Package 'ORFhunteR'

October 16, 2026

Type Package

biocViews Technology, StatisticalMethod, Sequencing, RNASeq, Classification, FeatureExtraction

Title Predict open reading frames in nucleotide sequences

Version 1.21.0

Description The ORFhunteR package is a R and C++ library for an automatic determination and annotation of open reading frames (ORF) in a large set of RNA molecules. It efficiently implements the machine learning model based on vectorization of nucleotide sequences and the random forest classification algorithm. The ORFhunteR package consists of a set of functions written in the R language in conjunction with C++. The efficiency of the package was confirmed by the examples of the analysis of RNA molecules from the NCBI RefSeq and Ensembl databases. The package can be used in basic and applied biomedical research related to the study of the transcriptome of normal as well as altered (for example, cancer) human cells.

BugReports https://github.com/rfctbio-bsu/ORFhunteR/issues

License MIT License

Encoding UTF-8

LazyData true

Imports Rcpp (>= 1.0.3), BSgenome.Hsapiens.UCSC.hg38, data.table, stringr, randomForest, xfun, stats, utils, parallel, graphics

Depends Biostrings, rtracklayer, Peptides

LinkingTo Rcpp

Suggests knitr, BiocStyle, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.1.1.9000

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

git_branch devel

git_last_commit 6287669

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-10-15

Author Vasily V. Grinev [aut, cre] (ORCID:
<https://orcid.org/0000-0001-9981-7333>),
Mikalai M. Yatskou [aut],
Victor V. Skakun [aut],
Maryna Chepeleva [aut] (ORCID: <https://orcid.org/0000-0003-3036-4916>),
Petr V. Nazarov [aut] (ORCID: <https://orcid.org/0000-0003-3443-0298>)

Maintainer Vasily V. Grinev <grinev_vv@bsu.by>

Contents

annotateORFs 2
classifyORFsCandidates 3
codonStartStop 4
findORFs 5
findPTCs 6
getSeqORFs 7
loadTrExper 8
predictORF 8
translateORFs 9
vectorizeORFs 10

Index **11**

`annotateORFs` *Annotate open reading frames*

Description

Annotate the open reading frames identified in nucleotide sequences of interest.

Usage

```
annotateORFs(orfs, tr, gtf = NULL, prts, workDir = NULL)
```

Arguments

`orfs`	character string giving the name of tab-delimited TXT file with coordinates of open reading frame(-s). This file should include three mandatory fields: i) transcript_id - transcript ID; ii) start - start coordinate of open reading frame in a transcript; iii) end - end coordinate of open reading frame in a transcript.
`tr`	character string giving the name of file with transcript(-s) of interest. This file must include the transcript(-s) for which the open reading frame(-s) was/were identified and listed in above orfs file. Valid format is "fasta" or "fa".
`gtf`	character string giving the name of GTF/GFF file with annotated transcripts of interest. Default value is NULL.
`prts`	character string giving the name of FASTA file with sequences of in silico translated proteins encoded by identified open reading frames.
`workDir`	character string giving the path to and name of work directory. NULL by default that mean the current working directory.

Value

data.frame object with columns:

transcript_id	transcript ID.
f_utr.length	length of 5'UTRs.
start.codon	type of start codon.
orf.start	start coordinate of open reading frames.
orf.stop	stop coordinate of open reading frames.
stop.codon	type of stop codon.
stop.status	PTC status of stop codon.
orf.length	length of open reading frames.
t_utr.length	length of 3'UTRs.
MW	molecular weight.
pI	isoelectic point of a protein sequence.
indexPPI	potential protein interaction index.

Author(s)

Vasily V. Grinev

Examples

```
orfs_path <- system.file("extdata",
                         "Set.trans_ORFs.coordinates.txt",
                         package="ORFhunteR")
tr_path <- system.file("extdata",
                       "Set.trans_sequences.fasta",
                       package="ORFhunteR")
gtf_path <- system.file("extdata",
                        "Set.trans_sequences.gtf",
                        package="ORFhunteR")
prts_path <- system.file("extdata",
                         "Set.trans_proteins.sequences.fasta",
                         package="ORFhunteR")
anno_orfs <- annotateORFs(orfs=orfs_path,
                          tr=tr_path,
                          gtf=gtf_path,
                          prts=prts_path,
                          workDir=NULL)
```

---

classifyORFsCandidates

Classify the pseudo and true ORF candidates derived from RNA molecules

---

Description

Clussify the pseudo and true ORF candidates.

