Package ‘BoSSA’

May 9, 2017

Type Package
Title A Bunch of Structure and Sequence Analysis
Version 2.1
Date 2017-05-03
Author pierre lefeuvre
Maintainer pierre lefeuvre <pierre.lefeuvre@cirad.fr>
Depends R (>= 3.3.1)
Imports ape, RSQLite, jsonlite, phangorn, plotrix
Suggests BiocStyle, knitr, rmarkdown, XML, rentrez, phyloseq
VignetteBuilder knitr
Description Reads and plots phylogenetic placements obtained using the 'pplacer' and 'guppy' soft-
License GPL
NeedsCompilation no
Repository CRAN
Date/Publication 2017-05-09 21:26:36 UTC

R topics documented:

BoSSA-package ................................................................. 2
plot.pplace ................................................................. 3
pplace ................................................................. 5
pplace_to_matrix ....................................................... 5
pplace_to_table ....................................................... 6
pplace_to_taxonomy .................................................. 7
print.pplace ............................................................. 8
print.protdb ............................................................ 8
read_protdb ............................................................ 9
read_sqlite ............................................................... 10
refpkg ................................................................. 11
sub_pplace ............................................................. 12

Index 14
**BoSSA-package**

---

**Description**

Reads and plots phylogenetic placements obtained using the 'pplacer' and 'guppy' softwares [https://matsen.github.io/pplacer/].

**Details**

The DESCRIPTION file:

- **Package:** BoSSA
- **Type:** Package
- **Title:** A Bunch of Structure and Sequence Analysis
- **Version:** 2.1
- **Date:** 2017-05-03
- **Author:** pierre lefeuvre
- **Maintainer:** pierre lefeuvre &lt;pierre.lefeuvre@cirad.fr&gt;
- **Depends:** R (&ge; 3.3.1)
- **Imports:** ape, RSQLite, jsonlite, phangorn, plotrix
- **Suggests:** BiocStyle, knitr, rmarkdown, XML, rentrez, phyloseq
- **VignetteBuilder:** knitr
- **Description:** Reads and plots phylogenetic placements obtained using the 'pplacer' and 'guppy' softwares [https://matsen.github.io/pplacer/].
- **License:** GPL

**Index of help topics:**

- BoSSA-package
- plot.pplace
- pplace
- pplace_to_matrix
- pplace_to_table
- pplace_to_taxonomy
- print.pplace
- print.protdb
- read_protdb
- read_sqlite
- refpkg
- sub_pplace

BoSSA offers functions to read and plot phylogenetic placement files obtained using pplacer and guppy. It works using the sqlite output from guppy.placements can be plotted in different fashion. Previously BoSSA integrate NCBI BLAST functions and access to NCBI CGI through R. All these functions have been removed. For BLAST, a good alternative is to use biopython. Although it’s not
R, the biopython cookbook gives really good examples making it quite easy to use. For access to the NCBI CGI, other packages such as "rentrez" or "reutils" are really well done and very handy.

Author(s)
	pierre lefeuvre Maintainer: pierre lefeuvre <pierre.lefeuvre@cirad.fr>

References

- protein data bank http://www.rcsb.org/pdb/home/home.do

---

plot.pplace  
Plot a pplace object

Description

Plot the tree and placements from a pplace object

Usage

```r
## 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, cex.number=0.5, cex.text=0.8,
     transp=80, add=FALSE, color=NULL, pch=16, ...)
```

Arguments

- **x**: A pplace object
- **type**: The type of plotting desired with either, "precise", "color" 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 optional title to plot along the tree
- **N**: An optional vector with a number of occurrence associated to each placed sequence
- **transfo**: An optional function to transform the placement size when type set to "precise". Beware that it is also applied to the legend text so that it does not anymore correspond to the placement size but to the transform dot size
- **legend**: Plot a legend
- **stl**: Show tip labels
- **asb**: Add scale bar
- **edge.width**: The tree edge width
cex.number  Control the size of the number 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 assignment when type is set to "color"
add        Add placement to an existing plot when type is set to precise. default is FALSE
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"
pch        The dot style used for placements when type is set to "precise"
...

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)

### 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 <- as.numeric(gsub("_*","",pplace$multiclass$name))
# 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 17 sequence reads are placed over a mastrevirus phylogeny. At the end of the read name, a number indicates the number of reads it represents (i.e. these reads are centroids of read clusters. the number indicate the size of the cluster).

