Package 'RDPutils'

October 23, 2018

Type Package

Title R Utilities for Processing RDPTool Output

Version 1.4.1

Date 2018-03-08

Author John Quensen

Maintainer John Quensen <quensenj@msu.edu>

Depends phyloseq, reshape2, Biostrings, R (>= 3.2.0)

Suggests knitr, rmarkdown

VignetteBuilder knitr

Description Parses OTU tables from RDP cluster files; associates OTU names with representative sequences; renames representative sequences with OTU names; converts RDP and USEARCH classifier outputs to phyloseq objects.

License GPL-2

NeedsCompilation no

R topics documented:

`RDPutils-package` *RDPutils*

Description

This package provides utilities for importing RDP output into R and the Bioconductor package phyloseq. Version 1.3.0 adds functions for importing USEARCH and iTagger output.

The RDP provides both web-based and command line tools for processing rRNA gene sequences from Bacteria, Archaea, and Fungi as well as functional genes. These tools cluster sequences, reformat cluster files into OTU tables that can be imported into R, retrieve representative sequences for each cluster, and classify sequences from one or more samples. RDP does not, however, provide means of attaching taxonomic information to clusters (OTUs), of renaming representative sequences to correspond to OTUs, nor of easily importing classifier results into R. The functions in this package fill these gaps. Renaming representative sequences makes it possible to assign taxonomy to OTUs and to label a tree of the representative sequences with OTU names. In this manner, a phyloseq object may be fully populated with an otu_table, tax_table, sample_data, and tree.

Phyloseq is a Bioconductor/R package that provides a convenient way of organizing data from a sequencing experiment and also provides wrappers for common analyses and plots. Additionally, functions from other R packages may be applied to phyloseq components. Thus RDPutils makes it possible to apply the full range of analytic procedures available in R to RDP's clustering and classifier results.

Details

Package:	RDPutils
Type:	Package
Version:	1.4.1
Date:	2018-03-08
License:	GPL-2

Brief descriptions of the more important functions follow.

For clustering results:

clstr2otu parses an OTU file for a given distance from a cluster file.

assoc_repseq_IDs_with_otus_by_fasta - parses representative sequence headers to give a table associating the machine names of representative sequences for each cluster with the OTU name in the OTU table. This function depends on the fasta IDs being formatted in certain ways.

assoc_repseq_IDs_with_otus_by_clstr - parses the cluster file to give a table associating the machine names of representative sequences for each cluster with the OTU name in the OTU table.

This function is more robust but slower than assoc_repseq_IDs_with_otus_by_fasta. It can be used in cases where the fasta IDs are formatted differently - i.e. for some legacy results, or in case the fasta ID format changes.

rename_fasta - renames the representative sequences from their machine names to their corresponding OTU names. These renamed sequences can then be classified and treed, and the results imported into a phyloseq object.

make_tax_table - reformats RDP classifier results into a phyloseq tax_table.

For classifier results:

hier2phyloseq - parses a hierarchical classification file to give a phyloseq object with otu_table and tax_table.

Additional functions are provided for dividing a cluster file into separate files for each distance and for modifying fasta files to prepare them for renaming and treeing.

For USEARCH results:

import_usearch_biom_file - For correctly importing biom files.

import_otutab_taxa - For importing OTU tables and combined OTU/taxonomy tables.

import_utax_file and import_sintax_file for importing taxonomy files.

For iTagger results:

import_itagger_otutab_taxa - For importing combined OTU/taxonomy tables created with iTagger.

Author(s)

John Quensen

Maintainer: John Quensen <quensenj@msu.edu>

References

Cole, J. R., Q. Wang, J. A. Fish, B. Chai, D. M. McGarrell, Y. Sun, C. T. Brown, A. Porras-Alfaro, C. R. Kuske, and J. M. Tiedje. 2014. Ribosomal Database Project: data and tools for high throughput rRNA analysis Nucl. Acids Res. 41(Database issue):D633-D642; doi: 10.1093/nar/gkt1244 [PMID: 24288368]

McMurdie and Holmes (2013) phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data. PLoS ONE. 8(4):e61217

Wang, Q, G. M. Garrity, J. M. Tiedje, and J. R. Cole. 2007. Native Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy. Appl Environ Microbiol. 73(16):5261-5267; doi: 10.1128/AEM.00062-07 [PMID: 17586664]

`assoc.table` *Example of an Association Table*

Description

This association table is a data frame for use in the example for the rename_fasta function in this package RDPutils. It is identical to that produced in the example for the function assoc_repseq_IDs_with_otus_by_clstr. The first column contains the machine names of representative sequences and the second column contains corresponding OTU names.

