

geneslator

August 12, 2026

GeneslatorDb

GeneslatorDb class

Description

The GeneslatorDb class is the container for storing annotation databases in the **geneslator** package.

Usage

```
GeneslatorDb(org, release.version = "latest")
```

Arguments

org	A character string specifying the scientific name of the organism (e.g. "Homo sapiens") or its Taxonomy ID. See availableDatabases() for the list of supported organisms.
release.version	A character string indicating the release version of the annotation database (e.g. "2025-12"). See availableVersions() for the list of available releases.

Details

The GeneslatorDb class is the container for storing annotation databases in the **geneslator** package. It wraps an OrgDb object, which represents the annotation database of a specific organism.

Annotation databases used by **geneslator** are updated on a monthly basis and available as different versions of a Zenodo record at <https://doi.org/10.5281/zenodo.20448208> as SQLite files. Each release refers to a specific version of the databases. Versions are indicated as year.month, where year and month denote the year and the month of the publication of the release (e.g. '2026.03'). Each database in a release refers to a specific organism.

The constructor method GeneslatorDb(org) creates a new GeneslatorDb object for the annotation database of organism org. Once created, the object is exported to the global environment of the user as a variable having the same name of the annotation database (e.g. org.Hsapiens.db for Human, org.Mmusculus.db for Mouse). By default, the constructor method considers the latest release of the database. An older version can be specified through parameter release.version. See [availableDatabases\(\)](#) and [availableVersions\(\)](#) for the list of available databases and release versions.

When called, the constructor method first look for a copy of the SQLite file in the R cache folder of the user. If the SQLite file exists and is up-to-date, the cached copy is used to create the GeneslatorDb object. Otherwise, upon request by the user, the database is downloaded from the remote release and copied in the **geneslator** package cache, before creating the object.

Value

A GeneslatorDb object.

Slots

db The annotation database represented as an OrgDb object.

Examples

```
# Create a GeneslatorDb object for Human
# First call: download human db (org.Hsapiens.db) from latest release and
# save it to R cache
GeneslatorDb("Homo sapiens")
org.Hsapiens.db
# Second call: load db from local cache
GeneslatorDb("Homo sapiens")
org.Hsapiens.db

# Create a GeneslatorDb object for Fly.
# Use taxonomy id and release version 2025.12
GeneslatorDb("7227", "2025.12")
org.Dmelanogaster.db
```

availableDatabases

Available databases in geneslator

Description

availableDatabases lists all possible annotation databases that can be queried in the **geneslator** package. Databases are updated on a monthly basis and available as different versions of a Zenodo record at <https://doi.org/10.5281/zenodo.20448208>. Each release refer to a specific version of the databases. Versions are indicated as year.month, where year and month denote the year and the month of the publication of the release (e.g. '2026.03'). Each database in a release refer to a specific organism.

Usage

```
availableDatabases(release.version = "latest")
```

Arguments

release.version

Release version of the databases. By default, the most recent version is considered ("latest"). Older versions must be indicated as year.month, where year and month denote the year and the month of the publication of the release (e.g. "2026.03"). See [availableVersions\(\)](#) for the list of available release versions.

Value

availableDatabases returns a dataframe which reports, for each annotation database: database name, scientific name of the organism, Taxonomy ID of the organism, MD5 security check of the SQLite database file and release version. Database info refer to the release version specified by the version parameter.

See Also

[GeneslatorDb](#), [availableVersions](#).

Examples

```
# List all databases included in the current geneslator release
availableDatabases()

# List all databases included in geneslator release version 2025.12
availableDatabases("2025.12")
```

availableVersions	<i>Available database versions in geneslator</i>
-------------------	--

Description

`availableVersions` lists all possible versions of the annotation databases that can be queried in the **geneslator** package. Databases are updated on a monthly basis and available as different versions of a Zenodo record at <https://doi.org/10.5281/zenodo.20448208>. Each release refer to a specific version of the databases. Versions are indicated as year.month, where year and month denote the year and the month of the publication of the release (e.g. '2026.03').

Usage

```
availableVersions()
```

Value

`availableVersions` returns a character vector with all available versions of the geneslator annotation databases.

See Also

[GeneslatorDb](#), [availableDatabases](#).

Examples

```
# List all available versions of geneslator databases
availableVersions()
```

keys	<i>List values of a column in the annotation databases of geneslator</i>
------	--

Description

The keys function lists of all possible values for a given column in the annotation database of a specific organism within the **geneslator** package.

