

targetscan.Hs.eg.db

June 22, 2022

targetscan.Hs.eg.db *Bioconductor annotation data package*

Description

Welcome to the targetscan.Hs.eg.db annotation Package. The purpose of this package is to provide detailed information about the latest version of the TargetScan miRNA target prediction database. This package is updated biannually.

You can learn what objects this package supports with the following command:

```
ls("package:targetscan.Hs.eg.db")
```

Each of these objects has their own manual page detailing where relevant data was obtained along with some examples of how to use it.

There are currently two annotation packages for TargetScan, one for human and another one for mouse. These correspond to the TargetScanHuman and the TargetScanMouse databases, see the TargetScan website for details.

Please note, that the package currently contains *_only_* the prediction for conserved miRNA targets, separately for each miRNA family. These are the predicted targets that are returned are the 'conserved across mammals' ones which are listed in the table 'Predicted Conserved Targets Info' from the download page.

References

<http://www.targetscan.org>

Examples

```
ls("package:targetscan.Hs.eg.db")
```

targetsca.Hs.egFAMILY2MIRBASE

An annotation data object that maps miRNA families to miRBase identifiers

Description

targetsca.Hs.egFAMILY2MIRBASE maps miRNA family names to miRBase identifiers. The mappings are taken from the TargetScan database.

Details

This is an R object containing key and value pairs. Keys are miRNA family names and values are miRBase identifiers. Values are vectors of various lengths, as a single miRNA family usually corresponds to many miRBase miRNAs.

References

<http://www.targetsca.org> for TargetScan and <http://www.mirbase.org/> for miRBase.

See Also

The miRBase database is also provided as a custom annotation package in Bioconductor: `mirbase.db`.

Examples

```
## Map some random miRNA families
fams <- sample(1s(targetsca.Hs.egFAMILY2MIRBASE), 3)
mget(fams, targetsca.Hs.egFAMILY2MIRBASE)
```

targetsca.Hs.egMAPCOUNTS

Number of mapped keys for the maps in package targetsca.Hs.eg.db

Description

targetsca.Hs.egMAPCOUNTS provides the "map count" (i.e. the count of mapped keys) for each map in package `targetsca.Hs.eg.db`.

Details

This "map count" information is precalculated and stored in the package annotation DB. This allows some quality control and is used by the `checkMAPCOUNTS` function defined in `AnnotationDbi` to compare and validate different methods (like `count.mappedkeys(x)` or `sum(!is.na(as.list(x)))`) for getting the "map count" of a given map.

See Also

[mappedkeys](#), [count.mappedkeys](#), [checkMAPCOUNTS](#)

Examples

```
targetscan.Hs.egMAPCOUNTS
mapnames <- names(targetscan.Hs.egMAPCOUNTS)
targetscan.Hs.egMAPCOUNTS[mapnames[1]]
x <- get(mapnames[1])
sum(!is.na(as.list(x)))
count.mappedkeys(x) # much faster!

## Check the "map count" of all the maps in package targetscan.Hs.eg.db
checkMAPCOUNTS("targetscan.Hs.eg.db")
```

```
targetscan.Hs.egMIRBASE2FAMILY
```

An annotation data object that maps miRBase identifiers to miRNA family names

Description

targetscan.Hs.egMIRBASE2FAMILY maps miRBase identifiers to miRNA family names. The mappings are taken from the TargetScan database.

Details

This is an R object containing key and value pairs. Keys are miRBase identifiers and values are miRNA family names. Values are vectors of length 1, as a miRBase miRNA belongs to a single miRNA family.

References

<http://www.targetscan.org> for TargetScan and <http://www.mirbase.org> for miRBase.

See Also

The miRBase database is also provided as a custom annotation package in Bioconductor: [mirbase.db](#).

Examples

```
## Map some random miRBase miRNAs
fams <- sample(1s(targetscan.Hs.egMIRBASE2FAMILY), 3)
mget(fams, targetscan.Hs.egMIRBASE2FAMILY)
```

targetscan.Hs.egMIRNA *Annotation of miRBase identifiers to associated data*

Description

This data set gives mappings between miRBase identifiers and their respective associated data.

