HDO.db

October 9, 2024

HDOALIAS *Map from HDO alias to HDO terms*

Description

HDOALIAS is an R object that provides mapping from HDO alias to HDO terms

Details

Mappings were based on data provided by: Disease Ontology With a date stamp from the source of: 20220706

References

http://hdo-wiki.nubic.northwestern.edu/index.php/Main_Page

Examples

```
# Convert the object to a list
xx <- as.list(HDOALIAS)
```

HDOANCESTOR *Annotation of HDO Identifiers to their Ancestors*

Description

This data set describes associations between HDO terms and their ancestor terms, based on the directed acyclic graph (DAG) defined by the Disease Ontology Consortium. The format is an R object mapping the HDO terms to all ancestor terms, where an ancestor term is a more general HDO term that precedes the given HDO term in the DAG (in other words, the parents, and all their parents, etc.).

Details

Each HDO term is mapped to a vector of ancestor HDO terms. Mappings were based on data provided by: Disease Ontology With a date stamp from the source of: 20220706

References

http://do-wiki.nubic.northwestern.edu/index.php/Main_Page

Examples

```
# Convert the object to a list
xx <- as.list(HDOANCESTOR)
```

HDO.db *Bioconductor annotation data package*

Description

Welcome to the HDO.db annotation Package. The purpose of this package is to provide detailed information about the latest version of the Disease Ontology. This package is updated biannually. You can learn what objects this package supports with the following command: `ls("package:HDO.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.

Examples

```
# Convert the object to a list
ls("package:HDO.db")
```

HDOCHILDREN *Annotation of HDO Identifiers to their Children*

Description

This data set describes associations between HDO terms and their direct children terms, based on the directed acyclic graph (DAG) defined by the Disease Ontology Consortium. The format is an R object mapping the HDO terms to all direct children terms, where a direct child term is a more specific HDO term that is immediately preceded by the given HDO term in the DAG.

Details

Each HDO term is mapped to a vector of children HDO terms. Mappings were based on data provided by: Disease Ontology With a date stamp from the source of: 20220706

References

http://do-wiki.nubic.northwestern.edu/index.php/Main_Page

Examples

```
# Convert the object to a list
xx <- as.list(HDOCHILDREN)
```

HDODb-objects *AnnotationDb objects and their progeny, methods etc.*

Description

AnnotationDb is the virtual base class for all annotation packages. It contain a database connection and is meant to be the parent for a set of classes in the Bioconductor annotation packages. These classes will provide a means of dispatch for a widely available set of select methods and thus allow the easy extraction of data from the annotation packages.

select, columns and keys are used together to extract data from an AnnotationDb object (or any object derived from the parent class). Examples of classes derived from the AnnotationDb object include (but are not limited to): ChipDb, OrgDb HDODb, InparanoidDb and ReactomeDb.

columns shows which kinds of data can be returned for the AnnotationDb object.

keytypes allows the user to discover which keytypes can be passed in to select or keys and the keytype argument.

keys returns keys for the database contained in the AnnotationDb object . This method is already documented in the keys manual page but is mentioned again here because it's usage with select is so intimate. By default it will return the primary keys for the database, but if used with the keytype argument, it will return the keys from that keytype.

select will retrieve the data as a data.frame based on parameters for selected keys columns and keytype arguments. Users should be warned that if you call select and request columns that have multiple matches for your keys, select will return a data.frame with one row for each possible match. This has the effect that if you request multiple columns and some of them have a many to one relationship to the keys, things will continue to multiply accordingly.

So it's not a good idea to request a large number of columns unless you know that what you are asking for should have a one to one relationship with the initial set of keys. In general, if you need to retrieve a column (like GO) that has a many to one relationship to the original keys, it is most useful to extract that separately.

