Package 'rRDP'

September 26, 2020

Title Interface to the RDP Classifier

Description Seamlessly interfaces RDP classifier (version 2.9).

Version 1.23.1

Date 2020-09-05

Author Michael Hahsler, Anurag Nagar

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

biocViews Genetics, Sequencing, Infrastructure, Classification, Microbiome, ImmunoOncology

Depends Biostrings (>= 2.26.2)

Suggests rRDPData

SystemRequirements Java

License GPL-2 | file LICENSE

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

git_branch master

git_last_commit 7c93cf2

git_last_commit_date 2020-09-05

Date/Publication 2020-09-25

R topics documented:

`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")
```

---

annotation *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

```
GenClass16S(Kingdom=NA, Phylum=NA, Class=NA, Order=NA,
            Family=NA, Genus=NA, Species=NA, Otu=NA,
            Org_name=NA, Id=NA)
decode_RDP(annotation)
encode_RDP(classification)
decode_Greengenes(annotation)
encode_Greengenes(classification)
```

Arguments

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.
annotation	Annotation from a FASTA file containing the classification information.
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 to classify 16S rRNA sequences. This package contains currently RDP version 2.9.

Usage

```
rdp(dir = NULL)
## S3 method for class 'RDPClassifier'
predict(object, newdata,
  confidence=.8, rdp_args="", java_args="-Xmx1g", ...)
trainRDP(x, dir="classifier", rank="genus", java_args="-Xmx1g")
removeRDP(object)
```

Arguments

dir	directory where the classifier information is stored.
object	a RDPClassifier object.
newdata	new data to be classified as a 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.
java_args	additional arguments for java (default sets the max. heap memory to 1GB).
x	an object of class DNAStringSet with the 16S rRNA sequences for training.
rank	Taxonomic rank at which the classification is learned.
...	additional arguments (currently unused).

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 attribure called "confidence" with a matrix containing the confidence values.

References

RDP Classifier http://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.

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)
pred <- predict(rdp(), 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