Usage

```
## S4 method for signature 'GeneslatorDb'
keys(x, keytype)
```

Arguments

x	A GeneslatorDb object returned by GeneslatorDb() . It represents the annotation database to query from.
keytype	Name of the column from which the list of values should be extracted. See keytypes() for the list of available columns for the annotation database x.

Value

keys returns a character vector of all possible values of the column keytype in database x.

See Also

[keytypes\(\)](#), [mapIds\(\)](#), [select\(\)](#)

Examples

```
# Get the list of all NCBI gene ids present in zebrafish annotation db
GeneslatorDb("Danio rerio")
geneslator::keys(org.Drerio.db, keytype = "ENTREZID")

# Get the list of all KEGG pathways present in rat annotation db
GeneslatorDb("Rattus norvegicus")
geneslator::keys(org.Rnorvegicus.db, keytype = "KEGGPATH")
```

keytypes	<i>List available columns in the annotation databases of geneslator</i>
----------	---

Description

Functions keytypes and columns are used to access the complete lists of input and output columns that can be queried in the annotation databases of the **geneslator** package through [mapIds\(\)](#) and [select\(\)](#) functions.

Usage

```
## S4 method for signature 'GeneslatorDb'
keytypes(x)

## S4 method for signature 'GeneslatorDb'
columns(x)
```

Arguments

x A GeneslatorDb object returned by `GeneslatorDb()`. It represents the annotation database to query from.

Details

`keytypes()` lists all possible columns of the annotation database `x` that can be used as input when querying `x`, i.e., all possible values of the `keytype` argument in `mapIds()` and `select()` functions.

`columns()` lists all possible columns of the annotation database `x` that can be used as output when querying `x`, i.e., all possible values of the `column` argument in `mapIds()` and `select()` functions.

The following is the complete list of columns defined in the annotation databases of **geneslator** package. Some of these columns may be missing in one or more organisms.

Column	Description
SYMBOL	Official gene symbol
ALIAS	Aliases of a gene
GENETYPE	Biological type of a gene (e.g. 'protein-coding', 'ncRNA')
GENENAME	Full name or description of a gene
ENTREZID	Gene ID in NCBI Gene
ENSEMBL	Gene ID in Ensembl
HGNC	Gene ID in HUGO Gene Nomenclature Committee (Human only)
MGI	Gene ID in Mouse Genome Informatics (Mouse only)
RGD	Gene ID in Rat Genome Database (Rat only)
SGD	Gene ID in Saccharomyces Genome Database (Yeast only)
WORMBASE	Gene ID in WormBase database (Worm only)
FLYBASE	Gene ID in FlyBase database (Fly only)
ZFIN	Gene ID in Zebrafish Information Network (Zebrafish only)
TAIR	Gene ID in The Arabidopsis Information Resource (Arabidopsis only)
UNIPROTKB	Uniprot IDs of proteins associated to a gene
ENTREZIDOLD	Archived IDs in NCBI Gene
ENSEMBLOLD	Archived IDs in Ensembl
ORTHOHUMAN	Orthologs in Human (absent in Human and Arabidopsis)
ORTHOMOUSE	Orthologs in Mouse (absent in Mouse and Arabidopsis)
ORTHORAT	Orthologs in Rat (absent in Rat and Arabidopsis)
ORTHOYEAST	Orthologs in Yeast (absent in Yeast and Arabidopsis)
ORTHOWORM	Orthologs in Worm (absent in Worm and Arabidopsis)
ORTHOFLY	Orthologs in Fly (absent in Fly and Arabidopsis)
ORTHOZEBRAFISH	Orthologs in Zebrafish (absent in Zebrafish and Arabidopsis)
GO	IDs of Gene Ontology (GO) terms associated to a gene
GONAME	Names of GO terms associated to a gene
GOEVIDENCE	Evidence codes of GO terms associated to a gene
GOTYPE	Types of GO terms ('BP'=biological process, 'CC'=cellular

component, 'MF'=molecular function) associated to a gene

KEGGPATH

KEGGPATHNAME

REACTOMEPATH

REACTOMEPATHNAME

WIKIPATH

WIKIPATHNAME

IDs of KEGG pathways associated to a gene

Names of KEGG pathways associated to a gene

IDs of Reactome pathways associated to a gene

Names of Reactome pathways associated to a gene

IDs of Wikipathways pathways associated to a gene

Names of Wikipathways pathways associated to a gene

Value

keytypes() and columns() return a character vector of column names of database x.