**Usage**

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

**Examples**

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

**pplace_to_matrix**

*Pplace to contingency matrix*

**Description**

Convert the pplace object into a contingency matrix OTUs / sample

**Usage**

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

**Arguments**

- `pplace`: A pplace object
- `sample_info`: A vector specifying the association between placement (in the multiclass table) and sample
- `N`: An optionnal vector with a number of occurence associated to each placed sequence
- `tax_name`: Either the tax ids (default) or the tax names are used as column names. the tax names are obtained form the "taxo" table of the pplace object

**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",7),rep("sample2",10)))

### using the N option to specify the number of sequence each placement represents
Npplace <- sample(1:20,17,replace=TRUE)
pplace_to_matrix(pplace,c(rep("sample1",7),rep("sample2",10)),N=Npplace)

### with tax_name=TRUE
pplace_to_matrix(pplace,c(rep("sample1",7),rep("sample2",10)),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")
```

Arguments

- **pplace**: a pplace object
- **type**: the placement type to consider

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 placed sequence is considered for the "best" type.

Value

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

Author(s)
pierre lefeuvre
**pplace_to_taxonomy**

### Examples

```r
data(pplace)

### with every placement
pplace_to_table(pplace)

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

---

**Description**

Convert a pplace object to a taxonomy table

**Usage**

```r
pplace_to_taxonomy(pplace, taxonomy,
rank = c("phylum", "class", "order", "family", "genus", "species"),
tax_name = TRUE)
```

**Arguments**

- `pplace`: A pplace object
- `taxonomy`: The taxonomy table as obtained using the refseq function with type set to taxonomy
- `rank`: The desired rank for the taxonomy table
- `tax_name`: Whether to use taxonomy names (default) or tax_id number

**Value**

A matrix with taxonomic ranks for each sequence

**Author(s)**

pierre lefeuvre

**Examples**

```r
data(pplace)

### getting the taxonomy file
# taxonomy <- refpkg(find.package("BoSSA"), "/extdata/example.refpkg", type = "taxonomy")

# pplace_to_taxonomy(pplace, taxonomy)
```
print.pplace

Compact display of pplace object

Description

Compact display of pplace object

Usage

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

Arguments

- `x`: a pplace object
- `...`: further arguments passed to or from other methods

Author(s)

pierre lefeuvre

Examples

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

print.protdb

Compact display of protdb object

Description

Function to print the header section of the protdb object on the console.

Usage

```r
## 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
**read_protdb**

**Examples**

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

---

**Description**

Read Protein Data Bank (PDB) file

**Usage**

```r
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 atom
- `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**

```r
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**

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

**Arguments**

- `sqlite_file`: A pplacer/guppy sqlite file name
- `jplace_file`: An optionnal jplace file name. By default, a jplace file with the same prefix as the sqlite file is made. If not, the jplace file path/name as to be specified.
- `rank`: The desired taxonomic assignation rank to extract

**Details**
As the tree informations are not available in the sqlite file, the jplace file is also required.

**Value**
A list with

- `run`: The command line used to obtained the sqlite file
- `taxo`: The taxonomic information table
- `multiclass`: The multiclass table
- `placement`: The placement table
- `arbre`: The tree in class "phylo" over wich placements are performed

**Author(s)**
pierre lefeuvre

**Examples**

```r
### 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=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)

Arguments

- **refpkg_path**  The path of the reference package directory
- **type**  The type of summary to perform
- **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 to color should follows taxa alphabetic 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

Value

A summary print on screen when type set to "summary". A data frame when type set to "taxonomy". A plot otherwise.

Author(s)

pierre lefeuvre
Examples

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

### summary
cdev

### taxonomy
taxonomy <- cdev
head(taxonomy)

### tree
cdev

### pie
cdev
```

---

sub_pplace

**Subsets a pplace object**

**Description**

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

**Usage**

```r
sub_pplace(pplace, placement_id = NULL, ech_id = NULL, ech_regexp = NULL)
```

**Arguments**

- **pplace**
  - The pplace 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

**Details**

Both the multiclass and the placement table are subseted. All the other compartiments of the list are left untouched.

**Value**

A pplace object
Examples

data(pplace)

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

### subsetting using sequenes ids
sub2 <- sub_pplace(pplace,ech_id=id)

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

BoSSA (BoSSA-package), 2
BoSSA-package, 2

plot.pplace, 3
pplace, 5
pplace_to_matrix, 5
pplace_to_table, 6
pplace_to_taxonomy, 7
print.pplace, 8
print.protdb, 8

read_protdb, 9
read_sqlite, 10
refpkg, 11

sub_pplace, 12