

Package ‘biofiles’

November 21, 2017

Type Package

Title An Interface for GenBank/GenPept Flat Files

Version 1.0.0

Author Gerhard Schöfl [aut, cre]

Maintainer Gerhard Schöfl <gerhard.schofl@gmail.com>

Description

Parse GenBank/GenPept records <<https://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html>> into native R objects to easily access and manipulate the sequence and annotation information.

Depends R (>= 3.3.1), Rcpp (>= 0.11.0)

LinkingTo Rcpp (>= 0.11.0)

Imports assertthat, BiocGenerics, Biostrings, foreach, GenomeInfoDb, GenomicRanges, IRanges, iterators, methods, parallel, RCurl, reutils, stats, S4Vectors, XVector

Suggests doParallel, testthat, BiocStyle, knitr, covr

License MIT + file LICENSE

Encoding UTF-8

URL <https://github.com/gschofl/biofiles>

BugReports <https://github.com/gschofl/biofiles/issues>

Collate 'RcppExports.R' 'utils.R' 'all-generics.R'
'biofiles-package.R' 'filter.R' 'flanking.R' 'gbHeader-class.R'
'gbLocation-class.R' 'gbFeature-class.R'
'gbFeatureTable-class.R' 'gbRecord-class.R'
'gbRecordList-class.R' 'genome-record-from-ncbi.R'
'make-ranges.R' 'misc-functions.R' 'parser-general.R'
'parser-embl.R' 'parser-gbk.R' 'select.R' 'shift.R'
'write-feature-table.R' 'write-genbank.R' 'zzz.R'

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-11-21 18:16:48 UTC

R topics documented:

as.gbLocation	3
biofiles	3
dbxref	4
end	5
featureTable	6
filter	7
fuzzy	8
gbFeature-class	9
gbFeatureTable-class	10
gbRecord	10
gbRecord-class	12
gbRecordList-class	13
geneID	13
getFeatures	14
getHeader	15
getLocus	16
getSequence	23
hasKey	24
hasQualif	25
index	26
key	26
length.gbRecord-method	27
location	28
locusTag	29
note	29
product	30
proteinID	30
qualif	31
qualifList	32
qualifTable	33
ranges	34
revcomp	35
saveRecord	36
shift	37
span	38
start	39
strand	40
summary	42
translation	43
uniqueQualifs	43
upstream	44
write.FeatureTable	45
write.GenBank	46
[[.gbFeature,character,missing-method	47

as.gbLocation	Create a gbLocation.
---------------	----------------------

Description

Create a gbLocation object out of a character string.

Usage

```
as.gbLocation(base_span)
```

Arguments

base_span A character string representation of GenBank feature location

Value

A `gbLocation` object.

Examples

```
as.gbLocation("join(1..10,12..20)")
```

biofiles	<i>biofiles is an R package for interfacing with GenBank or Embl flat file records.</i>
----------	---

Description

This is an R package for interfacing with GenBank, GenPept, Embl and IMGT/HLA flat file records. It includes utilities for reading and writing GenBank files, and methods for interacting with annotation and sequence data.

Author(s)

Gerhard Schöfl <gerhard.schofl@gmail.com>

References

For a sample GenBank record see <https://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html>

For a detailed description of the GenBank feature table format see <https://www.ncbi.nlm.nih.gov/collab/FT>

For a description of the EMBL flat file format see <ftp://ftp.ebi.ac.uk/pub/databases/embl/doc/usrman.txt>.

For a description of the format and conventions of IMGT/HLA flat files see <https://www.ebi.ac.uk/ipd/imgt/hla/docs/manual.html>.

dbxref *Access the db_xrefs of GenBank features*

Description

Access the db_xrefs of GenBank features

Usage

```
dbxref(x, db = NULL, ...)  
  
## S4 method for signature 'gbFeature'  
dbxref(x, db = NULL, ...)  
  
## S4 method for signature 'gbFeatureTable'  
dbxref(x, db = NULL, ...)  
  
## S4 method for signature 'gbRecord'  
dbxref(x, db = NULL, ...)
```

Arguments

x	A gbFeature , gbFeatureTable , or gbRecord object.
db	(Optional) A character vector giving the database names of the desired db_xrefs.
...	Additional arguments passed to methods.

Value

A named character vector (or list of named character vectors) of db_xrefs.

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))  
  
## all db_xrefs associated with CDSs  
dbxref(x["CDS"])  
  
## retrieve the TaxId from the "source" field.  
dbxref(x[[1]], "taxon")
```

end *Get or set the end position of genomic features*

Description

Get or set the end position of genomic features

Usage

```
end(x, ...)  
  
end(x, ...) <- value  
  
## S4 method for signature 'gbLocation'  
end(x, join = FALSE)  
  
## S4 replacement method for signature 'gbLocation'  
end(x, ...) <- value  
  
## S4 method for signature 'gbFeature'  
end(x, join = FALSE)  
  
## S4 replacement method for signature 'gbFeature'  
end(x, ...) <- value  
  
## S4 method for signature 'gbFeatureTable'  
end(x, join = FALSE)  
  
## S4 replacement method for signature 'gbFeatureTable'  
end(x, ...) <- value  
  
## S4 method for signature 'gbRecord'  
end(x, join = FALSE)  
  
## S4 method for signature 'gbRecordList'  
end(x, join = FALSE)
```

Arguments

x	A gbFeature , gbFeatureTable , gbRecord , or gbRecordList object.
...	Further arguments passed to methods.
value	The end information to set on x.
join	Join compound genomic locations into a single range.

Value

An integer vector or a list of integer vectors.

See Also

[start](#), [strand](#), [span](#), [ranges](#)

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))

## end
end(x)
cds <- x["CDS"]
end(cds)

## `end<-`
end(cds) <- 1000
end(cds)
ranges(cds)
```

featureTable

Tabulate Genbank features

Description

Extract a frequency table (or list of tables in the case of `gbRecordLists`) of feature keys.

Usage

```
featureTable(x, ...)
```

S4 method for signature 'gbFeatureTable'
featureTable(x)

S4 method for signature 'gbRecord'
featureTable(x)

S4 method for signature 'gbRecordList'
featureTable(x)

Arguments

x A [gbFeatureTable](#), [gbRecord](#), or [gbRecordList](#) object.

... Additional arguments to be passed to or from methods.

Value

A [table](#) (or list of tables) of feature keys.

See Also[qualifTable](#)**Examples**

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))
featureTable(x)
```

filter	<i>Return a subset of features or annotations from a GenBank Record</i>
--------	---

Description

filter returns a subset of features from [gbRecordList](#), [gbRecord](#) or [gbFeatureTable](#) objects, based on filters provided as *key*, *range*, or *qualifier* values.

select returns a specified subset of annotations from GenBank Features as a `data.frame`.

Usage

```
filter(x, ...)

select(x, ...)

## S4 method for signature 'gbFeatureTable'
filter(x, ..., .cols = NULL)

## S4 method for signature 'gbFeatureTable'
select(x, ..., .cols = NULL)

## S4 method for signature 'gbRecord'
filter(x, ..., .cols = NULL)

## S4 method for signature 'gbRecord'
select(x, ..., .cols = NULL)

## S4 method for signature 'gbRecordList'
filter(x, ..., .cols = NULL)

## S4 method for signature 'gbRecordList'
select(x, ..., .cols = NULL)
```

Arguments

x	A 'gbRecord' or 'gbFeatureTable' instance.
...	For filter: named values that specify the features to select. These are merged with the values of keys to create the actual query. See Details; for select: see <code>.cols</code> .

`.cols` A character vector of ‘keys’ that specify annotations to be returned as a data.frame from the filtered features. If NULL, a ‘gbFeatureTable’ is returned. Supported ‘keys’ are “index” or “idx”, “start”, tag (e.g., “locus_tag”, “product”, “db_xref”). Specific db_xrefs can be queried using, e.g. “db_xref.GI” or “db_xref.GeneID”.

Details

Filters are provided as named values using keywords and/or “qualifier = value” pairs:

Permissible keywords are:

index/idx For example: `idx = c(3,4,5,6)`, `idx = 100:150`, `index = c(1,12:20)`

range For example: `range = "10000..25000"`, `range = "..10000,20000..25000"`, `range = "30000.."`

key For example: `key = "CDS"`, `key = c("CDS", "gene")`

arbitrary qualifiers For example: `product = "ribosomal"`, `locus_tag = c("CPSIT_0123", "CPSIT_0124", "CPSIT_0125")`, `pseudo = TRUE`

Value

Depending on the value of `.col` a `gbRecordList`, `gbRecord`, `orgbFeatureTable` or a `data.frame`.

Examples