Details

This is an R object containing key and value pairs. Keys are miRBase identifiers and values are their associated data. Values are simple named lists of the following members:

MiRBase.ID The miRBase identifier, it starts with a three letter species code.

MiRBase.Accession Accession number of the miRNA.

Seed.m8 miRNA seed, seven bases from the 5' end of the miRNA.

Species Scientific name of the species.

Mature.sequence Mature sequence of the miRNA.

Family.conservation Level of conservation of the miRNA family.

The package contains the data for *all* miRBase identifiers, the ones for other species as well.

References

<http://www.targetscan.org> for TargetScan and <http://microrna.sanger.ac.uk/sequences/> for miRBase.

Examples

```
## Query some random miRBase identifiers
fams <- sample(ls(targetscan.Hs.egMIRNA), 3)
mget(fams, targetscan.Hs.egMIRNA)
```

targetscan.Hs.egTARGETS
Annotation data for TargetScan miRNA target predictions

Description

targetscan.Hs.egTARGETS maps Entrez gene identifiers to miRNA families. The mappings are taken from the TargetScan database.

Details

TargetScan predicts biological targets of miRNAs by searching for a presence of conserved 8mer and 7mer sites that match the seed region of each miRNA.

There are currently two annotation packages for TargetScan, one for human and another one for mouse. These correspond to the TargetScanHuman and the TargetScanMouse databases, see the TargetScan website for details.

targetscan.Hs.egTARGETS is an R object containing key and value pairs. Keys are Entrez gene identifiers and values are miRNA families. Values are vectors of various lengths, as some genes are predicted to be potentially regulated by many miRNA families. Note that a given miRNA family can come up many times in the value vector, if the miRNA seed matches the UTR of gene at multiple positions.

References

<http://www.targetscan.org> for TargetScan and <http://www.mirbase.org/> for miRBase.

See Also

[targetscan.Hs.egTARGETSFULL](#) for more information about the miRNA targets. The miRBase database is also provided as a custom annotation package in Bioconductor: `mirbase.db`.

Examples

```
## Get the miRNA families that regulate these genes
genes <- sample(ls(targetscan.Hs.egTARGETS), 3)
mget( genes, targetscan.Hs.egTARGETS )

## Get all targets of a given miRNA family
mget("miR-10abc/10a-5p", revmap(targetscan.Hs.egTARGETS))
```

targetscan.Hs.egTARGETSFULL

Annotation data for TargetScane miRNA target predictions

Description

targetscan.Hs.egTARGETSFULL maps Entrez gene identifiers to miRNA families. The mappings are taken from the TargetScan database.

Details

TargetScan predicts biological targets of miRNAs by searching for a presence of conserved 8mer and 7mer sites that match the seed region of each miRNA.

There are currently two annotation packages for TargetScan, one for human and another one for mouse. These correspond to the TargetScanHuman and the TargetScanMouse databases, see the TargetScan website for details.

targetscan.Hs.egTARGETSFULL is an R object containing key and value pairs. Keys are Entrez gene identifiers and values are lists, each list member corresponds to a predicted targeting miRNA family. Each list member is itself a list with the following members:

miR.Family Name of the miRNA family.

UTR.start Start position on UTR.

UTR.end End position on UTR.

MSA.start Start of multiple sequence alignment, with gaps.

MSA.end End of multiple sequence alignment.

Seed.match Type of the matching. Possible values are '8mer', which means that all eight bases match, 'm8', which means an exact match to positions 2-8 of the mature miRNA, and '1A', which means that positions 2-7 of the mature miRNA match, followed by an 'A'.

PCT The probability of conserved targeting, see Friedman et al. in the references section below for details.

References

Friedman RC, Farh KK, Burge CB, Bartel DP: Most mammalian mRNAs are conserved targets of microRNAs. *Genome Res.* 2009 Jan;19(1):92-105

<http://www.targetscan.org> for TargetScan and <http://www.mirbase.org/> for miRBase.

See Also

The miRBase database is also provided as a custom annotation package in Bioconductor: `mirbase.db`.