Usage

```
classifyORFsCandidates(
  ORFLncRNAs,
  ORFmRNAs,
  pLearn = 0.75,
  nTrees = 500,
  modelRF = NULL,
  workDir = NULL,
  showAccuracy = FALSE
)
```

Arguments

ORFLncRNAs	the oject of the class list of non-coding pseudo ORFs.
ORFmRNAs	the oject of the class list of coding true ORFs.
pLearn	probability threshold for the "traing" class of ORFs. Default value is 0.75.
nTrees	nuber of the decision trees in randomForest. Default value is 500.
modelRF	character string giving the name of RDS-file to store the classification model. NULL by default.
workDir	character string giving the path to and name of work directory. NULL by default that mean the current working directory.
showAccuracy	logic TRUE or FALSE. Use TRUE for print accuracy metrics. Default value is FALSE

Value

The classificator object of the class randomForest.

Author(s)

Mikalai M. Yatskou

Examples

```
## Not run:
clt <- classifyORFsCandidates(ORFLncRNAs, ORFmRNAs)

## End(Not run)
```

---

codonStartStop *Identify all potential start and stop codons in a nucleotide sequence*

Description

This function scans a nucleotide sequence of interest in the search of canonical start codon ATG or non-canonical start codons GTG, TTG and CTG as well as stop codons TAA, TAG and TGA.

Usage

```
codonStartStop(x)
```

Arguments

x	character string giving the nucleotide sequence.

Value

list of potential start and stop codons with their coordinates.

Author(s)

Vasily V. Grinev

Examples

```
codons <- codonStartStop(x = "AAAATGGCATGGTAAGTC")
```

---

findORFs *Identify all variants of open reading frames in a nucleotide sequence*

Description

Identify all possible variants of open reading frames in a nucleotide sequence of interest.

Usage

```
findORFs(x, codStart = "ATG")
```

Arguments

x	character string giving the nucleotide sequence of interest.
codStart	character string with type of start codon: "ATG", "GTG", "TTG" or "CTG". Default value is "ATG".

Value

matrix with start and stop positions, length and sequence of identified variants of open reading frames.

Author(s)

Vasily V. Grinev

Examples

```
x <- "AAAATGGCTGCGTAATGCAAAATGGCTGCGAATGCAAAATGGCTGCGAATGCCGGCACGTTGCTACGT"
orf <- findORFs(x = x, codStart = "ATG")
```

findPTCs *Identify the premature termination codons in nucleotide sequences*

Description

Identify the premature termination codons in nucleotide sequences of interest.

Usage

```
findPTCs(orfs, gtf, workDir = NULL)
```

Arguments

orfs	character string giving the name of tab-delimited TXT file with coordinates of open reading frame(-s). This file should include four mandatory fields: i) transcript_id - transcript ID; ii) start - start coordinate of open reading frame in a transcript; iii) end - end coordinate of open reading frame in a transcript; iv) length - length of open reading frame.
gtf	character string giving the name of GTF/GFF file with annotated transcripts of interest. Valid format is "gtf" or "gff".
workDir	character string giving the path to and name of work directory. NULL by default that mean the current working directory.

Value

data.frame object with start and stop positions, length and stop status of codons for each transcript ID.

Author(s)

Vasily V. Grinev

Examples

```
orfs_path <- system.file("extdata",
                         "Set.trans_ORFs.coordinates.txt",
                         package="ORFhunteR")
gtf_path <- system.file("extdata",
                        "Set.trans_sequences.gtf",
                        package="ORFhunteR")
ptcs <- findPTCs(orfs = orfs_path,
                 gtf = gtf_path,
                 workDir = NULL)
```

getSeqORFs *Extract the sequences of identified open reading frames*

Description

Extract the sequences of identified open reading frames.

Usage

```
getSeqORFs(orfs, tr, genome = "BSgenome.Hsapiens.UCSC.hg38", workDir = NULL)
```

Arguments

orfs	character string giving the name of tab-delimited TXT file with coordinates of open reading frame(-s). This file should include three mandatory fields: i) transcript_id - transcript ID; ii) start - start coordinate of open reading frame in a transcript; iii) end - end coordinate of open reading frame in a transcript.
tr	character string giving the name of file with transcript(-s) of interest. This file must include the transcript(-s) for which the open reading frame(-s) was/were identified and listed in above orfs file. Valid formats are "gtf", "gff", "fasta" and "fa".
genome	character string giving the name of BSgenome data package with full genome sequences. Default value is "BSgenome.Hsapiens.UCSC.hg38".
workDir	character string giving the path to and name of work directory. NULL by default that mean the current working directory.

