Package 'phylotypr'

October 22, 2026

Type Package

Title Classifying DNA Sequences to Taxonomic Groupings

Version 0.2.0

Description Classification based analysis of DNA sequences to taxonomic groupings. This package primarily implements Naive Bayesian Classifier from the Ribosomal Database Project. This approach has traditionally been used to classify 16S rRNA gene sequences to bacterial taxonomic outlines; however, it can be used for any type of gene sequence. The method was originally described by Wang, Garrity, Tiedje, and Cole in Applied and Environmental Microbiology 73(16):5261-7 <doi:10.1128/AEM.00062-07>. The package also provides functions to read in 'FASTA'-formatted sequence data.

License GPL-3

Encoding UTF-8

LazyData true

LazyDataCompression xz

URL https://github.com/mothur/phylotypr, https://mothur.org/phylotypr/

BugReports https://github.com/mothur/phylotypr/issues

Suggests knitr, rmarkdown, purrr, dplyr, testthat (>= 3.0.0), strollur

Config/testthat/edition 3

Imports Rcpp, readr (>= 2.1.0), stringi (>= 1.8.0), stats, parallelly

LinkingTo Rcpp, RcppThread

Depends R (>= 4.2)

VignetteBuilder knitr

Config/roxygen2/version 8.1.1

NeedsCompilation yes

Author Pat Schloss [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-6935-4275>)

Maintainer Pat Schloss <pschloss@umich.edu>

Repository CRAN

Date/Publication 2026-10-22 14:30:02 UTC

Contents

build_kmer_database 2
classify_sequence 3
classify_sequences 4
create_database 6
filter_taxonomy 7
phylotypr_example 8
print_taxonomy 9
rdp_trainset9 10
read_fasta 11
read_taxonomy 12
write_fasta 13

Index **15**

`build_kmer_database` *Build kmer database*

Description

Build kmer database for classifying 16S rRNA and other gene sequences to a genus when a kmer size is provided.

Usage

```
build_kmer_database(classification_database, kmer_size = 8)
```

Arguments

`classification_database`	A `classification_database` object, which is a list containing the fields `seq_taxonomy_data` and `reference_data`. These are both data.frame objects. The package includes rdp_trainset9 and rdp_trainset9_pds. Other databases can be obtained from **phylotyprrefdata** or by using `create_database()`
`kmer_size`	An integer indicating the length of the nucleotide word to base our classification on (default = 8)

Value

A list object containing the genus level conditional probability (`conditional_prob`) of seeing each kmer in a given genus as well as the genus names (`genera`)

References

Wang Q, Garrity GM, Tiedje JM, Cole JR. Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. Appl Environ Microbiol. 2007 Aug;73(16):5261-7. doi:10.1128/AEM.0006207 PMID: 17586664; PMCID: PMC1950982.

Examples

```
kmer_size <- 8

## Not run:
# Function takes >5 s to run on this database
build_kmer_database(rdp_trainset9_pds, kmer_size)

## End(Not run)
```

`classify_sequence` *Classify 16S rRNA gene sequence fragment*

Description

The `classify_sequence()` function implements the Wang et al. naive Bayesian classification algorithm for 16S rRNA gene sequences.

Usage

```
classify_sequence(sequence, database, kmer_size = 8, num_bootstraps = 100)
```

Arguments

`sequence`	A character object representing a DNA sequence that needs to be classified.
`database`	A kmer database generated using `build_kmer_database`
`kmer_size`	An integer value (default of 8) indicating the size of kmers to use for classifying sequences. Higher values use more RAM with potentially more specificity Lower values use less RAM with potentially less specificity. Benchmarking has found that the default of 8 provides the best specificity with the lowest possible memory requirement and fastest execution time.
`num_bootstraps`	An integer value (default of 100). The value of `num_bootstraps` is the number of randomizations to perform where `1/kmer_size` of all kmers are sampled (without replacement) from `sequence`. Higher values will provide greater precision on the confidence score.

Value

A list object of two vectors. One vector (`taxonomy`) is the taxonomic assignment for each level. The second vector (`confidence`) is the percentage of `num_bootstraps` that the classifier gave the same classification at that level.