Usage

```
columns(x)
keytypes(x)
keys(x, keytype, ...)
select(x, keys, columns, keytype, ...)
```

Arguments

- `x` — the `AnnotationDb` object. But in practice this will mean an object derived from an `AnnotationDb` object such as a `OrgDb` or `ChipDb` object.
- `keys` — the keys to select records for from the database. All possible keys are returned by using the `keys` method.
- `columns` — the columns or kinds of things that can be retrieved from the database. As with `keys`, all possible columns are returned by using the `columns` method.
- `keytype` — the keytype that matches the keys used. For the `select` methods, this is used to indicate the kind of ID being used with the keys argument. For the `keys` method this is used to indicate which kind of keys are desired from `keys`
- `...` — other arguments. These include:

  pattern: the pattern to match (used by keys)

  column: the column to search on. This is used by keys and is for when the thing you want to pattern match is different from the keytype, or when you want to simply want to get keys that have a value for the thing specified by the column argument.

  fuzzy: TRUE or FALSE value. Use fuzzy matching? (this is used with pattern by the keys method)

Value

`keys`,`columns` and `keytypes` each return a character vector or possible values. `select` returns a data.frame.

Author(s)

Marc Carlson

Examples

```
## display the columns
## use doid keys
dokeys <- head(keys(HDO.db))
res <- select(x = HDO.db, keys = dokeys, keytype = "doid",
    columns = c("offspring", "term", "parent"))

## use term keys
dokeys <- head(keys(HDO.db, keytype = "term"))
res <- select(x = HDO.db, keys = dokeys, keytype = "term",
    columns = c("offspring", "doid", "parent"))
```

HDOMAPCOUNTS — *Number of mapped keys for the maps in package HDO.db*

Description

HDOMAPCOUNTS provides the "map count" (i.e. the count of mapped keys) for each map in package HDO.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

```
# Convert the object to a list
xx <- as.list(HDOCHILDREN)
```

HDOOBSOLETE — *Annotation of HDO identifiers by terms defined by Disease Ontology Consortium and their status are obsolete*

Description

This is an R object mapping HDO identifiers to the specific terms in defined by Disease Ontology Consortium and their definition are obsolete

Details

All the obsolete HDO terms that are collected in this index will no longer exist in other mapping objects.

Mappings were based on data provided by: Disease Ontology With a date stamp from the source of: 220220706

References

http://do-wiki.nubic.northwestern.edu/index.php/Main_Page

HDOOFFSPRING *Annotation of HDO Identifiers to their Offspring*

Description

This data set describes associations between HDO terms and their offspring terms, based on the directed acyclic graph (DAG) defined by the Disease Ontology Consortium. The format is an R object mapping the HDO terms to all offspring terms, where an ancestor term is a more specific HDO term that is preceded by the given HDO term in the DAG (in other words, the children and all their children, etc.).

Details

Each HDO term is mapped to a vector of offspring HDO terms. Mappings were based on data provided by: Disease Ontology With a date stamp from the source of: 20220706

References

http://do-wiki.nubic.northwestern.edu/index.php/Main_Page

Examples

```
# Convert the object to a list
xx <- as.list(HDOOFFSPRING)
```

HDOPARENTS *Annotation of HDO Identifiers to their Parents*

Description

This data set describes associations between HDO terms and their direct parent terms, based on the directed acyclic graph (DAG) defined by the Disease Ontology Consortium. The format is an R object mapping the HDO terms to all direct parent terms, where a direct parent term is a more general HDO term that immediately precedes the given HDO term in the DAG.

Details

Each HDO term is mapped to a named vector of HDO terms. The name associated with the parent term will be either *isa*, *partof*, where *isa* indicates that the child term is a more specific version of the parent, and *partof* indicate that the child term is a part of the parent.

Mappings were based on data provided by: Disease Ontology With a date stamp from the source of: 220220706

References

http://do-wiki.nubic.northwestern.edu/index.php/Main_Page

Examples

```
xx <- as.list(HDOPARENTS)
```

HDOSYNONYM *Map from HDO synonyms to HDO terms*

Description

HDOSYNONYM is an R object that provides mapping from HDO synonyms to HDO terms

Details

Mappings were based on data provided by: Disease Ontology With a date stamp from the source of: 20220706

References

http://do-wiki.nubic.northwestern.edu/index.php/Main_Page

Examples

```
# Convert the object to a list
xx <- as.list(HDOSYNONYM)
```

HDOTERM *Annotation of HDO Identifiers to HDO Terms*

Description

This data set gives mappings between HDO identifiers and their respective terms.