```
load(system.file("extdata", "S_cerevisiae_mito.rda", package = "biofiles"))

## filter all hydrophobic tRNAs from the yeast mitochondrion
hydrophobic <- c("Val", "Ile", "Leu", "Met", "Phe", "Trp", "Cys")
trna <- filter(x, key = "tRNA", product = hydrophobic)

## select start, end, orientation, product, and GeneID
df <- select(trna, "start", "end", "strand", "product", "db_xref.GeneID")
df

## combine the above steps into one
cols <- c("start", "end", "strand", "product", "db_xref.GeneID")
filter(x, key = "tRNA", product = hydrophobic, .cols = cols)

## filter all CDS from position 60,000 bp onward
filter(x, key = "CDS", range = "60000..")
```

fuzzy

Has a feature fuzzy locations?

Description

With a GenBank location like *complement(<123..150)* we don’t know the exact start position of the feature. Use fuzzy to test for fuzzy locations.

Usage

```
fuzzy(x, ...)
```

```
## S4 method for signature 'gbLocation'
```

```
fuzzy(x)
```

```
## S4 method for signature 'gbFeature'
```

```
fuzzy(x)
```

```
## S4 method for signature 'gbFeatureTable'
```

```
fuzzy(x)
```

```
## S4 method for signature 'gbRecord'
```

```
fuzzy(x)
```

```
## S4 method for signature 'gbRecordList'
```

```
fuzzy(x)
```

Arguments

x A [gbFeature](#), [gbFeatureTable](#), [gbRecord](#), or [gbRecordList](#) object.

... Further arguments passed to methods.

Value

A logical matrix.

Examples

```
l <- as.gbLocation("complement(<123..150)")
```

```
fuzzy(l)
```

```
## note that start() or end() return exact positions even if they are fuzzy.
```

```
start(l)
```

gbFeature-class	<i>Class "gbFeature"</i>
-----------------	--------------------------

Description

“gbFeature” is an S4 class that provides a container for GenBank feature tables.

Slots

- .seqinfo An [seqinfo](#) object containing the full-length sequence of the GenBank record that the feature is part of as an [XStringSet](#) object, and sequence metadata as a [gbHeader](#) object.
- .id Identifier (index) of the feature in the GenBank record the feature is part of.
- key The feature key.
- location A [gbLocation](#) object.
- qualifiers A named character vector. Name attributes correspond to GenBank qualifier tags.

Accessor functions

[getHeader](#), [getSequence](#), [ranges](#), [key](#), [index](#), [qualif](#)

See Also

[gbFeatureTable](#), [gbRecord](#)

gbFeatureTable-class *Class "gbFeatureTable"*

Description

“gbFeatureTable” is an S4 class that provides a container for “[gbFeature](#)”s retrieved from GenBank flat files.

Slots

- .seqinfo A [seqinfo](#) object containing the genome sequence as an [XStringSet](#) object and sequence metadata as a [gbHeader](#) object.
- .id An integer vector the indices of the [gbFeatures](#) contained within a [gbFeatureTable](#) object.
- .Data A list of [gbFeature](#) objects.

gbRecord *Read a GenBank/GenPept or Embl format file.*

Description

Import data from GenBank/GenPept, Embl, or IMGT/HLA flat files into R, represented as an instance of the [gbRecord](#) or [gbRecordList](#) classes.

Usage

```
gbRecord(rcd, progress = FALSE)
```

Arguments

rcd	A vector of paths to GenBank/Embl format records, an <code>efetch</code> object containing GenBank record(s), or a <code>textConnection</code> to a character vector that can be parsed as a Genbank or Embl record.
progress	Print a nice progress bar if parsing multiple Genbank records. (This will not work if you process the records in parallel.)

Details

For a sample GenBank record see <https://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html>, for a detailed description of the GenBank feature table format see <https://www.ncbi.nlm.nih.gov/collab/FT/>.

For a description of the EMBL flat file format see <ftp://ftp.ebi.ac.uk/pub/databases/embl/doc/usrman.txt>.

For a description of the format and conventions of IMGT/HLA flat files see <https://www.ebi.ac.uk/ipd/imgt/hla/docs/manual.html>.

Value

An instance of the `gbRecord` or `gbRecordList` classes.

Note

The `gbRecord` class is modelled after the Genbank flat file format. Both Embl and IMGT/HLA files do not fit this model perfectly, so some pretty arbitrary choices were made to make the data from these files fitr the model.

See Also

[genomeRecordFromNCBI](#)

Examples

```
## Not run:
### import from file
gbk_file <- system.file("extdata", "marine_metagenome.gb", package = "biofiles")
x <- gbRecord(gbk_file)

## End(Not run)

load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))
getHeader(x)
getFeatures(x)

### quickly extract features as GRanges
ranges(x["CDS"], include = c("product", "note", "protein_id"))

## Directly subset features
x[[1]]
```

```

### import directly from NCBI
## Not run:
x <- gbRecord(reutils::efetch("139189709", "protein", rettype = "gp", retmode = "text"))
x

## End(Not run)

## import a file containing multiple GenBank records as a
## gbRecordList. With many short records it pays of to
## run the parsing in parallel
## Not run:
gss_file <- system.file("extdata", "gss.gb", package = "biofiles")
library(doParallel)
registerDoParallel(cores = 4)
gss <- gbRecord(gss_file)
gss

## End(Not run)

```

gbRecord-class	Class "gbRecord"
----------------	------------------

Description

“gbRecord” is an S4 class that provides a container for data parsed from GenBank, GenPept, Embl or IMGT records. For instantiation of a gbRecord object use the import function [gbRecord](#).

Slots

seqinfo A “[seqinfo](#)” instance; This is a reference class holding the sequence as an “[XStringSet](#)” instance and header of the file containing metadata as a “[gbHeader](#)” object.

features A “[gbFeatureTable](#)” instance.

contig If present, a CONTIG record.

Accessor functions

[header](#), [ft](#), [ranges](#)

See Also

The constructor, [gbRecord](#)

gbRecordList-class	Class "gbRecordList"
--------------------	----------------------

Description

“gbRecordList” is an S4 class that provides a container for “gbRecord”s retrieved from GenBank flat files. For instantiation of a gbRecordList object use the import function [gbRecord](#) or combine gbRecord objects using gbRecordList.

Usage

```
gbRecordList(...)
```

Arguments

... “gbRecord” elements.

Value

A “gbRecordList” instance.

geneID	Return the gene qualifiers from GenBank features.
--------	---

Description

Return the *gene* qualifiers from GenBank features.

Usage

```
geneID(...)
```

Arguments

... A [gbRecord](#), [gbFeatureTable](#), or, [gbFeature](#) instance.

Value

A character vector of *genes*.

getFeatures	<i>Get the feature table from a GenBank record.</i>
-------------	---

Description

Get the feature table from a GenBank record.

Usage

```
getFeatures(x, ...)  
  
ft(x, ...)  
  
## S4 method for signature 'gbRecord'  
getFeatures(x)  
  
## S4 method for signature 'gbRecord'  
ft(x)  
  
## S4 method for signature 'gbRecordList'  
getFeatures(x)  
  
## S4 method for signature 'gbRecordList'  
ft(x)
```

Arguments

x	A gbRecord instance.
...	Additional arguments passed to methods.

Value

The [gbFeatureTable](#) of a Genbank record.

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))  
getFeatures(x)  
ft(x)
```

getHeader	<i>Extract the header from a "gbRecord" object.</i>
-----------	---

Description

Extract the header from a "gbRecord" object.

