

# Package ‘Organism.dplyr’

September 30, 2024

**Title** dplyr-based Access to Bioconductor Annotation Resources

**Version** 1.32.0

**Description** This package provides an alternative interface to Bioconductor 'annotation' resources, in particular the gene identifier mapping functionality of the 'org' packages (e.g., org.Hs.eg.db) and the genome coordinate functionality of the 'TxDb' packages (e.g., TxDb.Hsapiens.UCSC.hg38.knownGene).

**Depends** R (>= 3.4), dplyr (>= 0.7.0), AnnotationFilter (>= 1.1.3)

**Imports** RSQLite, S4Vectors, GenomeInfoDb, IRanges, GenomicRanges, AnnotationDbi, rlang, methods, tools, utils, BiocFileCache, DBI, dbplyr, tibble

**Suggests** org.Hs.eg.db, TxDb.Hsapiens.UCSC.hg38.knownGene, org.Mm.eg.db, TxDb.Mmusculus.UCSC.mm10.ensGene, testthat, knitr, rmarkdown, magick, BiocStyle, ggplot2

**License** Artistic-2.0

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Collate** src.R filter.R table-handlers.R extractors.R  
extractor-methods.R select.R utils.R join.R zzz.R

**VignetteBuilder** knitr

**biocViews** Annotation, Sequencing, GenomeAnnotation

**BugReports** <https://github.com/Bioconductor/Organism.dplyr/issues>

**git\_url** <https://git.bioconductor.org/packages/Organism.dplyr>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** 0ac9982

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-09-29

**Author** Martin Morgan [aut, cre],  
Daniel van Twisk [ctb],  
Yubo Cheng [aut]

**Maintainer** Martin Morgan <martin.morgan@roswellpark.org>

## Contents

BasicFilter-class . . . . .	2
Genomic-Extractors . . . . .	4
hg38light . . . . .	6
keytypes,src_organism-method . . . . .	7
src_organism . . . . .	9

<b>Index</b>	<b>12</b>
--------------	-----------

---

BasicFilter-class	<i>Filtering src_organism objects</i>
-------------------	---------------------------------------

---

## Description

These functions create filters to be used by the "select" interface to src\_organism objects.

## Usage

```

AccnumFilter(value, condition = "==")
AliasFilter(value, condition = "==")
CdsChromFilter(value, condition = "==")
CdsIdFilter(value, condition = "==")
CdsNameFilter(value, condition = "==")
CdsStrandFilter(value, condition = "==")
EnsemblFilter(value, condition = "==")
EnsemblprotFilter(value, condition = "==")
EnsembltransFilter(value, condition = "==")
EnzymeFilter(value, condition = "==")
EvidenceFilter(value, condition = "==")
EvidenceallFilter(value, condition = "==")
ExonChromFilter(value, condition = "==")
ExonStrandFilter(value, condition = "==")
FlybaseFilter(value, condition = "==")
FlybaseCgFilter(value, condition = "==")
FlybaseProtFilter(value, condition = "==")
GeneChromFilter(value, condition = "==")
GeneStrandFilter(value, condition = "==")
GoFilter(value, condition = "==")
GoallFilter(value, condition = "==")
IpiFilter(value, condition = "==")
MapFilter(value, condition = "==")
MgiFilter(value, condition = "==")
OmimFilter(value, condition = "==")
OntologyFilter(value, condition = "==")

```

```

OntologyallFilter(value, condition = "==")
PfamFilter(value, condition = "==")
PmidFilter(value, condition = "==")
PrositesFilter(value, condition = "==")
RefseqFilter(value, condition = "==")
TxChromFilter(value, condition = "==")
TxStrandFilter(value, condition = "==")
TxTypeFilter(value, condition = "==")
WormbaseFilter(value, condition = "==")
ZfinFilter(value, condition = "==")

## S4 method for signature 'BasicFilter'
show(object)

## S4 method for signature 'src_organism'
supportedFilters(object)

```

### Arguments

object	A BasicFilter or GRangesFilter object
value	Value of the filter. For GRangesFilter value should be a GRanges object.
condition	The condition to be used in filter for genomic extractors, one of "=", "!=", "startsWith", "endsWith", ">", "<", ">=", "<=". For character values "=", "!=", "startsWith" and "endsWith" are allowed, for numeric values (CdsStartFilter, CdsEndFilter, ExonStartFilter, ExonEndFilter, GeneStartFilter, GeneEndFilter, TxStartFilter and TxEndFilter), "=", "!=", ">", ">=", "<" and "<=". Default condition is "=".