Details

Each HDO identifier is mapped to a `HDOTerms` object that has 2 slots: do_id: HDO Identifier; Term: The term for that HDO id

Mappings were based on data provided by: Disease Ontology With a date stamp from the source of: 220220706

References

http://do-wiki.nubic.northwestern.edu/index.php/Main_Page

Examples

```
# Convert the object to a list
xx <- as.list(HDOTERM)
```

HDO_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

```
HDO_dbconn()
HDO_dbfile()
HDO_dbschema(file="", show.indices=FALSE)
HDO_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

`HDO_dbconn` returns a connection object to the package annotation DB. IMPORTANT: HDOn't call `dbDisconnect` on the connection object returned by `HDO_dbconn` or you will break all the `AnnDbObj` objects defined in this package!

`HDO_dbfile` returns the path (character string) to the package annotation DB (this is an SQLite file).

`HDO_dbschema` prints the schema definition of the package annotation DB.

`HDO_dbInfo` prints other information about the package annotation DB.

Value

`HDO_dbconn`: a DBIConnection object representing an open connection to the package annotation DB.

`HDO_dbfile`: a character string with the path to the package annotation DB.

`HDO_dbschema`: none (invisible `NULL`).

`HDO_dbInfo`: none (invisible `NULL`).

See Also

`dbGetQuery`, `dbConnect`, `dbconn`, `dbfile`, `dbschema`, `dbInfo`

Examples

```
## Count the number of rows in the "hdo_term" table:
HDO_dbconn()
```

HDOmetadata *Annotation of HDO Identifiers to HDO Terms*

Description

This data set gives mappings between HDO identifiers and their respective terms.

Details

Each HDO identifier is mapped to a `HDOTerms` object that has 2 slots: name: HDO name; value: The value

Mappings were based on data provided by: Disease Ontology With a date stamp from the source of: 20220706

References

http://do-wiki.nubic.northwestern.edu/index.php/Main_Page

Examples

```
# Convert the object to a data.frame
library(AnnotationDbi)
xx <- toTable(HDOmetadata)
```

Index

∗ **classes**
HDODb-objects, 3
∗ **datasets**
HDO.db, 2
HDO_dbconn, 8
HDOALIAS, 1
HDOANCESTOR, 1
HDOCHILDREN, 2
HDOMAPCOUNTS, 5
HDOmetadata, 9
HDOOBSOLETE, 5
HDOOFFSPRING, 6
HDOPARENTS, 6
HDOSYNONYM, 7
HDOTERM, 7
∗ **methods**
HDODb-objects, 3
∗ **utilities**
HDO_dbconn, 8

AnnDbObj, *8*

cat, *8*
checkMAPCOUNTS, *5*
columns *(HDODb-objects)*, 3
columns,HDODb-method *(HDODb-objects)*, 3
count.mappedkeys, *5*

dbconn, *8*
dbConnect, *8*
dbDisconnect, *8*
dbfile, *8*
dbGetQuery, *8*
dbInfo, *8*
dbschema, *8*

HDO *(HDO.db)*, 2
HDO.db, 2
HDO_dbconn, 8
HDO_dbfile *(HDO_dbconn)*, 8
HDO_dbInfo *(HDO_dbconn)*, 8
HDO_dbschema *(HDO_dbconn)*, 8
HDOALIAS, 1
HDOANCESTOR, 1
HDOCHILDREN, 2
HDODb-class *(HDODb-objects)*, 3
HDODb-objects, 3
HDOMAPCOUNTS, 5
HDOmetadata, 9
HDOOBSOLETE, 5
HDOOFFSPRING, 6
HDOPARENTS, 6
HDOSYNONYM, 7
HDOTERM, 7

keys *(HDODb-objects)*, 3
keys,HDODb-method *(HDODb-objects)*, 3
keytypes *(HDODb-objects)*, 3
keytypes,HDODb-method *(HDODb-objects)*, 3

mappedkeys, *5*

select *(HDODb-objects)*, 3
select,HDODb-method *(HDODb-objects)*, 3