Package ‘miRBaseConverter’

January 31, 2025

Type Package

Title A comprehensive and high-efficiency tool for converting and retrieving the information of miRNAs in different miRBase versions

Version 1.30.0

Date 2020-12-31

Imports stats

Depends R (>= 3.4)

Description A comprehensive tool for converting and retrieving the miRNA Name, Accession, Sequence, Version, History and Family information in different miRBase versions. It can process a huge number of miRNAs in a short time without other depends.

License GPL (>= 2)

Suggests BiocGenerics, RUnit, knitr, rtracklayer, utils, rmarkdown

VignetteBuilder knitr

biocViews Software, miRNA

LazyData TRUE

URL https://github.com/taoshengxu/miRBaseConverter

BugReports https://github.com/taoshengxu/miRBaseConverter/issues

RoxygenNote 7.3.2

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

git_branch RELEASE_3_20

git_last_commit e1b103a

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2025-01-30

Author Taosheng Xu Taosheng Xu [aut, cre] (<https://orcid.org/0000-0001-8283-7274>)

Maintainer Taosheng Xu Taosheng Xu <taosheng.x@gmail.com>

Contents

checkMiRNAAlive . . . 2
checkMiRNAFamily . . . 3
checkMiRNAVersion . . . 4
getAllMiRNAs . . . 5
getAllSpecies . . . 6
getAllVersionInfo . . . 6
getMiRNAHistory . . . 7
getMiRNASequence . . . 7
getMiRNATable . . . 8
goTo_miRBase . . . 9
goTo_miRNAFamily . . . 10
miRNATest . . . 11
miRNAVersionConvert . . . 11
miRNA_AccessionToName . . . 12
miRNA_MatureToPrecursor . . . 13
miRNA_NameToAccession . . . 14
miRNA_PrecursorToMature . . . 15

Index **17**

`checkMiRNAAlive` *Check the miRNA status(Alive or Dead)*

Description

This function checks the miRNA status (Alive or Dead) in the latest miRBase version.

Usage

```
checkMiRNAAlive(Accessions, verbose = TRUE)
```

Arguments

`Accessions`	A character vector representing the miRNA Accessions in miRBase.
`verbose`	Logical value. If true, the dead miRNAs will be printed the console.

Value

A character vector of Accessions for all alive miRNAs. The names of the return vector are the position indexes in the input Accessions.

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
data(miRNATest)
## The input is miRNA Accessions
Accessions=miRNATest$Accession
alive_Accession1=checkMiRNAAlive(Accessions)

##The input is miRNA names
miRNANames=miRNATest$miRNA_Name
version=checkMiRNAVersion(miRNANames,verbose = TRUE)
result=miRNA_NameToAccession(miRNANames,version = version)
Accessions=result$Accession
alive_Accession2=checkMiRNAAlive(Accessions)
```

---

checkMiRNAFamily *Check the miRNA family*

Description

This function checks the miRNA family for a list of miRNA Names.

Usage

```
checkMiRNAFamily(Accessions)
```

Arguments

Accessions	A character vector representing the miRNA Accessions in miRBase.

Value

A data frame with four columns. The number of rows equal to the input Accessions. The four columns are defined as below:

- **Accession** : The input miRNA accessions.
- **miRNAName_v21** : The miRNA names (version 21) corresponding to the Accession.
- **FamilyAccession** : The accession of the family .
- **Family** : The family name.

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
data(miRNATest)
## The input is miRNA Accessions
Accessions=miRNATest$Accession
Family_Info1=checkMiRNAFamily(Accessions)

##The input is miRNA names
miRNANames=miRNATest$miRNA_Name
version=checkMiRNAVersion(miRNANames,verbose = TRUE)
result=miRNA_NameToAccession(miRNANames,version=version)
Accessions=result$Accession
Family_Info2=checkMiRNAFamily(Accessions)
```

---

checkMiRNAVersion *check the miRNA Version in miRBase*

Description

This function checks the most possible miRBase version for a list of miRNA names.

