

Package ‘AnnotationFilter’

October 17, 2017

Title Facilities for Filtering Bioconductor Annotation Resources

Version 1.0.0

URL <https://github.com/Bioconductor/AnnotationFilter>

BugReports <https://github.com/Bioconductor/AnnotationFilter/issues>

Description This package provides class and other infrastructure to implement filters for manipulating Bioconductor annotation resources. The filters will be used by `ensemblDb`, `Organism.dplyr`, and other packages.

Depends R (>= 3.4.0)

Imports `utils`, `methods`, `GenomicRanges`, `lazyeval`

Suggests `BiocStyle`, `knitr`, `testthat`, `RSQLite`, `org.Hs.eg.db`

VignetteBuilder `knitr`

License Artistic-2.0

biocViews Annotation, Infrastructure, Software

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Collate 'AllGenerics.R' 'AnnotationFilter.R' 'AnnotationFilterList.R'
'translate-utils.R'

NeedsCompilation no

Author Martin Morgan [aut],
Johannes Rainer [aut],
Bioconductor Maintainer [cre]

Maintainer Bioconductor Maintainer <maintainer@bioconductor.org>

R topics documented:

AnnotationFilter	2
AnnotationFilterList	4
Index	6

Description

The filters extending the base `AnnotationFilter` class represent a simple filtering concept for annotation resources. Each filter object is thought to filter on a single (database) table column using the provided values and the defined condition.

Filter instances created using the constructor functions (e.g. `GeneIdFilter`).

`supportedFilters()` lists all defined filters. Packages using `AnnotationFilter` should implement the `supportedFilters` for their annotation resource object (e.g. for `object = "Ensembl"` in the `ensembl` package) to list all supported filters for the specific resource.

`condition()` get the condition value for the filter object.

`value()` get the value for the filter object.

`field()` get the field for the filter object.

`feature()` get the feature for the `GRangesFilter` object.

`AnnotationFilter` *translates* a filter expression such as `~ gene_id == "BCL2"` into a filter object extending the `AnnotationFilter` class (in the example a `GeneIdFilter` object) or an `AnnotationFilterList` if the expression contains multiple conditions (see examples below).

Usage

```
CdsStartFilter(value, condition = "==")
CdsEndFilter(value, condition = "==")
ExonIdFilter(value, condition = "==")
ExonNameFilter(value, condition = "==")
ExonRankFilter(value, condition = "==")
ExonStartFilter(value, condition = "==")
ExonEndFilter(value, condition = "==")
GeneIdFilter(value, condition = "==")
GenenameFilter(value, condition = "==")
GeneBiotypeFilter(value, condition = "==")
GeneStartFilter(value, condition = "==")
GeneEndFilter(value, condition = "==")
EntrezFilter(value, condition = "==")
SymbolFilter(value, condition = "==")
TxIdFilter(value, condition = "==")
TxNameFilter(value, condition = "==")
TxBiotypeFilter(value, condition = "==")
TxStartFilter(value, condition = "==")
TxEndFilter(value, condition = "==")
ProteinIdFilter(value, condition = "==")
UniprotFilter(value, condition = "==")
SeqNameFilter(value, condition = "==")
SeqStrandFilter(value, condition = "==")

## S4 method for signature 'AnnotationFilter'
condition(object)
```

```

## S4 method for signature 'AnnotationFilter'
value(object)

## S4 method for signature 'AnnotationFilter'
field(object)

GRangesFilter(value, feature = "gene", type = c("any", "start", "end",
  "within", "equal"))

feature(object)

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

AnnotationFilter(expr)

```

Arguments

object	An AnnotationFilter object.
value	character(), integer(), or GRanges() value for the filter
feature	character(1) defining on what feature the GRangesFilter should be applied. Choices could be "gene", "tx" or "exon".
type	character(1) indicating how overlaps are to be filtered. See findOverlaps in the IRanges package for a description of this argument.
expr	A filter expression, written as a formula, to be converted to an AnnotationFilter or AnnotationFilterList class. See below for examples.
condition	character(1) defining the condition to be used in the filter. For IntegerFilter, one of "=", "!=", ">", "<", ">=" or "<=". For CharacterFilter, one of "=", "!=", "startsWith" or "endsWith". Default condition is "=".

Details

By default filters are only available for tables containing the field on which the filter acts (i.e. that contain a column with the name matching the value of the field slot of the object). See the vignette for a description to use filters for databases in which the database table column name differs from the default field of the filter.

Filter expressions for the AnnotationFilter class have to be written as formulas, i.e. starting with a ~.

Value

The constructor function return an object extending AnnotationFilter. For the return value of the other methods see the methods' descriptions.

AnnotationFilter returns an [AnnotationFilter](#) or an [AnnotationFilterList](#).

Note

Translation of nested filter expressions using the AnnotationFilter function is not yet supported.

See Also

[AnnotationFilterList](#) for combining AnnotationFilter objects.