### Details

All filters except GRangesFilter() takes value(s) from corresponding fields in the data base. For example, AccnumFilter() takes values of accession number(s), which come from field accnum. See keytypes() and keys() for possible values.

GRangesFilter() takes a GRanges object as filter, and returns genomic extractors (genes, transcripts, etc.) that are partially overlapping with the region.

supportedFilters() lists all available filters for src\_organism object.

### Value

A Filter object showing class, value and condition of the filter

### Author(s)

Yubo Cheng.

**See Also**

[src\\_organism](#) for creating a `src_organism` object.

[transcripts\\_tbl](#) for generic functions to extract genomic features from a `src_organism` object.

[select,src\\_organism-method](#) for "select" interface on `src_organism` objects.

**Examples**

```
src <- src_organism(dbpath=hg38light())
keytypes(src)
head(keys(src, "ensembl"))

## filter by ensembl
EnsemblFilter("ENSG00000171862")

## filter by gene symbol start with "BRAC"
SymbolFilter("BRCA", "startsWith")

## filter by GRanges
GRangesFilter(GenomicRanges::GRanges("chr10:87869000-87876000"))

## filter by transcript start position
TxStartFilter(87863438, ">")
```

---

Genomic-Extractors      *Extract genomic features from src\_organism objects*

---

**Description**

Generic functions to extract genomic features from an object. This page documents the methods for [src\\_organism](#) objects only.

These are the main functions for extracting transcript information from a [src\\_organism](#) object, inherited from [transcripts](#) in GenomicFeatures package. Two versions of results are provided: [tibble](#) (`transcripts_tbl()`) and [GRanges](#) or [GRangesList](#) (`transcripts()`).

**Usage**

```
cds(x, ...)
exons(x, ...)
genes(x, ...)
transcripts(x, ...)
cds_tbl(x, filter=NULL, columns=NULL)
exons_tbl(x, filter=NULL, columns=NULL)
genes_tbl(x, filter=NULL, columns=NULL)
transcripts_tbl(x, filter=NULL, columns=NULL)
cdsBy(x, by=c("tx", "gene"), ...)
exonsBy(x, by=c("tx", "gene"), ...)
```

```

transcriptsBy(x, by=c("gene", "exon", "cds"), ...)
cdsBy_tbl(x, by=c("tx", "gene"), filter=NULL, columns=NULL)
exonsBy_tbl(x, by=c("tx", "gene"), filter=NULL, columns=NULL)
transcriptsBy_tbl(x, by=c("gene", "exon", "cds"), filter=NULL, columns=NULL)
promoters_tbl(x, upstream, downstream, filter=NULL, columns=NULL)
intronsByTranscript_tbl(x, filter=NULL, columns=NULL)
fiveUTRsByTranscript(x, ...)
fiveUTRsByTranscript_tbl(x, filter=NULL, columns=NULL)
threeUTRsByTranscript(x, ...)
threeUTRsByTranscript_tbl(x, filter=NULL, columns=NULL)

## S4 method for signature 'src_organism'
promoters(x, upstream, downstream, filter = NULL, columns = NULL)

## S4 method for signature 'src_organism'
intronsByTranscript(x, filter = NULL, columns = NULL)

```

### Arguments

x	A <code>src_organism</code> object
upstream	For <code>promoters()</code> : An integer(1) value indicating the number of bases upstream from the transcription start site.
downstream	For <code>promoters()</code> : An integer(1) value indicating the number of bases downstream from the transcription start site.
filter	Either <code>NULL</code> , <code>AnnotationFilter</code> , or <code>AnnotationFilterList</code> to be used to restrict the output. Filters consists of <code>AnnotationFilters</code> and can be a <a href="#">GRanges</a> object using <code>"GRangesFilter"</code> (see examples).
columns	A character vector indicating columns to be included in output <code>GRanges</code> object or <code>tbl</code> .
by	One of "gene", "exon", "cds" or "tx". Determines the grouping.
...	Additional arguments to S4methods. In this case, the same as <code>filter</code> .

### Value

functions with `_tbl` return a [tibble](#) object, other methods return a [GRanges](#) or [GRangesList](#) object.

### Author(s)

Yubo Cheng.

### See Also

[src\\_organism](#) for creating a `src_organism` object.

