "matrix". It corresponds exactly to the CURVFILE file.

Methods

print signature(`x = "CONOP"`): print a summary of what contains the CONOP object.

Note

Objects of class CONOP are meant to be created by function [read.CONOP](#).

Author(s)

Johan Renaudie.

See Also

[read.CONOP](#)

Examples

```
showClass("CONOP")
```

CONOP9companion-package

Tools for integrating biostratigraphic software CONOP9 in a statistical workflow

Description

Help preparing input files of CONOP9, read output files of CONOP9 and a few functions to plot and analyze the data.

Details

Package:	CONOP9companion
Type:	Package
Version:	1.0
Date:	2013-09-24
License:	GPL
Depends:	methods

Two object classes are implemented ([CONOPinput-class](#) for input files and [CONOP-class](#) for output files of CONOP9). Functions [lofo](#) and [pacman](#) are used to prepare the data before the CONOP analysis. Function [prep.conop](#) is used to prepare the input files for CONOP9 while function [read.CONOP](#) is used to import the result of a CONOP analysis. Several classic plots, already presents in the CONOP9 GUI, are also implemented here, namely [draw.ranges](#), [sect.ranges](#), [fence](#), [printLOC](#) and [collection.quality](#), [culling](#) and [Section.vs.composite](#). Some additional functions are provided to help analysing the composite sequence of events ([compact.ordinal](#), [composite.age](#), [site2composite](#)).

Author(s)

Johan Renaudie.

Johan Renaudie <johan.renaudie@mfn-berlin.de> <lecoryphee@googlemail.com>

References

Sadler, P. M. 2007. CONOP9 version 7.43

CONOPinput-class	Class "CONOPinput"
------------------	--------------------

Description

A class that contains all inputs for a CONOP run.

Objects from the Class

Objects can be created by calls of the form `new("CONOPinput", ...)`.

Slots

Config: Object of class "list". It corresponds to the configuration file of the run.

Sectfile: Object of class "data.frame". It corresponds exactly to the SECTFILE file.

Eventfile: Object of class "data.frame". It corresponds exactly to the EVENTTFILE file.

Levelfile: Object of class "data.frame". It is a correspondance table giving the number of each sample in each section, according to its depth (height). It is meant to be the output of function [prep.levelfile](#).

Loadfile: Object of class "data.frame". It corresponds exactly to the LOADFILE file.

Methods

write signature(x = "CONOPinput"): write the input files to a folder, giving them names according to the configuration file. To run CONOP9, just add a copy of the software in the folder and launch it.

Author(s)

Johan Renaudie.

See Also

[prep.CONOP](#)

Examples

```
showClass("CONOPinput")
```

Description

`draw.ranges` plots a range-chart. `culling` shows the full range of positions where an event may be placed in the optimal sequence. `fence` plots a fence diagram. `collection.quality` plots the events and ranges observed in each given samples vs the events or ranges that should be observed in each given samples according to the solution of the optimization. `Section_vs_Composite` create a plot of the composite section with the range occupied by the queried section highlighted. `sect_range` create a plot of taxa ranges on a section, comparing observed range and predicted range.

Usage

```
draw.ranges(CONOP, order = "FAD", method = "depth", ...)
culling(CONOP, cpcht, FO.col = "blue", LO.col = "red")
fence(CONOP, events = "observed", tag = 1)
collection.quality(CONOP, section)
Section.vs.Composite(CONOP, section, col, cex=1, axes=TRUE)
sect.range(CONOP, section, cex=0.8)
```

Arguments

CONOP	An object of class CONOP, typically the output of function <code>read.CONOP</code> .
order	The rangechart can be ordered by "FAD" or by "LAD".
method	If "depth", then the rangechart is plotted according to the composite sequence 'depth'. If "age" it is plotted according to the age of the events.
cpcht	The output of function <code>read.cpcht</code> .
FO.col	Color for FO.
LO.col	Color for LO.
col	Color of the highlighted area.
events	A character string. Can be observed, in which case the events are plotted at the depth (or height) they have been observed, or placed, in which case they are plotted at their optimal depth (or height).
tag	Two possible values: if 1 then each sites will be named after its abridged name, if 2 after its complete name (respectively columns 2 and 4 of the input file SECT-FILE).
section	Section name. Should correspond to the complete name of the section as entered in the SECTFILE.
cex	Size factor for the annotations. To be decreased when plotting several sections on the same plot.
axes	Logical. TRUE if axes desired, FALSE otherwise. Two x-axes are to be plotted: one corresponding to the composite section, the other labels the extremes of the queried section. If only the composite section depth is desired, use <code>axes=FALSE</code> and then <code>axis(1,...)</code> or <code>axis(3,...)</code> after the call to <code>Section.vs.Composite</code> .
...	Any argument that could be passed to segments.

