Package 'rRDP'

September 19, 2024

Title Interface to the RDP Classifier

Description This package installs and interfaces the naive Bayesian classifier for 16S rRNA sequences developed by the Ribosomal Database Project (RDP). With this package the classifier trained with the standard training set can be used or a custom classifier can be trained.

Version 1.39.0

Date 2024-03-26

biocViews Genetics, Sequencing, Infrastructure, Classification, Microbiome, ImmunoOncology, Alignment, SequenceMatching, DataImport, Bayesian

Depends Biostrings (>= 2.26.2)

BugReports https://github.com/mhahsler/rRDP/issues

URL https://github.com/mhahsler/rRDP/

Imports rJava, utils

Suggests rRDPData, knitr, rmarkdown

SystemRequirements Java JDK 1.4 or higher

License GPL-2 + file LICENSE

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.1

Roxygen list(markdown = TRUE)

git_url https://git.bioconductor.org/packages/rRDP

git_branch devel

git_last_commit 1d3c5fd

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-09-18

Author Michael Hahsler [aut, cre] (<https://orcid.org/0000-0003-2716-1405>), Nagar Anurag [aut]

Maintainer Michael Hahsler <mhahsler@lyle.smu.edu>

Contents

`rRDP-package` *rRDP: Interface to the RDP Classifier*

Description

This package installs and interfaces the naive Bayesian classifier for 16S rRNA sequences developed by the Ribosomal Database Project (RDP). With this package the classifier trained with the standard training set can be used or a custom classifier can be trained.

Author(s)

Maintainer: Michael Hahsler <mhahsler@lyle.smu.edu> (ORCID)

Authors:

- Nagar Anurag

See Also

Useful links:

- https://github.com/mhahsler/rRDP/
- Report bugs at https://github.com/mhahsler/rRDP/issues

`accuracy` *Calculate Classification Accuracy*

Description

Calculate the classification accuracy at a given phylogenetic level.

Usage

```
accuracy(actual, predicted, rank)

confusionTable(actual, predicted, rank)
```

Arguments

`actual`	data.frame with the actual classification hierarchy.
`predicted`	data.frame with the predicted classification hierarchy.
`rank`	rank at which the accuracy should be evaluated.

Value

The accuracy or a confusion table.

Examples

```
seq <- readRNAStringSet(system.file("examples/RNA_example.fasta",
    package = "rRDP"
))

### decode the actual classification
actual <- decode_Greengenes(names(seq))

### use RDP to predict the classification
pred <- predict(rdp(), seq)

### calculate accuracy
confusionTable(actual, pred, "genus")
accuracy(actual, pred, "genus")
```

`classification` *Decoding and Encoding Phylogenetic Classification Annotations*

Description

Functions to represent, decode and encode phylogenetic classification annotations used in FASTA files by RDP and the Greengenes project.

Usage

```
decode_Greengenes(annotation)

GenClass16S(
  Kingdom = NA,
  Phylum = NA,
  Class = NA,
  Order = NA,
  Family = NA,
  Genus = NA,
  Species = NA,
  Otu = NA,
  Org_name = NA,
  Id = NA
)

encode_Greengenes(classification)

decode_RDP(annotation)

encode_RDP(classification)
```

Arguments

annotation	Annotation from a FASTA file containing the classification information.
Kingdom	Name of the kingdom to which the organism belongs.
Phylum	Name of the phylum to which the organism belongs.
Class	Name of the class to which the organism belongs.
Order	Name of the order to which the organism belongs.
Family	Name of the family to which the organism belongs.
Genus	Name of the genus to which the organism belongs.
Species	Name of the species to which the organism belongs.
Otu	Name of the otu to which the organism belongs.
Org_name	Name of the organism.
Id	ID of the sequence.
classification	A data.frame created with GenClass16S() with the classification information.

Value

GenClass16S() and decodeX() return a data.frame. encodeX() returns a string with the corresponding annotation.