Usage

```
getHeader(x, ...)  
  
header(x, ...)  
  
## S4 method for signature 'gbFeature'  
header(x)  
  
## S4 method for signature 'gbFeature'  
getHeader(x)  
  
## S4 method for signature 'gbFeatureTable'  
getHeader(x)  
  
## S4 method for signature 'gbFeatureTable'  
header(x)  
  
## S4 method for signature 'gbRecord'  
getHeader(x)  
  
## S4 method for signature 'gbRecord'  
header(x)
```

Arguments

x	A "gbRecord", "gbFeature", or "gbFeatureTable" instance.
...	Additional arguments passed to methods.

Value

A "gbHeader" instance

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))  
getHeader(x)  
header(x)
```

`getLocus`*Access the various fields of a GenBank record.*

Description

Access the various fields of a GenBank record.

Usage`getLocus(x, ...)``getLength(x, ...)``getMoltype(x, ...)``getTopology(x, ...)``getDivision(x, ...)``getDate(x)``getDefinition(x, ...)``getAccession(x, ...)``getVersion(x, ...)``getGeneID(x, ...)``getDBLink(x)``getDBSource(x)``getSource(x)``getOrganism(x)``getTaxonomy(x)``getKeywords(x)``getReference(x)``getComment(x)`

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



```
## S4 method for signature 'seqinfo'  
getLength(x)  
  
## S4 method for signature 'seqinfo'  
getMoltype(x)  
  
## S4 method for signature 'seqinfo'  
getTopology(x)  
  
## S4 method for signature 'seqinfo'  
getDivision(x)  
  
## S4 method for signature 'seqinfo'  
getDate(x)  
  
## S4 method for signature 'seqinfo'  
getDefinition(x)  
  
## S4 method for signature 'seqinfo'  
getAccession(x)  
  
## S4 method for signature 'seqinfo'  
getVersion(x)  
  
## S4 method for signature 'seqinfo'  
getGeneID(x, db = "gi")  
  
## S4 method for signature 'seqinfo'  
getDBLink(x)  
  
## S4 method for signature 'seqinfo'  
getDBSource(x)  
  
## S4 method for signature 'seqinfo'  
getSource(x)  
  
## S4 method for signature 'seqinfo'  
getOrganism(x)  
  
## S4 method for signature 'seqinfo'  
getTaxonomy(x)  
  
## S4 method for signature 'seqinfo'  
getReference(x)  
  
## S4 method for signature 'seqinfo'  
getKeywords(x)
```

```
## S4 method for signature 'seqinfo'  
getComment(x)  
  
## S4 method for signature 'gbLocation'  
getAccession(x)  
  
## S4 method for signature 'gbFeature'  
getLocus(x)  
  
## S4 method for signature 'gbFeature'  
getLength(x)  
  
## S4 method for signature 'gbFeature'  
getMoltype(x)  
  
## S4 method for signature 'gbFeature'  
getTopology(x)  
  
## S4 method for signature 'gbFeature'  
getDivision(x)  
  
## S4 method for signature 'gbFeature'  
getDate(x)  
  
## S4 method for signature 'gbFeature'  
getDefinition(x)  
  
## S4 method for signature 'gbFeature'  
getAccession(x)  
  
## S4 method for signature 'gbFeature'  
getVersion(x)  
  
## S4 method for signature 'gbFeature'  
getGeneID(x, db = "gi")  
  
## S4 method for signature 'gbFeature'  
getDBLink(x)  
  
## S4 method for signature 'gbFeature'  
getDBSource(x)  
  
## S4 method for signature 'gbFeature'  
getSource(x)  
  
## S4 method for signature 'gbFeature'  
getOrganism(x)
```

```
## S4 method for signature 'gbFeature'  
getTaxonomy(x)  
  
## S4 method for signature 'gbFeature'  
getReference(x)  
  
## S4 method for signature 'gbFeature'  
getKeywords(x)  
  
## S4 method for signature 'gbFeature'  
getComment(x)  
  
## S4 method for signature 'gbFeatureTable'  
getLocus(x)  
  
## S4 method for signature 'gbFeatureTable'  
getLength(x)  
  
## S4 method for signature 'gbFeatureTable'  
getMoltype(x)  
  
## S4 method for signature 'gbFeatureTable'  
getTopology(x)  
  
## S4 method for signature 'gbFeatureTable'  
getDivision(x)  
  
## S4 method for signature 'gbFeatureTable'  
getDate(x)  
  
## S4 method for signature 'gbFeatureTable'  
getDefinition(x)  
  
## S4 method for signature 'gbFeatureTable'  
getAccession(x)  
  
## S4 method for signature 'gbFeatureTable'  
getVersion(x)  
  
## S4 method for signature 'gbFeatureTable'  
getGeneID(x, db = "gi")  
  
## S4 method for signature 'gbFeatureTable'  
getDBLink(x)  
  
## S4 method for signature 'gbFeatureTable'  
getDBSource(x)
```

```
## S4 method for signature 'gbFeatureTable'  
getSource(x)  
  
## S4 method for signature 'gbFeatureTable'  
getOrganism(x)  
  
## S4 method for signature 'gbFeatureTable'  
getTaxonomy(x)  
  
## S4 method for signature 'gbFeatureTable'  
getReference(x)  
  
## S4 method for signature 'gbFeatureTable'  
getKeywords(x)  
  
## S4 method for signature 'gbFeatureTable'  
getComment(x)  
  
## S4 method for signature 'gbRecord'  
getLocus(x)  
  
## S4 method for signature 'gbRecord'  
getLength(x)  
  
## S4 method for signature 'gbRecord'  
getMoltype(x)  
  
## S4 method for signature 'gbRecord'  
getTopology(x)  
  
## S4 method for signature 'gbRecord'  
getDivision(x)  
  
## S4 method for signature 'gbRecord'  
getDate(x)  
  
## S4 method for signature 'gbRecord'  
getDefinition(x)  
  
## S4 method for signature 'gbRecord'  
getAccession(x)  
  
## S4 method for signature 'gbRecord'  
getVersion(x)  
  
## S4 method for signature 'gbRecord'  
getGeneID(x, db = "gi")
```

```
## S4 method for signature 'gbRecord'  
getDBLink(x)  
  
## S4 method for signature 'gbRecord'  
getDBSource(x)  
  
## S4 method for signature 'gbRecord'  
getSource(x)  
  
## S4 method for signature 'gbRecord'  
getOrganism(x)  
  
## S4 method for signature 'gbRecord'  
getTaxonomy(x)  
  
## S4 method for signature 'gbRecord'  
getReference(x)  
  
## S4 method for signature 'gbRecord'  
getKeywords(x)  
  
## S4 method for signature 'gbRecord'  
getComment(x)  
  
## S4 method for signature 'gbRecordList'  
getLocus(x)  
  
## S4 method for signature 'gbRecordList'  
getLength(x)  
  
## S4 method for signature 'gbRecordList'  
getMoltype(x)  
  
## S4 method for signature 'gbRecordList'  
getTopology(x)  
  
## S4 method for signature 'gbRecordList'  
getDivision(x)  
  
## S4 method for signature 'gbRecordList'  
getDate(x)  
  
## S4 method for signature 'gbRecordList'  
getDefinition(x)  
  
## S4 method for signature 'gbRecordList'  
getAccession(x)
```

```
## S4 method for signature 'gbRecordList'
getVersion(x)

## S4 method for signature 'gbRecordList'
getGeneID(x, db = "gi")

## S4 method for signature 'gbRecordList'
getDBLink(x)

## S4 method for signature 'gbRecordList'
getDBSource(x)

## S4 method for signature 'gbRecordList'
getSource(x)

## S4 method for signature 'gbRecordList'
getOrganism(x)

## S4 method for signature 'gbRecordList'
getTaxonomy(x)

## S4 method for signature 'gbRecordList'
getReference(x)

## S4 method for signature 'gbRecordList'
getKeywords(x)

## S4 method for signature 'gbRecordList'
getComment(x)
```

Arguments

x	A gbFeature , gbFeatureTable , gbRecord , or gbRecordList object.
...	Further arguments passed to methods.
db	Which database identifier (default: 'gi')

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))

getLocus(x)
getLength(x)
getGeneID(x)
getReference(x)
getDate(x)
```

getSequence	<i>Get the sequence from a GenBank record.</i>
-------------	--

Description

Get the sequence from a GenBank record.