Usage

```
checkMiRNAVersion(miRNANames, verbose = TRUE)
```

Arguments

miRNANames	A character vector representing the miRNA names.
verbose	Logical value. If true, the detail version information is printed in the console for user reference.

Value

A single character value or a character vector represent the most possible miRBase version for the list of miRNA names.

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
data(miRNATest)
miRNANames=miRNATest$miRNA_Name
version=checkMiRNAVersion(miRNANames,verbose=TRUE)
```

`getAllMiRNAs` *Get all miRNAs in the specified miRBase version*

Description

This function gets all miRNAs in the specified miRBase version.

Usage

```
getAllMiRNAs(version = "v22", type = "all", species = "all")
```

Arguments

`version` A character value representing the specified miRBase version for retrieval. Users can apply the function **getAllVersionInfo()** to get the available miRNA version names. The optional values are in below:
"v6","v7_1","v8","v8_1","v8_2","v9","v9_1",
"v9_2","v10","v10_1","v11","v12","v13","v14",
"v15","v16","v17","v18","v19","v20","v21","v22"

`type` A character value representing the miRNA type for retrieval.

- **"precursor"**
- **"mature"**
- **"all"** : precursor and mature

`species` A character value representing the abbreviation of species. Users can apply the **getAllSpecies()** function to get the available abbreviation of species. If species is set to **"all"**, the miRNAs of all species will return.

Value

A data frame with three columns. The three columns are defined as below:

- **Accession**
- **Name**
- **Sequence**

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
miRNAs=getAllMiRNAs(version="v22", type="all", species="hsa")
```

getAllSpecies — *Get all species of miRNAs embodied in miRBase repository*

Description

This function return a reference for all species of miRNAs including the abbreviation and full name.

Usage

```
getAllSpecies()
```

Value

A dataframe A data frame with two columns. The two columns are defined as below:

- **Species**
- **FullName**

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
allSpecies=getAllSpecies()
```

getAllVersionInfo — *Get all miRBase version information*

Description

This function return a reference for all miRBase versions' information including Version name, Release date, miRNA number and Status.

Usage

```
getAllVersionInfo()
```

Value

The detailed version information is printed in the console for user reference.

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
getAllVersionInfo()
```

getMiRNAHistory — *Get the detailed information of a single specified miRNA in all miRBase versions.*

Description

This function returns all available miRBase versions' information of a single specified miRNA.

Usage

```
getMiRNAHistory(Accession)
```

Arguments

Accession	A character representing the single Accession.

Value

A data frame including all the history information (Precursor, Mature, Sequence) of the specified miRNA. Each row represents a miRBase version.

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
#####1,The input is a miRNA Name
miRNAName="hsa-miR-26b-5p"
result1=miRNA_NameToAccession(miRNAName,version="v22")
Accession=result1$Accession
result2=getMiRNAHistory(Accession)

#####2,The input is miRNA Accession
Accession="MIMAT0000765"
result3=getMiRNAHistory(Accession)
```

getMiRNASequence — *Get the miRNA sequences*

Description

This function returns the miRNA sequences for a list of miRNAs.

Usage

```
getMiRNASequence(Accessions, targetVersion = "v22")
```

Arguments

`Accessions`	A character vector representing the miRNA Accessions in miRBase.
`targetVersion`	A character value representing the target miRBase version corresponding the Accessions. Users can apply the function **getAllVersionInfo()** to get the available miRNA version names. The optional values are in below: "v6","v7_1","v8","v8_1","v8_2","v9","v9_1","v9_2","v10","v10_1","v11","v12","v13","v14", "v15","v16","v17","v18","v19","v20","v21","v22"

Value

A nx2 data frame. The number of row equals to input miRNAs. The two columns are defined as below:

- **Accession** : The original miRNA (Column 1).
- **miRNASequence_{targetVersion}** : The return miRNA sequence (in specified version) corresponding to the input miRNAs (Column 2).

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
#####1, The input are miRNA Accessions
data(miRNATest)
Accessions=miRNATest$Accession
result1=getMiRNASequence(Accessions,targetVersion="v13")
result2=getMiRNASequence(Accessions,targetVersion="v22")

#####2, The input are miRNA Names
data(miRNATest)
miRNANames=miRNATest$miRNA_Name
result3=miRNAVersionConvert(miRNANames,targetVersion="v22",exact=TRUE)
Accessions=result3$Accession
result4=getMiRNASequence(Accessions,targetVersion="v22")
```

