Package 'funbarRF'

May 27, 2019

Type Package

Title Fungal Species Identification using DNA Barcode with Random Forest

Version 1.0.2

Date 2019-05-27

Author Prabina Kumar Meher <meherprabin@yahoo.com>

Maintainer Prabina Kumar Meher <meherprabin@yahoo.com>

Depends R(>= 3.3.0)

Imports randomForest,Biostrings, BioSeqClass

LazyData TRUE

Description A machine learning based approach for fungal species identification using barcode sequence data. The multi-class random forest model has been used for prediction purpose, where the gap-pair compositional feature was used to encode the barcode sequence data. The encoded dataset was used as input for prediction purpose. Though this approach has been developed for fungal species identification in particular, can be used for other species identification as well.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-05-27 14:10:03 UTC

R topics documented:

`data_barcode` *Barcode sequences for five differet taxonomical entities i.e., Fish, Bat, Inga, Drosophila and Cypraeidae*

Description

This dataset consists of barcode sequences for both reference and query sets pertaining to the above mentioned five taxonomical entities.

Usage

```
data (data_barcode)
```

Details

These datasets have been used in earlier studies for the species identificatin using DNA barcode sequences.

Source

These datasets can be retrieved from http://cabgrid.res.in:8080/spidbar/Dataset/ED-I/.

References

1. Weitschek E., Fiscon G., and Felici G. (2014). Supervised DNA barcodes species classification: analysis, comparisons and results. *BioData Mining*, 7, 4.
2. Van Velzen R., Weitschek E., Felici G., Bakker F.T. (2012). DNA barcoding of recently diverged species: relative performance of matching methods. *PLoS One*, 7 (1), e30490.
3. Meher P.K., Sahu T.K., and Rao A.R. (2016). Identification of species based on DNA barcode using k-mer feature vector and Random forest classifier. *Gene*, 592(2), 316-324.

Examples

```
data (data_barcode)
fish_reference <- data_barcode$Fish$train
fish_query <- data_barcode$Fish$test

######################
bat_reference <- data_barcode$Bat$train
bat_query <- data_barcode$Bat$test
############################
inga_reference <- data_barcode$Inga$train
inga_query <- data_barcode$Inga$test
##################################
drosophila_reference <- data_barcode$Drosophila$train
drosophila_query <- data_barcode$Drosophila$test
```

```
###################################
cypraeidae_reference <- data_barcode$Cypraeidae$train
cypraeidae_query <- data_barcode$Cypraeidae$test
```

---

encGPC *Encoding barcode sequences using gap-pair compositional features.*

Description

This function can be used for encoding the barcode sequences with gap-pair compositional features. This is an alternative function to `seq_funbarRF` and `seq_funbarRF_manual` functions.

Usage

```
encGPC (bar_seq)
```

Arguments

`bar_seq`	Barcode sequences of class DNAStringSet.

Details

The user has to supply the barcode sequences in FASTA format, the class of which must be of DNAString type. It can also be an object generated using `read_seq_txt` function.

Value

`test`	A dataframe of *N* rows and 96 columns, where *N* is the number of input sequences supplied by the user.

Author(s)

Prabina Kumar Meher, Division of Statistical Genetics,Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References

1. Li H. (2016). BioSeqClass: Classification for biological Sequences. R package version 1.32.0.

See Also

`seq_funbarRF`, `seq_funbarRF_manual`

Examples

```
data (fun_dat)
ms <- read_seq_txt (fun_dat$seq)[1:2]
res <- encGPC (ms)
head (res)
```

`fun_dat` *A dataset of 2726 fungal barcode sequences belonging to 1363 fungal species, where each species has exactly 2 sequence.*

Description

These sequences has been collected from BOLD systems.

Usage

```
data (fun_dat)
```

Source

BOLD systems: http://www.boldsystems.org/

References

1. Ratnasingham S., and Hebert P.D.N. (2007). BOLD: the barcode of life data system. *Mol. Ecol. Notes*, 7, 355-364.

Examples

```
data (fun_dat)
```

`predict_test_funbarRF` *Prediction of species label for the query fungal barcode sequences.*

Description

This function can be used for predicting the species labels for the fungal barcode sequences of the query set, using the model trained with reference barcode sequences.

Usage

```
predict_test_funbarRF (object1, object2, m_try = 10, n_tree = 500)
```

Arguments

object1	An object created by the function seq_funbarRF or seq_funbarRF_manual , with **reference dataset** as input.
object2	An object created by the function seq_funbarRF or seq_funbarRF_manual , with **query dataset** as input.
m_try	This parameter is required for randomForest. It represents the number (must be an integer) of variables to be randomly sampled at each split. Default value is 10.
n_tree	This is also a parameter for randomForest. It denotes the number (must be an integer) of tree-based classifiers to be built. This should not be set to too small, to ensure that every instance gets predicted at least a few times. Default is 500.