See Also

[availableDatabases](#), [mapIds](#), [select](#)

Examples

```
# Get the list of available keytypes in mouse
GeneslatorDb("Mus musculus")
geneslator::keytypes(org.Mmusculus.db)

# Get the list of available columns that can be mapped to keys in yeast
GeneslatorDb("Saccharomyces cerevisiae")
geneslator::columns(org.Scerevisiae.db)
```

mapIds

Map data from the annotation databases of geneslator

Description

mapIds maps key values of a column to values of another column in the annotation databases of **geneslator** package.

Usage

```
## S4 method for signature 'GeneslatorDb'
mapIds(
  x,
  keys,
  column,
  keytype,
  search.aliases = TRUE,
  search.archives = TRUE,
  ...,
  multiVals
)
```

Arguments

x	A GeneslatorDb object returned by GeneslatorDb() . It represents the annotation database to query from.
keys	Values used as keys to retrieve records from the annotation database.
column	Column to return as output of the query. See columns() for more details.
keytype	Column representing the type of values of keys parameter. See keytypes() for more details.
search.aliases	When no mapping is found using gene symbol (SYMBOL column), should select perform query using also ALIAS column? (default = TRUE). This parameter is used only in queries involving SYMBOL column.
search.archives	When no mapping is found using NCBI gene ids (ENTREZID column) and/or Ensembl gene ids (ENSEMBL column), should select perform query using also archived identifiers (columns ENTREZIDOLD and/or ENSEMBLOLD)? (default = TRUE). This parameter is used only in queries involving ENTREZID and/or ENSEMBL column.
...	Other arguments. See AnnotationDb for more info.
multiVals	What should mapIds do when there are multiple output values that could be returned for a specific input? Options include:

Option	Description
first	Return a vector object containing only the first match found
for each input (default behaviour).	
asNA	Return a vector object with NA values whenever there are
multiple matches for a given input.	
filter	Return a shorter vector object, excluding all inputs for which
multiple matches have been found.	
list	Return a list object with all matches found for each input.
CharacterList	Return a SimpleCharacterList object with all matches
found for each input.	
FUN	Supply a function to the multiVals argument for custom
behaviors.	

If using FUN, the function must take a single argument and return a single value. This function will be applied to all elements and will serve as a 'rule' for which item to keep when there is more than one match for a given input. For example, the following function grabs the last element in each result: `last <- function(x) { x[[length(x)]] }`.

Details

mapIds maps each key value to either a single value or a list of values of the type specified by column parameter, depending on the value of multiVals parameter.

Value

mapIds returns either a named vector, where each value is a possible mapping (if exists) for a given key, or a list of values, where each element of the list is the vector of all mappings found for a given key. The type of the return object depends on the value of the multiVals parameter.

See Also

[availableDatabases](#), [keytypes](#), [columns](#)

Examples

```
# Map NCBI gene ids to gene aliases in yeast.
# Return a named vector with 1st mapping found
GeneslatorDb("Saccharomyces cerevisiae")
geneslator::mapIds(org.Scerevisiae.db,
  keys = c("856781", "1466469"),
  column = "ALIAS", keytype = "ENTREZID"
)

# Map gene symbols to gene ontologies in mouse.
# Return a list with all possible mappings
GeneslatorDb("Mus musculus")
geneslator::mapIds(org.Mmusculus.db,
  keys = c("Grin2a", "Rev3l"), column = "GO",
  keytype = "SYMBOL", multiVals = "list"
)

# Map gene symbols to uniprot ids in rat. Apply a custom function to
# return the last mapping found and do not use Ensembl archive data.
GeneslatorDb("Rattus norvegicus")
last <- function(x) {
  x[[length(x)]]
}
geneslator::mapIds(org.Rnorvegicus.db,
  keys = c(
    "ENSRNOG00000003105",
    "ENSRNOG00000049505"
  ), column = "UNIPROT", keytype = "ENSEMBL",
  multiVals = "list", search.archives = FALSE
)

# Map gene symbols to reactome pathways in zebrafish.
# Return a CharacterList object with all possible mappings
GeneslatorDb("Danio rerio")
geneslator::mapIds(org.Drerio.db,
  keys = c("hoxc8a", "samhd1"),
  column = "REACTOME_PATH", keytype = "SYMBOL", multiVals = "CharacterList"
)
```

select

Extract data from the annotation databases of geneslator

Description

select query annotation databases of **geneslator** package, by mapping different types of gene annotation data from several source of data.