Usage

```
assoc.table
```

Format

A data frame with 443 rows and 4 columns.

---

assoc_repseq_IDs_with_otus_by_clstr

Associate Representative Sequence IDs with OTUs from Cluster File

---

Description

This function parses a cluster file for a single distance and makes a table associating the representative sequence machine names with the OTU names as given by RDP's cluster file formatter with options "R" or "biom."

Usage

```
assoc_repseq_IDs_with_otus_by_clstr(clstr_file, rep_seqs, otu_format)
```

Arguments

clstr_file	The name of a cluster file for a single distance.
rep_seqs	A vector of representative sequence machine names.
otu_format	When equal to "R" (default) OTU names have the form "OTUxxxnn." When equal to "biom", OTU names have the form "cluster_nn."

Details

The first input to this function is the name of a cluster file for a single distance. The function reads the cluster file from disk. The cluster file for a single distance is excised from the original cluster file which likely contained cluster information for several distances with either of the functions clstr2otu or split_clstr_file.

The second input to this function is a vector of representative sequence machine names corresponding to the input cluster file; that is, they are for the same distance. This vector is created from a fasta file of the representative sequences with the function get_repseq_IDs_from_fasta.

OTUs may be named in either of two formats, corresponding to those output by the RDP's cluster file formatter with options "R" and "biom." With option "R" (the default), OTU names begin with "OTU" and are padded to equal length with leading zeros, e.g. "OTU00067." Thus they can be sorted in numerical order. With option "biom," OTU names begin with "cluster_" and are not padded with leading zeros, e.g "cluster_67."

Value

This function returns a data frame. The first column contains the machine names of the representative sequences. The second column contains the names of the corresponding OTUs as given by clstr2otu, by RDP's web-based Cluster File Formatter with optons "R" or "biom", and by the command line version of function Cluster in RDPTools. The third column contains sample names and the fourth the number of sequences in the OTU for the sample in column three. These last two columns are provided as a means of checking the result.

Author(s)

John Quensen

References

The web-based tool for retrieving representative sequences is here: http://pyro.cme.msu.edu/

The RDPTools are available on GitHub: https://github.com/rdpstaff

Examples

```
repseq.file <- system.file("extdata", "all_seq_complete.clust_rep_seqs.fasta", package="RDPutils")
rep.seqs <- get_repseq_IDs_from_fasta(repseq_file = repseq.file)
clstr.file <- system.file("extdata", "dist_03.clust", package="RDPutils")
assoc.table <- assoc_repseq_IDs_with_otus_by_clstr(clstr_file = clstr.file, rep_seqs = rep.seqs)
head(assoc.table)
```

---

assoc_repseq_IDs_with_otus_by_fasta

Associate Representative Sequences with OTUs from Fasta Headers

---

Description

This function parses representative sequence IDs and makes a table associating the representative sequence machine names with the OTU names as given by RDP's cluster file formatter with options "R" or "biom," or function clstr2otu in this package.

Usage

```
assoc_repseq_IDs_with_otus_by_fasta(repseq_file="all_seq_complete.clust_rep_seqs.fasta", otu_for
```

Arguments

repseq_file	The name of the fasta file containing representative sequences.
otu_format	When equal to "R" (default) OTU names have the form "OTUxxxnn." When equal to "biom", OTU names have the form "cluster_nn."

Details

Representative sequences from clusters for a given distance may be obtained with either the web-based representative sequence tool currently on the rdpipeline page (http://pyro.cme.msu.edu/), or with the RDPTools' cluster function using a command similar to:

java -Xmx2g -jar $Clustering rep-seqs –one-rep-per-otu all_seq_complete.clust 0.03 merged_aligned.fasta

In these cases the fasta headers contain information on the cluster number and the size of the cluster. This function parses this information into a table that can be used as input to function rename_fasta, which renames the representative sequences with their corresponding OTU names.