Author(s)

Johan Renaudie, based on the CONOP9 functionalities created by Peter Sadler.

Examples

```
data(riley)
par(mfcol=c(2,1))
par(mar=c(0,5,1,1))
fence(riley,"placed")
par(mar=c(1,5,0,1))
fence(riley, "observed")

collection.quality(riley, "Wht")

par(mfrow=c(nrow(riley@Sectfile),1))
for(i in riley@Sectfile[,4]){
  par(mar=c(1,3,1,1))
  Section.vs.Composite(riley,i,col="lightblue",cex=0.8,axes=FALSE)
}

sect.range(riley,"Morgan")
```

LOC

Line of Correlation

Description

Functions to derive an age model from the composite sequence.

Usage

```
printLOC(CONOP, events = "observed")
composite.age(CONOP, agemodels, tie.points, age.span, pch, bg)
LOC.tiepoints(CONOP, saveLOC = FALSE)
```

Arguments

CONOP	An object of class CONOP, typically the output of function read.CONOP .
events	A character string. Can be observed, in which case the events are plotted at the depth (or height) they have been observed, or placed, in which case they are plotted at their optimal depth (or height).
agemodels	A list of known age models with one element for each site (in the order defined in sectfile). Each element is a Two-column matrix: first column is the age value (in Ma) and the second column is the depth (in mbsf). The user can provide existing age models for any number of sites that he want (max is the number of site included in sectfile). Sites that are included in sectfile but for which no age models is given needs to be NA. E. g. two sites are present in sectfile, the first site is not provided with an age models but the second site have two tie points (10mbsf -> 1Ma and 50mbsf ->3Ma), agemodels should therefore be as follow: <code>[[1]]NA [[2]][,1]10 50 [,2]1 3</code>
tie.points	Output from function LOC.tiepoints

age.span	Vector of two age values (oldest, youngest) that limit the plots.
pch	See par .
bg	See par .
saveLOC	If TRUE, plots and LOC are saved in the working directory.

Details

- `LOC.tiepoints`: Prompt the user to interactively draw a line of correlation on depth vs composite depth diagram and return the set of tie points thus selected. Press ESC when the line is drawn to move on to the next section.
- `printLOC`: Plot and print depth vs composite depth (or age) diagram.
- `composite.age`: From a given set of tie points, translate COMPOSITE from composite depth to numerical age values.

Value

The output of `LOC.tiepoints` is a list where each element corresponds to a site (the index is the same as in `sectfile`). In each element, an age-depth matrix is stored. The output of `codecomposite.age` is an object of class `CONOP` similar to the input with numerical age value instead of composite depth.

Author(s)

Johan Renaudie.

Examples

```
data(riley)
printLOC(riley, events="placed")
```

compact.ordinal	<i>Compact the ordinal composite sequence.</i>
-----------------	--

Description

Take the ordinal composite sequence of events and compact it so that events that are not differentiable temporally are projected to the same level.

Usage

```
compact.ordinal(CONOP)
```

Arguments

CONOP An object of class `CONOP`, typically the output of function [read.CONOP](#).

Value

The modified object of class `CONOP`.

Author(s)

Johan Renaudie.

lofo

*Bioevents***Description**

Find the FO and LO for each species of the dataset on the studied site.

Usage

```
lofo(site, mbsf, remove.na = TRUE)
```

Arguments

site	Sample by Taxa matrix (samples as rows, taxa as columns)
mbsf	Vector of sample depth (or height, or age if known).
remove.na	If the species is contained in the last sample or in the first sample, remove.na=TRUE will output the depth of the last or the first sample as its LO or FO.

Value

Output is a matrix with one row for each taxon concerned. Two columns: LO (Last Occurrence) and FO (First Occurrence).

