Package 'BoSSA'

May 6, 2026

Type Package

Title A Bunch of Structure and Sequence Analysis

Version 3.7

Date 2020-10-16

Author Pierre Lefeuvre

Maintainer Pierre Lefeuvre <pierre.lefeuvre@cirad.fr>

Depends R (>= 3.3.0)

Imports ape, RSQLite, jsonlite, phangorn, plotrix

Suggests prettydoc, knitr, rmarkdown, XML, rentrez, httr

VignetteBuilder knitr

Description Reads and plots phylogenetic placements.

License GPL

NeedsCompilation no

Repository CRAN

Date/Publication 2020-10-20 07:20:05 UTC

Contents

BoSSA-package *A Bunch of Structure and Sequence Analysis*

Description

Reads and plots phylogenetic placements.

Details

The DESCRIPTION file:

Package:	BoSSA
Type:	Package
Title:	A Bunch of Structure and Sequence Analysis
Version:	3.7
Date:	2020-10-16
Author:	Pierre Lefeuvre
Maintainer:	Pierre Lefeuvre <pierre.lefeuvre@cirad.fr>
Depends:	R (>= 3.3.0)
Imports:	ape, RSQLite, jsonlite, phangorn, plotrix
Suggests:	prettydoc, knitr, rmarkdown, XML, rentrez, httr
VignetteBuilder:	knitr
Description:	Reads and plots phylogenetic placements.
License:	GPL

Index of help topics:

BoSSA-package	A Bunch of Structure and Sequence Analysis
circular_tree	Plot an inside-out circular tree
plot.pplace	Plot a pplace or jplace object
pplace	A placement object as obtained with the read_sqlite function
pplace_to_matrix	Pplace to contingency matrix
pplace_to_table	Merge the multiclass and the placement table of pplace object
pplace_to_taxonomy	Convert a pplace object to a taxonomy table
print.pplace	Compact display of pplace and jplace objects
print.protdb	Compact display of protdb object
read_jplace	Read a jplace file
read_protdb	Read Protein Data Bank (PDB) file
read_sqlite	Read a pplacer/guppy sqlite file
refpkg	Summary data and plots for reference packages
sub_pplace	Subsets a pplace object
write_jplace	Write a jplace or pplace object to the disk

BoSSA contains functions to read and plot phylogenetic placement files obtained using softwares such as pplacer, guppy, EPA and RAPPAS.

Author(s)

Pierre Lefeuvre Maintainer: Pierre Lefeuvre <pierre.lefeuvre@cirad.fr>

References

- pplacer and guppy http://matsen.fhcrc.org/pplacer/ http://matsen.github.io/pplacer/ - EPA https://sco.h-its.org/exelixis/web/software/epa/index.html - RAPPAS https://github.com/benclaff/RAPPAS - Common file format http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0031009

---

circular_tree *Plot an inside-out circular tree*

Description

Plot a tree in a circular manner with the tips pointing inward

Usage

```
circular_tree(phy,ratio=0.5,def=1000,pos_out=FALSE,tip_labels=TRUE,cex_tips=0.5)
```

Arguments

phy	a class phylo object
ratio	the ratio of the tree size compared to the plot size
def	the def parameter controls the granularity of the curves
pos_out	a matrix with the x and y coordinates of the branches extremities (i.e. nodes and tips) is outputed when set to TRUE
tip_labels	whether or not the tiplabels should be plotted
cex_tips	the size of the tiplabels

Details

The function plot a tree in a circular manner. Note that the tree will produce a correct output only if there is no topology modifications after reading the original tree using the ape read.tree function.

