# Package: cpp11bigwig (via r-universe)

October 28, 2026

**Type** Package

**Title** Read bigWig and bigBed Files

**Version** 0.3.1

**Description** Read bigWig and bigBed files using ``libBigWig'' <https://github.com/dpryan79/libBigWig>. Provides lightweight access to the binary bigWig and bigBed formats developed by the UCSC Genome Browser group.

**License** MIT + file LICENSE

**URL** https://rnabioco.github.io/cpp11bigwig/, https://github.com/rnabioco/cpp11bigwig

**BugReports** https://github.com/rnabioco/cpp11bigwig/issues

**Imports** GenomicRanges, IRanges, S4Vectors, tibble

**Suggests** testthat (>= 3.0.0)

**LinkingTo** cpp11

**SystemRequirements** libcurl (optional, for remote file access): libcurl-devel (rpm) or libcurl4-openssl-dev (deb)

**Config/Needs/website** rnabioco/rbitemplate

**Config/testthat/edition** 3

**Encoding** UTF-8

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** yes

**Author** Jay Hesselberth [aut, cre], RNA Bioscience Initiative [fnd, cph], Devon Ryan [cph]

**Maintainer** Jay Hesselberth <jay.hesselberth@gmail.com>

**Config/pak/sysreqs** libssl-dev

**Repository** https://cran.r-universe.dev

**Date/Publication** 2026-06-26 08:00:08 UTC

**RemoteUrl** https://github.com/cran/cpp11bigwig

**RemoteRef** HEAD

**RemoteSha** 83ac26764ba81e256b61224b32c7caa390391bb4

# Contents

bigbed_info . . . . . 2
bigwig_info . . . . . 3
read_bigbed . . . . . 3
read_bigwig . . . . . 5

**Index** **7**

---

`bigbed_info` *Report header metadata for a bigBed file.*

---

## Description

Reads the bigBed header without loading any intervals. This is useful for identifying the BED variant a file holds before reading it: a genuine BED12 has `defined_field_count == 12`, whereas a `bed9+3` file (9 standard BED columns plus 3 custom fields) has `defined_field_count == 9` and `field_count == 12`.

## Usage

```
bigbed_info(bbfile)
```

## Arguments

| | |
|---|---|
| `bbfile` | path or URL for a bigBed file. Remote files (`http://`, `https://`, `ftp://`) are supported when the package was installed with libcurl available. |

## Value

A named list with elements `version`, `n_chroms`, `field_count`, `defined_field_count`, `n_bases_covered`, and `autosql` (the embedded autoSql schema string, or `""` when the file has none).

## See Also

`read_bigbed()`, `bigwig_info()`

## Examples

```
bb <- system.file("extdata", "test.bb", package = "cpp11bigwig")

info <- bigbed_info(bb)
info$defined_field_count
```

## bigwig_info *Report header metadata and summary statistics for a bigWig file.*

### Description

Reads the bigWig header without loading any intervals. The summary statistics (`min`, `max`, `mean`, `std`) are the file-level values stored in the header and computed over all covered bases.

### Usage

```
bigwig_info(bwfile)
```

### Arguments

| | |
|---|---|
| `bwfile` | path or URL for a bigWig file. Remote files (`http://`, `https://`, `ftp://`) are supported when the package was installed with libcurl available. |

### Value

A named list with elements `version`, `n_levels`, `n_chroms`, `n_bases_covered`, `min`, `max`, `mean`, and `std`.

### See Also

`read_bigwig()`, `bigbed_info()`

### Examples

```
bw <- system.file("extdata", "test.bw", package = "cpp11bigwig")

bigwig_info(bw)
```

## read_bigbed *Read data from bigBed files.*

### Description

Columns are automatically typed based on the autoSql schema embedded in the bigBed file. Integer types (`uint`, `int`) become R integers, floating point types (`float`, `double`) become R doubles, and all other types (including array types like `int[blockCount]`) remain as character strings.

### Usage

```
read_bigbed(bbfile, chrom = NULL, start = NULL, end = NULL)
```

## Arguments

| | |
|---|---|
| bbfile | path or URL for a bigBed file. Remote files (http://, https://, ftp://) are supported when the package was installed with libcurl available. |
| chrom | chromosome(s) to read. Either a character vector of chromosome names, or a GenomicRanges::GRanges of query regions (in which case start/end are ignored). As with read_bigwig(), GRanges 1-based coordinates are converted to bigBed's 0-based half-open coordinates. |
| start | start position(s) for data. May be a vector describing several ranges, recycled against chrom/end. |
| end | end position(s) for data. May be a vector describing several ranges, recycled against chrom/start. |

## Details

When a bigBed file has no embedded autoSql schema (for example one produced by bedToBigBed without -as), columns are still recovered using the standard BED field names (name, score, strand, thickStart, thickEnd, itemRgb, blockCount, blockSizes, blockStarts) derived from the file's field counts. Any additional (bedN+) fields beyond the standard BED columns are returned as generic fieldN character columns. Because those names are inferred rather than declared by the file, a message() is emitted in this case; silence it with base::suppressMessages().