Value

A dataframe consisting of predicted species label for each sequence of the query dataset.

Author(s)

Prabina Kumar Meher, Division of Statistical Genetics,Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References

1. Liaw A., and Wiener M. (2002). Classification and Regression by randomForest. *R News*, 2(3), 18-22.
2. Meher P.K., Sahu T.K., and Rao A.R. (2016). Identification of species based on DNA barcode using k-mer feature vector and Random forest classifier. *Gene*, 592(2), 316-324.

See Also

randomForest, predict_train_funbarRF, predict

Examples

```
data (data_barcode)
train1 <- seq_funbarRF_manual (manual_seq=data_barcode$Fish$train[1:30])
test1 <- seq_funbarRF_manual (manual_seq=data_barcode$Fish$test[1:3])
res1 <- predict_test_funbarRF (object1=train1, object2=test1, m_try = 10, n_tree = 5)
# kindly use large number of n_tree
print(res1)


##################################

data (data_barcode)
train2 <- seq_funbarRF_manual (manual_seq=data_barcode$Inga$train[1:30])
test2 <- seq_funbarRF_manual (manual_seq=data_barcode$Inga$test[1:3])
res2 <- predict_test_funbarRF (object1=train2, object2=test2, m_try = 10, n_tree = 20)
```

```
# kindly use large number of n_tree
print(res2)
```

`predict_train_funbarRF` *Prediction of species labels for the out-of-bag (OOB) reference barcode sequence using Random Forest.*

Description

Generally, training or reference dataset is used to train the model and not for prediction purpose. However, since Random Forest method is used here, prediction for the OOB instances is made. The OOB instances are the observations that are not participated in constructing tree-based classifiers.

Usage

```
predict_train_funbarRF (object, m_try = 10, n_tree = 500)
```

Arguments

`object`	An object created by the function `seq_funbarRF` or `seq_funbarRF_manual` .
`m_try`	This parameter is required for `randomForest`. It represents the number of variables to be randomly sampled at each split. Default value is 10.
`n_tree`	This is also a parameter for `randomForest`. It denotes the number of tree-based classifiers to be built. This should not be set to too small a number, to ensure that every instance gets predicted at least a few times. Default is 500.

Details

The user has to supply the reference sequence dataset to assess the accuracy of the developed prediction approach. Here, the prediction for the species label is made for the OOB instances and are then aggregated over all the classifiers for final prediction based on majority voting strategy.

Value

`result_train`	A dataframe consisting of species labels, number of species labels observed and correctly predicted.

Author(s)

Prabina Kumar Meher, Division of Statistical Genetics,Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References

1. Liaw A., and Wiener M. (2002). Classification and Regression by randomForest. *R News*, 2(3), 18-22.
2. Meher P.K., Sahu T.K., and Rao A.R. (2016). Identification of species based on DNA barcode using k-mer feature vector and Random forest classifier. *Gene*, 592(2), 316-324.

See Also

randomForest, predict_test_funbarRF

Examples

```
######################
data (fun_dat)
kk <- read_seq_txt (fun_dat$seq)[1:5]
zz <- as.factor(as.character (fun_dat$seq_name)[1:5])
train <- seq_funbarRF (reference_seq=kk, seq_id=zz)
res <- predict_train_funbarRF (object=train, m_try=10, n_tree=20)
# kindly use large number of n_tree
print(res)

######################

data (data_barcode)
tr_ss <- seq_funbarRF_manual (manual_seq=data_barcode$Fish$train[1:100])
prd1 <- predict_train_funbarRF (object=tr_ss, m_try=10, n_tree=500)
# kindly use large number of n_tree
print(prd1)
```

---

read_seq_txt *Conversion of DNA sequences of character types to DNAStringSet types.*

Description

This function can be used for conversion of sequence dataset of character types to DNAString format.

Usage

```
read_seq_txt (seq.file)
```

Arguments

seq.file	A character vector of barcode sequences.

Value

The function returns input sequences in DNAString format.

Author(s)

Prabina Kumar Meher, Division of Statistical Genetics,Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References

1. Pages H., Aboyoun P., Gentleman R., and DebRoy S. (2016). Biostrings: String objects representing biological sequences, and matching algorithms. R package version 2.42.1.