Value

a plot

Author(s)

pierre lefeuvre

Examples

```
library(ape)

test_tree <- rtree(20)

circular_tree(test_tree)
```

`plot.pplace` *Plot a pplace or jplace object*

Description

Plot the tree and placements from a pplace or a jplace object

Usage

```
## S3 method for class 'pplace'
plot(x,type="precise",simplify=FALSE,
main="",N=NULL,transfo=NULL,legend=TRUE,stl=FALSE,
asb=FALSE,edge.width=1,max_width=10,cex.number=0.5,
cex.text=0.8,transp=80,add=FALSE,color=NULL,discrete_col=FALSE,
pch=16,run_id=NULL, ...)
```

Arguments

`x`	A pplace or jplace object
`type`	The type of ploting desired with either, "precise", "color", "fattree" or "number". For each option, placement sizes represent the multiplication of the N value with the placement ML ratio.
`simplify`	If set to TRUE, only plot the best position for each placement. default is FALSE.
`main`	An optionnal title to plot along the tree
`N`	An optionnal vector of the weight of each placement. Must be of the same length and order as placements in the multiclass table. Note that the placement mass (potentially) available from the original files are imported into R but aren't use in the analysis. The N parameter should be used instead.
`transfo`	An optionnal function to transform the placement size when type set to "precise". Beware that it is also applied to the legend so that it does not anymore correspond to the placement size but to the transform dot size
`legend`	Plot a legend. Not available for type "number" or "fattree"
`stl`	Show tip labels
`asb`	Add scale bar
`edge.width`	The tree edge width
`max_width`	The maximum edge width when type is set to "fattree"

cex.number	Control the size of the text when type is set to "number"
cex.text	Control the size of the main
transp	Control the transparency of the placement when type is "precise" and the transparency of the branch without placement when type is set to "color". Encoded in hexadecimal scale (i.e. range from "00" to "FF")
add	Add placement to an existing plot when type is set to precise. Default is FALSE. If it was drawn, the legend won't be updated. Beware to use the same value for the "transfo" option in each plot. Dots color scale won't be accurate when using the "add" option. It is highly recommanded to use a single color.
color	The colors used for pendant branch length scale when type is set to "precise". Default is a color ramp with "blue", "green", "yellow" and "red"
discrete_col	Discretise the color scale for pendant branch length
pch	The dot style used for placements when type is set to "precise"
run_id	A vector of run_id to subset
...	Further arguments passed to or from other methods.

Author(s)

pierre lefeuvre

Examples

```
data(pplace)

### number type
plot(pplace,type="number",main="number")

### color type without and with legend
plot(pplace,type="color",main="color without legend",legend=FALSE)
plot(pplace,type="color",main="color with legend",legend=TRUE)

### fattree type
plot(pplace,type="fattree",main="fattree")

### precise type
plot(pplace,type="precise",main="precise vanilla")
plot(pplace,type="precise",simplify=TRUE,main="precise simplify")

# using the read number information encoded here in the name (if available)
Npplace <- sample(1:100,nrow(pplace$multiclass),replace=TRUE)
# in the following exemple, the dots are too large...
plot(pplace,type="precise",main="precise N",legend=TRUE,N=Npplace,simplify=TRUE)

# using the transfo option to modify dot sizes
# note that placements sizes inferior to 1 won't
# behave properly with log10 as a transformation function.
# In this case, you rather use simplify (all the placement
# will corresponds to at least one sequence).
# Beware that when using the transfo option,
```

```
# the legend does not anymore correspond to the actual placement
# size but to the transform placement size
# (i.e. the transform function applied to the dot size).
# we will use the the log10 function
plot(pplace,type="precise",main="precise log10",
legend=TRUE,N=Npplace,transfo=log10)
# or without simplify, you can use a custom function
# as transfo that will produce positive sized dots
plot(pplace,type="precise",main="precise custom"
,legend=TRUE,N=Npplace,transfo=function(X){log10(X+1)})
```

---

pplace *A placement object as obtained with the read_sqlite function*

Description

A placement object as obtained with the read_sqlite function. In this example, a set of 100 sequence reads are placed over a 16S phylogeny. This example is a subset of those available for download at http://fhcrc.github.io/microbiome-demo/

Usage

```
data("pplace")
```

References

http://fhcrc.github.io/microbiome-demo/

Examples

```
data(pplace)
str(pplace)
```

---

pplace_to_matrix *Pplace to contingency matrix*

Description

Convert the pplace object into a contingency matrix OTUs / sample

Usage

```
pplace_to_matrix(pplace, sample_info, N = NULL, tax_name = FALSE
 ,run_id=NULL,round_type=NULL)
```

