

Package: geobounds (via r-universe)

October 17, 2026

Title Download Administrative Boundary Data from 'geoBoundaries'

Version 1.0.1

Description Provides tools to download individual country boundaries and global composite boundaries from 'geoBoundaries' <<https://www.geoboundaries.org/>> across multiple administrative ('ADM') levels. Returns boundaries as 'sf' objects for mapping and spatial analysis. Runfola et al. (2020) <[doi:10.1371/journal.pone.0231866](https://doi.org/10.1371/journal.pone.0231866)> describe the underlying database.

License MIT + file LICENSE

URL <https://dieghernan.github.io/geobounds/>,
<https://github.com/dieghernan/geobounds>

BugReports <https://github.com/dieghernan/geobounds/issues>

Depends R (>= 4.1.0)

Imports cli, countrycode, dplyr, httr2 (>= 1.0.0), sf, tools, utils

Suggests ggplot2, jsonlite, knitr, quarto, testthat (>= 3.0.0),
tibble, withr

VignetteBuilder quarto

Config/Needs/website dieghernan/gitdevr, xfun, devtools, remotes,
reactable

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

Config/testthat/parallel true

Copyright Package code is MIT licensed. See file COPYRIGHTS for data and figure attribution.

Encoding UTF-8

Language en-US

X-schema.org-keywords administrative-boundaries, api, geoboundaries,
r, r-package, spatial-data, cran, cran-r

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

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

Date/Publication 2026-10-17 09:51:47 UTC

RemoteUrl <https://github.com/dieghernan/geobounds>

RemoteRef HEAD

RemoteSha 62afc1f033355579f0a247681e7f47ceb95bfe37

Contents

gb_clear_cache	2
gb_detect_cache_dir	3
gb_get	4
gb_get_adm	6
gb_get_max_adm_lvl	9
gb_get_metadata	10
gb_get_world	12
gb_set_cache_dir	13
Index	16

gb_clear_cache	<i>Clear the R</i> hrefhttps://CRAN.R-project.org/package=geobounds geobounds cache directory
----------------	--

Description

Use this function with caution. It clears cached archives and configuration by deleting the **geobounds** configuration directory returned by `tools::R_user_dir()`, deleting the active cache directory and clearing the GEOBOUNDS_CACHE_DIR environment variable. See `base::Sys.getenv()`.

Usage

```
gb_clear_cache(config = FALSE, cached_data = TRUE, quiet = TRUE)
```

Arguments

- `config` A logical value. If TRUE, delete the **geobounds** configuration directory.
- `cached_data` A logical value. If TRUE, delete the active cache directory and all its contents.
- `quiet` A logical value. If TRUE, suppress informational messages.

Details

This reset restores the cache state of a fresh **geobounds** installation. For safety, the function refuses to recursively delete a cache path that contains the home, working, temporary or package configuration directory. Use a dedicated cache subdirectory. Inspect the active directory with `gb_detect_cache_dir()` before clearing it. Use `gb_set_cache_dir()` to configure a cache directory again.

Value

`NULL`, returned invisibly. This function is called for its side effects.

See Also

Cache management functions: `gb_detect_cache_dir()`, `gb_set_cache_dir()`

Examples

```
# Caution: this may modify your current state.

## Not run:
my_cache <- gb_detect_cache_dir()
# Set an example cache directory.
ex <- file.path(tempdir(), "example", "cache")
gb_set_cache_dir(ex, quiet = TRUE)

gb_clear_cache(quiet = FALSE)

# Restore the initial cache.
gb_set_cache_dir(my_cache)
identical(my_cache, gb_detect_cache_dir())

## End(Not run)
```

<code>gb_detect_cache_dir</code>	<i>Detect the R</i> hrefhttps://CRAN.R-project.org/package=geobounds geobounds <i>cache directory</i>
----------------------------------	--

Description

Detects the active cache directory. See `gb_set_cache_dir()`.

Usage

```
gb_detect_cache_dir(x = NULL)
```

Arguments

<code>x</code>	An object. Ignored.
----------------	---------------------

Value

A [character](#) scalar containing the path to the active cache directory. The path is also printed as a clickable message. See [cli::inline-markup](#) from [cli](#).

See Also

Cache management functions: [gb_clear_cache\(\)](#), [gb_set_cache_dir\(\)](#)

Examples

```
gb_detect_cache_dir()
```

gb_get

*Download individual country boundaries from **geoBoundaries***

Description

Returns individual country boundaries that reflect how countries represent their own boundaries, without special identification of disputed areas.

Use [gb_get_world\(\)](#) for global composite boundaries that standardize disputed areas and fill gaps between borders.

Always acknowledge **geoBoundaries** and follow the boundary's license. See `vignette("metadata", package = "geobounds")` for sources and licensing.

The wrappers [gb_get_adm0\(\)](#), [gb_get_adm1\(\)](#), [gb_get_adm2\(\)](#), [gb_get_adm3\(\)](#), [gb_get_adm4\(\)](#) and [gb_get_adm5\(\)](#) are also available for requesting a single ADM level.