Value

This function returns a data frame with 4 columns: the machine name of the representative sequence, the corresponding OTU name as given by RDP's cluster file formatter with options "R" or "biom" and by function clstr2otu in this package, the cluster number, and the total number of sequences in the cluster (cluster size).

Note

The representative sequence tool on the FunGene pipeline page (http://fungene.cme.msu.edu/FunGenePipeline/) returns one representative sequences per sample, a format which is not compatible with this function.

This function expects the representative sequence IDs to be formatted in one of these ways:

>HC9DO0P01BCTC4 prefered=false,cluster=0,clustsize=2

>HC9DO0P01BCTC4 cluster_id=1,size=2

If the representative sequence IDs are not formatted as in these examples, or do not contain information on cluster number and size, a similar association table may be made using function assoc_repseq_IDs_with_otus_by_clstr.

Author(s)

John Quensen

See Also

assoc_repseq_IDs_with_otus_by_clstr, clstr2otu, rename_fasta

Examples

```
repseq.file <- system.file("extdata", "all_seq_complete.clust_rep_seqs.fasta", package="RDPutils")
assoc.table <- assoc_repseq_IDs_with_otus_by_fasta(repseq_file=repseq.file)
head(assoc.table)
```

---

clstr2otu *Create an OTU Table from a Cluster File*

Description

Parses a cluster file into an otu table with samples as rows and OTUs as columns. The input cluster file can contain data for more than one distance, or for a single distance. This function provides the same result as RDP's R-formatter, but for only one distance. If OutFile=TRUE, then the function also writes a cluster file for distance = dist to file.name.

Usage

```
clstr2otu(clstr_file = "all_seq_complete.clust", dist = 0.03, OutFile = FALSE, file_name = "dist_03.
```

Arguments

clstr_file	The output file given by RDP's cluster tool. This file usually contains cluster information for more than one distance.
dist	dist is the maximum distance between sequences in the same cluster. An OTU table will be created for the distance given here, which must be present in the cluster file.
OutFile	A logical. If OutFile is TRUE, then a cluster file is written to disk for the single distance given by dist. Default = FALSE.
file_name	The name of the cluster file written to disk if OutFile is TRUE. Otherwise ignored.

`otu_format` When equal to "R" (default) OTU names have the form "OTUxxxnn." When equal to "biom", OTU names have the form "cluster_nn."

Details

This function can take several minutes, depending on the number of OTUs and samples. The cluster file for a single distance output by the function can be used to associate the OTU names in the OTU table produced with the representative sequence machine names.

The sample names are shortened by removing common prefixes and suffixes introduced by RDP's tools. These include "nc_", "aligned_", "_trimmed", ".fasta" and ".fastq."

The two OTU formats correspond to those output by the RDP's cluster file formatter, which has the options "R"" and "biom." With option "R," OTU names begin with "OTU" and are padded to equal length with leading zeros, e.g. "OTU00067." Thus they can be sorted in numerical order. With option "biom," OTU names begin with "cluster_" and are not padded with leading zeros, e.g "cluster_67."

Value

Returns a numerical data frame, or OTU table, where samples are rows and OTUs are columns. This is the convention used by vegan. To import the result into phyloseq, it should first be transposed and converted to class matrix.

Author(s)

John Quensen

References

RDP web-based clustering services and tutorials on clustering sequence data are available at: http://rdp.cme.msu.edu/

RDPTools is an open source package available from https://github.com/rdpstaff; it includes a command line function for clustering, necessary for data sets too large to process with the web-based clustering tool.

Cole, J. R., Q. Wang, J. A. Fish, B. Chai, D. M. McGarrell, Y. Sun, C. T. Brown, A. Porras-Alfaro, C. R. Kuske, and J. M. Tiedje. 2014. Ribosomal Database Project: data and tools for high throughput rRNA analysis Nucl. Acids Res. 41(Database issue):D633-D642; doi: 10.1093/nar/gkt1244 [PMID: 24288368]

Examples

```
clstr.file <- system.file("extdata", "all_seq_complete.clust", package="RDPutils")
otu <- clstr2otu(clstr_file=clstr.file, dist=0.03, OutFile=TRUE, file_name= "dist_03.clust")
otu[ , 1:5]
```

count_char_occurrences *Count Occurrences of a Character*

Description

Count the number of occurrences of a character in a string.