Usage

```
getSequence(x, ...)  
  
## S4 method for signature 'gbFeature'  
getSequence(x)  
  
## S4 method for signature 'gbFeatureTable'  
getSequence(x)  
  
## S4 method for signature 'gbRecord'  
getSequence(x)  
  
## S4 method for signature 'gbRecordList'  
getSequence(x)
```

Arguments

x	A gbFeature , gbFeatureTable , gbRecord , or gbRecordList instance.
...	Additional arguments passed to methods.

Value

An [XStringSet](#) object, containing either the complete sequence(s) of the record(s), or of the selected feature(s)

Examples

```
## Not run:  
gbk_file <- system.file("extdata", "S_cerevisiae_mito.gb", package = "biofiles")  
x <- gbRecord(gbk_file)  
  
## extract the full-length sequence of the record.  
getSequence(x)  
  
## extract coding sequences only  
getSequence(x["CDS"])  
  
## End(Not run)
```

hasKey	<i>Has a feature a specific key?</i>
--------	--------------------------------------

Description

Has a feature a specific key?

Usage

```
hasKey(x, key, ...)  
  
## S4 method for signature 'gbFeature'  
hasKey(x, key)  
  
## S4 method for signature 'gbFeatureTable'  
hasKey(x, key)  
  
## S4 method for signature 'gbRecord'  
hasKey(x, key)  
  
## S4 method for signature 'gbRecordList'  
hasKey(x, key)
```

Arguments

x	A gbFeature , gbFeatureTable , gbRecord , or gbRecordList object.
key	A feature key.
...	Additional arguments to be passed to or from methods.

Value

A logical vector or a list of logical vectors.

See Also

[key](#)

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))  
hasKey(x, 'CDS')
```

hasQualif	<i>Has a feature a specific qualifier?</i>
-----------	--

Description

Has a feature a specific qualifier?

Usage

```
hasQualif(x, qualifier, ...)  
  
## S4 method for signature 'gbFeature'  
hasQualif(x, qualifier)  
  
## S4 method for signature 'gbFeatureTable'  
hasQualif(x, qualifier)  
  
## S4 method for signature 'gbRecord'  
hasQualif(x, qualifier)  
  
## S4 method for signature 'gbRecordList'  
hasQualif(x, qualifier)
```

Arguments

x	A gbFeature , gbFeatureTable , gbRecord , or gbRecordList object.
qualifier	A character string. Name of a qualifier.
...	Additional arguments to be passed to or from methods.

Value

A logical vector or a list of logical vectors.

See Also

[qualifList](#), to extract a list of available qualifiers for each feature; [uniqueQualifs](#), for a vector of all unique qualifiers present in an object.

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))  
hasQualif(x, 'CDS')
```

index	<i>Access the indices of GenBank features</i>
-------	---

Description

Access the indices of GenBank features

Usage

```
index(x, ...)  
  
## S4 method for signature 'gbFeature'  
index(x)  
  
## S4 method for signature 'gbFeatureTable'  
index(x)  
  
## S4 method for signature 'gbRecord'  
index(x)  
  
## S4 method for signature 'gbRecordList'  
index(x)
```

Arguments

x	A gbFeature , gbFeatureTable , gbRecord , or gbRecordList object.
...	Additional arguments passed to methods.

Value

A numeric vector of feature indices.

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))  
index(x)
```

key	<i>Get/set keys of GenBank features</i>
-----	---

Description

Get/set keys of GenBank features

Usage

```
key(x, ...)  
  
key(x, check = TRUE) <- value  
  
## S4 method for signature 'gbFeature'  
key(x)  
  
## S4 replacement method for signature 'gbFeature'  
key(x, check = TRUE) <- value  
  
## S4 method for signature 'gbFeatureTable'  
key(x)  
  
## S4 method for signature 'gbRecord'  
key(x)  
  
## S4 method for signature 'gbRecordList'  
key(x)
```

Arguments

x	A gbFeature , gbFeatureTable , gbRecord , or gbRecordList object.
...	Additional arguments passed to methods.
check	if FALSE, don't perform validity checks.
value	The key information to set on x.

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))  
key(x)
```

length.gbRecord-method

Get the number of gbFeatures.

Description

Get the number of gbFeatures.

Usage

```
## S4 method for signature 'gbRecord'  
length(x)
```

Arguments

x A gbRecord

Value

An integer

location	<i>Access genomic locations of GenBank features</i>
----------	---

Description

Access genomic locations of GenBank features

Usage

```
location(x, ...)

## S4 method for signature 'gbFeature'
location(x)

## S4 method for signature 'gbFeatureTable'
location(x, join = FALSE)

## S4 method for signature 'gbRecord'
location(x, join = FALSE)
```

Arguments

x A [gbFeature](#), [gbFeatureTable](#), or [gbRecord](#) object.
 ... Additional arguments passed to methods.
 join Join compound genomic locations to a single range.

Value

A list of [gbLocation](#) objects

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))
location(x)
```

locusTag	<i>Return the locus_tag qualifiers from GenBank features.</i>
----------	---

Description

Return the *locus_tag* qualifiers from GenBank features.

Usage

```
locusTag(...)
```

Arguments

... A [gbRecord](#), [gbFeatureTable](#), or, [gbFeature](#) instance.

Value

A character vector of *locus_tags*.

note	<i>Return the note qualifiers from GenBank features.</i>
------	--

Description

Return the *note* qualifiers from GenBank features.

Usage

```
note(...)
```

Arguments

... A [gbRecord](#), [gbFeatureTable](#), or, [gbFeature](#) instance.

Value

A character vector of *notes*.

product	<i>Return the product qualifiers from GenBank features.</i>
---------	---

Description

Return the *product* qualifiers from GenBank features.

Usage

```
product(...)
```

Arguments

... A [gbRecord](#), [gbFeatureTable](#), or, [gbFeature](#) instance.

Value

A character vector of *products*.

proteinID	<i>Return the protein_id qualifiers from GenBank features.</i>
-----------	--

Description

Return the *protein_id* qualifiers from GenBank features.

Usage

```
proteinID(...)
```

Arguments

... A [gbRecord](#), [gbFeatureTable](#), or, [gbFeature](#) instance.

Value

A character vector of *protein_ids*.

`qualif`*Get/set qualifiers of GenBank features*

Description

Get/set qualifiers of GenBank features

Usage

```
qualif(x, which = "", ...)  
  
qualif(x, which, check = TRUE) <- value  
  
## S4 method for signature 'gbFeature'  
qualif(x, which, fixed = FALSE, use.names = TRUE)  
  
## S4 replacement method for signature 'gbFeature'  
qualif(x, which, check = TRUE) <- value  
  
## S4 method for signature 'gbFeatureTable'  
qualif(x, which = "", fixed = FALSE,  
       use.names = TRUE)  
  
## S4 method for signature 'gbRecord'  
qualif(x, which = "", fixed = FALSE,  
       use.names = TRUE)
```

Arguments

<code>x</code>	A <code>gbFeature</code> , <code>gbFeatureTable</code> , or <code>gbRecord</code> object.
<code>which</code>	A character vector giving the name(s) of the qualifiers to retrieve or set.
<code>...</code>	Additional arguments passed to methods.
<code>check</code>	if FALSE, don't perform validity checks.
<code>value</code>	The qualifier information to set on <code>x</code> .
<code>fixed</code>	If TRUE, which is matched against qualifiers as is, if FALSE it is treated as a regular expression.
<code>use.names</code>	If TRUE, return a <code>data.frame</code> using which as column names, if FALSE return, if possible, a character vector or a list.

Value

A `data.frame`.

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))
qualif(x[[1]], 'db_xref')

## use shortcuts to common qualifiers
proteinID(x["CDS"])
locusTag(x["CDS"])
```

qualifList

List the names of Genbank qualifiers.