Author(s)

Johan Renaudie.

pacman

*Pacman profiling and trimming***Description**

Flag and trim anomalous occurrences given a certain percentage on an abundance matrix

Usage

```
pacman(neptune, perc_top = 5, perc_bottom = 3)
```

Arguments

ab_mat	An abundance matrix with samples as rows and taxa as columns.
perc_top	Percentage to trim at the top of the ranges, default to 5%
perc_bottom	Percentage to trim at the bottom of the ranges, default to 3%

Value

The returned value is a list:

Trimming	Trimmed Matrix
Profiling	Vector: number of outliers in each sample.

Author(s)

Lazarus et al. 2012 for the algorithm, Johan Renaudie for the code.

References

Lazarus et al. 2012. Pacman profiling: a simple procedure to identify stratigraphic outliers in high-density deep-sea microfossil data. *Paleobiology*, 38(1): 858-875.

prep.conop	<i>Prepares CONOP9 input file.</i>
------------	------------------------------------

Description

Functions to prepare CONOP input files referred to as eventfile and loadfile. Function prep.weight allow the user to modify loadfile to include weights (to deal with outliers).

Usage

```
prep.conop(Input, lofo, relaxed = FALSE, paired = TRUE)
prep.eventfile(Input, taxa_list)
prep.weight(Input, method = "badspecies", bad, weight.bad = "0.50",
            inconsistent, weight.inconsistent = "0.50",
            paclist, abmat, mbsf, pacman.sample.ratio = 0.1, weight.pacman = "0.50")
```

Arguments

Input	Object of class CONOPinput to modify.
lofo	Output of function lofo .
relaxed	If FALSE, FO will be able to move down a section but not up and LO inversely. If TRUE, they will both be able to move up or down a section.
paired	If TRUE, FO and LO are paired events.
taxa_list	Vector of taxa names to be used in the analysis (for non-bioevents, it will need to be modified later manually or with other functions such as prep.magn)
method	prep.weight should be used one method at a time. The methods are as follows: badspecies, badevent, inconsistent and pacman. For inconsistent and badspecies, the function weights down species that are thought to be diachronic (bad and weight.bad needed) or with inconsistent ranges (inconsistent and weight.inconsistent needed). For badevent, the function weights down an a priori diachronic event (bad and weight.bad needed). For pacman (paclist,), the function weights down a sample according to a pacman profiling (Lazarus et al. 2012).
bad	For badspecies method, a vector of species name to weight down. For badevent method, a two-column matrix: first column is the name of the species for which the FO will be weight down, second column is the name of the species for which the LO is desired to be weighted down.
weight.bad	Weight (from 0 to 1) for badspecies and badevent method.
inconsistent	For inconsistent, a vector of species names to be weighted down.

<code>weight.inconsistent</code>	Weight (from 0 to 1) for inconsitent method.
<code>paclist</code>	Output of function <code>pacman</code> .
<code>abmat</code>	For <code>pacman</code> method, same matrix as used to create <code>paclist</code>
<code>mbsf</code>	Vector of sample depth (same order as the samples in <code>abmat</code>).
<code>pacman.sample.ratio</code>	Ratio of anomalous occurrences per sample above which the function is desired to weight down the sample (numeric value from 0 to 1).
<code>weight.pacman</code>	Weight (from 0 to 1) for <code>pacman</code> method.

Author(s)

Johan Renaudie.

References

Lazarus et al. 2012. Pacman profiling: a simple procedure to identify stratigraphic outliers in high-density deep-sea microfossil data. *Paleobiology*, 38(1): 858-875. Sadler, 2007. CONOP version 7.43

See Also

[CONOPinput-class](#) and [read.config](#).

Examples

```
#Here is an example routine to prepare files for CONOP9
example <- new("CONOPinput") #Create an empty container
setwd("~/path/to/datasets/") #Set the wd to the folder containing the datasets to process
sites <- c("Morgan Creek","Potomac",...) #vector of section names
files <- c("morgan.csv","potomac.csv",...) #vector of dataset filenames
sectfile <- data.frame(seq_along(sites), substring(sites,1,3), seq_along(sites), #Create the SectFile
as.character(sites), rep(1,length(sites)), stringsAsFactors=FALSE)
colnames(sectfile) <- 1:5
example@Sectfile <- sectfile
example@Config <- read.config("/path/to/conop9.cfg") #No function to create the config file yet, so need to write it
l <- list() #will contains the LOFO table of each section
tax <- c() #will contains the unique, sorted taxa names
for(i in seq_along(files)){
  temp <- read.table(files[i], sep=",", header=TRUE, check.names=FALSE, row.names=1)
  l[[i]] <- lofo(temp[, -1], temp[, 1]) #Assuming the datasets are samples (rows) vs taxa (columns) with column 1 being the sample name
  tax <- sort(unique(c(rownames(l[[i]]), tax)))
}
example <- prep.eventfile(example, tax)
example <- prep.conop(example, l, relaxed=FALSE, paired=TRUE)
write(example)
```

read.CONOP	<i>Read the output of CONOP9.</i>
------------	-----------------------------------