Usage

```
count_char_occurrences(char, strng_x)
```

Arguments

char	Character to search for.
strng_x	String that is searched

Value

An integer value.

Author(s)

John Quensen

Examples

```
my.strng <- "abcdefabcaa"
count_char_occurrences("a", strng_x=my.strng)
```

---

get_repseq_IDs_from_fasta *Get Representative Sequence Names*

Description

This function retrieves a vector of the representative sequence machine names from a fasta file of representative sequences.

Usage

```
get_repseq_IDs_from_fasta(repseq_file = "all_seq_complete.clust_rep_seqs.fasta")
```

Arguments

repseq_file	The file name of the combined fasta file containing the representative sequences.

Details

The function reads the fasta file of representative sequences from disk.

Value

A vector of the machine names of the representative sequences.

Author(s)

John Quensen

References

The web-based tool for retrieving representative sequences is here: http://pyro.cme.msu.edu/

A command line version of the tool is included in RDPTools, available on GitHub: https://github.com/rdpstaff

Examples

```
repseq.file <- system.file("extdata", "all_seq_complete.clust_rep_seqs.fasta", package="RDPutils")
rep.seqs <- get_repseq_IDs_from_fasta(repseq_file = repseq.file)
rep.seqs
```

---

hier2phyloseq *Classification File Converter*

Description

Converts RDP's hierarchical classification file to a phyloseq object.

Usage

```
hier2phyloseq(hier_file = "test_hier.txt")
```

Arguments

hier_file	Tab-delimited text file from RDP's command line classifier with option filterbyconf.

Details

RDP's classifier generates two types of output. The detail format gives the classification of each sequence input. The hierachical format gives the number of sequences in each taxon. If the classifier is given a number of samples at the same time, the hierachical format can be filtered in Excel to give an otu table for a given rank, with taxa as rows and samples as columns.

The command line classifier takes arguments for confidence level and output format. For example, 28S fungal sequences can be classified with the commands:

cd c:\test

java -Xmx1g -jar /path_to_classfier.jar/classifier.jar classify -g fungallsu -c 0.5 -f filterbyconf -o test_classified.txt -h test_hier.txt *.fasta

All sample fasta files in directory c:\test are classified to a rank with confidence of 0.5 or more. By setting the format to filterbyconf, ranks not identified with a confidence of at least 0.5 are empty.

hier2phyloseq converts the hierachical result, file test_hier.txt in the example above, to a phyloseq object with otu_table and tax_table. Empty ranks are filled as unclassifed higher rank. For example, the family and genus assigned to sequences classified with confidence greater than 0.5 only as far as order "Pleosporales"" would be "unclass_Pleosporales."

The sample names are shortened by removing common prefixes and suffixes introduced by RDP's tools. These include "nc_", "aligned_", "_trimmed", ".fasta" and ".fastq."

This function is not appropriate for processing a single sample. To make a tax_table phyloseq object, use function "make_tax_table."

Value

Returns a phyloseq object with otu_table and tax_table.

Author(s)

John Quensen

See Also

make_tax_table

Examples

```
hier.file <- system.file("extdata", "test_hier.txt", package="RDPutils")
expt <- hier2phyloseq(hier_file=hier.file)
expt
```

---

import_itagger_otutab_taxa

Import iTagger File

Description

Converts the tab-delimited iTagger otu.tax.tsv file into a phyloseq object with otu_table and tax_table.

Usage

```
import_itagger_otutab_taxa(in_file)
```

Arguments

in_file	otu.tax.tav file

Details

The iTagger otu.tax.tsv file is similar to legacy QIIME format. This function expects six ranks: Kingdom, Phylum, Class, Order, Family, and Genus. The taxonomy field in otu.tax.tsv is parsed on semicolons.

Value

A phyloseq object with OTU and taxonomy tables.