Description

List the names of Genbank qualifiers.

Usage

```
qualifList(x, ...)

## S4 method for signature 'gbFeature'
qualifList(x)

## S4 method for signature 'gbFeatureTable'
qualifList(x)

## S4 method for signature 'gbRecord'
qualifList(x)
```

Arguments

x A [gbFeature](#), [gbFeatureTable](#), [gbRecord](#), or [gbRecord](#) object.
... Additional arguments to be passed to or from methods.

Value

A character vector (or list of character vectors) of qualifier names.

See Also

[uniqueQualifs](#), [hasQualif](#)

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))
qualifList(x["source"])
```

qualifTable	<i>Tabulate Genbank qualifiers</i>
-------------	------------------------------------

Description

Extract a frequency table (or list of tables in the case of `gbRecordLists`) of qualifier names.

Usage

```
qualifTable(x, ...)  
  
## S4 method for signature 'gbFeatureTable'  
qualifTable(x)  
  
## S4 method for signature 'gbRecord'  
qualifTable(x)  
  
## S4 method for signature 'gbRecordList'  
qualifTable(x)
```

Arguments

x	A gbFeatureTable , gbRecord , or gbRecordList object.
...	Additional arguments to be passed to or from methods.

Value

A [table](#) (or list of tables) of qualifiers names.

See Also

[uniqueQualifs](#), [hasQualif](#)

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))  
qualifTable(x)
```

ranges

Extract features as "GRanges" objects.

Description

Extract features as "GRanges" objects.

Usage

```
ranges(x, join = FALSE, key = TRUE, include = "none", exclude = "", ...)
```

```
## S4 method for signature 'gbFeature'  
ranges(x, join = FALSE, key = TRUE,  
       include = "none", exclude = "", ...)
```

```
## S4 method for signature 'gbFeatureTable'  
ranges(x, join = FALSE, key = TRUE,  
       include = "none", exclude = "", ...)
```

```
## S4 method for signature 'gbRecord'  
ranges(x, join = FALSE, key = TRUE, include = "none",  
       exclude = "", ...)
```

```
## S4 method for signature 'gbRecordList'  
ranges(x, join = FALSE, key = TRUE,  
       include = "none", exclude = "", ...)
```

Arguments

x	A gbFeature , gbFeatureTable , gbRecord , or gbRecordList object.
join	Join compound genomic locations into a single range.
key	Include feature keys with ranges.
include	Include qualifiers as metadata columns. Can be "none", "all", or a character vector of qualifier tags.
exclude	Exclude specific qualifiers.
...	Further arguments passed to methods.

Value

A [GRanges](#) or [GRangesList](#) object.

See Also

[start](#), [end](#), [span](#), [strand](#), [location](#), [key](#), [qualif](#)

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))

## default "GRanges" object.
ranges(x)

## subset CDSs and include "product", "note", "protein_id" as metadata.
ranges(x["CDS"], include = c("product", "note", "protein_id"))

## subset CDSs and exclude "translation"
ranges(x["CDS"], include = "all", exclude = "translation")
```

revcomp

Reverse-complement features in a GenBank record

Description

Reverse-complement features in a GenBank record

Usage

```
revcomp(x, order = TRUE, ...)
```

S4 method for signature 'gbFeatureTable'

```
revcomp(x, order = TRUE)
```

S4 method for signature 'gbRecord'

```
revcomp(x, order = TRUE)
```

Arguments

x	A gbFeatureTable or gbRecord object (gbFeatureTables must be complete and include a 'source' field).
order	Reorder features after reverse-complementing them.
...	Additional arguments passed to methods.

Examples

```
load(system.file("extdata", "S_cerevisiae_mito.rda", package = "biofiles"))
xr <- revcomp(x)
```

saveRecord	<i>Save and load gbRecord objects.</i>
------------	--

Description

Serialise and unserialise [gbRecords](#) using [saveRDS](#) and [readRDS](#)

Usage

```
saveRecord(x, file = NULL, dir = ".", ...)  
  
## S4 method for signature 'gbRecord'  
saveRecord(x, file = NULL, dir = ".", ...)  
  
## S4 method for signature 'gbRecordList'  
saveRecord(x, file = NULL, dir = ".", ...)  
  
loadRecord(file, ...)
```

Arguments

x	A gbRecord or gbRecordList instance.
file	A character string naming the file to write to or read from. If NULL, the accession number will be used to construct a file name.
dir	Target directory. (Default: current working directory)
...	Arguments passed to saveRDS .

Examples

```
## Not run:  
aca <- genomeRecordFromNCBI("Bacteria/Acaryochloris_marina", verbose = TRUE)  
aca  
saveRecord(aca)  
rm(aca)  
aca <- loadRecord("./NC_009925_NC_009926_NC_009927_NC_009928_NC_009929_NC_0099___.rds")  
aca  
  
## End(Not run)
```

`shift`*Shift the location of features in a GenBank record*

Description

Shift the location of features in a GenBank record

Usage

```
shift(x, shift = 0L, split = FALSE, order = TRUE, ...)
```

```
## S4 method for signature 'gbLocation'  
shift(x, shift = 0L, split = FALSE, order = TRUE,  
      ...)
```

```
## S4 method for signature 'gbFeature'  
shift(x, shift = 0L, split = FALSE, order = TRUE,  
      ...)
```

```
## S4 method for signature 'gbFeatureTable'  
shift(x, shift = 0L, split = FALSE,  
      order = FALSE)
```

```
## S4 method for signature 'gbRecord'  
shift(x, shift, split = FALSE, order = TRUE)
```

Arguments

<code>x</code>	A gbFeatureTable or gbRecord instance (gbFeatureTables must be complete and include a 'source' field).
<code>shift</code>	Number of basepairs (or aa residues) to shift.
<code>split</code>	Split features that after the shift extends across the end of the sequence.
<code>order</code>	Reorder features after the shift.
<code>...</code>	Additional arguments passed to methods.

Value

A [gbFeatureTable](#) object.

Note

`shift` does not currently handle compound locations In a shifted feature table compound locations get merged.

Examples

```
load(system.file("extdata", "S_cerevisiae_mito.rda", package = "biofiles"))

## shift the S. cerevisiae mitochondrion such that cytochrome b is the first CDS
cytb <- start(filter(x, product = "^cytochrome b$")[[1]])[1]
x2 <- shift(x, shift = -cytb + 1, split = TRUE)
```

span

Get the span of genomic features.

Description

Get the span of genomic features.

Usage

```
span(x, ...)
```

```
joint_range(x)
```

```
## S4 method for signature 'gbLocation'
```

```
span(x, join = FALSE)
```

```
## S4 method for signature 'gbLocation'
```

```
joint_range(x)
```

```
## S4 method for signature 'gbFeature'
```

```
span(x, join = FALSE)
```

```
## S4 method for signature 'gbFeature'
```

```
joint_range(x)
```

```
## S4 method for signature 'gbFeatureTable'
```

```
span(x, join = FALSE)
```

```
## S4 method for signature 'gbFeatureTable'
```

```
joint_range(x)
```

```
## S4 method for signature 'gbRecord'
```

```
span(x, join = FALSE)
```

```
## S4 method for signature 'gbRecordList'
```

```
span(x, join = FALSE)
```

Arguments

x A [gbFeature](#), [gbFeatureTable](#), [gbRecord](#), or [gbRecordList](#) object.
 ... Further arguments passed to methods.
 join Join compound genomic locations into a single range.

Value

An integer vector or a list of integer vectors.

See Also

[start](#), [end](#), [strand](#), [ranges](#)

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))
span(x)
```

start	<i>Get or set the start position of genomic features</i>
-------	--

Description

Get or set the start position of genomic features

Usage

```
start(x, ...)

start(x, ...) <- value

## S4 method for signature 'gbLocation'
start(x, join = FALSE)

## S4 replacement method for signature 'gbLocation'
start(x, ...) <- value

## S4 method for signature 'gbFeature'
start(x, join = FALSE)

## S4 replacement method for signature 'gbFeature'
start(x, ...) <- value

## S4 method for signature 'gbFeatureTable'
start(x, join = FALSE)

## S4 replacement method for signature 'gbFeatureTable'
```

```
start(x, ...) <- value

## S4 method for signature 'gbRecord'
start(x, join = FALSE)

## S4 method for signature 'gbRecordList'
start(x, join = FALSE)
```

Arguments

x	A gbFeature , gbFeatureTable , gbRecord , or gbRecordList object.
...	Further arguments passed to methods.
value	The start information to set on x.
join	Join compound genomic locations into a single range.

