# Package ‘primerTree’

February 20, 2015

**Version** 1.0.1  
**Date** 2013-8-13  
**Maintainer** Jim Hester <james.f. hester@gmail.com>  
**Author** Jim Hester  
**Title** primerTree  
**Description** Visually Assessing the Specificity and Informativeness of Primer Pairs.  
**Depends** R (>= 3.0.0), directlabels, gridExtra  
**Imports** ggplot2, grid, XML, ape, htr, lubridate, plyr, scales, stringr, foreach  
**LazyData** true  
**License** GPL-2  
**NeedsCompilation** yes  
**Repository** CRAN  
**Date/Publication** 2014-06-09 20:31:46

## R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>accession2gi</td>
<td>2</td>
</tr>
<tr>
<td>bryophytes_trnL</td>
<td>2</td>
</tr>
<tr>
<td>clustalo</td>
<td>3</td>
</tr>
<tr>
<td>get_sequence</td>
<td>3</td>
</tr>
<tr>
<td>get_sequences</td>
<td>4</td>
</tr>
<tr>
<td>get_taxonomy</td>
<td>4</td>
</tr>
<tr>
<td>gi2taxid</td>
<td>5</td>
</tr>
<tr>
<td>identify.primerTree_plot</td>
<td>5</td>
</tr>
<tr>
<td>layout_tree_ape</td>
<td>6</td>
</tr>
<tr>
<td>mammals_16S</td>
<td>6</td>
</tr>
<tr>
<td>plot.primerTree</td>
<td>6</td>
</tr>
<tr>
<td>plot_tree</td>
<td>7</td>
</tr>
<tr>
<td>plot_tree_ranks</td>
<td>8</td>
</tr>
<tr>
<td>primerTree</td>
<td>9</td>
</tr>
</tbody>
</table>
accession2gi

Maps a genbank accession to a nuclotide database gi.

Description

Maps a genbank accession to a nuclotide database gi.

Usage

accession2gi(accession)

Arguments

accession character vector to lookup.

Value

named vector of gis.

bryophytes_trnL

PrimerTree results for the bryophyte trnL primers

Description

PrimerTree results for the bryophyte trnL primers
**clustalo**  

*Multiple sequence alignment with clustal omega*

**Description**

Calls clustal omega to align a set of sequences of class DNAbin. Run without any arguments to see all the options you can pass to the command line clustal omega.

**Usage**

`clustalo(x, exec = "clustalo", quiet = TRUE, original.ordering = TRUE, ...)`

**Arguments**

- `x`: an object of class 'DNAbin'
- `exec`: a character string with the name or path to the program
- `quiet`: whether to suppress output to stderr or stdout
- `original.ordering`: use the original ordering of the sequences
- `...`: additional arguments passed to the command line clustalo

---

**get_sequence**  

*Retrieves a fasta sequence from NCBI nucleotide database.*

**Description**

Retrieves a fasta sequence from NCBI nucleotide database.

**Usage**

`get_sequence(gi, start = NULL, stop = NULL)`

**Arguments**

- `gi`: nucleotide gi to retrieve.
- `start`: start base to retrieve, numbered beginning at 1. If NULL the beginning of the sequence.
- `stop`: last base to retrieve, numbered beginning at 1. if NULL the end of the sequence.

**Value**

an DNAbin object.

**See Also**

DNAbin
get_sequences

Retrieves fasta sequences from NCBI nucleotide database.

Description

Retrieves fasta sequences from NCBI nucleotide database.

Usage

get_sequences(gi, start = NULL, stop = NULL, simplify = TRUE, .parallel = FALSE, .progress = "none")

Arguments

gi the gi number of the sequence to retrieve
start start bases to retrieve, numbered beginning at 1. If NULL the beginning of the sequence.
stop stop bases to retrieve, numbered beginning at 1. if NULL the stop of the sequence.
simplify simplify the FASTA headers to include only the genbank accession.
.parallel if 'TRUE', perform in parallel, using parallel backend provided by foreach
.progress name of the progress bar to use, see 'create_progress_bar'

Value

an DNAbin object.

See Also

DNAbin

get_taxonomy

Retrieve the taxonomy information from NCBI for a set of nucleotide gis.

Description

Retrieve the taxonomy information from NCBI for a set of nucleotide gis.

Usage

get_taxonomy(gis)
**gi2taxid**

**Arguments**

- **gis** a character vector of the gis to retrieve

**Value**

data.frame of the 'gis, taxIds, and taxonomy

---

Maps a nucleotide database gi to a taxonomy database taxId

**Description**

Maps a nucleotide database gi to a taxonomy database taxId

**Usage**

gi2taxid(gi)

**Arguments**

- **gi** gi character vector to lookup.

