Package 'refseqR'

May 22, 2024

Type Package

Title Common Computational Operations Working with RefSeq Entries (GenBank)

Version 1.0.2

Maintainer Jose V. Die <jose.die@uco.es>

Description Fetches NCBI data (RefSeq <https://www.ncbi.nlm.nih.gov/refseq/> database) and provides an environment to extract information at the level of gene, mRNA or protein accessions.

License MIT + file LICENSE

URL https://github.com/jdieramon/refseqR

BugReports https://github.com/jdieramon/refseqR/issues

Encoding UTF-8

Imports IRanges, rentrez, tibble, Biostrings

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Jose V. Die [aut, cre] (<https://orcid.org/0000-0002-7506-8590>),
Lluís Revilla Sancho [ctb] (<https://orcid.org/0000-0001-9747-2570>)

Repository CRAN

Date/Publication 2024-05-22 09:50:02 UTC

R topics documented:

---

refseqR   *refseqR: Common computational operations working with GenBank*

Description

refseqR is a framework of common computational operations working with RefSeq entries (Gen-Bank)

Author(s)

Jose V. Die <jose.die@uco.es>

See Also

Useful links:

- <https://github.com/jdieramon/refseqR>

- Report bugs at <https://github.com/jdieramon/refseqR/issues>

---

refseq_AAseq   *Extract the amino acid sequence into a Biostrings object*

Description

refseq_AAseq() Parses a single/multiple XP accessions (GenBank format) and extract the amino acid sequences into a AAStringSet object.

Usage

```
refseq_AAseq(xp)
```

Arguments

xp   A character string containing a single/multiple XP ids.

Value

An object of AAStringSet class.

Author(s)

Jose V. Die

Examples

```
xp = c("XP_004487758", "XP_004488550", "XP_004501961")
my_aa <- refseq_AAseq(xp)
# Now, the `AAStringSet`can be easily used to make a fasta file :
# writeXStringSet(x= my_aa, filepath = "aa_result")
```

---

refseq_CDScoords   *Extract the coding sequences (CDS) coordinates from a XM accession*

Description

`refseq_CDScoords()` Parses an XM accession (Genbank format) and extract the CDS coordinates. The CDS coordinates refer to the mRNA molecule.

Usage

```
refseq_CDScoords(xm)
```

Arguments

xm   A character string of the single/multiple XM id.

Value

An `IRanges` object with the start and end position of the CDS of the putative mRNAs.

Author(s)

Jose V. Die

See Also

refseq_CDSseq

Examples

```
xm = c("XM_004487701")
refseq_CDScoords(xm)


xm = c("XM_004487701", "XM_004488493")
refseq_CDScoords(xm)
```

---

refseq_CDSseq   *Extract the CDS nucleotide sequence into a Biostrings object*

Description

`refseq_CDSseq()` Parses a single/multiple XM acessions (Genbank format) and extract the CDS nucleotide sequences into a `DNAStringSet` object.

Usage

```
refseq_CDSseq(xm)
```

Arguments

xm   A character string of the single/multiple XM id.

Value

An object of `DNAStringSet` class.

Author(s)

Jose V. Die

See Also

refseq_CDScoords

Examples

```
xm <-  c("XM_004487701", "XM_004488493", "XM_004501904")
my_cds <- refseq_CDSseq(xm)
# Now, the `DNAStringSet` can easily used to make a fasta file :
# writeXStringSet(x= my_cds, filepath = "cds_result")
```

---

refseq_description *Get the sequence Description*

Description

refseq_description() Returns the sequence description from a single XM, XP, or Gene id. accession.

Usage

```
refseq_description(id)
```

Arguments

id   A character string of the XP, XM, or Gene id.

Value

A character vector containing the sequence description corresponding to the specified sequence as id.

Author(s)

Jose V. Die

See Also

refseq_XMfromXP to obtain the XM ids that encode a set of XP ids.

refseq_XPfromXM to obtain the XP ids encoded by a set of XM ids.

Examples

```
# Get the sequence descriptions from a set of XM accessions
xm = c("XM_004487701")
sapply(xm, function(x) refseq_description(x), USE.NAMES = FALSE)

# Get the sequence descriptions from a set of XP accessions
xp = c("XP_004487758")
sapply(xp, function(x) refseq_description(x), USE.NAMES = FALSE)


#' # Get the sequence descriptions from a set of Gene accessions
locs <- c("LOC101512347", "LOC101506901")
sapply(locs, function(x) refseq_description(x), USE.NAMES = FALSE)
```

---

refseq_fromGene   *Get the mRNA or protein accession*

Description

refseq_fromGene() Returns the XP or XM accession from a single gene id. accession.

Usage

```
refseq_fromGene(gene_symbol,sequence, retries)
```

Arguments

gene_symbol	A character string of the gene symbol.
sequence	A character string of the "XM" or "XP" to fetch data from mRNA or protein databases, respectively.
retries	A numeric value to control the number of retry attempts to handle 502 errors.

Value

A character vector containing the XP or XM id accessions corresponding to the especified gene symbol.

Author(s)

Jose V. Die

See Also

[refseq_XMfromXP](#) to obtain the XM ids that encode a set of XP ids.

[refseq_XPfromXM](#) to obtain the XP ids encoded by a set of XM ids.

