Package 'VirFinder'

April 24, 2017

Type Package

Title VirFinder: identifying viral sequences from metagenomic data using sequence signatures

Version 1.1

Date 2017-03-28

Author Jie Ren, Nathan Ahlgren, Jed Fuhrman, Fengzhu Sun

Maintainer Jie Ren <renj@usc.edu>

Description The package provides functions to predict viral sequences in a fasta file.

Depends R (>= 2.10), glmnet, qvalue, Matrix

License GPL (>= 2)

Imports Rcpp (>= 0.12.8)

LinkingTo Rcpp

NeedsCompilation yes

R topics documented:

`VirFinder-package` *An R package for predicting viral sequences in a fasta file.*

Description

The package provides functions to predict viral sequences in a fasta file, such as the assembled contigs from metagenomic data. The method has good prediction accuracy for short (~1kb) and novel viral sequences.

Details

The prediction is based on the sequence signatures (k-tuple word frequencies) that distinguish virus from host sequences. The model was trained using all known viral sequences obtained from complete virus refSeqs and the equal number of host sequences in NCBI. For a query sequence, the number of occurrences of k-tuple words are first counted by a c++ program using a hash table. Then the sequence is predicted based on the k-tuple word frequencies using a logistic regression model trained with previously known sequences.

Users can also train the prediction model using their own database of choice. One must be careful that the sequences in the database can be trusted as truly viral and prokaryotic, respectively.

The package contains the following functions: "VF.pred", "VF.qvalue", "VF.train.user", and "VF.pred.user". "VF.pred" predicts viral sequences using the defualt trained model. "VF.train.user" and "VF.pred.user" trains and then predicts viral sequences using the user's customized database.

The package contains five data files: "VF.trainMod8mer.rda", "contigs.fa", "crAssphage.fa", "tara_virus.fa", and "tara_host.fa". "VF.trainMod8mer.rda" is the default model trained using all known viral sequences and equal number of host sequences in NCBI. "contigs.fa" is a testing example with 30 contigs asembled from a human gut metagenome. "crAssphage.fa" is the genomic sequence of crAssphage, a newly discovered virus ubiquitous in healthy human gut. "tara_virus.fa" and "tara_host.fa" are small subsets of viral and prokaryotic contigs respectively, assembled using samples from Tara Ocean Expeditions. The two fasta file are used for testing the function of training model with customized database.

Author(s)

Jie Ren, Nathan Ahlgren, Jed Fuhrman, Fengzhu Sun

Maintainer: Jie Ren <renj@usc.edu>

References

Ren J, Ahlgren N, Fuhrman J and Sun F (2017). VirFinder: identifying viral sequences from metagenomic data using sequence signatures

---

`VF.pred` *Identify viral sequences in a fasta file*

Description

The function identifies viral sequences in the input fasta file using the trained model. A score (higher value suggests higher possibility) and a p-value (lower value suggests higher possibility) will be assigned to each query sequence. For a query sequence of length L, if L < 1 kb, the model trained using 0.5 kb sequence is used for prediction; if 1 kb <= L < 3 kb, the model trained using 1000 bp sequence is used; if L>=3 kb, the model trained using 3000 bp sequences is used for prediction. The prediction model uses the feature of k-tuple word frequency of a sequence to predict if it is a viral sequence. The k-tuple length used in the model is 8.

Usage

```
VF.pred(inFaFile)
```

Arguments

inFaFile	The full file name (including path) of the fasta file

Value

The function returns a data frame. The rows correspond to sequences, and the columns are from the left to the right, sequence name (name), sequence length (length), prediction score (score), and prediction p-value (pvalue).

Examples

```
## (1) set the input fasta file name.
library(VirFinder)
inFaFile <- system.file("data", "contigs.fa", package="VirFinder")

## (2) prediction
predResult <- VF.pred(inFaFile)
predResult

## sort sequences by p-value in ascending order
predResult[order(predResult$pvalue),]

## (3) predict for crAssphage
inFaFile <- system.file("data", "crAssphage.fa", package="VirFinder")
VF.pred(inFaFile)
```

VF.pred.user — *Identify viral sequences in a fasta file using user's trained prediction model*

Description

The function identifies viral sequences in the input fasta file using the model trained based on the user's database. A score (higher value suggests higher possibility) and a p-value (lower value suggests higher possibility) will be assigned to each query sequence. For a query sequence of length L, if L < 1 kb, the model trained using 0.5 kb sequence is used for prediction; if 1 kb <= L < 3 kb, the model trained using 1 kb sequence is used; if L>=3 kb, the model trained using 3000 bp sequences is used for prediction.