Author(s)

John Quensen

Examples

```
##---- Not run. ----
##-- expt <- import_itagger_otutab_taxa(in_file = "iTagger_otutab_taxa_file.txt")

## The function is currently defined as
function (in_file)
{
    temp <- read.table(file = in_file, comment.char = "", header = TRUE,
        row.names = 1, stringsAsFactors = FALSE, sep = "\t")
    otu <- temp[, 1:ncol(temp) - 1]
    otu <- otu_table(as.matrix(otu), taxa_are_rows = TRUE, errorIfNULL = TRUE)
    tax <- matrix(data = NA, nrow = nrow(temp), ncol = 6)
    rownames(tax) <- rownames(temp)
    colnames(tax) <- c("Kingdom", "Phylum", "Class", "Order",
        "Family", "Genus")
    for (i in 1:nrow(temp)) {
        l <- temp[i, ncol(temp)]
        l.s <- strsplit(l, split = ";", fixed = TRUE)
        n <- length(l.s[[1]])
        for (j in 1:n) {
            tax[i, j] <- l.s[[1]][j]
        }
    }
    for (i in 1:nrow(tax)) {
        for (j in 1:ncol(tax)) {
            if (is.na(tax[i, j])) {
                t <- paste("uncl", tax[i, j - 1], sep = "_")
                for (n in j:ncol(tax)) {
                  tax[i, n] <- t
                }
            }
        }
    }
    tax <- tax_table(tax, errorIfNULL = TRUE)
    expt <- phyloseq(otu, tax)
    return(expt)
  }
```

`import_otutab_taxa` *Import USEARCH otutab_taxa File into phyloseq*

Description

Converts USEARCH's usearch_global output option otutabout to a phyloseq object with otu_table and tax_table.

Usage

```
import_otutab_taxa(in_file)
```

Arguments

in_file	An otutab_taxa file generated by USEARCH 8.1+

Details

The USEARCH usearch_global function has an option to output an OTU table with taxonomy assignments in a tab-delimited text file. The output resembles the legacy QIIME format. This function imports such a file into a phyloseq object with otu_table and tax_table.

Value

A phyloseq object with otu_table and tax_table.

Author(s)

John Quensen

Examples

```
##---- Not run. ----
##-- expt <- import_otutab_taxa(in_file = "otutab_taxa_03.txt")
```

---

import_sintax_file *Import sintax Taxonomy File*

Description

Converts the tab-delimited output of USEARCH's (version 9+) sintax command to a phyloseq tax_table object. The confidence level for taxonomic assignment is chosen on import.

Usage

```
import_sintax_file(in_file, confidence = 0.8)
```

Arguments

in_file	A sintax file.
confidence	The confidence level to use in assigning taxonomic categories

Details

The sintax algorithm is new to USEARCH 9. It does not require training and still calculates the confidence with which taxonomic assignments are made. It does this by a method purported to be more accurate than the RDP classifier method. This function allows choice of the confidence level (0.8 by default) at the time the file is imported as a phyloseq tax_table.

Value

A phyloseq tax_table.

Author(s)

John Quensen

References

Edgar, R.C. (2016), SINTAX, a simple non-Bayesian taxonomy classifier for 16S and ITS sequences, http://dx.doi.org/10.1101/074161.

Examples

```
##---- Not run. ----
##-- import_sintax_file(in_file="sintax_result.txt", confidence = 0.8)
```

`import_usearch_biom_file`

Import a USEARCH biom File

Description

Converts a biom file produced with USEARCH's usearch_global command into a phyloseq object.

Usage

```
import_usearch_biom_file(in_file, ...)
```

Arguments

`in_file`	A biom file produced with USEARCH's usearch_global command.
`...`	Other objects accepted by phyloseq, e.g. sample data table, representative sequences.

Details

This function will import the OTU table contained in the USEARCH biom file, and the taxonomy table if it is present. The function will also accept arguments for sample data table, reference sequences, and a phyloseq class taxonomy table from another source, if one is not present in the biom file.

Value

An experiment level phyloseq object (i.e. with at least an otu_table).

Author(s)

John Quensen

Examples

```
##---- Not run. ----
##-- expt <- import_usearch_biome(infile = "otutab_taxa_03.json")

## The function is currently defined as
import_usearch_biome_file <- function(in_file) {
  parse_taxonomy_usearch <- function (char.vec){
    parse_taxonomy_default(strsplit(char.vec, ",", TRUE)[[1]])
  }
  expt <- import_biom(in_file, parseFunction = parse_taxonomy_usearch, ...)
  return(expt)
}
```

`import_utax_file` *Import utax Taxonomy File*

Description

Converts the tab-delimited output of USEARCH's utax command to a phyloseq tax_table object. The confidence level for taxonomic assignment is chosen on import.

