HPO.db

November 11, 2026

HPOALIAS *Map from HPO alias to HPO terms*

Description

HPOALIAS is an R object that provides mapping from HPO alias to HPO terms

Details

Mappings were based on data provided by: Human Phenotype Ontology With a date stamp from the source of: 20230405

Value

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

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

HPO_dbschema: none (invisible NULL).

HPO_dbInfo: none (invisible NULL).

References

https://hpo.jax.org/app/data/ontology

Examples

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

HPOANCESTOR *Annotation of HPO Identifiers to their Ancestors*

Description

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

Details

Each HPO term is mapped to a vector of ancestor HPO terms. Mappings were based on data provided by: Human Phenotype Ontology With a date stamp from the source of: 20230405

Value

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

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

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

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

References

https://hpo.jax.org/app/data/ontology

Examples

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

HPO.db *Bioconductor annotation data package*

Description

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

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

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

HPO_dbschema: none (invisible NULL).

HPO_dbInfo: none (invisible NULL).

Examples

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

HPOCHILDREN *Annotation of HPO Identifiers to their Children*

Description

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

Details

Each HPO term is mapped to a vector of children HPO terms. Mappings were based on data provided by: Human Phenotype Ontology With a date stamp from the source of: 20230405

Value

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

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

HPO_dbschema: none (invisible NULL).

HPO_dbInfo: none (invisible NULL).

References

https://hpo.jax.org/app/data/ontology

Examples

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

HPODO *Annotation of HPO Identifiers to DO ID*

Description

This data set describes associations between HPO ids and do ids, based on the Human Phenotype Ontology. The format is an R object mapping the HPO ids to do ids.

Details

Each HPO id is mapped to a vector of do ids. Mappings were based on data provided by: The Human Phenotype Ontology

Value

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

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

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

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

References

https://github.com/mapping-commons/mh_mapping_initiative

Examples

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

HPODb-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` `HPODb`, `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 hpoid keys
dokeys <- head(keys(HPO.db))
res <- select(x = HPO.db, keys = dokeys, keytype = "hpoid",
    columns = c("offspring", "term", "parent"))

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

HPOGENE *Annotation of HPO Identifiers to gene ID*

Description

This data set describes associations between HPO ids and gene ids, based on the Human Phenotype Ontology. The format is an R object mapping the HPO ids to gene ids.

Details

Each HPO id is mapped to a vector of gene ids. Mappings were based on data provided by: The Human Phenotype Ontology

Value

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

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

HPO_dbschema: none (invisible NULL).

HPO_dbInfo: none (invisible NULL).

References

https://hpo.jax.org/app/data/ontology

Examples

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

HPOMAPCOUNTS *Number of mapped keys for the maps in package HPO.db*

Description

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

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

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

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

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

See Also

mappedkeys, count.mappedkeys, checkMAPCOUNTS

Examples

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

HPOMPO *Annotation of HPO Identifiers to MPO ID*

Description

This data set describes associations between HPO ids and MPO ids, based on the Human Phenotype Ontology. The format is an R object mapping the HPO ids to MPO ids.

Details

Each HPO id is mapped to a vector of MPO ids. Mappings were based on data provided by: The Human Phenotype Ontology

Value

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

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

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

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

References

https://github.com/mapping-commons/mh_mapping_initiative

Examples

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

HPOOBSOLETE *Annotation of HPO identifiers by terms defined by Human Phenotype Ontology Consortium and their status are obsolete*

Description

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

Details

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

Mappings were based on data provided by: Human Phenotype Ontology With a date stamp from the source of: 220230405

Value

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

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

HPO_dbschema: none (invisible NULL).

HPO_dbInfo: none (invisible NULL).

References

https://hpo.jax.org/app/data/ontology

HPOOFFSPRING *Annotation of HPO Identifiers to their Offspring*

Description

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

Details

Each HPO term is mapped to a vector of offspring HPO terms. Mappings were based on data provided by: Human Phenotype Ontology With a date stamp from the source of: 20230405

Value

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

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

HPO_dbschema: none (invisible NULL).

HPO_dbInfo: none (invisible NULL).