See Also

DNAStringSet, readDNAStringSet, XStringSet, writeXStringSet

Examples

```
data (fun_dat)
bs <- c("AATG","ATTGCCGTA","TTGAACGAAT","TGGCATTG")
read_seq_txt (bs)
kk <- read_seq_txt (fun_dat$seq)
```

`seq_funbarRF` *Conversion of barcode sequences into numeric vectors based on gap pair compositions, with user supplied barcode sequences and species labels.*

Description

This function can be used to map the sequence dataset onto numeric feature vectors, based on gap pair composition features. This function requires the barcode sequences in DNAString format and the species label of each sequence as factor. The resultant output can be directly used as input to train the random forest based prediction model.

Usage

```
seq_funbarRF (reference_seq, seq_id)
```

Arguments

`reference_seq`	Barcode sequences of class DNAStringSet. It can also be an object generated using the function `read_seq_txt`.
`seq_id`	A vctor of species labels as factor. The length of the vector must be equal to the number of sequences in reference_seq.

Details

For the argument seq_id, user has to supply the species label for each sequence in the specified format. For example, the species label *Absidia caerulea* should written as *Absidia_caerulea*. The class of the `seq_id` must be of factor type.

Value

`ref_label`	Species labels of barcode sequences as factor.
`ref_gpc`	A matrix of dimension *N**96, where *N* is the number of sequences and 96 columns represent the gap pair composition features for 0, 1, 2, 3, 4 and 5 gaps together.

Author(s)

Prabina Kumar Meher, Division of Statistical Genetics,Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References

1. Yu C.S., Chen Y.C., Lu C.H., and Hwang J.K. (2006). Prediction of protein subcellular localization. *Proteins*, 64(3), 643-651.
2. Meher P.K., Sahu T.K., Gahoi S., and Rao A.R. (2018). ir-HSP: Improved recognition of heat shock proteins, their families and sub-types based on g-spaced di-peptide features and support vector machine. *Front. Genet*., 8, 235.
3. Li H. (2016). BioSeqClass: Classification for biological Sequences. R package version 1.32.0.

See Also

`seq_funbarRF_manual`, `featureGapPairComposition`

Examples

```
data(fun_dat)
kk <- read_seq_txt (fun_dat$seq)[1:2]
zz <- as.factor (as.character(fun_dat$seq_name[1:2]))
res <- seq_funbarRF (reference_seq=kk, seq_id=zz)
print (res$ref_label)
head (res$ref_gpc)
```

`seq_funbarRF_manual` *Conversion of barcode sequences manually collected from BOLD database into numeric features based on gap pair compositions.*

Description

This function resulted in similar output as that of `seq_funbarRF` function. The only difference is in input sequences. To execute this function, the user has to collect the barcode sequences manually from the BOLD database and the same has to be supplied as input to this function.

Usage

```
seq_funbarRF_manual (manual_seq)
```

Arguments

manual_seq	Barcode sequences manually collected from the BOLD database.

Details

This function is a supplemnt to the seq_funbarRF function for mapping the manually collected barcode sequences from BOLD database into numeric feature vectors based on gap-pair compositional features.

Value

ref_label	Species labels of barcode sequences as factor.
ref_gpc	A matrix of dimension N*96, where N is the number of sequences and 96 columns represent the gap pair composition features for 0, 1, 2, 3, 4 and 5 gaps together.

Author(s)

Prabina Kumar Meher, Division of Statistical Genetics,Indian Agricultural Statistics Research Institute, New Delhi-110012, INDIA

References

1. Yu C.S., Chen Y.C., Lu C.H., and Hwang J.K. (2006). Prediction of protein subcellular localization. *Proteins*, 64(3), 643-651.
2. Meher P.K., Sahu T.K., Gahoi S., and Rao A.R. (2018). ir-HSP: Improved recognition of heat shock proteins, their families and sub-types based on g-spaced di-peptide features and support vector machine. *Front. Genet*., 8, 235.
3. Li H. (2016). BioSeqClass: Classification for biological Sequences. R package version 1.32.0.