`getMiRNATable` *Get the full miRNAs information table of the specified miRBase version*

Description

This function returns the full miRNAs information table of the specified miRBase version

Usage

```
getMiRNATable(version = "v22", species = "all")
```

Arguments

`version` A character value representing the specified miRBase version for retrieval. Users can apply the function **getAllVersionInfo()** to get the available miRNA version names. The optional values are in below:
"v6","v7_1","v8","v8_1","v8_2","v9","v9_1","v9_2","v10","v10_1","v11","v12","v13","v14", "v15","v16","v17","v18","v19","v20","v21","v22"

`species` A character value representing the abbreviation of species. Users can apply the function **getAllSpecies()** to get the available abbreviation of species. If species is set to **"all"**, the miRNAs of all species will return.

Value

A data frame

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
miRNA_Tab=getMiRNATable(version="v22",species="hsa")
```

---

goTo_miRBase *Open the miRBase webpages of the specified miRNAs*

Description

This function redirects the miRBase webpage of the specified miRNAs

Usage

```
goTo_miRBase(Accessions)
```

Arguments

`Accessions` A character vector representing the miRNA Accessions in miRBase. We restict the number of queried miRNAs each time. The maximum number of the input miRNAs is 15.

Value

No values

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
#### 1. A step-loop
Accession1="MI0000447"
goTo_miRBase(Accession1)

#### 2. A mature miRNA
Accession2="MIMAT0026477"
goTo_miRBase(Accession2)

#### 3. A list of miRNAs
Accession3=miRNATest$Accession[1:10]
goTo_miRBase(Accession3)
```

---

goTo_miRNAFamily *Open the miRNA family webpages of the specified miRNAs*

Description

This function redirects the miRBase miRNA family webpages of the specified miRNA families

Usage

```
goTo_miRNAFamily(FamilyAccessions, verbose = TRUE)
```

Arguments

`FamilyAccessions`	A character vector representing the miRNA family Accessions in miRBase. We restict the queried number of miRNA family each time. The maximum number of the input miRNA families is 15.
`verbose`	Logical value. If true, the invalid miRNA Family will be printed the console.

Value

No values

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
data(miRNATest)
Accessions=miRNATest$Accession
Family_Info=checkMiRNAFamily(Accessions)
FamilyAccessions=Family_Info$FamilyAccession[1:15]
goTo_miRNAFamily(FamilyAccessions)
```

miRNATest — *Dataset: miRNA Test Dataset*

Description

- Accession : miRNA Accessions in miRBase
- miRNA_Name : miRNA names

Format

A data frame

Note

This is a mixed miRNA name collection of some species for the examples' test in the package.

Examples

```
data(miRNATest)
```

miRNAVersionConvert — *miRBase version convert for miRNA Names*

Description

This function converts a group of any species' miRNA names (including precursor and mature miRNA) to the specified miRBase version if the miRNAs have been defined in miRBase.

Usage

