

# Package ‘brotli’

March 18, 2025

**Type** Package

**Title** A Compression Format Optimized for the Web

**Version** 1.3.2

**Description** A lossless compressed data format that uses a combination of the LZ77 algorithm and Huffman coding <<https://www.rfc-editor.org/rfc/rfc7932>>. Brotli is similar in speed to deflate (gzip) but offers more dense compression.

**License** MIT + file LICENSE

**URL** <https://jeroen.r-universe.dev/brotli>

**BugReports** <https://github.com/jeroen/brotli/issues>

**VignetteBuilder** knitr, R.rsp

**Suggests** spelling, knitr, R.rsp, microbenchmark, rmarkdown, ggplot2

**RoxygenNote** 6.0.1

**Language** en-US

**NeedsCompilation** yes

**Author** Jeroen Ooms [aut, cre] (<<https://orcid.org/0000-0002-4035-0289>>),  
Google, Inc [aut, cph] (Brotli C++ library)

**Maintainer** Jeroen Ooms <jeroenooms@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-03-18 11:20:02 UTC

## Contents

brotli . . . . .	2
<b>Index</b>	<b>4</b>

---

`brotli`*Brotli Compression*

---

## Description

Brotli is a compression algorithm optimized for the web, in particular small text documents.

## Usage

```
brotli_compress(buf, quality = 11, window = 22)
```

```
brotli_decompress(buf)
```

## Arguments

<code>buf</code>	raw vector with data to compress/decompress
<code>quality</code>	value between 0 and 11
<code>window</code>	log of window size

## Details

Brotli decompression is at least as fast as for gzip while significantly improving the compression ratio. The price we pay is that compression is much slower than gzip. Brotli is therefore most effective for serving static content such as fonts and html pages.

For binary (non-text) data, the compression ratio of Brotli usually does not beat bz2 or xz (lzma), however decompression for these algorithms is too slow for browsers in e.g. mobile devices.

## References

J. Alakuijala and Z. Szabadka (July 2016). *Brotli Compressed Data Format*. IETF Internet Draft <https://www.rfc-editor.org/rfc/rfc7932>.

## See Also

[memCompress](#)

## Examples

```
# Simple example
myfile <- file.path(R.home(), "COPYING")
x <- readBin(myfile, raw(), file.info(myfile)$size)
y <- brotli_compress(x)
stopifnot(identical(x, brotli_decompress(y)))

# Compare to other algorithms
length(x)
length(brotli_compress(x))
length(memCompress(x, "gzip"))
```

```
length(memCompress(x, "bzip2"))  
length(memCompress(x, "xz"))
```

# Index

`brotn`, [2](#)

`brotn_compress (brotn)`, [2](#)

`brotn_decompress (brotn)`, [2](#)

`memCompress`, [2](#)