Examples

```
## Get the miRNA families that regulate these genes
genes <- sample(1s(targetscan.Hs.egTARGETSFULL), 3)
mget(genes, targetscan.Hs.egTARGETSFULL)
```

targetscan.Hs.eg_dbconn

Collect information about the package annotation DB

Description

Some convenience functions for getting a connection object to (or collecting information about) the package annotation DB.

Usage

```
targetscan.Hs.eg_dbconn()
targetscan.Hs.eg_dbfile()
targetscan.Hs.eg_dbschema(file="", show.indices=FALSE)
targetscan.Hs.eg_dbInfo()
```

Arguments

file	A connection, or a character string naming the file to print to (see the file argument of the cat function for the details).
show.indices	The CREATE INDEX statements are not shown by default. Use show.indices=TRUE to get them.

Details

targetscan.Hs.eg_dbconn returns a connection object to the package annotation DB. **IMPORTANT:** Don't call [dbDisconnect](#) on the connection object returned by targetscan.Hs.eg_dbconn or you will break all the [AnnDbObj](#) objects defined in this package!

targetscan.Hs.eg_dbfile returns the path (character string) to the package annotation DB (this is an SQLite file).

targetscan.Hs.eg_dbschema prints the schema definition of the package annotation DB.

targetscan.Hs.eg_dbInfo prints other information about the package annotation DB.

Value

targetscan.Hs.eg_dbconn: a DBIConnection object representing an open connection to the package annotation DB.

targetscan.Hs.eg_dbfile: a character string with the path to the package annotation DB.

targetscan.Hs.eg_dbschema: none (invisible NULL).

targetscan.Hs.eg_dbInfo: none (invisible NULL).

See Also

[dbGetQuery](#), [dbConnect](#), [dbconn](#), [dbfile](#), [dbschema](#), [dbInfo](#)

Examples

```
## Count the number of rows in the "mirna_family" table:
dbGetQuery(targetscan.Hs.eg_dbconn(), "SELECT COUNT(*) FROM mirna_family")

## The connection object returned by targetscan.Hs.eg_dbconn() was
## created with:
dbConnect(SQLite(), dbname=targetscan.Hs.eg_dbfile(), cache_size=64000,
synchronous=0)

targetscan.Hs.eg_dbschema()

targetscan.Hs.eg_dbInfo()
```

Index

* datasets

- targetscan.Hs.eg.db, [1](#)
- targetscan.Hs.eg_dbconn, [6](#)
- targetscan.Hs.egFAMILY2MIRBASE, [2](#)
- targetscan.Hs.egMAPCOUNTS, [2](#)
- targetscan.Hs.egMIRBASE2FAMILY, [3](#)
- targetscan.Hs.egMIRNA, [4](#)
- targetscan.Hs.egTARGETS, [4](#)
- targetscan.Hs.egTARGETSFULL, [5](#)

* utilities

- targetscan.Hs.eg_dbconn, [6](#)

AnnDbObj, [7](#)

cat, [7](#)

checkMAPCOUNTS, [2](#), [3](#)

count.mappedkeys, [3](#)

dbconn, [7](#)

dbConnect, [7](#)

dbDisconnect, [7](#)

dbfile, [7](#)

dbGetQuery, [7](#)

dbInfo, [7](#)

dbschema, [7](#)

mappedkeys, [3](#)

targetscan.Hs.eg (targetscan.Hs.eg.db),
[1](#)

targetscan.Hs.eg.db, [1](#)

targetscan.Hs.eg_dbconn, [6](#)

targetscan.Hs.eg_dbfile
(targetscan.Hs.eg_dbconn), [6](#)

targetscan.Hs.eg_dbInfo
(targetscan.Hs.eg_dbconn), [6](#)

targetscan.Hs.eg_dbschema
(targetscan.Hs.eg_dbconn), [6](#)

targetscan.Hs.egFAMILY2MIRBASE, [2](#)

targetscan.Hs.egMAPCOUNTS, [2](#)

targetscan.Hs.egMIRBASE2FAMILY, [3](#)

targetscan.Hs.egMIRNA, [4](#)

targetscan.Hs.egTARGETS, [4](#)

targetscan.Hs.egTARGETSFULL, [5](#), [5](#)