Examples

```
## filter by GRanges
GRangesFilter(GenomicRanges::GRanges("chr10:87869000-87876000"))
## Create a SymbolFilter to filter on a gene's symbol.
sf <- SymbolFilter("BCL2")
sf

## Create a GeneStartFilter to filter based on the genes' chromosomal start
## coordinates
gsf <- GeneStartFilter(10000, condition = ">")
gsf

supportedFilters()

## Convert a filter expression based on a gene ID to a GeneIdFilter
gnf <- AnnotationFilter(~ gene_id == "BCL2")
gnf

## Same conversion but for two gene IDs.
gnf <- AnnotationFilter(~ gene_id %in% c("BCL2", "BCL2L11"))
gnf

## Converting an expression that combines multiple filters. As a result we
## get an AnnotationFilterList containing the corresponding filters.
## Be aware that nesting of expressions/filters does not work.
flt <- AnnotationFilter(~ gene_id %in% c("BCL2", "BCL2L11") &
                        tx_biotype == "nonsense_mediated_decay" |
                        seq_name == "Y")

flt
```

AnnotationFilterList *Combining annotation filters*

Description

The AnnotationFilterList allows to combine filter objects extending the [AnnotationFilter](#) class to construct more complex queries. Consecutive filter objects in the AnnotationFilterList can be combined by a logical *and* (&) or *or* (|). The AnnotationFilterList extends list, individual elements can thus be accessed with `[[`.

`value()` get a list with the AnnotationFilter objects. Use `[[` to access individual filters.

Usage

```
AnnotationFilterList(..., logOp = character())

## S4 method for signature 'AnnotationFilterList'
value(object)
```


Index

- .GRangesFilter (AnnotationFilter), 2
- AnnotationFilter, 2, 2, 3–5
- AnnotationFilter-class (AnnotationFilter), 2
- AnnotationFilterList, 2–4, 4
- AnnotationFilterList-class (AnnotationFilterList), 4
- CdsEndFilter (AnnotationFilter), 2
- CdsEndFilter-class (AnnotationFilter), 2
- CdsStartFilter (AnnotationFilter), 2
- CdsStartFilter-class (AnnotationFilter), 2
- CharacterFilter-class (AnnotationFilter), 2
- condition (AnnotationFilter), 2
- condition, AnnotationFilter-method (AnnotationFilter), 2
- EntrezFilter (AnnotationFilter), 2
- EntrezFilter-class (AnnotationFilter), 2
- ExonEndFilter (AnnotationFilter), 2
- ExonEndFilter-class (AnnotationFilter), 2
- ExonIdFilter (AnnotationFilter), 2
- ExonIdFilter-class (AnnotationFilter), 2
- ExonNameFilter (AnnotationFilter), 2
- ExonNameFilter-class (AnnotationFilter), 2
- ExonRankFilter (AnnotationFilter), 2
- ExonRankFilter-class (AnnotationFilter), 2
- ExonStartFilter (AnnotationFilter), 2
- ExonStartFilter-class (AnnotationFilter), 2
- feature (AnnotationFilter), 2
- field (AnnotationFilter), 2
- field, AnnotationFilter-method (AnnotationFilter), 2
- GeneBiotypeFilter (AnnotationFilter), 2
- GeneBiotypeFilter-class (AnnotationFilter), 2
- GeneEndFilter (AnnotationFilter), 2
- GeneEndFilter-class (AnnotationFilter), 2
- GeneIdFilter, 2
- GeneIdFilter (AnnotationFilter), 2
- GeneIdFilter-class (AnnotationFilter), 2
- GenenameFilter (AnnotationFilter), 2
- GenenameFilter-class (AnnotationFilter), 2
- GeneStartFilter (AnnotationFilter), 2
- GeneStartFilter-class (AnnotationFilter), 2
- GRangesFilter (AnnotationFilter), 2
- GRangesFilter-class (AnnotationFilter), 2
- IntegerFilter-class (AnnotationFilter), 2
- ProteinIdFilter (AnnotationFilter), 2
- ProteinIdFilter-class (AnnotationFilter), 2
- SeqNameFilter (AnnotationFilter), 2
- SeqNameFilter-class (AnnotationFilter), 2
- SeqStrandFilter (AnnotationFilter), 2
- SeqStrandFilter-class (AnnotationFilter), 2
- show, AnnotationFilter-method (AnnotationFilter), 2
- show, AnnotationFilterList-method (AnnotationFilterList), 4
- show, CharacterFilter-method (AnnotationFilter), 2
- show, GRangesFilter-method (AnnotationFilter), 2
- show, IntegerFilter-method (AnnotationFilter), 2
- supportedFilters, 5
- supportedFilters (AnnotationFilter), 2
- supportedFilters, missing-method (AnnotationFilter), 2
- SymbolFilter (AnnotationFilter), 2

SymbolFilter-class (AnnotationFilter), 2

TxBiotypeFilter (AnnotationFilter), 2

TxBiotypeFilter-class
 (AnnotationFilter), 2

TxEndFilter (AnnotationFilter), 2

TxEndFilter-class (AnnotationFilter), 2

TxIdFilter (AnnotationFilter), 2

TxIdFilter-class (AnnotationFilter), 2

TxNameFilter (AnnotationFilter), 2

TxNameFilter-class (AnnotationFilter), 2

TxStartFilter (AnnotationFilter), 2

TxStartFilter-class (AnnotationFilter),
 2

UniprotFilter (AnnotationFilter), 2

UniprotFilter-class (AnnotationFilter),
 2

value (AnnotationFilter), 2

value, AnnotationFilter-method
 (AnnotationFilter), 2

value, AnnotationFilterList-method
 (AnnotationFilterList), 4