Usage

```
## S4 method for signature 'GeneslatorDb'
select(
  x,
  keys,
  columns,
  keytype,
  search.aliases = TRUE,
  search.archives = TRUE,
  orthologs.mapping = "multiple",
  ...
)
```

Arguments

<code>x</code>	A <code>GeneslatorDb</code> object returned by <code>GeneslatorDb()</code> . It represents the annotation database to query from.
<code>keys</code>	Values used as keys to retrieve records from the annotation database.
<code>columns</code>	Columns to return as output of the query. See <code>columns()</code> for more details.
<code>keytype</code>	Column representing the type of values of keys parameter. See <code>keytypes()</code> for more details.
<code>search.aliases</code>	When no mapping is found using gene symbol (SYMBOL column), should select perform query using also ALIAS column? (default = TRUE). This parameter is used only in queries involving SYMBOL column.
<code>search.archives</code>	When no mapping is found using NCBI gene ids (ENTREZID column) and/or Ensembl gene ids (ENSEMBL column), should select perform query using also archived identifiers (columns ENTREZIDOLD and/or ENSEMBLOLD)? (default = TRUE). This parameter is used only in queries involving ENTREZID and/or ENSEMBL column.
<code>orthologs.mapping</code>	Return all orthologs ("multiple") or just the first ortholog ("single") of a gene? (default = "multiple"). Used only in queries where the output columns include ORTHO columns (e.g. ORTHOMOUSE, ORTHOYEAST).
<code>...</code>	Other arguments. See AnnotationDb for more info.

Details

`select` collects all possible mappings between values of the column specified by `keytype` parameter and values of the columns specified by the `columns` parameter.

Value

`select` returns a dataframe with all columns specified by `keytype` and `columns` parameters and one row for each mapping found between keys and column values.

See Also

[availableDatabases](#), [keytypes](#), [columns](#)

Examples

```
# Lookup NCBI gene ids for a given list of gene symbols in fly
GeneslatorDb("Drosophila melanogaster")
geneslator::select(org.Dmelanogaster.db,
  keys = c("CG14883", "GstE2"),
  columns = "ENTREZID", keytype = "SYMBOL"
)

# Lookup KEGG pathway ids and their relative full names for a given list
# of ensembl gene ids in worm
GeneslatorDb("Caenorhabditis elegans")
geneslator::select(org.Celegans.db,
  keys = c(
    "ENSDARG00000013522",
    "ENSDARG000000103044"
  ), columns = c("KEGGPATH", "KEGGPATHNAME"),
  keytype = "ENSEMBL"
)

# Lookup mouse orthologs for a list of human gene symbols.
# Ignore aliases and return only the first ortholog found for each gene
GeneslatorDb("Homo sapiens")
geneslator::select(org.Hsapiens.db,
  keys = c("BRCA1", "PTEN"),
  columns = "ORTHOMOUSE", keytype = "SYMBOL", search.aliases = FALSE,
  orthologs.mapping = "single"
)

# Lookup gene ontologies for a list of entrez ids in arabidopsis.
# Do not use NCBI archive data
GeneslatorDb("Arabidopsis thaliana")
geneslator::select(org.Athaliana.db,
  keys = c("820005", "831939"),
  columns = c("GO", "GONAME", "GOTYPE"), keytype = "ENTREZID",
  search.archives = FALSE
)
```

show, GeneslatorDb-method

Show method for GeneslatorDb

Description

Show method for GeneslatorDb

Usage

```
## S4 method for signature 'GeneslatorDb'
show(object)
```

Arguments

object A GeneslatorDb object.

Value

Invisibly returns NULL. Called for its side effect of printing a summary of the object to the console.

Index

AnnotationDb, [7](#), [9](#)
availableDatabases, [2](#), [3](#), [6](#), [8](#), [9](#)
availableDatabases(), [1](#)
availableVersions, [3](#), [3](#)
availableVersions(), [1](#), [2](#)

columns, [8](#), [9](#)
columns (keytypes), [4](#)
columns(), [7](#), [9](#)
columns, GeneslatorDb-method (keytypes),
[4](#)

GeneslatorDb, [1](#), [3](#)
GeneslatorDb(), [4](#), [5](#), [7](#), [9](#)
GeneslatorDb-class (GeneslatorDb), [1](#)

keys, [4](#)
keys, GeneslatorDb-method (keys), [4](#)
keytypes, [4](#), [8](#), [9](#)
keytypes(), [4](#), [7](#), [9](#)
keytypes, GeneslatorDb-method
(keytypes), [4](#)

mapIds, [6](#), [6](#)
mapIds(), [4](#), [5](#)
mapIds, GeneslatorDb-method (mapIds), [6](#)

select, [6](#), [8](#)
select(), [4](#), [5](#)
select, GeneslatorDb-method (select), [8](#)
show, GeneslatorDb-method, [10](#)