References

https://hpo.jax.org/app/data/ontology

Examples

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

HPOPARENTS *Annotation of HPO Identifiers to their Parents*

Description

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

Details

Each HPO term is mapped to a named vector of HPO 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: Human Phenotype Ontology With a date stamp from the source of: 220230405

Value

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

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

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

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

References

https://hpo.jax.org/app/data/ontology

Examples

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

HPOSYNONYM *Map from HPO synonyms to HPO terms*

Description

HPOSYNONYM is an R object that provides mapping from HPO synonyms to HPO terms

Details

Mappings were based on data provided by: Human Phenotype Ontology With a date stamp from the source of: 20230405

Value

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

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

HPO_dbschema: none (invisible NULL).

HPO_dbInfo: none (invisible NULL).

References

https://hpo.jax.org/app/data/ontology

Examples

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

HPOTERM *Annotation of HPO Identifiers to HPO Terms*

Description

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

Details

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

Mappings were based on data provided by: Human Phenotype Ontology With a date stamp from the source of: 220230405

Value

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

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

HPO_dbschema: none (invisible NULL).

HPO_dbInfo: none (invisible NULL).

References

https://hpo.jax.org/app/data/ontology

Examples

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

HPO_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

```
HPO_dbconn()
HPO_dbfile()
HPO_dbschema(file="", show.indices=FALSE)
HPO_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

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

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

HPO_dbschema prints the schema definition of the package annotation DB.

HPO_dbInfo prints other information about the package annotation DB.

Value

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

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

HPO_dbschema: none (invisible NULL).

HPO_dbInfo: none (invisible NULL).

See Also

dbGetQuery, dbConnect, dbconn, dbfile, dbschema, dbInfo

Examples

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

HPOmetadata *Annotation of HPO Identifiers to HPO Terms*

Description

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

Details

Each HPO identifier is mapped to a HPOTerms object that has 2 slots: name: HPO name; value: The value

Mappings were based on data provided by: Human Phenotype Ontology With a date stamp from the source of: 20230405

Value

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

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

HPO_dbschema: none (invisible NULL).

HPO_dbInfo: none (invisible NULL).

References

https://hpo.jax.org/app/data/ontology

Examples

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

Index

∗ **classes**
HPODb-objects, 4
∗ **datasets**
HPO.db, 2
HPO_dbconn, 12
HPOALIAS, 1
HPOANCESTOR, 2
HPOCHILDREN, 3
HPODO, 4
HPOGENE, 6
HPOMAPCOUNTS, 7
HPOmetadata, 13
HPOMPO, 8
HPOOBSOLETE, 8
HPOOFFSPRING, 9
HPOPARENTS, 10
HPOSYNONYM, 10
HPOTERM, 11
∗ **methods**
HPODb-objects, 4
∗ **utilities**
HPO_dbconn, 12

AnnDbObj, *12*

cat, *12*
checkMAPCOUNTS, *7*
columns *(*HPODb-objects*)*, 4
columns,HPODb-method *(*HPODb-objects*)*, 4
count.mappedkeys, *7*

dbconn, *12*
dbConnect, *12*
dbDisconnect, *12*
dbfile, *12*
dbGetQuery, *12*
dbInfo, *12*
dbschema, *12*

HPO *(*HPO.db*)*, 2
HPO.db, 2
HPO_dbconn, 12
HPO_dbfile *(*HPO_dbconn*)*, 12
HPO_dbInfo *(*HPO_dbconn*)*, 12
HPO_dbschema *(*HPO_dbconn*)*, 12
HPOALIAS, 1
HPOANCESTOR, 2
HPOCHILDREN, 3
HPODb-class *(*HPODb-objects*)*, 4
HPODb-objects, 4
HPODO, 4
HPOGENE, 6
HPOMAPCOUNTS, 7
HPOmetadata, 13
HPOMPO, 8
HPOOBSOLETE, 8
HPOOFFSPRING, 9
HPOPARENTS, 10
HPOSYNONYM, 10
HPOTERM, 11

keys *(*HPODb-objects*)*, 4
keys,HPODb-method *(*HPODb-objects*)*, 4
keytypes *(*HPODb-objects*)*, 4
keytypes,HPODb-method *(*HPODb-objects*)*, 4

mappedkeys, *7*

select *(*HPODb-objects*)*, 4
select,HPODb-method *(*HPODb-objects*)*, 4