

Package: Rtwobitlib (via r-universe)

May 10, 2026

Title '2bit' 'C' Library

Description A trimmed down copy of the ``kent-core source tree" turned into a 'C' library for manipulation of '.2bit' files. See <https://genome.ucsc.edu/FAQ/FAQformat.html#format7> for a quick overview of the '2bit' format. The ``kent-core source tree" can be found here: <https://github.com/ucscGenomeBrowser/kent-core/>. Only the '.c' and '.h' files from the source tree that are related to manipulation of '.2bit' files were kept. Note that the package is primarily useful to developers of other R packages who wish to use the '2bit' 'C' library in their own 'C'/'C++' code.

URL <https://github.com/hpages/Rtwobitlib>

BugReports <https://github.com/hpages/Rtwobitlib/issues>

Version 0.3.10

License MIT + file LICENSE

Encoding UTF-8

Imports tools

Suggests testthat, knitr, rmarkdown

SystemRequirements GNU make

VignetteBuilder knitr

Config/pak/sysreqs make

Repository <https://hpages.r-universe.dev>

Date/Publication 2025-05-05 22:58:33 UTC

RemoteUrl <https://github.com/hpages/rtwobitlib>

RemoteRef HEAD

RemoteSha 6cc5aa293dcf2e6c50ebdab98ac23c01acffe09

Contents

| | |
|----------------------------|---|
| pkgconfig | 2 |
| twobit_roundtrip | 2 |
| twobit_seqstats | 4 |

| | |
|-----------|---|
| pkgconfig | <i>Compiler configuration arguments for use of Rtwobitlib</i> |
|-----------|---|

Description

The `pkgconfig` function prints values for `PKG_LIBS` and `PKG_CPPFLAGS` variables for use in Makevars files. It is not meant for the end user. See `vignette("Rtwobitlib")` for more information.

Usage

```
pkgconfig(opt=c("PKG_LIBS", "PKG_CPPFLAGS"))
```

Arguments

`opt` Either `"PKG_LIBS"` or `"PKG_CPPFLAGS"`

Value

The function prints the `PKG_LIBS` or `PKG_CPPFLAGS` value and returns an invisible `NULL`.

Examples

```
pkgconfig("PKG_LIBS")
pkgconfig("PKG_CPPFLAGS")
```

| | |
|------------------|--------------------------------|
| twobit_roundtrip | <i>Read/write a .2bit file</i> |
|------------------|--------------------------------|

Description

Read/write a character vector representing DNA sequences from/to a file in *2bit* format.

Usage

```
twobit_read(filepath)

twobit_write(x, filepath, use.long=FALSE, skip.dups=FALSE)
```

Arguments

| | |
|-----------|--|
| filepath | A single string (character vector of length 1) containing a path to the file to read or write. |
| x | A named character vector representing DNA sequences. The names on the vector should be unique and the sequences should only contain A's, C's, G's, T's, or N's, in uppercase or lowercase. |
| use.long | By default the <i>2bit</i> format cannot store more than 4Gb of sequence data in total. Set <code>use.long</code> to TRUE if your sequence data is bigger than that. |
| skip.dups | By default duplicate sequence names are an error. By setting <code>skip.dups</code> to FALSE, sequences with a duplicated name will be skipped with a warning. |

Value

For `twobit_read()`: A named character vector containing the DNA sequences loaded from the file.

For `twobit_write()`: `filepath` returned invisibly.

References

A quick overview of the *2bit* format: <https://genome.ucsc.edu/FAQ/FAQformat.html#format7>

See Also

[twobit_seqstats](#) and [twobit_seqlengths](#) to extract the sequence lengths and letter counts from a *.2bit* file.

Examples

```
## Read:
inpath <- system.file(package="Rtwobitlib", "extdata", "sacCer2.2bit")
dna <- twobit_read(inpath)
names(dna)
nchar(dna)

## Write:
outpath <- twobit_write(dna, tempfile())

## Sanity checks:
library(tools)
stopifnot(md5sum(inpath) == md5sum(outpath))
stopifnot(identical(nchar(dna), twobit_seqlengths(inpath)))
```

| | |
|-----------------|---|
| twobit_seqstats | <i>Extract sequence lengths and letter counts from a .2bit file</i> |
|-----------------|---|

Description

Extract the lengths and letter counts of the DNA sequences stored in a .2bit file.

Usage

```
twobit_seqstats(filepath)
```

```
twobit_seqlengths(filepath)
```

Arguments

filepath A single string (character vector of length 1) containing a path to a .2bit file.

Details

twobit_seqlengths(filepath) is a shortcut for twobit_seqstats(filepath)[, "seqlengths"] that is also a much more efficient way to get the sequence lengths as it does not need to load the sequence data in memory.

Value

For twobit_seqstats(): An integer matrix with one row per sequence in the .2bit file and 6 columns. The rownames on the matrix are the sequence names and the colnames are: seqlengths, A, C, G, T, N. Columns A, C, G, T, and N contain the letter count for each sequence.

For twobit_seqlengths(): A named integer vector where the names are the sequence names and the values the corresponding lengths.

References

A quick overview of the *2bit* format: <https://genome.ucsc.edu/FAQ/FAQformat.html#format7>

See Also

[twobit_read](#) and [twobit_write](#) to read/write a character vector representing DNA sequences from/to a file in *2bit* format.

Examples

```
filepath <- system.file(package="Rtwobitlib", "extdata", "sacCer2.2bit")
```

```
twobit_seqstats(filepath)
```

```
twobit_seqlengths(filepath)
```

```
## Sanity checks:
sacCer2_seqstats <- twobit_seqstats(filepath)
stopifnot(
  identical(sacCer2_seqstats[ , 1], twobit_seqlengths(filepath)),
  all.equal(rowSums(sacCer2_seqstats[ , -1]), sacCer2_seqstats[ , 1])
)
```

Index

* **manip**

pkgconfig, [2](#)

twobit_roundtrip, [2](#)

twobit_seqstats, [4](#)

pkgconfig, [2](#)

twobit_read, [4](#)

twobit_read(twobit_roundtrip), [2](#)

twobit_roundtrip, [2](#)

twobit_seqlengths, [3](#)

twobit_seqlengths(twobit_seqstats), [4](#)

twobit_seqstats, [3](#), [4](#)

twobit_write, [4](#)

twobit_write(twobit_roundtrip), [2](#)