

# Package ‘utr.annotation’

August 23, 2021

**Type** Package

**Title** Annotate Variants in the Untranslated Regions

**Version** 1.0.4

**Maintainer** Yating Liu <yatinghust@gmail.com>

**Description** A fast, user-friendly tool for annotating potential deleterious variants in the untranslated regions for both human and mouse species. Y Liu, JD Dougherty (2021) <[doi:10.1101/2021.06.23.449510](https://doi.org/10.1101/2021.06.23.449510)>.

**License** GPL (>= 3)

**Encoding** UTF-8

**SystemRequirements** Linux or MacOS Intel chip tensorflow (1.14.0)

**Depends** R (>= 3.5.0)

**Imports** parallel, doParallel, data.table, readr, stringr, biomaRt, vcfR, Biostrings, dplyr, tidyr, keras, AnnotationHub, ensemblDb, AnnotationFilter, BiocGenerics, GenomicFeatures, GenomicRanges, IRanges, foreach, rtracklayer, stats, utils, xml2

**Suggests** testthat, knitr, rmarkdown, ggplot2

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Yating Liu [aut, cre] (<<https://orcid.org/0000-0001-7425-0422>>), Joseph D. Dougherty [aut] (<<https://orcid.org/0000-0002-6385-3997>>)

**Repository** CRAN

**Date/Publication** 2021-08-23 06:40:02 UTC

## R topics documented:

concatenateAnnotationResult . . . . .	2
initUTRAnnotation . . . . .	3
partitionVariantFile . . . . .	4
runUTRAnnotation . . . . .	5

---

concatenateAnnotationResult

*Concatenate annotation result files into one file*

---

### Description

concatenateAnnotationResult Concatenate annotation result files into one file

### Usage

```
concatenateAnnotationResult(  
  varResultPath,  
  annotationFinalResult,  
  cores = NULL,  
  verbose = FALSE  
)
```

### Arguments

varResultPath    annotation result directory  
annotationFinalResult  
                  final annotation file  
cores            (optional) number of cores you'd like to use, by default would use a single core.  
verbose          Whether print diagnostic messages. The default is FALSE.

### Value

write concatenated annotation file to annotationFinalResult

### Examples

```
## Not run:  
# After we run UTR annotation on all partition variant files and  
# store the annotations in "partition_results" for example,  
# we can concatenate the annotation results into one file and  
# store it in concatenated_annotation.csv  
concatenateAnnotationResult(varResultPath = "partition_results",  
                              annotationFinalResult = "concatenated_annotation.csv")  
  
## End(Not run)
```

---

initUTRAnnotation	<i>Query transcripts regions and sequences from Ensembl database</i>
-------------------	--

---

### Description

initUTRAnnotation query transcripts regions, UTRs and coding sequences from Ensembl database, which will be used by runUTRAnnotation to do UTR annotation.

### Usage

```
initUTRAnnotation(  
  variantFile,  
  species,  
  ensemblVersion,  
  getTranscript = TRUE,  
  format = "csv",  
  dataDir = NULL,  
  verbose = FALSE  
)
```

### Arguments

variantFile	a CSV file with Chr, Pos, Ref, Alt
species	either human or mouse
ensemblVersion	(optional) a number specifying which version of Ensembl annotation you'd like to use, by default use the latest version
getTranscript	(optional) Whether to get ids of the transcripts that overlap with all the variants. If the number of variants is too large (for example > 100,000), set it to FALSE and do this in runUTRAnnotation on each partition in parallel.
format	(optional) csv or vcf, the default is csv
dataDir	(optional) path to the store the database information, if not specified will create a folder named as input variant file name with a "db_" prefix
verbose	Whether print diagnostic messages. The default is FALSE.

### Value

A variant table with Transcript column which contains the ids of the transcripts that overlap with the variants

### Examples

```
test_variant_file <- system.file("extdata", "variants_sample.csv", package = "utr.annotation")  
initUTRAnnotation(variantFile = test_variant_file,  
  species = "human",  
  ensemblVersion = 93,
```

```
dataDir = "test_db")
```

---

partitionVariantFile *Partition the variant file to run in parallel*

---

### Description

partitionVariantFile is used to partition the variant file into small partition files by user defined size or number

### Usage

```
partitionVariantFile(
  variantFile,
  chunkSize = NULL,
  chunkNum = NULL,
  chunkPath = "chunks",
  species,
  ensemblVersion = NULL,
  overwrite = FALSE,
  dataDir = NULL,
  getTranscript = TRUE,
  format = "csv",
  verbose = FALSE
)
```