Usage

```
gb_get(
  country,
  adm_lvl = "adm0",
  simplified = FALSE,
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),
  quiet = TRUE,
  overwrite = FALSE,
  cache_dir = NULL
)
```

Arguments

country	A character vector of country names or ISO 3166-1 alpha-3 country codes. Use "all" to return boundaries for all countries. See also countrycode::countrycode() from countrycode .
---------	---

adm_lvl	ADM level. Accepted values are "all" (all available boundaries) or the ADM level ("adm0" is the country boundary, "adm1" is the first level of subnational boundaries, "adm2" is the second level and so on). Uppercase versions ("ADM1") and level numbers (0, 1, 2, 3, 4, 5) are also accepted, including numbers supplied as text (for example, "1").
simplified	A logical value. If TRUE, read simplified boundaries that are faster to render. Both options download the complete ZIP archive.
release_type	A character string, one of "gbOpen" (the default), "gbHumanitarian" or "gbAuthoritative". Selects a product, not a dataset version. "gbAuthoritative" is restricted to non-commercial use. See <code>vignette("metadata", package = "geobounds")</code> for product sources and licenses.
quiet	A logical value. If TRUE, suppress informational messages.
overwrite	A logical value. If TRUE, force a fresh download of the source .zip archive.
cache_dir	A path to a cache directory. If NULL, use the configured directory, or a temporary directory when none is configured. See <code>gb_set_cache_dir()</code> .

Details

These are terrestrial boundaries, not maritime boundaries. For product selection and mapping workflows, see `vignette("geobounds", package = "geobounds")`.

This function uses current API metadata and reuses cached archives without checking for upstream changes. Historical versions cannot be selected. See `vignette("metadata", package = "geobounds")` for cache limitations and provenance.

Value

An `sf` object from **sf** containing the requested boundaries. Returns `NULL` if no boundaries match the request or the downloads return no geometries.

Source

geoBoundaries API.

References

Runfola et al. (2020) "geoBoundaries: A global database of political administrative boundaries." *PLOS ONE*, **15**(4), e0231866. doi:[10.1371/journal.pone.0231866](https://doi.org/10.1371/journal.pone.0231866).

See Also

`gb_get_metadata()` inspects boundary metadata and licensing. `gb_get_max_adm_lvl()` checks the ADM levels available for individual country boundaries. `gb_set_cache_dir()` configures where downloaded archives are cached.

Boundary download functions: `gb_get_adm`, `gb_get_world()`

Examples

```
# Map ADM2 in Sri Lanka.
sri_lanka <- gb_get(
  "Sri Lanka",
  adm_lvl = 2,
  simplified = TRUE
)

sri_lanka

library(ggplot2)
ggplot(sri_lanka) +
  geom_sf() +
  labs(
    caption = paste(
      "Sources: geoBoundaries, OpenStreetMap and Wambacher,",
      "license: ODbL 1.0"
    )
  )
)
```

gb_get_adm

Download country boundaries for one ADM level

Description

These functions call [gb_get\(\)](#) for a single ADM level. [gb_get_adm0\(\)](#) returns country boundaries, [gb_get_adm1\(\)](#) returns first-level subnational boundaries (for example, states in the United States) and [gb_get_adm2\(\)](#) returns second-level subnational boundaries (for example, counties in the United States). [gb_get_adm3\(\)](#), [gb_get_adm4\(\)](#) and [gb_get_adm5\(\)](#) return third-, fourth- and fifth-level administrative boundaries, respectively.

Use [gb_get_metadata\(\)](#) to check availability for the requested country, ADM level and release type.

See [gb_get\(\)](#) for download behavior and licensing, and [vignette\("geobounds", package = "geobounds"\)](#) for worked examples.

Usage

```
gb_get_adm0(
  country,
  simplified = FALSE,
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),
  quiet = TRUE,
  overwrite = FALSE,
  cache_dir = NULL
)
```

```
gb_get_adm1(  
    country,  
    simplified = FALSE,  
    release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),  
    quiet = TRUE,  
    overwrite = FALSE,  
    cache_dir = NULL  
)  
  
gb_get_adm2(  
    country,  
    simplified = FALSE,  
    release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),  
    quiet = TRUE,  
    overwrite = FALSE,  
    cache_dir = NULL  
)  
  
gb_get_adm3(  
    country,  
    simplified = FALSE,  
    release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),  
    quiet = TRUE,  
    overwrite = FALSE,  
    cache_dir = NULL  
)  
  
gb_get_adm4(  
    country,  
    simplified = FALSE,  
    release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),  
    quiet = TRUE,  
    overwrite = FALSE,  
    cache_dir = NULL  
)  
  
gb_get_adm5(  
    country,  
    simplified = FALSE,  
    release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative"),  
    quiet = TRUE,  
    overwrite = FALSE,  
    cache_dir = NULL  
)
```

Arguments

country	A character vector of country names or ISO 3166-1 alpha-3 country codes. Use "all" to return boundaries for all countries. See also countrycode::countrycode() from countrycode .
simplified	A logical value. If TRUE, read simplified boundaries that are faster to render. Both options download the complete ZIP archive.
release_type	A character string, one of "gbOpen" (the default), "gbHumanitarian" or "gbAuthoritative". Selects a product, not a dataset version. "gbAuthoritative" is restricted to non-commercial use. See <code>vignette("metadata", package = "geobounds")</code> for product sources and licenses.
quiet	A logical value. If TRUE, suppress informational messages.
overwrite	A logical value. If TRUE, force a fresh download of the source .zip archive.
cache_dir	A path to a cache directory. If NULL, use the configured directory, or a temporary directory when none is configured. See gb_set_cache_dir() .