```
miRNAVersionConvert(
  miRNANames,
  targetVersion = "v22",
  exact = TRUE,
  verbose = TRUE
)
```

Arguments

`miRNANames`	A character vector representing the source miRNA names needed to be convert.
`targetVersion`	A character value representing the target miRBase version corresponding the source miRNA names. Users can apply the function **getAllVersionInfo()** to get the available miRNA version names. The optional values are in below: "v6","v7_1","v8","v8_1","v8_2","v9","v9_1","v9_2","v10","v10_1","v11","v12","v13","v14", "v15","v16","v17","v18","v19","v20","v21","v22"
`exact`	Logical value. If true, the result will be the most exactly matched result. If FALSE, the result will include all the possible matched miRNA name. If one miRNA can match multiple names. All the matched names are concatenated with "&".
`verbose`	Logical value. If true, it will print the multiple matched miRNA Names and Accessions to the console.

Value

A data frame with three columns. The number of rows equal to the input miRNA names. The three columns are defined as below:

- **OriginalName** : The original miRNA names (Column 1).
- **TargetName** : The converted miRBase names (in specified version) corresponding to the original miRNA names (Column 2).
- **Accession** : The corresponding miRBase Accessions (Column 3).

Note

Please note: Due to some miRNA names changing many times in history. Even if choose the third parameter "exact"=TRUE, it may still have some miRNAs that can't match the unique name in the target version. In order to return the accurate result as possible, we also concatenate the multiple matched miRNA names with "&". This is the rare case but it happens sometimes.

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
data(miRNATest)
miRNANames=miRNATest$miRNA_Name
result1=miRNAVersionConvert(miRNANames,targetVersion="v13",exact=TRUE,verbose=TRUE)
result2=miRNAVersionConvert(miRNANames,targetVersion="v22",exact=TRUE,verbose=TRUE)
result3=miRNAVersionConvert(miRNANames,targetVersion="v22",exact=FALSE,verbose=TRUE)

miRNANames=c( "hsa-let-7c","hsa-miR-3190-3p","hsa-let-7c","hsa-miR-34b","hsa-miR-378",
"hsa-miR-499a-3p","hsa-miR-499a-5p","hsa-miR-500","hsa-miR-516a-5p","hsa-miR-550","hsa-miR-589")
result4=miRNAVersionConvert(miRNANames, targetVersion="v22", exact=TRUE, verbose=TRUE)
```

---

miRNA_AccessionToName *miRBase Accession to miRNA Name in specified version*

Description

This function converts a group of any species' miRNA Accessions (including precursor and mature miRNA) to a specified miRBase version if the Accessions have been defined in miRBase.

Usage

```
miRNA_AccessionToName(Accessions, targetVersion = "v22")
```

Arguments

`Accessions`	A character vector representing the miRNA Accessions needed to be convert.
`targetVersion`	A character value representing the target miRBase version corresponding the Accessions. The optional values are in below: "v6","v7_1","v8","v8_1","v8_2","v9","v9_1", "v9_2","v10","v10_1","v11","v12","v13","v14", "v15","v16","v17","v18","v19","v20","v21","v22"

Value

A nx2 data frame. The number of rows equal to the input miRNA names. The two columns are defined as below:

- **Accession** : The Accession of miRNAs (Column 1).
- **TargetName** : The converted miRBase names (in specified version) corresponding to the Accessions (Column 2).

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
data(miRNATest)
Accessions=miRNATest$Accession
result1=miRNA_AccessionToName(Accessions,targetVersion="v13")
result2=miRNA_AccessionToName(Accessions,targetVersion="v22")
```

---

miRNA_MatureToPrecursor

Convert the mature miRNAs to the corresponding precursors

Description

This function converts the mature miRNAs to the corresponding precursors in the specified miRBase version.

Usage

```
miRNA_MatureToPrecursor(miRNANames, version = NULL)
```

Arguments

`miRNANames`	A character vector representing the miRNA names.
`version`	The default is **NULL** representing the most possible latest version of the input miRNA Names will be checked automatically. Otherwise, a character value representing the version corresponding the input miRNA Names. Users can apply the function **getAllVersionInfo()** to get the available miRNA version names. The optional values are in below: "v6","v7_1","v8","v8_1","v8_2","v9","v9_1","v9_2","v10","v10_1","v11","v12","v13","v14", "v15","v16","v17","v18","v19","v20","v21","v22"

Value

A data frame(nx2). The number of rows equal to the input miRNA Names. The two columns are defined as below:

- **OriginalName** : The input miRNA Names.
- **Precursor** : The corresponding precursors of the mature miRNAs.

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
data(miRNATest)
miRNANames=miRNATest$miRNA_Name
result=miRNA_MatureToPrecursor(miRNANames)
```

`miRNA_NameToAccession` *The miRBase miRNA names with specified version to Accessions*

Description

This function converts a group of any species' miRNA name to the Accessions defined in miRBase.