Value

An integer vector or a list of integer vectors.

See Also

[end](#), [strand](#), [span](#), [ranges](#)

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))

## start
start(x)
cds <- x["CDS"]
start(cds)

## `start<-`
start(cds) <- 10
start(cds)
```

strand

Get or set the strand information of genomic features

Description

Get or set the strand information of genomic features

Usage

```
strand(x, ...)  
  
strand(x, ...) <- value  
  
## S4 method for signature 'gbLocation'  
strand(x, join = FALSE)  
  
## S4 replacement method for signature 'gbLocation'  
strand(x, ...) <- value  
  
## S4 method for signature 'gbFeature'  
strand(x, join = FALSE)  
  
## S4 replacement method for signature 'gbFeature'  
strand(x, ...) <- value  
  
## S4 method for signature 'gbFeatureTable'  
strand(x, join = FALSE)  
  
## S4 replacement method for signature 'gbFeatureTable'  
strand(x, ...) <- value  
  
## S4 method for signature 'gbRecord'  
strand(x, join = FALSE)  
  
## S4 method for signature 'gbRecordList'  
strand(x, join = FALSE)
```

Arguments

x	A gbFeature , gbFeatureTable , gbRecord , or gbRecordList object.
...	Further arguments passed to methods.
value	The strand information to set on x.
join	Join compound genomic locations into a single range.

Value

An integer vector (or a list thereof) of 1 (plus strand), -1 (minus strand), or NA

See Also

[start](#), [end](#), [span](#), [ranges](#)

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))
```

```
## strand
strand(x)

## `strand<-`
showMethods("strand<-")
```

summary	<i>Summarise a GenBank record.</i>
---------	------------------------------------

Description

Summarise a GenBank record.

Usage

```
summary(object, ...)

## S4 method for signature 'gbFeature'
summary(object, ...)

## S4 method for signature 'gbFeatureTable'
summary(object, n = 8, ...)

## S4 method for signature 'gbRecord'
summary(object, n = 7, ...)

## S4 method for signature 'gbRecordList'
summary(object, n = 2, ...)
```

Arguments

object	An object of class <code>gbFeature</code> , <code>gbFeatureTable</code> , <code>gbRecord</code> , or <code>gbRecordList</code> .
...	Arguments to be passed to methods.
n	How many elements should be summarized in head and tail.

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))
summary(x)
```

translation	<i>Return the translations from GenBank features.</i>
-------------	---

Description

Return the *translations* from GenBank features.

Usage

```
translation(x)
```

Arguments

x A [gbRecord](#), [gbFeatureTable](#), or, [gbFeature](#) instance.

Value

An [AAStringSet](#).

uniqueQualifs	<i>Quickly list all qualifier names.</i>
---------------	--

Description

Quickly list all qualifier names.

Usage

```
uniqueQualifs(...)
```

Arguments

... A [gbRecord](#), [gbFeatureTable](#), or, [gbFeature](#) instance.

Value

A character vector of qualifier names.

upstream	<i>Find flanking features.</i>
----------	--------------------------------

Description

Find flanking features.

Usage

```
upstream(query, subject, n = 5, include_key = "all", exclude_key = "none")
```

```
downstream(query, subject, n = 5, include_key = "all",  
            exclude_key = "none")
```

```
flanking(query, subject, n = 5, include_key = "all", exclude_key = "none")
```

Arguments

query	A gbFeature or gbFeatureTable object.
subject	A gbRecord or gbFeatureTable object within which the n nearest upstream features are found.
n	The number of upstream features to be returned.
include_key	Which features should be returned. Defaults to "all".
exclude_key	Which feature(s) should be excluded from the search. Defaults to "none".

Value

A (list of) [gbFeatureTables](#).

Examples

```
load(system.file("extdata", "S_cerevisiae_mito.rda", package = "biofiles"))  
cytb <- ft(filter(x, product = "^cytochrome b$"))
```

```
## find the three nearest upstream neighbor CDS to CYTB  
upstream(cytb, x["CDS"], n = 3)
```

write.FeatureTable *Write GenBank records or features to file in Feature Table format*

Description

Feature Tables are simple five-column tab-delimited tables specifying the location and type of each feature. They can be used as input for tbl2asn or Sequin to generate annotation.

Usage

```
write.FeatureTable(x, file, tablename = "", dbname = "", sequence = FALSE,
  append = FALSE, ...)
```

```
## S4 method for signature 'gbRecord'
write.FeatureTable(x, file, tablename = "",
  dbname = "", sequence = FALSE, append = FALSE)
```

```
## S4 method for signature 'gbFeatureTable'
write.FeatureTable(x, file, tablename = "",
  dbname = "", sequence = FALSE, append = FALSE)
```

Arguments

x	A gbRecord instance.
file	A connection or a character string naming the file to write to.
tablename	Optional table name to appear in the first line of the feature table.
dbname	Data base name associated with the CDS qualifier protein_id.
sequence	if TRUE, additionally output a fasta file.
append	if TRUE the data is appended to the connection.
...	Additional arguments passed to methods.

Examples

```
## Not run:
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))
write.FeatureTable(x, file = "data/marine_metagenome.tbl")

## End(Not run)
```

write.GenBank	<i>Write GenBank records or features to file in GenBank format</i>
---------------	--

Description

Write GenBank records or features to file in GenBank format

Usage

```
write.GenBank(x, file, append = FALSE, ...)

## S4 method for signature 'gbRecord'
write.GenBank(x, file, header = TRUE, sequence = TRUE,
  append = FALSE)

## S4 method for signature 'gbFeatureTable'
write.GenBank(x, file, header = TRUE,
  sequence = TRUE, append = FALSE)
```

Arguments

x	A gbRecord instance.
file	A connection or a character string naming the file to write to.
append	if TRUE the data is appended to the connection.
...	Additional arguments passed to methods.
header	if FALSE exclude the Genbank header.
sequence	if FALSE exclude the sequence.

Details

For a description of the GenBank format see <http://www.ncbi.nlm.nih.gov/collab/FT/>

Examples

```
## Not run:
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))
write.GenBank(x, file = "data/marine_metagenome.gb")

## write selected features to file.
write.GenBank(x["CDS"], file = "data/marine_metagenome_cds.gb", header = FALSE, sequence = FALSE)

## End(Not run)
```

[[,gbFeature,character,missing-method

Method extensions to extraction operator for gbRecord objects.

Description

See the documentation for the [Extract](#) generic, defined in the R [base-package](#) for the expected behavior.

Usage

```
## S4 method for signature 'gbFeature,character,missing'
x[[i, j]]

## S4 method for signature 'gbFeature'
x$name

## S4 method for signature 'gbFeatureTable,character,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'gbFeatureTable,numeric,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'gbFeatureTable,logical,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'gbFeatureTable,missing,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'gbFeatureTable,ANY,ANY'
x[[i, j, ...]]