Value

DNAStringSet object with sequences of extracted open reading frames.

Author(s)

Vasily V. Grinev

Examples

```
orfs_path <- system.file("extdata",
                         "Set.trans_ORFs.coordinates.txt",
                         package = "ORFhunteR")
tr_path <- system.file("extdata",
                       "Set.trans_sequences.fasta",
                       package = "ORFhunteR")
seq_orfs <- getSeqORFs(orfs = orfs_path,
                       tr = tr_path,
                       genome = "BSgenome.Hsapiens.UCSC.hg38",
                       workDir = NULL)
```

loadTrExper *Load a set of transcripts*

Description

Load a set of experimental transcripts.

Usage

```
loadTrExper(tr, genome = "BSgenome.Hsapiens.UCSC.hg38", workDir = NULL)
```

Arguments

tr	character string giving the name of file with experimental transcripts. Allowed file formats are "fasta", "fa", "gtf" or "gff".
genome	character string giving the name of BSgenome data package with full genome sequences. Default value is "BSgenome.Hsapiens.UCSC.hg38".
workDir	character string giving the path to and name of work directory. NULL by default that mean the current working directory.

Value

List of loaded transcript sequences.

Author(s)

Vasily V. Grinev

Examples

```
trans <- system.file("extdata",
                     "Set.trans_sequences.fasta",
                     package = "ORFhunteR")
trans_seq <- loadTrExper(tr = trans)
```

predictORF *Predict the true ORFs in mRNA molecules*

Description

Predict the true ORFs in mRNA molecules.

Usage

```
predictORF(
  tr,
  genome = "BSgenome.Hsapiens.UCSC.hg38",
  model = NULL,
  prThr = 0.5,
  workDir = NULL
)
```

Arguments

tr	character string giving the name of file with experimental transcripts. Allowed file formats are "fasta", "fa", "gtf" or "gff".
genome	character string giving the name of BSgenome data package with full genome sequences. Default value is "BSgenome.Hsapiens.UCSC.hg38".
model	character string giving the connection or full path to the file from which the classification model is read. Use default NULL value to use our default model.
prThr	probability threshold for the "winning" class of ORFs. Default value is 0.5.
workDir	character string giving the path to and name of work directory. NULL by default that mean the current working directory.

Value

The coordinates of ORFs in mRNA molecules of interest.

Author(s)

Mikalai M. Yatskou

Examples

```
## Not run:
tr_path <- system.file("extdata",
                       "Set.trans_sequences.fasta",
                       package="ORFhunteR")
model <- "http://www.sstcenter.com/download/ORFhunteR/classRFmodel_1.rds"
ORFs <- predictORF(tr=tr_path, model=model)

## End(Not run)
```

---

translateORFs *Translate open reading frames to proteins*

Description

Translate the identified open reading frames to proteins.

Usage

```
translateORFs(seqORFs, aaSymbol = 1, workDir = NULL)
```

Arguments

seqORFs	character string giving the name of FASTA/FA file with sequences of identified open reading frames.
aaSymbol	type of amino acid symbols (one- or three-letter coding). Default value is 1.
workDir	character string giving the path to and name of work directory. NULL by default that mean the current working directory.

Value

AAStringSet object with protein sequences.

Author(s)

Vasily V. Grinev

Examples

```
seq_orf_path <- system.file("extdata",
                            "Set.trans_ORFs.sequences.fasta",
                            package = "ORFhunteR")
prot_seqs <- translateORFs(seqORFs = seq_orf_path)
```

`vectorizeORFs` *Vectorize the ORF(-s) sequence(-s) to sequence features*

Description

Vectorize the ORF(-s) sequence(-s) to sequence features.

Usage

```
vectorizeORFs(x)
```

Arguments

x	DNAStringSet object with ORF(-s) sequence(-s).

Value

The object of class data.frame with ORF(-s) vectorized into sequence features.

Author(s)

Mikalai M. Yatskou

Examples

```
x = DNAStringSet(x = "ATGGGCCTCA")
feats <- vectorizeORFs(x = x)
```

Index

annotateORFs, 2

classifyORFsCandidates, 3
codonStartStop, 4

findORFs, 5
findPTCs, 6

getSeqORFs, 7

loadTrExper, 8

predictORF, 8

translateORFs, 9

vectorizeORFs, 10