Examples

```
kmer_size <- 3
reference_names <- c("seq_1", "seq_2", "seq_3")
reference_sequences <- c("ATGCGCTA", "ATGCGCTC", "ATGCGCTC")
reference_genera <- c("A", "B", "B")
reference <- create_database(reference_names, reference_sequences, reference_genera)

db <- build_kmer_database(reference, kmer_size)
unknown_sequence <- "ATGCGCTC"

classify_sequence(
  sequence = unknown_sequence,
  database = db,
  kmer_size = kmer_size
)
```

---

classify_sequences *Classify a vector of 16S rRNA gene sequence fragments*

Description

The `classify_sequences()` function implements the Wang et al. naive Bayesian classification algorithm for 16S rRNA gene sequence.

Usage

```
classify_sequences(
  data = NULL,
  sequence,
  sequence_name = as.character(seq_along(sequence)),
  database,
  kmer_size = 8,
  num_bootstraps = 100,
  min_confidence = 80,
  seed = 1,
  num_processors = parallelly::availableCores()
)
```

Arguments

`data`	A data frame containing a column of `sequence_name` and `sequence`. If provided column names need to be provided to `sequence_name` and `sequence`.
`sequence`	A vector of character objects representing DNA sequence that need to be classified. If `data` is provided then only the name of the column containing sequence needs to be provided, quotation marks are optional. This argument can be ignored when using a `{strollur}` object

sequence_name	A vector of character objects containing the sequence name that need to be classified. Optional. Defaults to the position of the sequence in the vector of sequence. If data is provided then only the name of the column containing the sequence name needs to be provided, quotation marks are optional. This argument can be ignored when using a {strollur} object
database	A kmer database generated using build_kmer_database
kmer_size	An integer value (default of 8) indicating the size of kmers to use for classifying sequence. Higher values use more RAM with potentially more specificity Lower values use less RAM with potentially less specificity. Benchmarking has found that the default of 8 provides the best specificity with the lowest possible memory requirement and fastest execution time.
num_bootstraps	An integer value (default of 100). The value of num_bootstraps is the number of randomizations to perform where 1/kmer_size of all kmers are sampled (without replacement) from unknown_sequence. Higher values will provide greater precision on the confidence score.
min_confidence	An integer value (default of 80). The value of min_confidence represents the minimum number of bootstraps that must have a classification that agree with each other.
seed	An integer value (default of 1). The value of seed sets the seed for the random number generator.
num_processors	An integer value. The value of num_processors indicates the number of desired processors. Defaults to all available cores using parallelly::availableCores()

Value

A data frame object with three columns: the sequence name (sequence_name), the taxonomic assignment for each level (taxonomy), and the confidence associated with that classification level (confidence). The value in confidence represents the percentage of num_bootstraps that the classifier gave the same classification at that level.

References

Wang Q, Garrity GM, Tiedje JM, Cole JR. Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. Appl Environ Microbiol. 2007 Aug;73(16):5261-7. doi:10.1128/AEM.0006207 PMID: 17586664; PMCID: PMC1950982.

Examples

```
kmer_size <- 3
reference_names <- c("seq_1", "seq_2", "seq_3")
reference_sequences <- c("ATGCGCTA", "ATGCGCTC", "ATGCGCTC")
reference_genera <- c("A", "B", "B")
reference <- create_database(reference_names, reference_sequences, reference_genera)

db <- build_kmer_database(reference, kmer_size)

unknown_sequence <- c("ATGCGCTC", "ATGCGCTC", "ATGCGCTC")
unknown_id <- c("seq_1", "seq_2", "seq_3")
```

```
classify_sequences(
  sequence = unknown_sequence,
  database = db,
  kmer_size = kmer_size,
  seed = 1,
  num_processors = 1
)

classify_sequences(
  sequence = unknown_sequence,
  sequence_name = unknown_id,
  database = db,
  kmer_size = kmer_size
)

classify_sequences(
  sequence = "ATGATCGAGTC",
  database = db,
  kmer_size = kmer_size,
  min_confidence = 95
)

sequence_df <- data.frame(
  seq_names = unknown_id,
  seqs = unknown_sequence
)

classify_sequences(sequence_df,
  sequence_name = seq_names, sequence = seqs,
  database = db, kmer_size = 3
)
```

`create_database` *Build custom database object for use with phylotypr*

Description

The `create_database()` function builds a `classification_database` object that can be used to generate the kmer database for downstream use in `classify_sequences()`. This function is only necessary if you are building your own database or modifying one of the existing databases provided in {phylotypr} or {phylotyprrefdata}.