Usage

```
VF.pred.user(inFaFile, VF.trainModUser)
```

Arguments

inFaFile	The full file name (including path) of the fasta file
VF.trainModUser	The trained model using user's database

Value

The function returns a data frame. The rows correspond to sequences, and the columns are from the left to the right, sequence name (name), sequence length (length), prediction score (score), and prediction p-value (pvalue).

Note

To train models using user's database of viral and prokaryotic sequences, please use the function "VF.train.user".

Examples

```
## (1) specifiy the fasta files of the training contigs
#### (1.1) one for virus and one for prokaryotic hosts
trainFaFileHost <- system.file("data", "tara_host.fa", package="VirFinder")
trainFaFileVirus <- system.file("data", "tara_virus.fa", package="VirFinder")

#### (1.2) specify the directory where the trained model will be saved to, and the name of the model
userModDir <- file.path(find.package("VirFinder"))
userModName <- "modTara"

## (2) train the model using user's database
w <- 4  # the length of the k-tuple word
VF.trainModUser <- VF.train.user(trainFaFileHost, trainFaFileVirus, userModDir,
userModName, w, equalSize=TRUE)

## (3) predict the contigs using the customized model
#### (3.1) specify the fasta file containing contigs for prediction
inFaFile <- system.file("data", "contigs.fa", package="VirFinder")

#### (3.2) prediction
predResultUser <- VF.pred.user(inFaFile, VF.trainModUser)
predResultUser

#### (3.3) sort sequences by p-value in ascending order
predResultUser[order(predResultUser$pvalue),]
```

VF.qvalue *Estimate the false discovery rates (q-values) using p-values*

Description

The function estimates the false discovery rates (q-values) for a given set of p-values. The q-value measures the proportion of false positives incurred when predicting viral sequences using the corresponding p-value as a threshold. The function VF.qvalue uses the function "qvalue" in the R package "qvalue" by John D. Storey.

Usage

```
VF.qvalue(pvalue, fdr.level = NULL, pfdr = FALSE, ...)
```

Arguments

pvalue	The predicted p-values for sequences obtained from the function "VF.pred"
fdr.level	A level at which to control the FDR. Must be in (0,1]. Optional; if this is selected, a vector of TRUE and FALSE is returned that specifies whether each q-value is less than fdr.level or not.
pfdr	An indicator of whether it is desired to make the estimate more robust for small p-values and a direct finite sample estimate of pFDR - optional.
...	Additional arguments passed to 'pi0est' and 'lfdr'.

Details

Please refer to the function "qvalue" in the package "qvalue" for details.

<URL:http://www.bioconductor.org/packages/release/bioc/html/qvalue.html>

Value

the q-values for controling the porportion of host sequences in the predicted viral sequences.

References

Storey JD. (2002) A direct approach to false discovery rates. Journal of the Royal Statistical Society, Series B, 64: 479-498. <URL: http://onlinelibrary.wiley.com/doi/10.1111/1467-9868.00346/abstract>
Storey JD and Tibshirani R. (2003) Statistical significance for genome-wide experiments. Proceedings of the National Academy of Sciences, 100: 9440-9445. <URL: http://www.pnas.org/content/100/16/9440.full>

Examples

```
## (1) set the input fasta file name.
library(VirFinder)
inFaFile <- system.file("data", "contigs.fa", package="VirFinder")

## (2) prediction
predResult <- VF.pred(inFaFile)
predResult

#### (2.1) sort sequences by p-value in ascending order
predResult[order(predResult$pvalue),]

#### (2.2) estimate q-values (false discovery rates) based on p-values
predResult$qvalue <- VF.qvalue(predResult$pvalue)

#### (2.3) sort sequences by q-value in ascending order
predResult[order(predResult$qvalue),]
```

VF.train.user *Train virus prediction model using user's database*

Description

The function trains the viral contig prediction model using user's customized database containing viral sequences and prokaryotic sequences.

