Package 'VirFinder'

November 29, 2016

Type Package

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

Version 1.0

Date 2016-11-28

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 glmnet, qvalue

License GPL (>= 2)

Imports Rcpp (>= 0.12.8)

LinkingTo Rcpp

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 noval 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 equal number of known viral and host sequences. 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.

The package contains, two functions: VF.pred, VF.qvalue, and

three data: VF.mod8mer, contigs.fa, crAssphage.fasta

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 < 1000, the model trained using 500 bp sequence is used for prediction; if 1000 <= L < 3000, the model trained using 1000 bp sequence is used; if L>=3000, the model trained using 3000 bp sequences is used for prediction.

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

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

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

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

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

`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

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

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

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

`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