## Value

tibble

## See Also

https://github.com/dpryan79/libBigWig

https://github.com/brentp/bw-python

## Examples

```
bb <- system.file("extdata", "test.bb", package = "cpp11bigwig")

read_bigbed(bb)

read_bigbed(bb, chrom = "chr10")

# query several chromosomes in one call
read_bigbed(bb, chrom = c("chr1", "chr10"))

# restrict each query to a window
read_bigbed(bb, chrom = c("chr1", "chr10"), start = c(0, 0), end = c(5e6, 5e6))

# pass a GRanges of regions; 1-based coords are converted automatically
gr <- GenomicRanges::GRanges(
  c("chr1", "chr10"),
  IRanges::IRanges(start = 1, width = 1e7)
)
```

```
read_bigbed(bb, chrom = gr)
```

---

`read_bigwig` *Read data from bigWig files.*

---

## Description

Read data from bigWig files.

## Usage

```
read_bigwig(
  bwfile,
  chrom = NULL,
  start = NULL,
  end = NULL,
  as = NULL,
  fill = 0
)
```

## Arguments

| | |
|---|---|
| `bwfile` | path or URL for a bigWig file. Remote files (`http://`, `https://`, `ftp://`) are supported when the package was installed with libcurl available. |
| `chrom` | chromosome(s) to read. Either a character vector of chromosome names, or a GenomicRanges::GRanges of query regions (in which case `start`/`end` are ignored; see Details). |
| `start` | start position(s) for data. May be a vector, recycled against `chrom`/`end` to describe several ranges. |
| `end` | end position(s) for data. May be a vector, recycled against `chrom`/`start` to describe several ranges. |
| `as` | return data as a specific type. One of `"tbl"` (the default tibble), `"GRanges"`, or `"Rle"`. `"Rle"` returns a per-base run-length-encoded vector spanning the requested range (see Details). |
| `fill` | value used for bases with no data when `as = "Rle"`. Defaults to `0` (the convention for coverage); use `NA` to mark uncovered bases as missing. Ignored for other `as` values. |

## Details

Multiple ranges can be queried in one call by passing equal-length (or length-1, recycled) `chrom`, `start`, and `end` vectors, where range `i` is `(chrom[i], start[i], end[i])`. Alternatively, pass a GenomicRanges::GRanges as `chrom`; its regions are used directly. Because `GRanges` is 1-based and inclusive while bigWig is 0-based and half-open, a region is converted as `start(gr) - 1` to `end(gr)`.

When `as = "Rle"`, the result is an S4Vectors::Rle whose expanded length equals the queried range, i.e. `end - start` when both are supplied, otherwise the extent of the returned data for each chromosome. Bases with no data in the file are set to `fill`. bigWig coordinates are 0-based and half-open, so element `i` corresponds to genomic position `start + i - 1`. A single-range query returns a bare `Rle`; a multi-range (or multi-chromosome) query returns a named IRanges::RleList with one element per range.

## Value

A `tibble`, `GRanges`, or `Rle`/`RleList` depending on `as`.

## See Also

https://github.com/dpryan79/libBigWig

https://github.com/brentp/bw-python

## Examples

```
bw <- system.file("extdata", "test.bw", package = "cpp11bigwig")

read_bigwig(bw)

read_bigwig(bw, chrom = "10")

read_bigwig(bw, chrom = "1", start = 100, end = 130)

read_bigwig(bw, as = "GRanges")

read_bigwig(bw, chrom = "1", start = 100, end = 130, as = "Rle")

# query several ranges in one call with equal-length vectors
read_bigwig(bw, chrom = c("1", "10"), start = c(0, 0), end = c(50, 50))

# multiple windows on the same chromosome (chrom recycles)
read_bigwig(bw, chrom = "1", start = c(0, 100), end = c(50, 130))

# a multi-range "Rle" query returns a named RleList, one element per range
read_bigwig(bw, chrom = "1", start = c(0, 100), end = c(50, 130), as = "Rle")

# pass a GRanges of regions; 1-based coords are converted automatically
gr <- GenomicRanges::GRanges(
  c("1", "10"),
  IRanges::IRanges(start = c(1, 1), end = c(50, 50))
)
read_bigwig(bw, chrom = gr)
```

# Index

base::suppressMessages(), *4*
bigbed_info, 2
bigbed_info(), *3*
bigwig_info, 3
bigwig_info(), *2*

GenomicRanges::GRanges, *4, 5*

IRanges::RleList, *6*

read_bigbed, 3
read_bigbed(), *2*
read_bigwig, 5
read_bigwig(), *3, 4*

S4Vectors::Rle, *6*