Usage

```
import_utax_file(in_file, confidence)
```

Arguments

`in_file`	The utaxout result from USEARCH's utax command.
`confidence`	The confidence level to use in assigning taxonomic categories.

Details

The default confidence level is 0.8.

Value

A phyloseq tax_table.

Author(s)

John Quensen

Examples

```
##---- Not run. ----
##-- tax.table <- import_utax_file(in_file="utax_result.txt", confidence = 0.8)
```

make_framebot_tax_table *make_framebot_tax_table*

Description

Create a taxonomy table from FunGene Pipeline output.

Usage

```
make_framebot_tax_table(clstr_machine, taxa_machine)
```

Arguments

clstr_machine	match_cluster_machine_name.txt from my FunGene Pipeline script
taxa_machine	match_taxa_machine_names.txt from my FunGene Pipeline script

Details

My FunGene pipeline script (see john-quensen.com) produces one file matching representative sequence machine names with cluster numbers and a second matching representative sequence machine names with taxa names found by FrameBot. This function parses and combines the two files to produce a phyloseq tax_table. Ranks are Genus, Species, and Strain. Classification is the closest match to sequence in the FrameBot reference database. The percent identity to the closest match is appended to the name of the closest match (strain level in the tax_table).

Value

A tax_table of class phyloseq.

Note

See the workshop page Command Line FunGene Pipeline at john-quensen.com for the FunGene Pipeline script.

Author(s)

John Quensen

Examples

```
match.clst <- system.file("extdata", "match_cluster_machine_name.txt", package="RDPutils")
match.taxa <- system.file("extdata", "match_taxa_machine_names.txt", package="RDPutils")
my_taxa <- make_framebot_tax_table(clstr_machine=match.clst, taxa_machine=match.taxa)
```

make_otu_names *Make OTU Names*

Description

This function takes a vector of integers and makes a vector of OTU names.

Usage

```
make_otu_names(otu_nums, otu_format="R")
```

Arguments

otu_nums	A vector of integers.
otu_format	When equal to "R" (default) OTU names have the form "OTUxxxnn." When equal to "biom", OTU names have the form "cluster_nn."

Details

This function is used by the function clstr2otu to name the columns of the OTU table it returns, but make_otu_names can be used separately if desired.

OTUs may be named in either of two formats, corresponding to those output by the RDP's cluster file formatter with options "R" and "biom." With option "R" (the default), OTU names begin with "OTU" and are padded to equal length with leading zeros, e.g. "OTU00067." Thus they can be sorted in numerical order. With option "biom," OTU names begin with "cluster_" and are not padded with leading zeros, e.g "cluster_67."

Value

A vector of character strings of the form "OTU_xxxnn"" or "cluster_nn."

Author(s)

John Quensen

Examples

```
n <- c(1:10)
otu.names <- make_otu_names(n)
otu.names
```

make_tax_table *Make a Phyloseq tax_table from Classifier Result*

Description

This function creates a phyloseq tax_table from the RDP classifier result (detail format) for renamed representative sequences.

Usage

```
make_tax_table(in_file="fixrank_classified.txt", confidence=0.5)
```

Arguments

in_file	File name for RDP classifier result in detail format.
confidence	Confidence level for construction of tax_table.

Details

The input file is the file designated by the "-o" argument to the command line RDP classifer. The format given by "-f" must be "fixrank." The confidence level "-c" in the classifier command is unimportant; instead, the tax_table will be constructed according to the confidence argument to make_tax_table. This confidence argument is similar in concept to "-c" which actually affects only the hierarchical (-h) and fillterbyconf (-f) output of the classifier. For example, after classifying the file "renamed_repseqs.fasta" (16S rRNA DNA sequences) with the commands:

cd <to directory with fasta file to be classified>

java -Xmx1g -jar /path_to_classfier.jar/classifier.jar classify -g 16srrna -f fixrank -o test_classify.txt renamed_repseqs.fasta

use "test_classify.txt" as the in_file to make_tax_table.