### Arguments

variantFile	variant file in CSV format
chunkSize	Partition the variant file into chunk files, which has a certain number of rows
chunkNum	Partition the variant file into a certain number of chunk files
chunkPath	(optional) A file directory to store all partition files. By default will output to "chunks" folder
species	human or mouse
ensemblVersion	(optional) a number specifying which version of Ensembl annotation you'd like to use, by default use the latest version
overwrite	(optional) If chunkPath already exists and not empty, whether or not to overwrite it. By default, do not overwrite and will raise error.
dataDir	(optional) path to store database information, if not specified will create a folder named as input variant file name with a "db_" prefix
getTranscript	(optional) Whether to get ids of the transcripts that overlap with all the variants. The default value is TRUE. If the number of variants is too large (for example > 100,000), set it to FALSE and do this in runUTRAnnotation on each partition in parallel.
format	(optional) csv or vcf, the default is csv
verbose	Whether print diagnostic messages. The default is FALSE.

**Value**

a list of partition variant files

**Examples**

```
variants_sample <- system.file("extdata", "variants_sample.csv", package = "utr.annotation")

# Partition variants_sample file equally into 3 variant files
# and store them in user specified chunkPath folder
partitionVariantFile(variantFile = variants_sample,
                     chunkNum = 3,
                     chunkPath = "chunks_3",
                     species = "human",
                     ensemblVersion = 93,
                     dataDir = "db_all_variants")

# Partition variants_sample file into smaller variant files each of which contains 7 variants,
# and store them in user specified chunkPath folder
partitionVariantFile(variantFile = variants_sample,
                     chunkSize = 7,
                     chunkPath = "chunks_7_vars",
                     species = "human",
                     ensemblVersion = 93,
                     dataDir = "db_all_variants")

unlink("db_all_variants", recursive = TRUE)
unlink("chunks_3", recursive = TRUE)
unlink("chunks_7_vars", recursive = TRUE)
```

---

runUTRAnnotation

*Run UTR annotation on a variant file*

---

**Description**

Run UTR annotation on a variant file to check whether a variant alter any of the feature element in UTRs, return a CSV file with annotated information

**Usage**

```
runUTRAnnotation(
  variantFile,
  annotationResult,
  species,
  ensemblVersion = NULL,
  dataDir = NULL,
  conservationBwFiles = NULL,
  cores = NULL,
  format = "csv",
```

```

    mrl_prediction = TRUE,
    verbose = FALSE
)

```

### Arguments

**variantFile** a CSV file with Chr, Pos, Ref, Alt

**annotationResult** output annotation file

**species** species, either human or mouse

**ensemblVersion** (optional) a number specifying which version of Ensembl annotation you'd like to use, by default use the latest version

**dataDir** (optional) path to the store / extract database information, if not specified will create a folder named as input variant file name with a "db\_" prefix

**conservationBwFiles** (optional) path to a folder which contains one or more conservation bigWig files. If not set, will skip conservation analysis.

**cores** (optional) number of cores to use for parallel computing. If not specified, use a single core by default.

**format** (optional) csv or vcf, the default is csv

**mrl\_prediction** (optional) Whether do MRL prediction and check if it increase or decrease. The default is TRUE.

**verbose** Whether print diagnostic messages. The default is FALSE.

### Value

A CSV file with annotated variants

### Examples

```

test_variant_file <- system.file("extdata", "variants_sample.csv", package = "utr.annotation")

# run UTR annotation with 1 core and skip getting conservation scores for variant positions
runUTRAnnotation(variantFile = test_variant_file,
                 annotationResult = "annotated_variants_sample.csv",
                 species = "human",
                 ensemblVersion = 93,
                 dataDir = "test_db")
unlink("test_db", recursive = TRUE)
unlink("annotated_variants_sample.csv")

## Not run:
# Download BigWig format conservation track files hg38.phastCons100way.bw and hg38.phyloP100way.bw
# from UCSC Genome Browser to a folder (here we name it folder_store_conservation_bw_files)
Conservation_scores <- "folder_store_conservation_bw_files"
runUTRAnnotation(variantFile = test_variant_file,
                 annotationResult = "annotated_variants_sample.csv",

```

*runUTRAnnotation*

7

```
species = "human",  
ensemblVersion = 93,  
dataDir = "test_db",  
cores = 8,  
conservationBwFiles = Conservation_scores)
```

```
## End(Not run)
```

# Index

`concatenateAnnotationResult`, [2](#)

`initUTRAnnotation`, [3](#)

`partitionVariantFile`, [4](#)

`runUTRAnnotation`, [5](#)