Examples

```
seq <- readRNAStringSet(system.file("examples/RNA_example.fasta",
    package = "rRDP"
))

### the FASTA annotation is read as names. This data has a Greengenes format
### annotation
names(seq)

classification <- decode_Greengenes(names(seq))
classification

### look at the Genus of all sequences
classification[, "Genus"]

### to train the RDP classifier, the annotations need to be in RDP format
annotation <- encode_RDP(classification)
names(seq) <- annotation
seq

### now we can train the classifier
customRDP <- trainRDP(seq)
customRDP

## clean up
removeRDP(customRDP)
```

rdp *Ribosomal Database Project (RDP) Classifier for 16S rRNA*

Description

Use the RDP classifier (Wang et al, 2007) to classify 16S rRNA sequences. This package contains currently RDP version 2.14 released in August 2023. The associated data package `rRDPData` contains models trained on the bacterial and archaeal taxonomy training set No. 19 (see Wang and Cole, 2024).

Usage

```
rdp(dir = NULL)

## S3 method for class 'RDPClassifier'
predict(object, newdata, confidence = 0.8, rdp_args = "", verbose = FALSE, ...)

trainRDP(x, dir = "classifier", rank = "genus", verbose = FALSE)

removeRDP(object)
```

Arguments

`dir`	directory where the classifier information is stored.
`object`	a RDPClassifier object.
`newdata`	new data to be classified as a Biostrings::DNAStringSet.
`confidence`	numeric; minimum confidence level for classification. Results with lower confidence are replaced by NAs. Set to 0 to disable.
`rdp_args`	additional RDP arguments for classification (e.g., `"-minWords 5"` to set the minimum number of words for each bootstrap trial.). See RDP documentation.
`verbose`	logical; print additional information.
`...`	additional arguments (currently unused).
`x`	an object of class Biostrings::DNAStringSet with the 16S rRNA sequences for training.
`rank`	Taxonomic rank at which the classification is learned.

Details

RDP is a naive Bayes classifier using 8-mers as features.

`rdp()` creates a default classifier trained with the data shipped with RDP. Alternatively, a directory with the data for an existing classifier (created with `trainRDP()`) can be supplied.

`trainRDP()` creates a new classifier for the data in `x` and stores the classifier information in `dir`. The data in `x` needs to have annotations in the following format:

`"<ID> <Kingdom>;<Phylum>;<Class>;<Order>;<Family>;<Genus>"`

A created classifier can be removed with `removeRDP()`. This will remove the directory which stores the classifier information.

The data for the default 16S rRNA classifier can be found in package `rRDPData`.

Value

`rdp()` and `trainRDP()` return a `RDPClassifier` object.

`predict()` returns a data.frame containing the classification results for each sequence (rows). The data.frame has an attribute called `"confidence"` with a matrix containing the confidence values.

References

Hahsler M, Nagar A (2020). "rRDP: Interface to the RDP Classifier." R Package, Bioconductor. doi:10.18129/B9.bioc.rRDP.

RDP classifier software: https://sourceforge.net/projects/rdp-classifier/

Qiong Wang, George M. Garrity, James M. Tiedje and James R. Cole. Naive Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy, Appl. Environ. Microbiol. August 2007 vol. 73 no. 16 5261-5267. doi:10.1128/AEM.0006207

Qiong W. and Cole J.R. Updated RDP taxonomy and RDP Classifier for more accurate taxonomic classification, Microbial Ecology, Announcement, 4 March 2024. doi:10.1128/mra.0106323

Examples

```
### Use the default classifier
seq <- readRNAStringSet(system.file("examples/RNA_example.fasta",
    package = "rRDP"
))

## shorten names
names(seq) <- sapply(strsplit(names(seq), " "), "[", 1)
seq

## use rdp for classification (this needs package rRDPData installed)
## > BiocManager::install("rRDPData")

cl_16S <- rdp()
cl_16S

pred <- predict(cl_16S, seq)
pred

attr(pred, "confidence")

### Train a custom RDP classifier on new data
trainingSequences <- readDNAStringSet(
    system.file("examples/trainingSequences.fasta", package = "rRDP")
)

customRDP <- trainRDP(trainingSequences)
customRDP

testSequences <- readDNAStringSet(
    system.file("examples/testSequences.fasta", package = "rRDP")
)
predict(customRDP, testSequences)

## clean up
removeRDP(customRDP)
```

Index