**Value**

named vector of taxIds.

---

identify.primerTree_plot

identify the point closest to the mouse click only works on single ranks

**Description**

identify the point closest to the mouse click only works on single ranks

**Usage**

```r
## S3 method for class 'primerTree_plot'
identify(x, ...)
```

**Arguments**

- **x** the plot to identify
- **...** additional arguments passed to annotate
layout_tree_ape  
\textit{layout a tree using ape, return an object to be plotted by plot.tree}

**Description**

layout a tree using ape, return an object to be plotted by \texttt{plot.tree}

**Usage**

\texttt{layout_tree_ape(tree, \ldots)}

**Arguments**

- \texttt{tree}  
The \texttt{phylo} tree to be plotted
- \texttt{\ldots}  
additional arguments to \texttt{plot.phylo}

**Value**

- \texttt{edge}  
list of x, y and xend, yend coordinates as well as ids for the edges
- \texttt{tips}  
list of x, y, label and id for the tips
- \texttt{nodes}  
list of x, y and id for the nodes

---

\texttt{mammals_16S}  
\textit{PrimerTree results for the mammalian 16S primers}

**Description**

PrimerTree results for the mammalian 16S primers

---

\texttt{plot.primerTree}  
\textit{plot function for a primerTree object, calls plot.tree.ranks}

**Description**

plot function for a primerTree object, calls plot.tree.ranks

**Usage**

\texttt{## S3 method for class 'primerTree'  
plot(x, ranks = NULL, main = NULL, \ldots)}
plot_tree

**Arguments**

- **x**: primerTree object to plot
- **ranks**: The ranks to include, defaults to all common ranks, if NULL print all ranks. If 'none' just print the layout.
- **main**: an optional title to display, if NULL displays the name as the title
- **...**: additional arguments passed to plot_tree_ranks

**See Also**

plot_tree_ranks, plot_tree

**Examples**

```r
library(gridExtra)
library(directlabels)

# plot with all common ranks
plot(mammals_16S)

# plot only the class
plot(mammals_16S, 'class')

# plot the layout only
plot(mammals_16S, 'none')
```

---

**plot_tree**

plots a tree, optionally with colored and labeled points by taxonomic rank

**Description**

plots a tree, optionally with colored and labeled points by taxonomic rank

**Usage**

```r
plot_tree(tree, type = "unrooted", main = NULL, guide_size = NULL,
          rank = NULL, taxonomy = NULL, size = 2, legend_cutoff = 25, ...)
```

**Arguments**

- **tree**: to be plotted, use layout_tree to layout tree.
- **taxonomy**: A data.frame with an accession field corresponding to the tree tip labels.
- **main**: An optional title for the plot
- **type**: The type of tree to plot, default unrooted.
- **rank**: The rank to include, if null only the tree is plotted
- **size**: The size of the colored points
plot_tree_ranks

Plots a tree along with a series of taxonomic ranks.

**Description**

Plots a tree along with a series of taxonomic ranks.

**Usage**

```r
plot_tree_ranks(tree, taxonomy, main = NULL, type = "unrooted",
                ranks = common_ranks, size = 2, guide_size = NULL, legend_cutoff = 25,
                ...)```

**Arguments**

- `ranks`: The ranks to include, defaults to all common ranks, if null print all ranks.
- `tree`: To be plotted, use `layout_tree` to layout tree.
- `taxonomy`: A data.frame with an accession field corresponding to the tree tip labels.
- `main`: An optional title for the plot.
- `type`: The type of tree to plot, default unrooted.
- `size`: The size of the colored points.
- `guide_size`: The size of the length guide. If NULL auto detects a reasonable size.
- `legend_cutoff`: The number of different taxa names after which the names are no longer printed.
- `...`: Additional arguments passed to `layout_tree_ape`.

**Value**

The plot to be printed.

**Examples**

```r
library(gridExtra)
library(directlabels)

# Plot all the common ranks
plot_tree_ranks(mammals_16S$tree, mammals_16S$taxonomy)

# Plot specific ranks, with a larger dot size
plot_tree_ranks(mammals_16S$tree, mammals_16S$taxonomy,
                ranks=c('kingdom', 'class', 'family'), size=3)
```
**primerTree**

**Visually Assessing the Specificity and Informativeness of Primer Pairs**

**Description**

primerTree has two main commands: `search_primer_pair` which takes a primer pair and returns a primerTree object of the search results. `plot.primerTree` a S3 method for plotting the primerTree object obtained using `search_primer_pair`.

**primer_search**

Query a pair of primers using ncbi's Primer-BLAST, if primers contain iupac ambiguity codes, enumerate all possible combinations and combine the results.

**Usage**