**Examples**

```
## Not run: src <- src_ucsc("human")
src <- src_organism(dbpath=hg38light())

## transcript coordinates with filter in tibble format
filters <- AnnotationFilter(~symbol == c("A1BG", "CDH2"))
transcripts_tbl(src, filters)

transcripts_tbl(src, AnnotationFilter(~symbol %startsWith% "SNORD"))
transcripts_tbl(src, AnnotationFilter(~go == "GO:0005615"))
transcripts_tbl(src, filter=AnnotationFilter(
  ~symbol %startsWith% "SNORD" & tx_start < 25070000))

## transcript coordinates with filter in granges format
filters <- GRangesFilter(GenomicRanges::GRanges("chr15:1-25070000"))
transcripts(src, filters)

## promoters
promoters(src, upstream=100, downstream=50,
  filter = SymbolFilter("ADA"))

## transcriptsBy
transcriptsBy(src, by = "exon", filter = SymbolFilter("ADA"))

## exonsBy
exonsBy(src, filter = SymbolFilter("ADA"))

## intronsByTranscript
intronsByTranscript(src, filter = SymbolFilter("ADA"))
```

---

hg38light

*Utilities used in examples, vignettes, and tests*


---

**Description**

These functions are primarily for illustrating functionality. `hg38light()` and `mm10light()` provide access to trimmed-down versions of `Organism.dplyr` data based derived from the `TxDb.Hsapiens.UCSC.hg38.knownGene` and `TxDb.Mmusculus.UCSC.mm10.ensGene` data bases.

**Usage**

```
hg38light()
```

```
mm10light()
```

**Value**

character(1) file path to the trimmed-down data base

## Examples

```
hg38light()
mm10light()
```

---

```
keytypes,src_organism-method
```

*Using the "select" interface on src\_organism objects*

---

## Description

select, columns and keys can be used together to extract data from a [src\\_organism](#) object.

## Usage

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

## S4 method for signature 'src_organism'
columns(x)

## S4 method for signature 'src_organism'
keys(x, keytype, ...)

select_tbl(x, keys, columns, keytype)

## S4 method for signature 'src_organism'
select(x, keys, columns, keytype)

## S4 method for signature 'src_organism'
mapIds(x, keys, column, keytype, ..., multiVals)
```

## Arguments

x	a <code>src_organism</code> object
keytype	specifies the kind of keys that will be returned. By default keys will return the keys for schema of the <code>src_organism</code> object.
...	other arguments. These include: pattern: the pattern to match. column: the column to search on. fuzzy: TRUE or FALSE value. Use fuzzy matching? (this is used with pattern)
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.

<code>column</code>	character(1) the column to search on, can only have a single element for the value
<code>multiVals</code>	<p>What should <code>mapIds</code> do when there are multiple values that could be returned. Options include:</p> <p><code>first</code>: when there are multiple matches only the 1st thing that comes back will be returned. This is the default behavior.</p> <p><code>list</code>: return a list object to the end user</p> <p><code>filter</code>: remove all elements that contain multiple matches and will therefore return a shorter vector than what came in whenever some of the keys match more than one value</p> <p><code>asNA</code>: return an NA value whenever there are multiple matches</p> <p><code>CharacterList</code>: returns a <code>SimpleCharacterList</code> object</p> <p><code>FUN</code>: can also supply a function to the <code>multiVals</code> argument for custom behaviors. The function must take a single argument and return a single value. This function will be applied to all the elements and will serve a 'rule' that for which thing to keep when there is more than one element. So for example this example function will always grab the last element in each result: <code>last &lt;- function(x){x[[length(x)]]}</code></p>

### Details

`keytypes()`: discover which keytypes can be passed to `keytype` argument of methods `select` or `keys`.

`keys()`: returns keys for the `src_organism` object. By default it returns the primary keys for the database, and returns the keys from that keytype when the `keytype` argument is used.

`columns()`: discover which kinds of data can be returned for the `src_organism` object.

`select()`: retrieves the data as a `tibble` based on parameters for selected keys columns and keytype arguments. If requested columns that have multiple matches for the keys, '`select()`' will return a `tibble` with one row for each possible match.

`mapIds()`: gets the mapped ids (column) for a set of keys that are of a particular keytype. Usually returned as a named character vector.

### Value

`keys`, `columns` and `keytypes` each returns a character vector of possible values. `select` returns a `tibble`.

### Author(s)

Yubo Cheng.

### See Also

[AnnotationDb-class](#) for more description of methods `select`, `keytypes`, `keys` and `columns`.

[src\\_organism](#) for creating a `src_organism` object.

[transcripts\\_tbl](#) for generic functions to extract genomic features from a `src_organism` object.



**Examples**

```
## Not run: src <- src_organism("TxDb.Hsapiens.UCSC.hg38.knownGene")
src <- src_organism(dbpath=hg38light())

## keytypes
keytypes(src)

## columns
columns(src)

## keys
keys(src, "entrez")

keytype <- "symbol"
keys <- c("ADA", "NAT2")
columns <- c("entrez", "tx_id", "tx_name", "exon_id")