Usage

```
create_database(sequence_name, sequence, taxonomy, reference_data = NULL)
```

Arguments

sequence_name	A character vector of identifiers for each sequence in the database. Should have the same number of values as found in the sequence and taxonomy vectors.
sequence	A character vector of the sequences that are included in the database. Should have the same number of values as found in the sequence_name and taxonomy vectors.
taxonomy	A character vector of taxonomy data for each sequence in the database. Should have the same number of values as found in the sequence_name and sequence vectors. The format of each value should be with each taxonomic level separated by a semicolon. Ideally each value in taxonomy will have the same number of levels.
reference_data	single row data frame object that has columns vendor, name, version, usage, notes, method_url, documentation_url, and citation. This is supplied with databases obtained from **phylotyprrefdata**. This information is only used if user supplied sequences are supplied as a **strollur** object.

Value

A classification_database object, which is a list containing the fields seq_taxonomy_data and reference_data. These are both data.frame objects.

Examples

```
sequence_name <- c("seq_a", "seq_b", "seq_c")
sequence <- c("AAAATCGAGTA", "ATGCGCTCCCC", "ATGCGCTCCCC")
genera <- c("A;b;C;d", "A;b;C;e", "A;b;C;e")

create_database(
  sequence_name = sequence_name,
  sequence = sequence,
  taxonomy = genera
)
```

---

filter_taxonomy *Filter taxonomy*

Description

The filter_taxonomy() function will filter a consensus taxonomy to remove any taxonomic levels where the confidence score is below a min_confidence level

Usage

```
filter_taxonomy(consensus, min_confidence = 80)
```

Arguments

consensus	A data frame that contains at least three columns: the sequence_name column that contains an identifier for the sequence, the taxonomy column contains the classification at each taxonomic level, and the confidence column contains the percentage of bootstraps that had the specified classification. The level column is optional and contains the numeric level of the level (e.g., Kingdom = 1, Genus = 6)
min_confidence	A double value between 0 and 100 (default = 80). The minimum percentage of bootstrap replicates that had the same classification. Any confidence score below this value will have the corresponding taxonomy removed

Value

A data frame object that is filtered to remove low confidence taxonomies. One column, taxonomy, contains the taxonomy at each taxonomic level and the other column, confidence contains the confidence score for that taxonomy. There will be no taxonomies or confidence scores below min_confidence

Examples

```
oscillospiraceae <- data.frame(
  sequence_name = c("seq_1", "seq_1", "seq_1", "seq_1", "seq_1", "seq_1"),
  level = 1:6,
  taxonomy = c(
    "Bacteria", "Bacillota", "Clostridia",
    "Eubacteriales", "Oscillospiraceae",
    "Flintibacter"
  ),
  confidence = c(100, 100, 99, 99, 98, 58)
)

filter_taxonomy(oscillospiraceae, min_confidence = 80)
```

---

phylotypr_example *Get path to phylotypr example*

Description

phylotypr comes bundled with some example files in its inst/extdata directory. This function make them easy to access.

Usage

```
phylotypr_example(path = NULL)
```

Arguments

path	Name of file. If NULL, the example files will be listed.

Value

A string indicating path to the file listed in `path`. If `NULL` is given then the return value is a vector of file names in the `extdata/` directory

Examples

```
phylotypr_example()
phylotypr_example("miseq_sop.fasta.gz")
```

`print_taxonomy` *Print taxonomy for an unknown sequence*

Description

The `print_taxonomy()` will output the consensus taxonomy for an unknown sequence with confidence scores for each taxonomic level and each taxonomic level separated by semi-colons

Usage

```
print_taxonomy(
  consensus,
  n_levels = 0,
  fill_levels = TRUE,
  show_confidence = TRUE
)
```

