

# Package: bedbaser (via r-universe)

February 22, 2025

**Title** A BEDbase client

**Version** 0.99.23

**Description** A client for BEDbase. bedbaser provides access to the API at [api.bedbase.org](https://api.bedbase.org). It also includes convenience functions to import BED files into GRanges objects and BEDsets into GRangesLists.

**URL** <https://github.com/waldronlab/bedbaser>

**BugReports** <https://github.com/waldronlab/bedbaser/issues>

**biocViews** Software, DataImport, ThirdPartyClient

**License** Artistic License 2.0

**Imports** AnVIL (>= 1.16.0), BiocFileCache, dplyr, GenomeInfoDb, GenomicRanges, httr, methods, purrr, rtracklayer, rlang, R.utils, stats, stringr, tibble, tidyr, tools, utils

**Suggests** BiocStyle, knitr, liftOver, testthat (>= 3.0.0)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Config/pak/sysreqs** make libicu-dev libxml2-dev libssl-dev zlib1g-dev

**Repository** <https://bioc.r-universe.dev>

**RemoteUrl** <https://github.com/bioc/bedbaser>

**RemoteRef** HEAD

**RemoteSha** 466392b408898fcdce9f220a3580f41dfc7a9435

## Contents

bb_beds_in_bedset . . . . .	2
bb_bed_text_search . . . . .	3

bb_example . . . . .	3
bb_list_beds . . . . .	4
bb_list_bedsets . . . . .	5
bb_metadata . . . . .	5
bb_save . . . . .	6
bb_to_granges . . . . .	7
bb_to_grangeslist . . . . .	8
BEDbase . . . . .	8
BEDbase-class . . . . .	9
getCache,BEDbase-method . . . . .	10
operations,BEDbase-method . . . . .	10
schemas,BEDbase-method . . . . .	11
setCache,BEDbase-method . . . . .	12
tags,BEDbase-method . . . . .	12

## Index 14

---

bb_beds_in_bedset	<i>Get BEDs associated with BEDset</i>
-------------------	--

---

### Description

Return a tibble of BED files in BEDset given its id.

### Usage

```
bb_beds_in_bedset(bedbase, bedset_id)
```

### Arguments

bedbase	BEDbase() object
bedset_id	integer(1) BEDset record identifier

### Value

tibble of BED files in BEDset

### Examples

```
bedbase <- BEDbase()
ex_bedset <- bb_example(bedbase, "bedset")
bb_beds_in_bedset(bedbase, ex_bedset$id)
```

---

bb\_bed\_text\_search      *Search BED files by text*

---

### Description

Return all available BED files ranked by relevance to the keywords. Uses the [bedhost API](#) default of 10 records and an initial offset of 0.

### Usage

```
bb_bed_text_search(bedbase, query, limit = 10, offset = 0)
```

### Arguments

bedbase	BEDbase() object
query	character() keywords to search
limit	integer(1) (default 10) maximum number of results
offset	integer(1) (default 0) page offset of results

### Value

[tibble](#) of results

### Examples

```
bedbase <- BEDbase()
bb_bed_text_search(bedbase, "hg38")
```

---

bb\_example      *Get the example BED file or BEDset with metadata*

---

### Description

Get the example BED file or BEDset available through [bedhost](#). Useful for an initial exploration of bedbaser with an example BED file and BEDset in BEDbase.

### Usage

```
bb_example(bedbase, rec_type = c("bed", "bedset"))
```

### Arguments

bedbase	BEDbase() object
rec_type	character(1) bed or bedset

**Value**

list() bed files or bedsets

**Examples**

```
bedbase <- BEDbase()
ex_bed <- bb_example(bedbase, "bed")
str(ex_bed)
ex_bedset <- bb_example(bedbase, "bedset")
str(ex_bedset)
```

---

bb\_list\_beds

*List BEDs*


---

**Description**

List BED files available through **bedhost**. By default uses the bedhost default of 1000 records and an initial offset of 0.

**Usage**

```
bb_list_beds(bedbase, genome = NULL, bed_type = NULL, limit = 1000, offset = 0)
```

**Arguments**

bedbase	BEDbase() object
genome	character(1) (default NULL) genome keyword
bed_type	character(1) (default NULL) bed file type
limit	integer(1) (default 1000) maximum records
offset	integer(1) (default 0) page token of records

**Value**

[tibble](#) of BED records

**Examples**

```
bedbase <- BEDbase()
bb_list_beds(bedbase)
```

---

bb_list_bedsets	<i>List BEDsets</i>
-----------------	---------------------

---

**Description**

List BEDsets available through `bedhost`. By default uses the `bedhost` default of 1000 records and an initial offset of 0.