Description

read.CONOP reads all output of CONOP9 and create an object of class CONOP. read.config is a function to read the configuration file (both input and output). read.cpchrt is a function to read the CONOP9 additional output file cpchrt. Needed for [culling](#).

Usage

```
read.CONOP(dir)
read.config(file = file.choose())
read.cpchrt(file = file.choose())
```

Arguments

dir	Directory of the CONOP9 run to input.
file	For read.config, configuration file (usually CONOP9.CFG); for read.copchrt, composite ranges file (usually cpcht.txt)

Value

An object of class CONOP for read.CONOP, a list for read.config and a matrix, equivalent to that found on the file for read.cpchrt.

Author(s)

Johan Renaudie.

References

Sadler, 2007. CONOP version 7.43.

See Also

[CONOP-class](#).

Examples

```
download.file("http://geology.usf.edu/faculty/data/harries/CD_2007.zip", "CONOP9.zip")
unzip("CONOP9.zip")
riley1 <- read.CONOP("CD_2007/SAMPLES/RILEY/Augmented")

#read.CONOP includes read.config. Here is an example for read.config alone.

riley.config <- read.config("CD_2007/SAMPLES/RILEY/Augmented/CONOP9.CFG")
```

riley

Riley Cambrian trilobite dataset

Description

A classic dataset for biostratigraphy from Palmer (1954). Cambrian trilobites from the Riley formation (7 sections, 62 taxa) augmented with a small number of hypothetical mid-range events and unpaired range-end events. Data set included in CONOP9 CD.

Usage

```
data(riley)
```

Details

An object of class 'CONOP'.

Source

Palmer, 1954. The faunas of the Riley Formation in Central Texas. *Journal of Paleontology*, 28: 709-786.

References

Sadler, 2007. CONOP version 7.43.

Examples

```
data(riley)
```

site2composite

From site depth to composite depth.

Description

Translate the depth of a sample in its 'depth' on the composite sequence.

Usage

```
site2composite(CONOP, mbsf, section)
```

Arguments

CONOP	An object of class CONOP, typically the output of function read.CONOP .
mbsf	Vector of depths to translate.
section	Abridged name of the section (should correspond to one of the elements of column 2 of SECTFILE).

Value

Vector of composite depths.

Author(s)

Johan Renaudie.

Index

*Topic **IO**

prep.conop, 8
read.CONOP, 10

*Topic **classes**

CONOP-class, 1
CONOPinput-class, 3

*Topic **datasets**

riley, 11

*Topic **misc**

compact.ordinal, 6
Graphs, 4
LOC, 5
lofo, 7
pacman, 7
site2composite, 11

*Topic **package**

CONOP9companion-package, 2

collection.quality, 2
collection.quality (Graphs), 4
compact.ordinal, 2, 6
composite.age, 2
composite.age (LOC), 5
CONOP-class, 1
CONOP9companion
 (CONOP9companion-package), 2
CONOP9companion-package, 2
CONOPinput-class, 3
culling, 2, 10
culling (Graphs), 4

draw.ranges, 2
draw.ranges (Graphs), 4

fence, 2
fence (Graphs), 4

Graphs, 4

LOC, 5
LOC.tiepoints (LOC), 5
lofo, 2, 7, 8

pacman, 2, 7
par, 6

prep.CONOP, 3
prep.conop, 2, 8
prep.eventfile (prep.conop), 8
prep.levelfile, 3
prep.weight (prep.conop), 8
print, CONOP-method (CONOP-class), 1
printLOC, 2
printLOC (LOC), 5

read.config, 9
read.config (read.CONOP), 10
read.CONOP, 1, 2, 4–6, 10, 11
read.cpchrt (read.CONOP), 10
riley, 11

sect.range (Graphs), 4
sect.ranges, 2
Section.vs.Composite (Graphs), 4
Section.vs.composite, 2
site2composite, 2, 11

write, CONOPinput-method
 (CONOPinput-class), 3