Arguments

pplace	A pplace object
sample_info	A vector or list specifying the association between placement (in the multiclass table) and sample. In the case of a list, multiple sample can be associated with a single placement.
N	An optionnal vector or list with a number of occurence (or weight) associated to each placed sequence. If "sample_info" is a list, "N" must also be a list. Note that the placement mass (potentially) available from the original files are imported into R but aren't use in the analysis. The N parameter should be used instead.
tax_name	Either the tax ids (when set to FALSE, default) or the tax names (when set to TRUE) are used as column names. The tax names are obtained form the "taxo" table of the pplace object.
run_id	A vector of run_id to subset
round_type	The name of the rounding fonction to apply to the product of the number of individuals classified in a given category and the likelihood ratio of this classification. Should be set to NULL (no rounding) or one of "trunc", "round", "ceiling" or "floor".

Value

A contingency matrix with OTUs / species in rows and samples in columns.

Author(s)

pierre lefeuvre

Examples

```
data(pplace)

### simple example
pplace_to_matrix(pplace,c(rep("sample1",27),rep("sample2",50),rep("sample3",23)))

### using the N option to specify the number of sequence each placement represents
Npplace <- sample(1:20,100,replace=TRUE)
pplace_to_matrix(pplace,c(rep("sample1",27),rep("sample2",50),rep("sample3",23)),N=Npplace)

### with tax_name=TRUE
pplace_to_matrix(pplace,c(rep("sample1",27),rep("sample2",50),rep("sample3",23)),tax_name=TRUE)
```

pplace_to_table *Merge the multiclass and the placement table of pplace object*

Description

Merge the multiclass and the placement table of pplace object

Usage

```
pplace_to_table(pplace, type = "full",run_id=NULL)
```

Arguments

pplace	a pplace object
type	the placement type to consider
run_id	A vector of run_id to subset

Details

For the type argument, either "full" or "best" are accepted. Whereas for the "full" type, all the placements are considered, only the best placement for each sequence is considered for the "best" type.

Value

a data frame with the same column names as the mutliclass and placements tables

Author(s)

pierre lefeuvre

Examples

```
data(pplace)

### with every placement
pplace_to_table(pplace)

### keeping only the best placement for each sequence
pplace_to_table(pplace,type="best")
```

pplace_to_taxonomy *Convert a pplace object to a taxonomy table*

Description

Convert a pplace object to a taxonomy table

Usage

```
pplace_to_taxonomy(pplace,taxonomy,
rank=c("phylum","class","order","family","genus","species"),
type="all",tax_name=TRUE,run_id=NULL)
```

Arguments

pplace	A pplace object
taxonomy	The taxonomy table as obtained using the refseq fonction with type set to taxonomy
rank	The desired rank for the taxonomy table
type	Wether all the possible classification available in the multiclass table are outputed (type="all") or only the best (type="best")
tax_name	Wether to use taxonomy names (default) or tax_id number
run_id	A vector of run_id to subset

Value

A matrix with taxonomic ranks for each sequence

Author(s)

pierre lefeuvre

print.pplace *Compact display of pplace and jplace objects*

Description

Compact display of pplace and jplace objects

Usage

```
## S3 method for class 'pplace'
print(x, ...)
```

Arguments

x	a pplace or jplace object
...	further arguments passed to or from other methods

Author(s)

pierre lefeuvre

Examples

```
data(pplace)
print(pplace)
```

---

print.protdb *Compact display of protdb object*

Description

Function to print the header section of the protdb object.

Usage

```
## S3 method for class 'protdb'
print(x, ...)
```

Arguments

x	a protdb class object
...	further arguments passed to or from other methods

Author(s)

pierre lefeuvre

Examples

```
pdb_file <- system.file("extdata", "1L2M.pdb", package = "BoSSA")
pdb <- read_protdb(pdb_file)
print(pdb)
```

`read_jplace` *Read a jplace file*

Description

Read a jplace file

Usage

```
read_jplace(jplace_file, full = TRUE)
```

Arguments

`jplace_file`	A jplace file name
`full`	If set to FALSE, only the tree is read from the jplace file

Details

When the jplace or sqlite files are imported into R, the node numbering available in the original file is converted to the class "phylo" numbering. The class phylo is defined in the "ape" package.