Ranks classified with confidence less than that specified are filled as unclassifed higher rank. For example, the family and genus assigned to OTUs classified with confidence greater than 0.5 only as far as order "Actinomycetales" would be "unclass_Actinomycetales."

Value

A phyloseq tax_table object.

Author(s)

John Quensen

Examples

```
my.in.file <- system.file("extdata", "fixrank_classified.txt", package="RDPutils")
my.tax.table <- make_tax_table(in_file = my.in.file, confidence=0.5)
my.tax.table
```

otu *Example of an OTU Table*

Description

This OTU table is for use with the examples in this package RDPutils. There are 4 samples in rows by 443 OTUs in columns. It was made by clustering 4 samples of 250 partial 16S rRNA gene sequences at a distance of 0.03.

Usage

```
otu
```

Format

A data frame with 4 rows and 443 columns.

remove_model_seqs *Remove Model Sequences*

Description

This function removes model sequences from a combined fasta file.

Usage

```
remove_model_seqs(in_file, out_file = in_file)
```

Arguments

in_file	The name of a combined fasta file with model sequences.
out_file	The name of the corresponding fasta file written to disk without model sequences.

Details

This function operates on files. It is not normally assigned to a variable. By default, the input file is overwritten. If no model sequence is found in the input file, a message to that effect is returned.

RDP aligns 16S rRNA gene sequences using the Infernal aligner with model sequences for Bacteria or Archaea. A model sequence is introduced into each alignment. These model sequences are required by the cluster tool and the merge alignment tool. If a model sequence is present in the fasta file of representative sequences, it needs to be removed before the sequences can be treed with FastTree; this function provides the means to do so.

Value

The function returns a message that it has completed.

Author(s)

John Quensen

Examples

```
in.file <- system.file("extdata", "all_seq_complete.clust_rep_seqs.fasta", package="RDPutils")
remove_model_seqs(in_file=in.file, out_file = file2tree.fasta)
```

---

rename_fasta *Rename Representative Sequences*

Description

This function renames representative sequences from their machine names to their OTU names.

Usage

```
rename_fasta(in_file = "names_trimmed.fasta", out_file = "renamed.fasta", rename_table)
```

Arguments

in_file	The name of the fasta file containing representative sequences with their machine names.
out_file	The name of the fasta file containing representative sequences with their OTU names.
rename_table	A data frame associating the machine names of representative sequences with their corresponding OTU names.

Details

The output of the function assoc_repseq_IDs_with_otus_by_fasta or assoc_repseq_IDs_with_otus_by_clstr can be used as the rename_table.

Before representative sequences can be renamed with this function, their IDs have to be shortened to include only their machine names with the function trim_fasta_names.

Value

A fasta file with renamed representative sequences is written to disk. This function is not normally assigned to a variable. The function returns a message that it has completed.

Author(s)

John Quensen

See Also

assoc_repseq_IDs_with_otus_by_clstr

assoc_repseq_IDs_with_otus_by_fasta

trim_fasta_names

Examples

```
in.file <- system.file("extdata", "names_trimmed.fasta", package="RDPutils")
data(assoc.table)
rename_fasta(in_file = in.file, out_file = "renamed.fasta", rename_table = assoc.table)
```

---

sam.data *An example of a Sample Data Table*

Description

This sample data table is for use with the examples in this package RDPutils. There are 4 samples in rows by 4 factors and variables in columns.

Usage

```
data(sam.data)
```

Format

A data frame with 4 observations on the following 4 variables.

`Treatment` a factor with levels `A` `B`

`Replicate` a factor with levels `1` `2`

`Variable_A` a numeric vector

`Variable_B` a numeric vector

Examples

```
data(sam.data)
sam.data
```

---

select_seqs *Select Fasta Sequences*

Description

This function subsets a combined fasta file.

Usage

```
select_seqs(in_file, select_list, out_file)
```

Arguments

`in_file`	The name of a combined fasta file to be read from disk.
`select_list`	A vector of the names of the individual fasta files to be kept.
`out_file`	The name of the modified combined fasta file to be written to disk.

Details

This function can be used to select a subset of the renamed representative sequences corresponding to OTUs containing at least n sequences. See the example section below.

Reducing the number of representative sequences in this manner makes several subsequent steps go faster: classifying the representative sequences, making a phyloseq tax_table, and treeing the representative sequences.

