Package 'Lncident'

October 8, 2016

Type Package

Title Long Non-Coding RNA identification and Find Open Reading Frame

Version 1.3.3

Author Han Siyu

Maintainer Han Siyu <han.siyu@outlook.com>

Description Functions for predicting sequences are mRNAs or long non-coding RNAs. The default model is trained on human dataset by employing SVM, but the models can be built on users' own data. Also has function to help find the ORFs of the sequences. Supports the format of FASTA.

Depends R (>= 3.1.0), seqinr (>= 3.1-3), e1071 (>= 1.6-7), parallel (>= 3.1.0)

License GPL-3

LazyData TRUE

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

`extract_features` *Extract the Sequence Features*

Description

This is a funcion to extract the features when users want to build their own model.

Usage

```
extract_features(Seq, label = NULL, with.parallel = TRUE,
  cl.core = parallel::detectCores())
```

Arguments

Seq	Are the sequences that users want to extract the features. Should be an object of the class SeqFastadna.
label	Optional. A character which indicates the label of the sequences.
with.parallel	Logical. If TRUE (Default), the process will be run in parallel.
cl.core	The number of cores used to create cluster. Defualt value is all the CPU cores available. (Obtain by function parallel::detectCores())

Details

This is a funcion to extract the features. For each sequence, there will be 1366 features which consist of the length, coverage of the longest ORF and the frequencies of 1~5 adjoining-base(s) in the longest ORF region. For 1 base, there will be A/C/G/T, i.e. 4 features, and for 2 adjoining-bases, there will be AA/AC/AG...TC/TG/TT, 4^2 features. That is to say there will be 4+4^2+4^3+4^4+4^5 features of frequency. If there are more than one longest ORF, only the first one will be considered. And if there is no ORF in the sequence, the length and coverage of the sequence will be 0 while the frequencies will be calculated on the whole sequence. Users can use the data frame returned by this function to build their own model. To build a svm model, users can use "svm" function of package "e1071" and please refer to its documentation for further details. The package "e1071" will be loaded antumatically when users employ the package "lncpred". The extraction of ORF features is based on Avril Coghlan's R code. The original code is to find the positions of the start and stop codons of the ORFs, and the code is slightly modified because the package it depends on has updated.

Value

Returns a data frame contains the features. The values are numeric. If uses provide a label, the values of the "label" column are factors.

Author(s)

Han Siyu (The extraction of ORF is based on the code by Avril Coghlan)

Examples

```
### Use the "read.fasta" function of package "seqinr" to read the file: ###
Seqs <- read.fasta(file =
"http://www.ncbi.nlm.nih.gov/WebSub/html/help/sample_files/nucleotide-sample.txt")
### Without the label: ###
features_data <- extract_features(Seqs)
### Label attached to every sequence: ###
features_data1 <- extract_features(Seqs[1:3], "label one")
features_data2 <- extract_features(Seqs[4:6], "label two")
training_set <- rbind(features_data1, features_data2)
### Users can use "svm" function of package "e1071" to build a new model. ###
### The label needs to be attached before training the new model. ###
### This is only an example and you may see some warnings.  ###
new_model <- svm(label ~ ., data = training_set, scale = TRUE,
                 kernel = "radial", probability = TRUE)
test_set <- extract_features(Seqs[7])
pred <- predict(new_model, test_set, probability = TRUE)
### For further details of function "svm" please refer to the documentation of package "e1071". ###
```

find_orfs *Find the ORFs*

Description

This is a function to find the ORFs in one sequence.

Usage

```
find_orfs(OneSeq)
```

Arguments

OneSeq	Is one sequence. Can be an object of the class SeqFastadna or just the class character contains the sequence.

Details

This function can extract all ORFs of one sequence. It returns the regions, lengths and coverages of ORFs. Coverage is the the ratio of ORF to transcript lengths. If there are no ORF in one sequence, the first row will display "No ORF is found in the sequence" while the length and coverage will be zero. The package "seqinr" will be attached automatically when the package "lncpred" are loaded. Users can use the function of "seqinr" to handle their data. This function is developed from the Avril Coghlan's R code which is used to find the positions of ORFs. The original code is slightly modified here because the package it depends on has updated.

Value

Returns a data frame. The first row is the ORF regions, the second row is the lengths of the ORFs and the third row is the ORFs' coverages.

Author(s)

Han Siyu (Developed from the R code by Avril Coghlan)

Examples

```
### For one sequence: ###
OneSeq <- c("cccatgcccagctagtaagcttagcc")
Seq_ORF1 <- find_orfs(OneSeq)
### For a FASTA file contains several sequences: ###
### Use the "read.fasta" function of package "seqinr" to read the file: ###
Seqs <- read.fasta(file =
"http://www.ncbi.nlm.nih.gov/WebSub/html/help/sample_files/nucleotide-sample.txt")
### Use apply function to find ORFs: ###
Seq_ORF2 <- sapply(Seqs, find_orfs)
```

`lncident` *LncRNAs Identification (Default Model)*

Description

Using the default model to identify the sequences.

Usage

```
lncident(Seq, species = "human", with.parallel = TRUE,
  cl.core = parallel::detectCores(), detail = FALSE)
```

Arguments

`Seq`	The sequence(s) needed to be identified. Can be an object of the class SeqFastadna or the class character.
`species`	A String indicates the species name. Use "human", "mouse" or "c.elegans" to specify which model is used to predict the sequences.
`with.parallel`	Logical. If TRUE (Default), the process will be run in parallel.
`cl.core`	The number of cores used to create cluster. Defualt value is all the CPU cores available. (Obtain by function parallel::detectCores())
`detail`	If TRUE, the result will provide the class of the sequence as well as the coding potential and the length and coverage of the longest ORF. Else, only the class of the sequences will be returned.

Details

Utilizing the default model to predict the sequences. The default model for "human" is trained on human's long non-coding RNAs(lncRNAs) and protein-coding transcripts, the model for "mouse" is trained on mouse's lncRNAs and coding sequences(CDs); and the model for "c.elegans" is trained on non-coding RNAs (ncRNAs) and CDs of Caenorhabditis elegans, this model can also be applied to some invertebrata such as Saccharomyces cerevisiae.

Value

Returns a data frame indicates the class of the sequence. The coding potential and information of the longest ORF will also be provided if the "detail" option is set as TRUE.

Author(s)

Default training model is developed by Han siyu. The model is built with the LIBSVM in e1071 package.

Examples

```
### Use the "read.fasta" function of package "seqinr" to read the file: ###
Seqs <- read.fasta(file =
"http://www.ncbi.nlm.nih.gov/WebSub/html/help/sample_files/nucleotide-sample.txt")
### Predict the Sequences: ###
pred1 <- lncident(Seqs, detail = TRUE)
### Predict one Sequence: ###
```

```
pred2 <- lncident(c("cccatgcccagctagtaagcttagcc"), species = "c.elegans", with.parallel = TRUE, detail = FALS
### Note that species name needs to be string and has no mistake. ###
pred3 <- lncident(c("cccatgcccagctagtaagcttagcc"), species = "C.elegans", detail = FALSE)
### Will get a warning. ###
```

Index