Value

A list with

`arbre`	The tree in class "phylo" over wich placements are performed
`placement`	The placement table
`multiclass`	The multiclass table
`run`	The command line used to obtained the jplace file

Author(s)

pierre lefeuvre

See Also

read_sqlite

read_protdb *Read Protein Data Bank (PDB) file*

Description

Read Protein Data Bank (PDB) file

Usage

```
read_protdb(X)
```

Arguments

X	The path/name of a pdb file.

Value

The output is a list of objects

header	The header of the pdb file
compound	A data frame summarizing the CMPND part of the pdb file. This include the molecule ID, the molecule name and the chain ID
atom	A data frame with the atom type, the amino acid, the amino acid number, the chain and the euclidian X, Y, Z coordinates of the atoms
sequence	A list with the numbering of the amino acid and the amino acid sequence for each chain

Author(s)

pierre lefeuvre

References

http://www.rcsb.org/pdb/home/home.do

Examples

```
pdb_file <- system.file("extdata", "1L2M.pdb", package = "BoSSA")
pdb <- read_protdb(pdb_file)
pdb
```

read_sqlite *Read a pplacer/guppy sqlite file*

Description

Read a pplacer/guppy sqlite file

Usage

```
read_sqlite(sqlite_file,jplace_file=gsub("sqlite$","jplace",sqlite_file),
rank="species")
```

Arguments

sqlite_file	A pplacer/guppy sqlite path/file name
jplace_file	An optionnal jplace file name. By default, the sqlite file name with the suffix changed from "sqlite" to jplace" is used. If different, the jplace path/name must be specified.
rank	The desired taxonomic assignation rank to extract. default is "species".

Details

As the tree informations are not available in the sqlite file, the jplace file is also required. When the jplace or sqlite files are import into R, the node numbering available in the original file is converted to the class "phylo" numbering.

Value

A list with

runs	The command line used to obtained the sqlite file
taxa	The taxonomic information table
multiclass	The multiclass table
placement_positions	A data frame with the posiotn of each placement in the reference tree
arbre	The tree in class "phylo" over wich placements are performed
edge_key	A matrix with correspondance of node numbering between the original tree in the jplace file and the class phylo tree of the "arbre" component
original_tree	The tree string from the jplace file

For details on the other components (i.e. "placements, "placement_classifications", "placement_evidence", "placement_median_identities", "placement_names", "placement_nbc", "placements", "ranks" and "sqlite_sequence", please, refer to http://erick.matsen.org/pplacer/generated_rst/guppy_classify.html)

Author(s)

pierre lefeuvre

References

http://erick.matsen.org/pplacer/generated_rst/guppy_classify.html

Examples

```
### the path to the sqlite and jplace files
sqlite_file <- system.file("extdata", "example.sqlite", package = "BoSSA")
jplace_file <- system.file("extdata", "example.jplace", package = "BoSSA")
pplace <- read_sqlite(sqlite_file,jplace_file)
```

`refpkg` *Summary data and plots for reference packages*

Description

Summary data and plots for reference packages

Usage

```
refpkg(refpkg_path,type="summary",rank_tree="species",
rank_pie=c("phylum","class","order","family","genus"),
scale_pie=TRUE,alpha_order=TRUE,cex.text=0.7,
cex.legend=1,asb=TRUE,rotate_label=TRUE,
out_krona="for_krona.txt",text2krona=NULL)
```

Arguments

`refpkg_path`	The path of the reference package directory
`type`	The type of summary to perform with "summary", "taxonomy", "info", "tree", "pie" or "krona" available
`rank_tree`	The desired rank for tree coloring
`rank_pie`	The ranks to be plot for the taxonomy pie chart
`scale_pie`	Wether or not to take into account the number of sequences available within the reference package for the pie chart
`alpha_order`	Wether or not the color should follow taxa alpahabetic order when type set to "tree"
`cex.text`	The tip labels cex parameter when type is set to "tree" and the text cex parameter when type is set to "pie"
`cex.legend`	The size of the legend when type set to "tree"
`asb`	Add a scale bar on the tree
`rotate_label`	Rotates the pie slice labels
`out_krona`	The name of the output file when type is set to "krona".
`text2krona`	The full path to the krona "ImportText.pl" script when KronaTools is installed and you wish to directly produce the html krona file.