## select
select_tbl(src, keys, columns, keytype)
select(src, keys, columns, keytype)

## mapIds
mapIds(src, keys, column = "tx_name", keytype)
```

---

src_organism	<i>Create a sqlite database from TxDb and corresponding Org packages</i>
--------------	--

---

**Description**

The database provides a convenient way to map between gene, transcript, and protein identifiers. `'select_tbl_organism()'` is DEPRECATED, please use `'select()'`.

**Usage**

```
src_organism(txdb = NULL, dbpath = NULL, overwrite = FALSE)

src_ucsc(organism, genome = NULL, id = NULL, dbpath = NULL, verbose = TRUE)

supportedOrganisms()

## S3 method for class 'tbl_organism'
select_(.data, ...)

## S3 method for class 'src_organism'
src_tbls(x, ...)

## S3 method for class 'src_organism'
```

```
tbl(src, from, ...)

## S4 method for signature 'src_organism'
orgPackageName(x)

## S4 method for signature 'src_organism'
seqinfo(x)
```

### Arguments

txdb	character(1) naming a TxDb.* package (e.g., TxDb.Hsapiens.UCSC.hg38.knownGene) or a TxDb object instantiating the content of a TxDb.* package.
dbpath	character(1) path or BiocFileCache instance representing the location where an Organism.dplyr SQLite database will be accessed or created. If no path is specified, the SQLite file is created in the default BiocFileCache() location.
overwrite	logical(1) overwrite an existing 'dbpath' contains an Organism.dplyr SQLite database different from the version implied by 'txdb'?
organism	organism or common name
genome	genome name
id	choose from "knownGene", "ensGene" and "refGene"
verbose	logical. Should R report extra information on progress? Default is TRUE.
.data	A tbl.
...	Comma separated list of unquoted expressions. You can treat variable names like they are positions. Use positive values to select variables; use negative values to drop variables.
x	A src_organism object
src	An src_organism object
from	character(1) name of temporary table in 'src'.

### Details

src\_organism() and src\_ucsc() are meant to be a building block for [src\\_organism](#), which provides an integrated presentation of identifiers and genomic coordinates.

src\_organism() creates a dplyr database integrating org.\* and TxDb.\* information by given TxDb. And src\_ucsc() creates the database by given organism name, genome and/or id.

supportedOrganisms() provides all supported organisms in this package with corresponding OrgDb and TxDb.

The 'tbl.src\_organism()' parameter 'load\_tbl\_only' has been removed. The function behaves as 'load\_tbl\_only = FALSE' (the previous default); for 'load\_tbl\_only = TRUE', use 'tbl(src\$con, ...)'

### Value

src\_organism() and src\_ucsc() returns a dplyr src\_dbi instance representing the data tables. A tibble of the requested table coming from the temporary database of the src\_organism object.

**Author(s)**

Yubo Cheng.

**See Also**

[dplyr](#) for details about using dplyr to manipulate data.

[transcripts\\_tbl](#) for generic functions to extract genomic features from a src\_organism object.

[select,src\\_organism-method](#) for "select" interface on src\_organism objects.

**Examples**

```
## create human sqlite database with TxDb.Hsapiens.UCSC.hg38.knownGene and
## corresponding org.Hs.eg.db
## Not run: src <- src_organism("TxDb.Hsapiens.UCSC.hg38.knownGene")
src <- src_organism(dbpath=hg38light())

## query using dplyr
inner_join(tbl(src, "id"), tbl(src, "id_go")) %>%
  filter(symbol == "ADA") %>%
  dplyr::select(entrez, ensembl, symbol, go, evidence, ontology)