Usage

```
miRNA_NameToAccession(miRNANames, version = "v22")
```

Arguments

`miRNANames`	A character vector representing the source miRNA names needed to be convert.
`version`	A character value representing the version corresponding the miRNANames. Users can apply the function **getAllVersionInfo()** to get the available miRNA version names. The optional values are in below: "v6","v7_1","v8","v8_1","v8_2","v9","v9_1","v9_2","v10","v10_1","v11","v12","v13","v14", "v15","v16","v17","v18","v19","v20","v21","v22"

Value

A nx2 data frame. The number of rows equal to the input miRNA names. The two columns are defined as below:

- **miRNAName_{Version}** : The input miRNA names (Column 1).
- **Accession** : The convert Accession(Column 2).

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
data(miRNATest)
miRNANames=miRNATest$miRNA_Name
version=checkMiRNAVersion(miRNANames,verbose=TRUE)
result1=miRNA_NameToAccession(miRNANames,version=version)
result2=miRNA_AccessionToName(result1[,2],targetVersion="v22")
result3=miRNAVersionConvert(miRNANames,targetVersion="v22",exact=TRUE)
```

---

miRNA_PrecursorToMature

Convert the precursors to the corresponding mature miRNAs

Description

This function converts the precursors to the corresponding mature miRNAs in the specified miRBase version.

Usage

```
miRNA_PrecursorToMature(miRNANames, version = NULL)
```

Arguments

miRNANames	A character vector representing the miRNA names.
version	The default is **NULL** representing the most possible latest version of the input miRNA Names will be checked automatically. Otherwise, a character value representing the version corresponding the input miRNA Names. Users can apply the function **getAllVersionInfo()** to get the available miRNA version names. The optional values are in below: "v6","v7_1","v8","v8_1","v8_2","v9","v9_1","v9_2","v10","v10_1","v11","v12","v13","v14", "v15","v16","v17","v18","v19","v20","v21","v22"

Value

A data frame(nx3). The number of rows equal to the input miRNA names. The three columns are defined as below:

- **OriginalName** : The input miRNA Names.
- **Mature1** : The corresponding mature miRNAs (always "-5p") .
- **Mature2** : The corresponding mature miRNAs (always "-3p") .

Note

If the input miRNA Names mingle mature miRNA names, the mature miRNA names will match to themselves in the output.

Author(s)

Xu, Taosheng <taosheng.x@gmail.com>

Examples

```
miRNANames=c("pma-mir-100a","sko-mir-92a","hsa-mir-6131","mtr-MIR2655i",
"mmu-mir-153","mtr-MIR2592am","mml-mir-1239","xtr-mir-128-2","oan-mir-100",
"mmu-mir-378b","hsa-miR-508-5p","mmu-miR-434-3p")
result=miRNA_PrecursorToMature(miRNANames)
```

Index

∗ **datasets**
    miRNATest, 11

checkMiRNAAlive, 2
checkMiRNAFamily, 3
checkMiRNAVersion, 4

getAllMiRNAs, 5
getAllSpecies, 6
getAllVersionInfo, 6
getMiRNAHistory, 7
getMiRNASequence, 7
getMiRNATable, 8
goTo_miRBase, 9
goTo_miRNAFamily, 10

miRNA_AccessionToName, 12
miRNA_MatureToPrecursor, 13
miRNA_NameToAccession, 14
miRNA_PrecursorToMature, 15
miRNATest, 11
miRNAVersionConvert, 11