Examples

```
# Get the XM accessions from a set of gene ids
locs <- c("LOC101512347")
sapply(locs, function(x) refseq_fromGene (x, sequence = "XM", retries = 3), USE.NAMES = FALSE)

# Get the XP accessions from a set of gene ids
locs <- c("LOC101512347")
sapply(locs, function(x) refseq_fromGene (x, sequence = "XP", retries = 3), USE.NAMES = FALSE)
```

---

refseq_fromXM *Get mRNA features*

Description

refseq_fromXM() Returns a number of features from a single/multiple mRNA accession(s).

Usage

```
refseq_fromXM(xm , feat)
```

Arguments

xm	A character string of the XM id.
feat	A character string of the selected features. Allowed features: 'caption', 'moltype', 'sourcedb', 'updatedate', 'slen', 'organism', 'title'.

Value

A `tibble` of summarized results including columns:

- caption, mRNA accession
- moltype, type of molecule
- sourcedb, database (GenBank)
- updatedate, date of updated record
- slen, molecule length (in bp)
- organism
- title, sequence description

Author(s)

Jose V. Die

See Also

[refseq_fromGene](#) to obtain the XP or XM accession from a single gene id. accession.

[refseq_XPfromXM](#) to obtain the XP ids encoded by a set of XM ids.

Examples

```
# Get several molecular features from a set of mRNA accessions
xm = c("XM_004487701", "XM_004488493", "XM_004501904")
feat = c("caption", "moltype", "sourcedb", "slen")
refseq_fromXM(xm ,feat)
```

refseq_geneSymbol   *Get the gene symbol*

Description

`refseq_geneSymbol()` Returns the gene symbol from a single Gene id. accession.

Usage

```
refseq_geneSymbol (id, db, retries)
```

Arguments

id	A character string of the XP or XM id.
db	A character string of the "nuccore" or "protein" database.
retries	A numeric value to control the number of retry attempts to handle internet errors.

Value

A character vector containing the gene symbol corresponding to the especified accession as `id`.

Author(s)

Jose V. Die

See Also

[refseq_XMfromXP](#) to obtain the XM ids that encode a set of XP ids.

[refseq_XPfromXM](#) to obtain the XP ids encoded by a set of XM ids.

Examples

```
# Get the gene symbol from a set of XM accessions
xm = c("XM_004487701", "XM_004488493")
sapply(xm, function(x) refseq_geneSymbol (x, db = "nuccore", retries = 3), USE.NAMES = FALSE)

# Get the gene symbol from a set of XP accessions
xp = c("XP_004487758")
sapply(xp, function(x) refseq_geneSymbol (x, db = "protein", retries = 3), USE.NAMES = FALSE)
```

---

refseq_mol_wt *Extract the molecular weigth from an XP accession*

Description

refseq_mol_wt() Parses an XP accession output (Genbank format) and extract the molecular weight (in Daltons).

Usage

```
refseq_mol_wt(xp)
```

Arguments

xp   A character string of the XP id.

Details

First, get the character vector containing the fetched record. Then, this function parses the fectched record and returns the molecular weight.

Value

A numeric vector represeting the molecular weight of the protein(s) especified as xp.

Author(s)

Jose V. Die

Examples

```
# Get the molecular weight from a single XP accession
xp <- "XP_020244413"
refseq_mol_wt(xp)

# Get the molecular weight from from a set of XP accessions
xp = c("XP_004487758", "XP_004488550")
sapply(xp, function(x) refseq_mol_wt(x), USE.NAMES = TRUE)
```

---

refseq_XMfromXP   *Get the XM accession from XP accession*

Description

refseq_XMfromXP() Returns the XM accession from a single XP accession.

Usage

```
refseq_XMfromXP(xp)
```

Arguments

xp   A character string of the XP id.

Value

A character vector containing the XM ids that encode the XP especified as xp.

Author(s)

Jose V. Die

See Also

[refseq_XPfromXM](#) to obtain the XP ids encoded by a set of XM ids.

Examples

```
# Get the XM id from a single XP accession
xp <- "XP_020244413"
refseq_XMfromXP(xp)


# Get the XM ids from a set of XP accessions
xp = c("XP_004487758", "XP_004488550")
sapply(xp, function(x) refseq_XMfromXP(x), USE.NAMES = FALSE)
```

refseq_XPfromXM *Get the XP accession from XM accession*

Description

refseq_XPfromXM() Returns the XP accession from a single XM accession.

Usage

```
refseq_XPfromXM(xm)
```

Arguments

xm   A character string of the XP id.

Value

A character vector containing the XP id encoded by the XM especified as xm.

Author(s)

Jose V. Die

See Also

[refseq_XMfromXP](#) to obtain the XM ids that encode a set of XP ids.

Examples

```
# Get the XP id from a single XM accession
xm <- "XM_004487701"
refseq_XPfromXM(xm)


# Get the XP ids from a set of XM accessions
xm = c("XM_004487701", "XM_004488493")
sapply(xm, function(x) refseq_XPfromXM(x), USE.NAMES = FALSE)
```

---

refseq_XPlength   *Get the protein length from an XP accession*

Description

refseq_XPlength() Returns the amino acid length from a single XP accession.

Usage

```
refseq_XPlength(xp, retries)
```

Arguments

xp	A character string of the XP id.
retries	A numeric value to control the number of retry attempts to handle internet errors.

Value

A numeric value representing the aa length of the protein especified as xp.

Author(s)

Jose V. Die

See Also

[refseq_XPfromXM](#) to obtain the XP ids encoded by a set of XM ids.

Examples

```
# Get the XM ids from a set of XP accessions
xp = c("XP_004487758", "XP_004488550")
sapply(xp, function(x) refseq_XPlength(x, retries = 3), USE.NAMES = FALSE)
```

Index