## S4 method for signature 'gbRecord,ANY,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'gbRecord,missing,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'gbRecord,ANY,ANY'
x[[i, j, ...]]
```

Arguments

x	A gbFeature , gbFeatureTable , or gbRecord object.
i	indices specifying elements to extract. With gbFeatureTables and gbRecords , a character index is matched against feature keys; gbFeatures a character index is matched against qualifiers.

j	Not used.
name	The name of the element to extract.
...	Not used.
drop	Not used.

Value

A [gbFeatureTable](#) object or elements of a [gbFeature](#) object.

See Also

[Extract](#)

Examples

```
load(system.file("extdata", "marine_metagenome.rda", package = "biofiles"))

## Extract a gbFeatureTable from a gbRecord:
x[1:4]

## Extract a gbFeature
x[[1]]

## Extract ggFeatures by Feature Key
x["CDS"]
```


Index

*Topic **package**
 biofiles, 3

[,gbFeatureTable,character,missing-method
 ([[,gbFeature,character,missing-method),
 47

[,gbFeatureTable,logical,missing-method
 ([[,gbFeature,character,missing-method),
 47

[,gbFeatureTable,missing,missing-method
 ([[,gbFeature,character,missing-method),
 47

[,gbFeatureTable,numeric,missing-method
 ([[,gbFeature,character,missing-method),
 47

[,gbRecord,ANY,missing-method
 ([[,gbFeature,character,missing-method),
 47

[,gbRecord,missing,missing-method
 ([[,gbFeature,character,missing-method),
 47

[[,gbFeature,character,missing-method,
 47

[[,gbFeatureTable,ANY,ANY-method
 ([[,gbFeature,character,missing-method),
 47

[[,gbRecord,ANY,ANY-method
 ([[,gbFeature,character,missing-method),
 47

\$.gbFeature-method
 ([[,gbFeature,character,missing-method),
 47

AAStringSet, 43

as.gbLocation, 3

biofiles, 3

biofiles-package (biofiles), 3

dbxref, 4

dbxref,gbFeature-method (dbxref), 4

dbxref,gbRecord-method (dbxref), 4

downstream (upstream), 44

efetch, 11

end, 5, 34, 39–41

end,gbFeature-method (end), 5

end,gbFeatureTable-method (end), 5

end,gbLocation-method (end), 5

end,gbRecord-method (end), 5

end,gbRecordList-method (end), 5

end<- (end), 5

end<-,gbFeature-method (end), 5

end<-,gbFeatureTable-method (end), 5

end<-,gbLocation-method (end), 5

Extract, 47, 48

featureTable, 6

featureTable,gbFeatureTable-method
 (featureTable), 6

featureTable,gbRecord-method
 (featureTable), 6

featureTable,gbRecordList-method
 (featureTable), 6

filter, 7

filter,gbFeatureTable-method (filter), 7

filter,gbRecord-method (filter), 7

filter,gbRecordList-method (filter), 7

flanking (upstream), 44

ft, 12

ft (getFeatures), 14

ft,gbRecord-method (getFeatures), 14

ft,gbRecordList-method (getFeatures), 14

fuzzy, 8

fuzzy,gbFeature-method (fuzzy), 8

fuzzy,gbFeatureTable-method (fuzzy), 8

fuzzy,gbLocation-method (fuzzy), 8

fuzzy,gbRecord-method (fuzzy), 8

fuzzy,gbRecordList-method (fuzzy), 8

- gbFeature, [4](#), [5](#), [9](#), [10](#), [13](#), [15](#), [22–32](#), [34](#), [39–44](#), [47](#), [48](#)
- gbFeature-class, [9](#)
- gbFeatureTable, [4–7](#), [9](#), [10](#), [12–15](#), [22–35](#), [37](#), [39–44](#), [47](#), [48](#)
- gbFeatureTable-class, [10](#)
- gbHeader, [10](#), [12](#), [15](#)
- gbLocation, [3](#), [10](#), [28](#)
- gbRecord, [4–7](#), [9](#), [10](#), [11–15](#), [22–37](#), [39–47](#)
- gbRecord-class, [12](#)
- gbRecordList, [5–7](#), [9–11](#), [13](#), [22–27](#), [33](#), [34](#), [36](#), [39–42](#)
- gbRecordList (gbRecordList-class), [13](#)
- gbRecordList-class, [13](#)
- geneID, [13](#)
- genomeRecordFromNCBI, [11](#)
- getAccession (getLocus), [16](#)
- getAccession,gbFeature-method (getLocus), [16](#)
- getAccession,gbFeatureTable-method (getLocus), [16](#)
- getAccession,gbLocation-method (getLocus), [16](#)
- getAccession,gbRecord-method (getLocus), [16](#)
- getAccession,gbRecordList-method (getLocus), [16](#)
- getAccession,seqinfo-method (getLocus), [16](#)
- getComment (getLocus), [16](#)
- getComment,gbFeature-method (getLocus), [16](#)
- getComment,gbFeatureTable-method (getLocus), [16](#)
- getComment,gbRecord-method (getLocus), [16](#)
- getComment,gbRecordList-method (getLocus), [16](#)
- getComment,seqinfo-method (getLocus), [16](#)
- getDate (getLocus), [16](#)
- getDate,gbFeature-method (getLocus), [16](#)
- getDate,gbFeatureTable-method (getLocus), [16](#)
- getDate,gbRecord-method (getLocus), [16](#)
- getDate,gbRecordList-method (getLocus), [16](#)
- getDate,seqinfo-method (getLocus), [16](#)
- getDBLink (getLocus), [16](#)
- getDBLink,gbFeature-method (getLocus), [16](#)
- getDBLink,gbFeatureTable-method (getLocus), [16](#)
- getDBLink,gbRecord-method (getLocus), [16](#)
- getDBLink,gbRecordList-method (getLocus), [16](#)
- getDBLink,seqinfo-method (getLocus), [16](#)
- getDBSource (getLocus), [16](#)
- getDBSource,gbFeature-method (getLocus), [16](#)
- getDBSource,gbFeatureTable-method (getLocus), [16](#)
- getDBSource,gbRecord-method (getLocus), [16](#)
- getDBSource,gbRecordList-method (getLocus), [16](#)
- getDBSource,seqinfo-method (getLocus), [16](#)
- getDefinition (getLocus), [16](#)
- getDefinition,gbFeature-method (getLocus), [16](#)
- getDefinition,gbFeatureTable-method (getLocus), [16](#)
- getDefinition,gbRecord-method (getLocus), [16](#)
- getDefinition,gbRecordList-method (getLocus), [16](#)
- getDefinition,seqinfo-method (getLocus), [16](#)
- getDivision (getLocus), [16](#)
- getDivision,gbFeature-method (getLocus), [16](#)
- getDivision,gbFeatureTable-method (getLocus), [16](#)
- getDivision,gbRecord-method (getLocus), [16](#)
- getDivision,gbRecordList-method (getLocus), [16](#)
- getDivision,seqinfo-method (getLocus), [16](#)
- getFeatures, [14](#)
- getFeatures,gbRecord-method (getFeatures), [14](#)
- getFeatures,gbRecordList-method (getFeatures), [14](#)
- getGeneID (getLocus), [16](#)
- getGeneID,gbFeature-method (getLocus),

16
 getGeneID, gbFeatureTable-method
 (getLocus), 16
 getGeneID, gbRecord-method (getLocus), 16
 getGeneID, gbRecordList-method
 (getLocus), 16
 getGeneID, seqinfo-method (getLocus), 16
 getHeader, 10, 15
 getHeader, gbFeature-method (getHeader),
 15
 getHeader, gbFeatureTable-method
 (getHeader), 15
 getHeader, gbRecord-method (getHeader),
 15
 getKeywords (getLocus), 16
 getKeywords, gbFeature-method
 (getLocus), 16
 getKeywords, gbFeatureTable-method
 (getLocus), 16
 getKeywords, gbRecord-method (getLocus),
 16
 getKeywords, gbRecordList-method
 (getLocus), 16
 getKeywords, seqinfo-method (getLocus),
 16
 getLength (getLocus), 16
 getLength, gbFeature-method (getLocus),
 16
 getLength, gbFeatureTable-method
 (getLocus), 16
 getLength, gbRecord-method (getLocus), 16
 getLength, gbRecordList-method
 (getLocus), 16
 getLength, seqinfo-method (getLocus), 16