Arguments

`consensus`	A data frame that contains at least three columns: the `sequence_name` column that contains an identifier for the sequence, the `taxonomy` column contains the classification at each taxonomic level, and the `confidence` column contains the percentage of bootstraps that had the specified classification. The `level` column is optional and contains the numeric level of the level (e.g., Kingdom = 1, Genus = 6)
`n_levels`	An integer indicating the number of taxonomic levels to expect. If the number of observed levels is less than this value, and `fill_levels` is `TRUE` then missing levels will have "_unclassified" appended to the end of the last named classification until the desired level is reached. If `n_levels` is less than the observed number of levels, then the string of the output will be truncated to `n_levels`. If 0 is given then we will report the longest possible taxonomy. Default is 0.
`fill_levels`	A logical indicating whether or not to extend the classifications of sequences that don't meet `n_levels`. Default is `TRUE`.
`show_confidence`	A logical indicating whether or not to include the confidence scores. Default is `TRUE`

Value

A vector of character strings indicating the classification at each taxonomic level with the corresponding confidence in parentheses (if show_confidence is FALSE. Each taxonomic level is separated by a semi-colon

Examples

```
oscillospiraceae <- data.frame(
  sequence_name = c("seq_1", "seq_1", "seq_1", "seq_1", "seq_1"),
  level = 1:5,
  taxonomy = c(
    "Bacteria", "Bacillota", "Clostridia",
    "Eubacteriales", "Oscillospiraceae"
  ),
  confidence = c(100, 100, 99, 99, 98)
)

print_taxonomy(oscillospiraceae, n_levels = 6)
print_taxonomy(oscillospiraceae, n_levels = 4)
print_taxonomy(oscillospiraceae, n_levels = 6, show_confidence = FALSE)
print_taxonomy(oscillospiraceae, n_levels = 7, fill_levels = FALSE)
```

---

rdp_trainset9 *RDP training set v9*

Description

The sequence and taxonomy data for the 10,049 sequences found in the Ribosomal Database Project's trainset9_032012 training set for use with the naive Bayesian classifier as implemented in the {phylyotypr} R package. Originally released by the RDP in September 2012. The rdp version contains the same sequences as provided by the official RDP version (9,665 bacterial and 384 archaeal). The pds version contains extra eukaryotic sequences including 119 chloroplasts and mitochondria (10,168 total sequences). See the mothur reference file page in "Sources" for more information. Be sure to see the mothur GitHub project where you can find the phylotyprrefdata package (https://github.com/mothur/phylotyprrefdata) for access to other taxonomic reference data.

Usage

```
rdp_trainset9

rdp_trainset9_pds
```

Format

A classification_database object that holds two slots: seq_taxonomy_data and reference_data. The seq_taxonomy_data slot is a data.frame containing

sequence_name Sequence accession identifier

sequence DNA sequence string

taxonomy Taxonomic string with each level separated with a ;

The `reference_data` slot contains metadata about the database and includes fields:

reference_vendor Entity that created original database (e.g., RDP)

reference_versions The version of database provided by original entity

reference_usages For classification or alignment

reference_notes Any additional notes about the database. Currently the version of phylotypr that provided the database

reference_urls URLs where database can be obtained

reference_citation Citation for database in BibTeX format

An object of class `classification_database` of length 2.

Source

- mothur-formatted files
- RDP sourceforge page # nolint: line_length_linter

`read_fasta` *Read in a FASTA-formatted file containing DNA sequences*

Description

Given a standard FASTA-formatted file, `read_fasta` will read in the contents of the file and create a three column data frame with columns for the sequence name, the sequence itself, and any comments found in the header line for each sequence.

Usage

```
read_fasta(file, degap = TRUE)
```

Arguments

`file`	Either a path to a file, a connection, or literal data (either a single string or a raw vector) containing DNA sequences in the standard FASTA format. There are no checks to determine whether the data are DNA or amino acid sequences. Files ending in .gz, .bz2, .xz, or .zip will be automatically uncompressed. Files starting with `http://`, `https://`, `ftp://`, or `ftps://` will be automatically downloaded. Remote gz files can also be autom downloaded and decompressed.
`degap`	Logical value (default = TRUE) Removes gap characters from sequences indicated by "." or "-"

Value

A data frame object with three columns. The `sequence_name` column will contain the non-space characters following the > in the header line of each sequence; the `sequence` column will contain the sequence; and the `comment` column will contain any text found after the first whitespace character on the header line.