```r
primer_search(forward, reverse, num_aligns = 500, num_permutations = 25,
...,.parallel = FALSE,.progress = "none")
```

**Arguments**

- `forward`: forward primer to search by 5'-3' on plus strand
- `reverse`: reverse primer to search by 5'-3' on minus strand
- `num_aligns`: number of alignment results to keep
- `num_permutations`: the number of primer permutations to search, if the degenerate bases cause more than this number of permutations to exist, this number will be sampled from all possible permutations.
- `...`: additional arguments passed to Primer-Blast
- `.parallel`: if TRUE, perform in parallel, using parallel backend provided by foreach
- `.progress`: name of the progress bar to use, see `create_progress_bar`

**Value**

data.frame of primer hits
search_primer_pair

Automatic primer searching Search a given primer pair, retrieving the alignment results, their product sequences, the taxonomic information for the sequences, a multiple alignment of the products.

Description

Automatic primer searching Search a given primer pair, retrieving the alignment results, their product sequences, the taxonomic information for the sequences, a multiple alignment of the products.

Usage

search_primer_pair(forward, reverse, name = NULL, num_aligns = 500, num_permutations = 25, simplify = TRUE, clustal_options = list(), distance_options = list(model = "raw"), ..., parallel = FALSE, progress = "none")

Arguments

name name to give to the primer pair
simplify use simple names for primer hit results or complex
.progress name of the progress bar to use, see create_progress_bar
clustal_options a list of options to pass to clustal omega, see link{clustalo} for a list of options
distance_options a list of options to pass to dist.dna, see link{dist.dna} for a list of options
forward forward primer to search by 5'-3' on plus strand
reverse reverse primer to search by 5'-3' on minus strand
num_aligns number of alignment results to keep
num_permutations the number of primer permutations to search, if the degenerate bases cause more than this number of permutations to exist, this number will be sampled from all possible permutations.
... additional arguments passed to Primer-Blast
.parallel if 'TRUE', perform in parallel, using parallel backend provided by foreach

Value

A list with the following elements,

name name of the primer pair
BLAST_result html blast results from Primer-BLAST as 'a response object.
summary.primerTree

taxonomy  taxonomy for the primer products from NCBI
sequence  sequence of the primer products
alignment multiple alignment of the primer products
tree  phylogenetic tree of the reconstructed from the `multiple alignment

See Also

primer_search, clustalo

Examples

```r
## Not run:
# simple search
mammals_16s = search_primer_pair(name='Mammals 16S',
   'CGTGGGTAACCTCGGA', 'GCTGTATCCCTAGGTAACT')
# returning 1000 alignments, allow up to 3 mismatches in primer
mammals_16s = search_primer_pair(name='Mammals 16S',
   'CGTGGGTAACCTCGGA', 'GCTGTATCCCTAGGTAACT',
   num_aligns=1000, total_primer_specificity_mismatch=3)
## End(Not run)
```

summary.primerTree  
Summary a primerTree result, printing quantiles of sequence length and pairwise differences.

Description

Summarize a primerTree result, printing quantiles of sequence length and pairwise differences.

Usage

```
## S3 method for class 'primerTree'
summary(object, ..., probs = c(0, 0.05, 0.5, 0.95, 1),
   ranks = common_ranks)
```

Arguments

- **object**  the primerTree object to summarise
- **...**  Ignored options
- **probs**  quantile probabilities to compute, defaults to 0, 5, 50, 95, and 100 probabilities.
- **ranks**  ranks to show unique counts for, defaults to the common ranks

Value

invisibly returns a list containing the printed results
tree_from_alignment  

Construct a neighbor joining tree from a dna alignment

Description

Construct a neighbor joining tree from a dna alignment

Usage

tree_from_alignment(dna, ...)

Arguments

dna          fasta dna object the tree is to be constructed from
...

furthur arguments to dist.dna

See Also

dist.dna, nj
Index

accession2gi, 2
bryophytes_trnL, 2
clustalo, 3, 11
create_progress_bar, 10
dist.dna, 12
DNAbin, 3, 4
get_sequence, 3
get_sequences, 4
get_taxonomy, 4
gi2taxid, 5
identify.primerTree_plot, 5
layout_tree_ape, 6, 8
mammals_16S, 6
nj, 12
phylo, 6
plot.phylo, 6
plot.primerTree, 6, 9
plot_tree, 6, 7, 7, 8
plot_tree_ranks, 7, 8
primer_search, 9, 11
primerTree, 9
primerTree-package (primerTree), 9
response, 10
search_primer_pair, 9, 10
summary.primerTree, 11
tree_from_alignment, 12