Value

This function operates on disk files. It is not normally assigned to a variable. It returns a message that it has completed.

Author(s)

John Quensen

Examples

```
renamed.fasta <- system.file("extdata", "renamed.fasta", package="RDPutils")
data(otu)
otu <- otu[ , colSums(otu)>=5]
select.list <- colnames(otu)
select_seqs(in_file=renamed.fasta, select_list=select.list, out_file="subset.renamed.repseqs.fasta")
```

---

simple_cap *simple_cap*

Description

Capitalizes the first letter of each word in a string.

Usage

```
simple_cap(x)
```

Arguments

x	A text string containing one or more words.

Details

This function is used by the function make_tax_table to make sure that the rank names begin with capital letters. Makes for prettier plotting when rank names are used as legend headers, etc.

Value

A text vector with first letter of each word capitalized.

Note

This function is given as an example in the documentation for the base function toupper.

See Also

make_tax_table

Examples

```
x <- "The quick brown fox jumped over the lazy dog."
simple_cap(x)
```

`split_clstr_file` *Split a Cluster File*

Description

This function splits a cluster file containing cluster information for several distances into separate cluster files for each distance.

Usage

```
split_clstr_file(clstr_file = "all_seq_complete.clust", file.prefix = "dist_")
```

Arguments

`clstr_file`	The name of the cluster file for multiple distances.
`file.prefix`	The prefix for the individual cluster files to be written to disk.

Details

Output files have names of the form prefix0.nn.clust where nn is the decimal distance at which sequences were clustered. The resulting cluster files may be used as input to the function assoc_repseq_IDs_with_otus_by_clstr which requires that the input cluster file be for a single distance. The resulting files may also be used as input to clstr2otu with some gain in speed compared to inputting a cluster file with multiple distances.

Value

This function operates on files. It returns a message when it has completed. It is not normally assigned to a variable; if so, the variable contains only the message.

Author(s)

John Quensen

See Also

`clstr2otu`

`assoc_repseq_IDs_with_otus_by_clstr`

Examples

```
clstr.file <- system.file("extdata", "all_seq_complete.clust", package="RDPutils")
split_clstr_file(clstr_file = clstr.file, file.prefix = "dist_")
```

`trim_fasta_names` *Trim Representative Sequence Names*

Description

This function trims the fasta IDs of the representative sequences to include only the machine name.

Usage

```
trim_fasta_names(repseq_file, trimmed_names, strip = FALSE)
```

Arguments

`repseq_file`	The name of the combined fasta file of representative sequences.
`trimmed_names`	The name of the modified fasta file written to disk.
`strip`	A logical. If TRUE, then the output fasta file is unaligned and any model sequences are removed. The default is FALSE.

Details

This function is a necessary prerequisite to renaming representative sequences with their corresponding OTU names.

Unaligning the sequences results in a smaller file size and is acceptable if the sequences are to be classified only. It may be desirable if using the web-based classifier because the smaller file size makes for shorter upload time. Unaligning must not be done if the sequences are to be treed.

Value

This function operates on files. It returns a message that it has completed. It is not normally assigned to a variable; if so, the variable contains only the message.

Author(s)

John Quensen

Examples

```
repseq.file <- system.file("extdata", "all_seq_complete.clust_rep_seqs.fasta", package="RDPutils")
trim_fasta_names(repseq_file = repseq.file, trimmed_names = "names_trimmed.fasta", strip = FALSE)
```

unalign_fasta *Unalign a Fasta File*

Description

This function unaligns a combined fasta file and removes any model sequences present. The modified fasta file is written to disk.

Usage

```
unalign_fasta(in_file, out_file)
```

Arguments

in_file	The name of the input fasta file. The file is read from disk.
out_file	The name of the modified fasta file written to disk.

Details

This function can be used to unalign renamed representative sequences in order to reduce the file size before uploading the file to the web-based RDP classifier.

Value

This function operates on files. It returns a message that it has completed. It is not normally assigned to a variable; if so, the variable contains only the message.

Author(s)

John Quensen

Examples

```
in.file <- system.file("extdata", "merged_aligned.fasta", package="RDPutils")
unalign_fasta(in_file=in.file, out_file="unaligned.fasta")
```

Index