Value

A summary print on screen when type set to "summary". A data frame when type set to "taxonomy" or "info". A file written to the disk when type is set to "krona". A plot otherwise.

Author(s)

pierre lefeuvre

References

https://github.com/marbl/Krona/wiki/KronaTools http://fhcrc.github.io/taxtastic/

Examples

```
refpkg_path <- paste(find.package("BoSSA"),"/extdata/example.refpkg",sep="")

### summary
refpkg(refpkg_path)

### taxonomy
taxonomy <- refpkg(refpkg_path,type="taxonomy")
head(taxonomy)

### info
refpkg(refpkg_path,type="info")

### tree
refpkg(refpkg_path,type="tree",rank_tree="order",cex.text=0.5)

### pie
refpkg(refpkg_path,type="pie",rank_pie=c("class","order","family"),cex.text=0.6)

### krona
# it will produce a flat text file
# this file can be use as input for the the "ImportText.pl" krona script
# see https://github.com/marbl/Krona/wiki/KronaTools for more details on krona
## Not run:
refpkg(refpkg_path,type="krona",out_krona="for_krona.txt")

## End(Not run)
```

---

sub_pplace *Subsets a pplace object*

Description

Subsets a pplace or jplace object based on the placement_id, the name of the placement or a regular expression of the name of the placement

Usage

```
sub_pplace(x, placement_id = NULL, ech_id = NULL, ech_regexp = NULL, run_id = NULL)
```

Arguments

x	The pplace or jplace object to subset
placement_id	A vector of the placement_id to subset
ech_id	A vector of the names of the placement to subset
ech_regexp	A regular expression of the name of the placement to subset
run_id	A vector of run_id to subset

Details

When using placement_id, the subset is performed based on the placement_id column of the multiclass, placements, placement_positions, placement_names, placement_classifications, placement_evidence, placement_median_identities and placement_nbc data frames. When using ech_id and ech_regexp, the subset is performed from the multiclass$name column. When using run_id, the subset is performed based on the placements$run_id column.

Value

A pplace object

Author(s)

pierre lefeuvre

Examples

```
data(pplace)

### subsetting using placement ids. Here placements 1 to 5
sub1 <- sub_pplace(pplace,placement_id=1:5)
sub1

### subsetting using sequenes ids
id <- c("GWZHISEQ01:514:HMCLFBCXX:2:1108:1739:60356_90",
"GWZHISEQ01:514:HMCLFBCXX:2:1114:13665:31277_80")
sub2 <- sub_pplace(pplace,ech_id=id)
sub2

### subsetting using a regular expression of sequence ids
sub3 <- sub_pplace(pplace,ech_regexp="^HWI")
sub3
```

`write_jplace` *Write a jplace or pplace object to the disk*

Description

Write a jplace or pplace object to the disk in the jplace JSON format

Usage

```
write_jplace(x,outfile)
```

Arguments

`x`	A pplace or jplace object
`outfile`	The name of the output file

Note

Note that the placement mass (potentially) available from the original files are imported into R but aren't use in the analysis. Anyway, the write_jplace function takes into account possible weight/mass information available in the the "nm" column of the multiclass table for jplace objects and in the "mass" column from the placement_names table for the pplace objects. The values in these column can be edited before writing the jplace file if one want to use distinct mass/weight in downtstream analysis (e.g. using the guppy program functionalities).

Author(s)

pierre lefeuvre

Examples

```
data(pplace)
## Not run:
write_jplace(pplace,"test.jplace")

## End(Not run)
```

Index