 getLocus, 16
 getLocus, gbFeature-method (getLocus), 16
 getLocus, gbFeatureTable-method
 (getLocus), 16
 getLocus, gbRecord-method (getLocus), 16
 getLocus, gbRecordList-method
 (getLocus), 16
 getLocus, seqinfo-method (getLocus), 16
 getMoltype (getLocus), 16
 getMoltype, gbFeature-method (getLocus),
 16
 getMoltype, gbFeatureTable-method
 (getLocus), 16
 getMoltype, gbRecord-method (getLocus),

16
 getMoltype, gbRecordList-method
 (getLocus), 16
 getMoltype, seqinfo-method (getLocus), 16
 getOrganism (getLocus), 16
 getOrganism, gbFeature-method
 (getLocus), 16
 getOrganism, gbFeatureTable-method
 (getLocus), 16
 getOrganism, gbRecord-method (getLocus),
 16
 getOrganism, gbRecordList-method
 (getLocus), 16
 getOrganism, seqinfo-method (getLocus),
 16
 getReference (getLocus), 16
 getReference, gbFeature-method
 (getLocus), 16
 getReference, gbFeatureTable-method
 (getLocus), 16
 getReference, gbRecord-method
 (getLocus), 16
 getReference, gbRecordList-method
 (getLocus), 16
 getReference, seqinfo-method (getLocus),
 16
 getSequence, 10, 23
 getSequence, gbFeature-method
 (getSequence), 23
 getSequence, gbFeatureTable-method
 (getSequence), 23
 getSequence, gbRecord-method
 (getSequence), 23
 getSequence, gbRecordList-method
 (getSequence), 23
 getSource (getLocus), 16
 getSource, gbFeature-method (getLocus),
 16
 getSource, gbFeatureTable-method
 (getLocus), 16
 getSource, gbRecord-method (getLocus), 16
 getSource, gbRecordList-method
 (getLocus), 16
 getSource, seqinfo-method (getLocus), 16
 getTaxonomy (getLocus), 16
 getTaxonomy, gbFeature-method
 (getLocus), 16
 getTaxonomy, gbFeatureTable-method

- (getLocus), 16
- getTaxonomy, gbRecord-method (getLocus), 16
- getTaxonomy, gbRecordList-method (getLocus), 16
- getTaxonomy, seqinfo-method (getLocus), 16
- getTopology (getLocus), 16
- getTopology, gbFeature-method (getLocus), 16
- getTopology, gbFeatureTable-method (getLocus), 16
- getTopology, gbRecord-method (getLocus), 16
- getTopology, gbRecordList-method (getLocus), 16
- getTopology, seqinfo-method (getLocus), 16
- getVersion (getLocus), 16
- getVersion, gbFeature-method (getLocus), 16
- getVersion, gbFeatureTable-method (getLocus), 16
- getVersion, gbRecord-method (getLocus), 16
- getVersion, gbRecordList-method (getLocus), 16
- getVersion, seqinfo-method (getLocus), 16
- GRanges, 34
- GRangesList, 34
- hasKey, 24
- hasKey, gbFeature-method (hasKey), 24
- hasKey, gbFeatureTable-method (hasKey), 24
- hasKey, gbRecord-method (hasKey), 24
- hasKey, gbRecordList-method (hasKey), 24
- hasQualif, 25, 32, 33
- hasQualif, gbFeature-method (hasQualif), 25
- hasQualif, gbFeatureTable-method (hasQualif), 25
- hasQualif, gbRecord-method (hasQualif), 25
- hasQualif, gbRecordList-method (hasQualif), 25
- header, 12
- header (getHeader), 15
- header, gbFeature-method (getHeader), 15
- header, gbFeatureTable-method (getHeader), 15
- header, gbRecord-method (getHeader), 15
- index, 10, 26
- index, gbFeature-method (index), 26
- index, gbFeatureTable-method (index), 26
- index, gbRecord-method (index), 26
- index, gbRecordList-method (index), 26
- joint_range (span), 38
- joint_range, gbFeature-method (span), 38
- joint_range, gbFeatureTable-method (span), 38
- joint_range, gbLocation-method (span), 38
- key, 10, 24, 26, 34
- key, gbFeature-method (key), 26
- key, gbFeatureTable-method (key), 26
- key, gbRecord-method (key), 26
- key, gbRecordList-method (key), 26
- key<- (key), 26
- key<- , gbFeature-method (key), 26
- length, gbRecord-method, 27
- loadRecord (saveRecord), 36
- location, 28, 34
- location, gbFeature-method (location), 28
- location, gbFeatureTable-method (location), 28
- location, gbRecord-method (location), 28
- locusTag, 29
- new_gbRecord (gbRecord-class), 12
- note, 29
- product, 30
- proteinID, 30
- qualif, 10, 31, 34
- qualif, gbFeature-method (qualif), 31
- qualif, gbFeatureTable-method (qualif), 31
- qualif, gbRecord-method (qualif), 31
- qualif<- (qualif), 31
- qualif<- , gbFeature-method (qualif), 31
- qualifList, 25, 32
- qualifList, gbFeature-method (qualifList), 32

- qualifList,gbFeatureTable-method
(qualifList), 32
- qualifList,gbRecord-method
(qualifList), 32
- qualifTable, 7, 33
- qualifTable,gbFeatureTable-method
(qualifTable), 33
- qualifTable,gbRecord-method
(qualifTable), 33
- qualifTable,gbRecordList-method
(qualifTable), 33

- ranges, 6, 10, 12, 34, 39–41
- ranges,gbFeature-method (ranges), 34
- ranges,gbFeatureTable-method (ranges),
34
- ranges,gbRecord-method (ranges), 34
- ranges,gbRecordList-method (ranges), 34
- readRDS, 36
- revcomp, 35
- revcomp,gbFeatureTable-method
(revcomp), 35
- revcomp,gbRecord-method (revcomp), 35

- saveRDS, 36
- saveRecord, 36
- saveRecord,gbRecord-method
(saveRecord), 36
- saveRecord,gbRecordList-method
(saveRecord), 36
- select (filter), 7
- select,gbFeatureTable-method (filter), 7
- select,gbRecord-method (filter), 7
- select,gbRecordList-method (filter), 7
- seqinfo, 10, 12
- shift, 37
- shift,gbFeature-method (shift), 37
- shift,gbFeatureTable-method (shift), 37
- shift,gbLocation-method (shift), 37
- shift,gbRecord-method (shift), 37
- span, 6, 34, 38, 40, 41
- span,gbFeature-method (span), 38
- span,gbFeatureTable-method (span), 38
- span,gbLocation-method (span), 38
- span,gbRecord-method (span), 38
- span,gbRecordList-method (span), 38
- start, 6, 34, 39, 39, 41
- start,gbFeature-method (start), 39
- start,gbFeatureTable-method (start), 39
- start,gbLocation-method (start), 39
- start,gbRecord-method (start), 39
- start,gbRecordList-method (start), 39
- start<- (start), 39
- start<-,gbFeature-method (start), 39
- start<-,gbFeatureTable-method (start),
39
- start<-,gbLocation-method (start), 39
- strand, 6, 34, 39, 40, 40
- strand,gbFeature-method (strand), 40
- strand,gbFeatureTable-method (strand),
40
- strand,gbLocation-method (strand), 40
- strand,gbRecord-method (strand), 40
- strand,gbRecordList-method (strand), 40
- strand<- (strand), 40
- strand<-,gbFeature-method (strand), 40
- strand<-,gbFeatureTable-method
(strand), 40
- strand<-,gbLocation-method (strand), 40
- summary, 42
- summary,gbFeature-method (summary), 42
- summary,gbFeatureTable-method
(summary), 42
- summary,gbRecord-method (summary), 42
- summary,gbRecordList-method (summary),
42

- table, 6, 33
- translation, 43

- uniqueQualifs, 25, 32, 33, 43
- upstream, 44

- write.FeatureTable, 45
- write.FeatureTable,gbFeatureTable-method
(write.FeatureTable), 45
- write.FeatureTable,gbRecord-method
(write.FeatureTable), 45
- write.GenBank, 46
- write.GenBank,gbFeatureTable-method
(write.GenBank), 46
- write.GenBank,gbRecord-method
(write.GenBank), 46

- XStringSet, 10, 12, 23