

# PANTHER.db

December 31, 2025

---

PANTHER.db-objects

*PANTHER.db objects and their related methods and functions*

---

## Description

In much the same way as an `AnnotationDb` object allows access to select for many other annotation packages, `PANTHER.db` is meant to allow usage of `select` methods and other supporting methods to enable the easy extraction of data from the PANTHER gene ontology.

`select`, `columns` and `keys` are used together to extract data via an `PANTHER.db` object.

`columns` shows which kinds of data can be returned for the `PANTHER.db` object.

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

`keys` returns keys for the database contained in the `PANTHER.db` object. By default it will return the primary keys for the database, which are `FAMILY_ID` keys, 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 and columns and `keytype` arguments.

`pthOrganisms` shows the organism label currently attached to the `PANTHER.db` objects database. The organism supported by `PANTHER.db` is currently restricted to the organisms supported by the main annotation packages in Bioconductor.

The `PANTHER.db` object will be loaded whenever you load the `PANTHER.db` package. This object will be set up to return information on PANTHER ontology terms and accessions based on all available organisms. The `pthOrganisms<-` method can be used to restrict queries to a specific organism and `resetPthOrganisms` can be used to remove the organism restriction.

`availablePthOrganisms` is a helper function to list out the available Species currently supported by the `PANTHER.db` package along with their official taxonomy IDs that are available by Uniprot.

The PANTHER Protein Class ontology was adapted from the PANTHER/X molecular function ontology, and includes commonly used classes of protein functions, many of which are not covered by GO molecular function. `traverseClassTree` can be used to traverse the tree structure of the PANTHER Protein Class ontology.

**Usage**

```

columns(x)
keytypes(x)
keys(x, keytype,...)
select(x, keys, columns, keytype, ...)
pthOrganisms(x)
availablePthOrganisms(x)
resetPthOrganisms(x)
traverseClassTree(x,query,scope)

```

**Arguments**

x	the PANTHER.db 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
query	character vector of valid PANTHER class identifiers e.g. PC000000
scope	character CHILD, PARENT, ANCESTOR or OFFSPRING to define the scope of the PANTHER class identifiers returned.
...	other arguments

**Value**

keys,columns,keytypes, traverseClassTree and pthOrganisms each return a character vector of possible values.

select and availablePthOrganisms each return a data.frame.

**Author(s)**

Julius Muller

**See Also**

select

**Examples**

```

## look at the object
PANTHER.db

## get the current organism
pthOrganisms(PANTHER.db)

## list the possible key types
keytypes(PANTHER.db)

## list the columns that can be retrieved
columns(PANTHER.db)

```

```
## list all possible keys of type entrez gene ID.  
head(keys(PANTHER.db))
```

```
## use select to extract some data  
keys <- c("E1C9F4", "O14618")  
cols <- c("FAMILY_ID", "SPECIES", "FAMILY_TERM")  
kt <- "UNIPROT"  
select(PANTHER.db, keys, cols, kt)
```

# Index

## \* classes

PANTHER.db-objects, [1](#)

## \* methods

PANTHER.db-objects, [1](#)

availablePthOrganisms

(PANTHER.db-objects), [1](#)

availablePthOrganisms, PANTHER.db-method

(PANTHER.db-objects), [1](#)

class:PANTHER.db (PANTHER.db-objects), [1](#)

columns (PANTHER.db-objects), [1](#)

columns, PANTHER.db-method

(PANTHER.db-objects), [1](#)

keys (PANTHER.db-objects), [1](#)

keys, PANTHER.db-method

(PANTHER.db-objects), [1](#)

keytypes (PANTHER.db-objects), [1](#)

keytypes, PANTHER.db-method

(PANTHER.db-objects), [1](#)

PANTHER (PANTHER.db-objects), [1](#)

PANTHER, PANTHER.db-method

(PANTHER.db-objects), [1](#)

PANTHER.db (PANTHER.db-objects), [1](#)

PANTHER.db-class (PANTHER.db-objects), [1](#)

PANTHER.db-objects, [1](#)

PANTHER\_dbconn (PANTHER.db-objects), [1](#)

PANTHER\_dbconn, PANTHER.db-method

(PANTHER.db-objects), [1](#)

PANTHER\_dbfile (PANTHER.db-objects), [1](#)

PANTHER\_dbfile, PANTHER.db-method

(PANTHER.db-objects), [1](#)

PANTHER\_dbInfo (PANTHER.db-objects), [1](#)

PANTHER\_dbInfo, PANTHER.db-method

(PANTHER.db-objects), [1](#)

PANTHER\_dbschema (PANTHER.db-objects), [1](#)

PANTHER\_dbschema, PANTHER.db-method

(PANTHER.db-objects), [1](#)

pthOrganisms (PANTHER.db-objects), [1](#)

pthOrganisms, PANTHER.db-method

(PANTHER.db-objects), [1](#)

pthOrganisms<- (PANTHER.db-objects), [1](#)

pthOrganisms<-, PANTHER.db-method

(PANTHER.db-objects), [1](#)

resetPthOrganisms (PANTHER.db-objects),

[1](#)

resetPthOrganisms, PANTHER.db-method

(PANTHER.db-objects), [1](#)

select (PANTHER.db-objects), [1](#)

select, PANTHER.db-method

(PANTHER.db-objects), [1](#)

show, PANTHER.db-method

(PANTHER.db-objects), [1](#)

traverseClassTree (PANTHER.db-objects),

[1](#)

traverseClassTree, PANTHER.db, character, character-method

(PANTHER.db-objects), [1](#)