The sequences are first splitted into fragments of fixed lengths, such as 0.5 kb, 1 kb and 3kb. , and the k-tuple word frequency are counted for each fragment. The prediction model is trained based on the k-tuple frequencies of viral and prokaryotic fragments.

Usage

```
VF.train.user(trainFaFileHost, trainFaFileVirus, userModDir, userModName, w, equalSize)
```

Arguments

trainFaFileHost	The fasta file (including path) of the prokaryotic sequences.
trainFaFileVirus	The fasta file (including path) of the viral sequences.
userModDir	The directory where the trained model will be saved.
userModName	The name of the model file.
w	The length of the k-tuple word.
equalSize	If use the same number of virus and host fragments to train? Default is FALSE.

Value

The function returns the trained model. There models are trained based on three different fragment lengths, 0.5 kb, 1 kb and 3 kb, respectively. The model for 0.5 kb will be used for predicting query sequences of length < 1 kb, the model for 1 kb will be used for predicting sequences of length ranging from 1 kb to 3 kb, and the model for 3 kb will be used for predicting sequences of length > 3kb. The model is trained based on k-tuple frequencies, which can be specified using w.

The trained model will also be saved into the user's specified directory, for easily loading and calling it in the future.

Note

One must be careful that the sequences in the database can be trusted as truly viral and prokaryotic, respectively. The assembled contigs in metagenome could possibly have viral sequences, and similarity the contigs in metavirome could also have host sequences.

The longer k-tuple can describe better the difference between virus and host sequences, but if the data is not enough, it can lead to an overfitting problem. Given the database, users are suggested to test different lengths of k-tuple in order to get the best model.

Examples

```
## (1) train the model using user's database
#### (1.1) specifiy the fasta files of the training contigs, one for virus and one for prokaryotic hosts
trainFaFileHost <- system.file("data", "tara_host.fa", package="VirFinder")
trainFaFileVirus <- system.file("data", "tara_virus.fa", package="VirFinder")

#### (1.2) specify the directory where the trained model will be saved to, and the name of the model
userModDir <- file.path(find.package("VirFinder"))
userModName <- "modTara"

## (2) train the model using user's database
w <- 4  # the length of the k-tuple word
VF.trainModUser <- VF.train.user(trainFaFileHost, trainFaFileVirus, userModDir,
userModName, w, equalSize=TRUE)

## (3) load the trained model based on user's database
modFile <- list.files(userModDir, userModName, full.names=TRUE)
load(modFile)
```

`VF.trainMod8mer` *The prediction models trained using sequences of various lengths from virus and host complete genomes.*

Description

The data contains three prediction models trained using 500, 1000 and 3000 bp viral and host sequences respectively. The viral sequences were generated by spliting virus complete genomes into non-overlapping fragments. The same number of host non-overlapped fragments were choosen to pair with the viral fragments as training data. Each sequence in the training data is then represented using a 8-tuple word frequency vector. The classification model for virus and host sequences is trained based on their word frequencies.

The data also contains three distributions of prediction scores for 500, 1000 and 3000 bp host sequences respectively. The distributions are used as null distributions to compute p-values for query sequences.

Usage

```
data("VF.trainMod8mer")
```

Format

This variable is for internal use. If users are interested, here is the description of the data.

The value of VF.trainMod8mer is 8, because the model was constructed using 8-tuple word frequencies.

The three prediction models are saved in attr(VF.trainMod8mer, "lasso.mod_0.5k"), attr(VF.trainMod8mer, "lasso.mod_1k"), and attr(VF.trainMod8mer, "lasso.mod_3k")

The three null distributions are saved in attr(VF.trainMod8mer, "nullDis_0.5k"), attr(VF.trainMod8mer, "nullDis_1k"), and attr(VF.trainMod8mer, "nullDis_3k")

Examples

```
data(VF.trainMod8mer)
```

Index