## create human sqlite database using hg38 genome
## Not run: human <- src_ucsc("human")

## all supported organisms with corresponding OrgDb and TxDb
supportedOrganisms()

## Look at all available tables
src_tbls(src)

## Look at data in table "id"
tbl(src, "id")

## Look at fields of one table
colnames(tbl(src, "id"))

## name of org package of src_organism object
orgPackageName(src)

## seqinfo of src_organism object
seqinfo(src)
```

# Index

AccnumFilter (BasicFilter-class), 2  
AliasFilter (BasicFilter-class), 2  
BasicFilter-class, 2  
cds (Genomic-Extractors), 4  
cds\_tbl (Genomic-Extractors), 4  
cdsBy (Genomic-Extractors), 4  
cdsBy\_tbl (Genomic-Extractors), 4  
CdsChromFilter (BasicFilter-class), 2  
CdsIdFilter (BasicFilter-class), 2  
CdsNameFilter (BasicFilter-class), 2  
CdsStrandFilter (BasicFilter-class), 2  
CharacterFilter-class  
    (BasicFilter-class), 2  
columns,src\_organism-method  
    (keytypes,src\_organism-method),  
    7  
dplyr, 11  
EnsemblFilter (BasicFilter-class), 2  
EnsemblprotFilter (BasicFilter-class), 2  
EnsembltransFilter (BasicFilter-class),  
    2  
EnzymeFilter (BasicFilter-class), 2  
EvidenceallFilter (BasicFilter-class), 2  
EvidenceFilter (BasicFilter-class), 2  
ExonChromFilter (BasicFilter-class), 2  
exons (Genomic-Extractors), 4  
exons\_tbl (Genomic-Extractors), 4  
exonsBy (Genomic-Extractors), 4  
exonsBy\_tbl (Genomic-Extractors), 4  
ExonStrandFilter (BasicFilter-class), 2  
fiveUTRsByTranscript  
    (Genomic-Extractors), 4  
fiveUTRsByTranscript\_tbl  
    (Genomic-Extractors), 4  
FlybaseCgFilter (BasicFilter-class), 2  
FlybaseFilter (BasicFilter-class), 2  
FlybaseProtFilter (BasicFilter-class), 2  
GeneChromFilter (BasicFilter-class), 2  
genes (Genomic-Extractors), 4  
genes\_tbl (Genomic-Extractors), 4  
GeneStrandFilter (BasicFilter-class), 2  
Genomic-Extractors, 4  
GoallFilter (BasicFilter-class), 2  
GoFilter (BasicFilter-class), 2  
GRanges, 4, 5  
GRangesList, 4, 5  
hg38light, 6  
IntegerFilter-class  
    (BasicFilter-class), 2  
intronsByTranscript  
    (Genomic-Extractors), 4  
intronsByTranscript,src\_organism-method  
    (Genomic-Extractors), 4  
intronsByTranscript\_tbl  
    (Genomic-Extractors), 4  
IpiFilter (BasicFilter-class), 2  
keys,src\_organism-method  
    (keytypes,src\_organism-method),  
    7  
keytypes,src\_organism-method, 7  
MapFilter (BasicFilter-class), 2  
mapIds,src\_organism-method  
    (keytypes,src\_organism-method),  
    7  
MgiFilter (BasicFilter-class), 2  
mm10light (hg38light), 6  
OmicFilter (BasicFilter-class), 2  
OntologyallFilter (BasicFilter-class), 2  
OntologyFilter (BasicFilter-class), 2  
orgPackageName,src\_organism-method  
    (src\_organism), 9

PfamFilter (BasicFilter-class), 2  
PmidFilter (BasicFilter-class), 2  
promoters (Genomic-Extractors), 4  
promoters,src\_organism-method  
    (Genomic-Extractors), 4  
promoters\_tbl (Genomic-Extractors), 4  
PrositeFilter (BasicFilter-class), 2  
  
RefseqFilter (BasicFilter-class), 2  
  
select,src\_organism-method  
    (keytypes,src\_organism-method),  
    7  
select\_.tbl\_organism (src\_organism), 9  
select\_tbl  
    (keytypes,src\_organism-method),  
    7  
seqinfo,src\_organism-method  
    (src\_organism), 9  
show,BasicFilter-method  
    (BasicFilter-class), 2  
show,CharacterFilter-method  
    (BasicFilter-class), 2  
show,IntegerFilter-method  
    (BasicFilter-class), 2  
src\_organism, 4, 5, 7, 8, 9, 10  
src\_tbls.src\_organism (src\_organism), 9  
src\_ucsc (src\_organism), 9  
supportedFilters,src\_organism-method  
    (BasicFilter-class), 2  
supportedOrganisms (src\_organism), 9  
  
tbl.src\_organism (src\_organism), 9  
threeUTRsByTranscript  
    (Genomic-Extractors), 4  
threeUTRsByTranscript\_tbl  
    (Genomic-Extractors), 4  
tibble, 4, 5  
transcripts, 4  
transcripts (Genomic-Extractors), 4  
transcripts\_tbl, 4, 8, 11  
transcripts\_tbl (Genomic-Extractors), 4  
transcriptsBy (Genomic-Extractors), 4  
transcriptsBy\_tbl (Genomic-Extractors),  
    4  
TxChromFilter (BasicFilter-class), 2  
TxStrandFilter (BasicFilter-class), 2  
TxTypeFilter (BasicFilter-class), 2  
  
WormbaseFilter (BasicFilter-class), 2  
  
ZfinFilter (BasicFilter-class), 2