**Usage**

```
bb_list_bedsets(bedbase, query = NULL, limit = 1000, offset = 0)
```

**Arguments**

bedbase	BEDbase() object
query	character() (default NULL) keyword
limit	integer(1) (default 1000) maximum records
offset	integer(1) (default 0) page token of records

**Value**

[tibble](#) of BEDset records

**Examples**

```
bedbase <- BEDbase()
bb_list_bedsets(bedbase)
```

---

bb_metadata	<i>Get metadata for a BED file or BEDset</i>
-------------	--

---

**Description**

Get metadata for a BED file or BEDset given its id. Abort if not found or id is not 32 characters.

**Usage**

```
bb_metadata(bedbase, id, full = FALSE)
```

**Arguments**

bedbase	BEDbase() object
id	integer(1) record or object identifier
full	logical(1) (default FALSE) include full record with stats, files, and metadata

**Value**

list() metadata

**Examples**

```
bedbase <- BEDbase()

ex_bed <- bb_example(bedbase, "bed")
bb_metadata(bedbase, ex_bed$id)

ex_bedset <- bb_example(bedbase, "bedset")
bb_metadata(bedbase, ex_bedset$id)
```

---

<code>bb_save</code>	<i>Save a BED file or BEDset to a path given an id</i>
----------------------	--

---

**Description**

Save a BED file or a BEDset to a local path. If the path does not exist, `bb_save()` will abort.

**Usage**

```
bb_save(bedbase, bed_or_bedset_id, path, file_type = "bed", quietly = TRUE)
```

**Arguments**

<code>bedbase</code>	BEDbase() object
<code>bed_or_bedset_id</code>	integer(1) BED or BEDset record identifier
<code>path</code>	character(1) directory to save file
<code>file_type</code>	character(1) (default "bed") bed, bigbed, etc.
<code>quietly</code>	logical(1) (default TRUE) display messages

**Value**

An invisible NULL

**Examples**

```
bedbase <- BEDbase()
ex_bed <- bb_example(bedbase, "bed")
bb_save(bedbase, ex_bed$id, tempdir())
```

---

`bb_to_granges`*Create a GRanges object given a BED id*

---

## Description

Create a GRanges object given a BED id. Columns and types are generated for broad and narrow peak files. Known columns and types can be passed as a named vector through `extra_cols`. Otherwise, `bb_to_granges()` attempts to determine the column type and substitute dummy column names.

## Usage

```
bb_to_granges(  
  bedbase,  
  bed_id,  
  file_type = "bed",  
  extra_cols = NULL,  
  quietly = TRUE  
)
```

## Arguments

<code>bedbase</code>	BEDbase() object
<code>bed_id</code>	integer(1) BED record identifier
<code>file_type</code>	character(1) bed or bigbed
<code>extra_cols</code>	character() (default NULL) extra column names to construct GRanges objects
<code>quietly</code>	logical(1) (default TRUE) display messages

## Value

[GRanges](#)

## Examples

```
bedbase <- BEDbase()  
ex_bed <- bb_example(bedbase, "bed")  
bb_to_granges(bedbase, ex_bed$id)
```

---

`bb_to_grangeslist`      *Create a GRangesList object given a BEDset id*

---

### Description

Create a GRangesList object given a BEDset id

### Usage

```
bb_to_grangeslist(bedbase, bedset_id, quietly = TRUE)
```

### Arguments

<code>bedbase</code>	BEDbase() object
<code>bedset_id</code>	integer(1) BEDset record identifier
<code>quietly</code>	logical(1) (default TRUE) display messages

### Value

[GRangesList](#)

### Examples

```
bedbase <- BEDbase()
bb_to_grangeslist(bedbase, "lola_hg38_ucsc_features")
```

---

`BEDbase`      *An R client for BEDbase*

---

### Description

bedbaser exposes the [bedhost API](#) and includes convenience functions for common tasks, such as to import a BED file by id into a GRanges object and a BEDset by its id into a GRangesList.

### Usage

```
BEDbase(cache_path, quietly = FALSE)

getCache(x, cache_type = c("bedfiles", "bedsets"))

setCache(x, cache_path, quietly = TRUE)
```



**Arguments**

cache_path	character(1)
quietly	logical(1) (default TRUE) display messages
x	BEDbase() object
cache_type	character(1) bedfiles or bedsets

**Details**

BEDbase() creates a cache similar to that of the [Geniml BBClient's cache](#).