Note

The sequences in the FASTA file can have line breaks within them and `read_fasta()` will put those separate lines into the same sequence

Examples

```
temp <- tempfile()
write(">seqA\nATGCATGC\n>seqB\nTACGTACG", file = temp)
write(">seqC\nTCCGATGC", file = temp, append = TRUE)
write(">seqD B.ceresus UW85\nTCCGATGC", file = temp, append = TRUE)
write(">seq4\tE. coli K12\tBacteria;Proteobacteria;\nTCCGATGC",
  file = temp,
  append = TRUE
)
write(">seq_4\tSalmonella LT2\tBacteria;Proteobacteria;\nTCCGATGC",
  file = temp, append = TRUE
)
write(">seqE B.ceresus UW123\nTCCGATGC\nTCCGATGC",
  file = temp,
  append = TRUE
)

sequence_df <- read_fasta(temp)
```

`read_taxonomy` *Read in taxonomy files*

Description

Read a mothur-formatted taxonomy file into R as a data frame

Usage

```
read_taxonomy(file)
```

Arguments

`file`	Either a path to a file, a connection, or literal data (either a single string or a raw vector) containing the sequence name and the taxonomy information for each sequence.

Files ending in .gz, .bz2, .xz, or .zip will be automatically uncompressed. Files starting with `http://`, `https://`, `ftp://`, or `ftps://` will be automatically downloaded. Remote gz files can also be autom downloaded and decompressed.

Value

A data frame with two columns. The `sequence_name` column contains a name for each sequence and the `taxonomy` column, which contains the taxonomy for each sequence. The string in the `taxonomy` column is a series of taxonomic names separated by semi-colons. The string does not have a semi-colon at the end of the sequence

Note

There are no checks to insure that each sequence has a unique name. It is also assumed that each sequence has the same number of taxonomic levels represented in the second column of the input file.

Examples

```
temp <- tempfile()
write("seqA\tA;B;C;", file = temp)
write("seqB\tA;B; C;", file = temp, append = TRUE)
write("seqC\tA; B;C;", file = temp, append = TRUE)
write("seqD\tA;B;C", file = temp, append = TRUE)
write("seqE\tA;B; C", file = temp, append = TRUE)
write("seqF\tA; B;C", file = temp, append = TRUE)
write("seq G\tA;B;C;", file = temp, append = TRUE)

read_taxonomy(temp)
```

---

write_fasta *Write to a FASTA-formatted file*

Description

Writes a data frame containing sequence_name, sequence, and comment columns, `write_fasta` will write the data frame out to a standard FASTA-formatted file. The header will have a tab character between the sequence name and any comments. There won't be a tab if there's no comment for the sequence. All sequence data will be on a single line

Usage

```
write_fasta(data_frame, file = NULL)
```

Arguments

`data_frame`	A data frame object with three columns. The `sequence_name` column will contain the non-space characters following the > in the header line of each sequence; the `sequence` column will contain the sequence; and the `comment` column will contain any text found after the first whitespace character on the header line. The `comment` column is optional.
`file`	Either a path to a file, a connection, or literal data (either a single string or a raw vector) to write to a standard FASTA formatted file. There are no checks to determine whether the data are DNA or amino acid sequences. Files ending in .gz, .bz2, .xz, or .zip will be automatically compressed. Files starting with `http://`, `https://`, `ftp://`, or `ftps://` will be automatically downloaded. Remote gz files can also be autom downloaded and decompressed. If the value of `file` is `NULL` (default), the string will be written out to the screen

Value

Sequence data is either written out to the screen (`file = NULL`) or to a file.

Examples

```
df_d <- data.frame(
  sequence_name = c("seqA", "seqB", "seqC"),
  sequence = c("ATGCATGC", "ATGCATGA", "ATGCATGT"),
  comment = c("comment 1", "", "comment 3")
)

string_d <- write_fasta(df_d)
```

Index

∗ **datasets**
rdp_trainset9, 10

build_kmer_database, 2, *3*, *5*

classify_sequence, 3
classify_sequences, 4
create_database, 6

filter_taxonomy, 7

phylotypr_example, 8
print_taxonomy, 9

rdp_trainset9, 10
rdp_trainset9_pds *(*rdp_trainset9*)*, 10
read_fasta, 11
read_taxonomy, 12

write_fasta, 13