See Also

seq_funbarRF, data_barcode, featureGapPairComposition

Examples

```
data (data_barcode)
tr_ss <- seq_funbarRF_manual (manual_seq=data_barcode$Fish$train[1:2])
print (tr_ss$ref_label)
head (tr_ss$ref_gpc)
#####################################

ts_ss <- seq_funbarRF_manual (manual_seq=data_barcode$Inga$test[1:2])
print (tr_ss$ref_label)
```

```
head (tr_ss$ref_gpc)
```

`Unite` *UNITE training dataset of 143723 sequences belonging to 9001 species.*

Description

The UNITE training dataset used in the RDP classifier was obtained from https://rdp.cme.msu.edu/classifier/classifier.jsp In the collected dataset, 145019 ITS sequences belonging to 10297 species were present. After removing 1296 species with single sequences, a dataset comprising of 143723 sequences belonging to 9001 species was prepared. This dataset can be used to train the prediction model using the proposed approach i.e., gap-pair compositional features and Random Forest method. We couldnot able to train the model with UNITE training dataset due to lack of super computing facility. However, the user can use this dataset to train the model in local server, details of which is provided in details section.

Usage

```
data (Unite)
```

Details

By using the 143723 sequences of 9001 species, the random Forest model can be trained with gap-pair compositional features. The developed trained model can be subsequently used for predicting the species labels of query barcode sequences. For step-by-step procedure, see the examples section.

References

1. Deshpande V., Wang Q., Greenfield P., Charleston M., Porras-Alfaro A., Kuske C.R., Cole J.R., Midgley D.J., and Tran-Dinh N. (2016). Fungal identification using a Bayesian classifier and the Warcup training set of internal transcribed spacer sequences. *Mycologia*, 108(1), 1-5.
2. Koljalg U., Nilsson R. H., Abarenkov K. et al. (2013). Towards a unified paradigm for sequence-based identification of fungi. *Mol. Ecol.*, 22, 5271-5277.

Examples

```
#Prepararing the trained model.
data (Unite) # Loading UNITE dataset.
trs <- Unite # Reading UNITE dataset into R.
tr<- trs[1:100]
en_tr <- encGPC (tr) # Encoding of UNITE dataset with gap-pair compositional features.
y1 <- as.factor (rownames(en_tr)) # preparing response vector.
x1 <- en_tr # Preparing predictors.
library(randomForest) # Install the "randomForest" package from CRAN.
```

```
ff <- randomForest (y=y1, x=x1, mtry=10, ntree=500) # Training with random forest technique.

#Preparing the test set.
data (fun_dat)
ms <- read_seq_txt (fun_dat$seq)[1:2] #test/query sequences.
res_enc <- encGPC (ms) #encoding of the query sequences with gap-pair compositionsl features.

#Prediction of species labels for the test set.
test_res <- predict (ff, res_enc, type="response") #prediction of labels for the query sequences.
print (test_res) #priniting the predicted labels.
```

`WarcupRDS` *Warcup training dataset which is trained with funbarRF.*

Description

The RDP Warcup ITS training set was retrieved from https://rdp.cme.msu.edu/classifier/classifier.jsp. The collected dataset comprises 17878 sequences belonging to 8551 species. After removing the 2262 singletons, a final dataset comprising 15616 sequences belonging to 6289 species was prepared.

Usage

```
data (WarcupRDS)
```

Details

This dataset can be used to train the Random Forest prediction model in a local server after installing the **funbarRF** package, which can be subsequently used for prediction of the species labels for unknown specimen. For predicting the species labels of unknown specimen, see examples section.

Source

RDP classifier Warcup ITS training dataset (https://rdp.cme.msu.edu/classifier/classifier.jsp)

References

1. Deshpande V., Wang Q., Greenfield P., Charleston M., Porras-Alfaro A., Kuske C.R., Cole J.R., Midgley D.J., and Tran-Dinh N. (2016) .Fungal identification using a Bayesian classifier and the Warcup training set of internal transcribed spacer sequences. *Mycologia*. 108(1): 1-5.

Examples

```
#Prepararing the trained model.
data (WarcupRDS) # Loading Warcup ITS training dataset.
trs <- WarcupRDS # Reading Warcup dataset into R.
tr<- trs[1:100]
en_tr <- encGPC (tr) # Encoding of Warcup dataset with gap-pair compositional features.
y1 <- as.factor (rownames(en_tr)) # preparing response vector.
x1 <- en_tr # Preparing predictors.
library(randomForest) # Install the "randomForest" package from CRAN.
ff <- randomForest (y=y1, x=x1, mtry=10, ntree=500)
# Training with random forest technique. User has to use sufficient number of ntree.
#Preparing the test set.
data (fun_dat)
ms <- read_seq_txt (fun_dat$seq)[1:2] #test/query sequences.
res_enc <- encGPC (ms) #encoding of the query sequences with gap-pair compositionsl features.
#Prediction of species labels for the test set.
test_res <- predict (ff, res_enc, type="response") #prediction of labels for the query sequences.
print (test_res) #priniting the predicted labels.
```

Index