The convenience functions are as follows

- bedbaser::BEDbase(): API service constructor
- bedbaser::getCache(): Retrieve cache
- bedbaser::setCache(): Set path to cache
- bedbaser::bb\_example(): Retrieve an example BED file or BEDset
- bedbaser::bb\_metadata(): Retrieve metadata for a BED file or BEDset
- bedbaser::bb\_list\_beds(): List all BED files
- bedbaser::bb\_list\_bedsets(): List all BEDsets
- bedbaser::bb\_beds\_in\_bedset(): List BED files in BEDset
- bedbaser::bb\_bed\_text\_search(): Search BED files by text
- bedbaser::bb\_to\_granges(): Create a GRanges object from a BED id
- bedbaser::bb\_to\_grangeslist(): Create a GRangesList from a BEDset id
- bedbaser::bb\_save(): Save a BED file to a path.

**Value**

BEDbase object

**Examples**

```
bedbase <- BEDbase(cache_path = tempdir())
ex_bed <- bb_example(bedbase, "bed")
bb_metadata(bedbase, ex_bed$id)
```

---

BEDbase-class

*BEDbase class*

---

**Description**

BEDbase class

**Value**

BEDbase class instance

---

getCache, BEDbase-method

*Return cache path*

---

### Description

Return cache path

### Usage

```
## S4 method for signature 'BEDbase'
getCache(x, cache_type = c("bedfiles", "bedsets"))
```

### Arguments

x	BEDbase() object
cache_type	character(1) bedfiles or bedsets

### Value

BiocFileCache() object of BED files

### Examples

```
bedbase <- BEDbase(tempdir())
getCache(bedbase, "bedfiles")
```

---

operations, BEDbase-method

*Display API functions*

---

### Description

Display functions defined through the [bedhost API](#) and their corresponding parameters.

### Usage

```
## S4 method for signature 'BEDbase'
operations(x, ..., .deprecated = FALSE)
```

### Arguments

x	BEDbase() object
...	other options
.deprecated	(default FALSE) if deprecated

**Value**

list() API endpoints

**Examples**

```
bedbase <- BEDbase()  
operations(bedbase)
```

---

schemas,BEDbase-method  
*Display bedhost API schemas*

---

**Description**

Display bedhost API schemas

**Usage**

```
## S4 method for signature 'BEDbase'  
schemas(x)
```

**Arguments**

x                   BEDbase() object

**Value**

list() API endpoints

**Examples**

```
bedbase <- BEDbase()  
schemas(bedbase)
```

---

setCache, BEDbase-method  
*Set cache along path*

---

**Description**

Create a cache for BED files and BEDsets like [Geniml BBClient's cache](#).

**Usage**

```
## S4 method for signature 'BEDbase'
setCache(x, cache_path, quietly = TRUE)
```

**Arguments**

x	BEDbase() object
cache_path	character(1)
quietly	logical(1) (default TRUE) display messages

**Value**

[BiocFileCache\(\)](#) object of BED files

**Examples**

```
bedbase <- BEDbase(tempdir())
bedbase <- setCache(bedbase, "/tmp")
```

---

tags, BEDbase-method    *Display functions for a tag*

---

**Description**

Display functions available through the API associated with a tag keyword in [bedhost](#).

**Usage**

```
## S4 method for signature 'BEDbase'
tags(x, .tags, .deprecated = FALSE)
```

**Arguments**

x	BEDbase() object
.tags	character() tags for filtering operations
.deprecated	(default FALSE) if deprecated

**Value**

list() API endpoints

**Examples**

```
bedbase <- BEDbase()  
unique(tags(bedbase)$tag)  
tags(bedbase, "bedset")
```

# Index

[.BEDbase \(BEDbase-class\), 9](#)

[bb\\_bed\\_text\\_search, 3](#)  
[bb\\_beds\\_in\\_bedset, 2](#)  
[bb\\_example, 3](#)  
[bb\\_list\\_beds, 4](#)  
[bb\\_list\\_bedsets, 5](#)  
[bb\\_metadata, 5](#)  
[bb\\_save, 6](#)  
[bb\\_to\\_granges, 7](#)  
[bb\\_to\\_grangeslist, 8](#)  
[BEDbase, 8](#)  
[BEDbase-class, 9](#)  
[BiocFileCache\(\), 12](#)

[getCache \(BEDbase\), 8](#)  
[getCache, BEDbase-method, 10](#)  
[GRanges, 7](#)  
[GRangesList, 8](#)

[operations, BEDbase-method, 10](#)

[schemas, BEDbase-method, 11](#)  
[setCache \(BEDbase\), 8](#)  
[setCache, BEDbase-method, 12](#)

[tags, BEDbase-method, 12](#)  
[tibble, 2–5](#)