MPO.db

May 1, 2024

`MPOALIAS` *Map from MPO id to MPO alias*

Description

MPOALIAS is an R object that provides mapping from MPO id to MPO alias

Details

Mappings were based on data provided by: Mouse Phenotype Ontology With a date stamp from the source of: 20230302

Value

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

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

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

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

References

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

Examples

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

MPOANATOMY *Annotation of MPO id to their anatomy*

Description

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

Details

Each MPO id is mapped to an anatomy. Mappings were based on data provided by: MGI database.

Value

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

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

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

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

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno

Examples

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

MPOANCESTOR *Annotation of MPO Identifiers to their Ancestors*

Description

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

Details

Each MPO term is mapped to a vector of ancestor MPO terms. Mappings were based on data provided by: Mouse Phenotype Ontology With a date stamp from the source of: 20230302

Value

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

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

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

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

References

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

Examples

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

`MPO.db` *Bioconductor annotation data package*

Description

Welcome to the MPO.db annotation Package. The purpose of this package is to provide detailed information about the latest version of the Mouse Phenotype Ontology. It also provides the gene annotation of phenotype and disease. This package is updated biannually. You can learn what objects this package supports with the following command: `ls("package:MPO.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.

Value

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

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

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

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

Examples

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

MPOCHILDREN *Annotation of MPO Identifiers to their Children*

Description

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

Details

Each MPO term is mapped to a vector of ancestor MPO terms. Mappings were based on data provided by: Mouse Phenotype Ontology With a date stamp from the source of: 20230302

Value

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

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

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

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

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno

Examples

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

MPODb-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` `MPODb`, `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 mpid keys
mpkeys <- head(keys(MPO.db))
res <- select(x = MPO.db, keys = mpkeys, keytype = "mpid",
    columns = c("offspring", "term", "parent"))

## use term keys
mpkeys <- head(keys(MPO.db, keytype = "term"))
res <- select(x = MPO.db, keys = mpkeys, keytype = "term",
    columns = c("offspring", "mpid", "parent"))
```

MPOMAPCOUNTS *Number of mapped keys for the maps in package MPO.db*

Description

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

Value

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

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

MPO_dbschema: none (invisible NULL).

MPO_dbInfo: none (invisible NULL).

See Also

mappedkeys, count.mappedkeys, checkMAPCOUNTS

Examples

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

MPOMGIDO *Annotation of Gene ENTREZID ID to DO id*

Description

This data set describes associations between gene ENTREZID id and DO ids, based on mh_mapping_initiative and HumanDiseaseOntology.

Details

Each gene ENTREZID id is mapped to a vector of DO ids. Mappings were based on data provided by: MGI, mh_mapping_initiative(Github), and HumanDiseaseOntology(Github).

Value

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

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

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

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

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno and https://github.com/DiseaseOntology/HumanDiseaseOntology and https://github.com/mapping-commons/mh_mapping_initiative

Examples

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

MPOMPDO *Annotation of MPO Identifiers to DO ID*

Description

This data set describes associations between MPO ids and DO ids, based on mh_mapping_initiative and HumanDiseaseOntology. The format is an R object mapping the MPO ids to DO ids.

Details

Each MPO id is mapped to a vector of DO ids. Mappings were based on data provided by: mh_mapping_initiative and HumanDiseaseOntology.

Value

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

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

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

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

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno and https://github.com/DiseaseOntology/HumanDiseaseOntology and https://github.com/mapping-commons/mh_mapping_initiative

Examples

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

MPOMPMGI *Annotation of MPO Ids to Gene ENTREZID Ids*

Description

This data set describes associations between MPO ids and gene ENTREZID ids, based on the MGI database.

Details

Each MPO id is mapped to a vector of gene ENTREZID ids. Mappings were based on data provided by: MGI database.

Value

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

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

MPO_dbschema: none (invisible NULL).

MPO_dbInfo: none (invisible NULL).

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno

Examples

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

MPOOBSOLETE *Annotation of MPO Ids by terms defined by Mouse Phenotype Ontology Consortium and their status are obsolete*

Description

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

Details

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

Mappings were based on data provided by: Mouse Phenotype Ontology With a date stamp from the source of: 20230302

Value

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

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

MPO_dbschema: none (invisible NULL).

MPO_dbInfo: none (invisible NULL).

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno

MPOOFFSPRING *Annotation of MPO Ids to their Offspring*

Description

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

Details

Each MPO term is mapped to a vector of offspring MPO terms. Mappings were based on data provided by: Mouse Phenotype Ontology With a date stamp from the source of: 20230302

Value

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

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

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

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

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno

Examples

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

MPOPARENTS *Annotation of MPO Identifiers to their Parents*

Description

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

Details

Each MPO term is mapped to a named vector of MPO 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: Mouse Phenotype Ontology With a date stamp from the source of: 20230302

Value

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

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

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

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

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno

Examples

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

MPOSYNONYM *Map from MPO synonyms to MPO terms*

Description

MPOSYNONYM is an R object that provides mapping from MPO synonyms to MPO terms

Details

Mappings were based on data provided by: Mouse Phenotype Ontology With a date stamp from the source of: 20230302

Value

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

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

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

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

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno

Examples

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

MPOTERM *Annotation of MPO Ids to MPO Terms*

Description

This data set gives mappings between MPO ids and their respective terms.

Details

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

Mappings were based on data provided by: Mouse Phenotype Ontology With a date stamp from the source of: 20230302

Value

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

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

MPO_dbschema: none (invisible NULL).

MPO_dbInfo: none (invisible NULL).

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno

Examples

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

MPO_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

```
MPO_dbconn()
MPO_dbfile()
MPO_dbschema(file="", show.indices=FALSE)
MPO_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

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

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

MPO_dbschema prints the schema definition of the package annotation DB.

MPO_dbInfo prints other information about the package annotation DB.

Value

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

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

MPO_dbschema: none (invisible NULL).

MPO_dbInfo: none (invisible NULL).

See Also

dbGetQuery, dbConnect, dbconn, dbfile, dbschema, dbInfo

Examples

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

MPOmetadata *Annotation of MPO Meta Data*

Description

This data set gives information of MPO meta data.

Details

Mappings were based on data provided by: Mouse Phenotype Ontology With a date stamp from the source of: 20230302

Value

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

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

MPO_dbschema: none (invisible NULL).

MPO_dbInfo: none (invisible NULL).

References

http://www.informatics.jax.org/downloads/reports/index.html#pheno

Examples

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

Index