Value

An [sf](#) object from **sf** containing the requested boundaries. Returns `NULL` if no boundaries match the request or the downloads return no geometries.

Source

geoBoundaries API.

References

Runfola et al. (2020) "geoBoundaries: A global database of political administrative boundaries." *PLOS ONE*, **15**(4), e0231866. doi:[10.1371/journal.pone.0231866](https://doi.org/10.1371/journal.pone.0231866).

See Also

[gb_get_metadata\(\)](#) lists the layers available for a country and release type. [gb_get_max_adm_lvl\(\)](#) summarizes the highest available ADM level.

Boundary download functions: [gb_get\(\)](#), [gb_get_world\(\)](#)

Examples

```
lev2 <- gb_get_adm2(
  c("Italia", "Suiza", "Austria"),
  simplified = TRUE
)

library(ggplot2)

ggplot(lev2) +
  geom_sf(aes(fill = shapeGroup)) +
  labs(
```

```

title = "Second-level administrative boundaries",
subtitle = "Selected countries",
caption = paste(
  "Sources: geoBoundaries and the original boundary providers,",
  "check gb_get_metadata() for licenses"
)
)

```

gb_get_max_adm_lvl *Find the highest available ADM level*

Description

Returns a summary of selected country codes and their highest available ADM level in **geoBoundaries**. ADM0 represents the country boundary. Use [gb_get_metadata\(\)](#) to list available layers in the selected release type.

Usage

```

gb_get_max_adm_lvl(
  country = "all",
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative")
)

```

Arguments

country	A character vector of country names or ISO 3166-1 alpha-3 country codes. Use "all" to return boundaries for all countries. See also countrycode::countrycode() from countrycode .
release_type	A character string, one of "gbOpen" (the default), "gbHumanitarian" or "gbAuthoritative". Selects a product, not a dataset version. "gbAuthoritative" is restricted to non-commercial use. See vignette("metadata", package = "geobounds") for product sources and licenses.

Value

A [tibble](#) from **tibble** with one row per matching country. The boundaryISO column contains ISO 3166-1 alpha-3 country codes and maxBoundaryType contains the highest available ADM level as an integer.

Source

geoBoundaries API.

References

Runfola et al. (2020) "geoBoundaries: A global database of political administrative boundaries." *PLOS ONE*, **15**(4), e0231866. doi:[10.1371/journal.pone.0231866](https://doi.org/10.1371/journal.pone.0231866).

See Also

[gb_get\(\)](#) downloads boundaries for the available ADM levels. The [ADM wrappers](#) request a single administrative level. See `vignette("metadata", package = "geobounds")` for availability and metadata fields.

Metadata and licensing functions: [gb_get_metadata\(\)](#)

Examples

```
all <- gb_get_max_adm_lvl()
library(dplyr)

# Countries whose highest available level is ADM1.
all |>
  filter(maxBoundaryType == 1)

# Countries whose highest available level is ADM4.
all |>
  filter(maxBoundaryType == 4)
```

gb_get_metadata	Retrieve boundary metadata from geoBoundaries
-----------------	--

Description

Returns boundary metadata from the **geoBoundaries API**.

Usage

```
gb_get_metadata(
  country = "all",
  adm_lvl = "all",
  release_type = c("gbOpen", "gbHumanitarian", "gbAuthoritative")
)
```

Arguments

country	A character vector of country names or ISO 3166-1 alpha-3 country codes. Use "all" to return boundaries for all countries. See also countrycode::countrycode() from countrycode .
---------	--

adm_lvl	ADM level. Accepted values are "all" (all available boundaries) or the ADM level ("adm0" is the country boundary, "adm1" is the first level of subnational boundaries, "adm2" is the second level and so on). Uppercase versions ("ADM1") and level numbers (0, 1, 2, 3, 4, 5) are also accepted, including numbers supplied as text (for example, "1").
release_type	A character string, one of "gbOpen" (the default), "gbHumanitarian" or "gbAuthoritative". Selects a product, not a dataset version. "gbAuthoritative" is restricted to non-commercial use. See vignette("metadata", package = "geobounds") for product sources and licenses.

Details

This function queries `/api/current` and does not select historical versions. See `vignette("metadata", package = "geobounds")` for column definitions, type conversions, availability, licensing and reproducible workflows.

Value

A [tibble](#) from [tibble](#) with one row per matching boundary. Numeric statistics are converted to numeric values, `sourceDataUpdateDate` to [POSIXlt](#) in GMT and `buildDate` to [Date](#). Literal "nan" values become NA.

Source

[geoBoundaries API](#).

References

Runfola et al. (2020) "geoBoundaries: A global database of political administrative boundaries." *PLOS ONE*, **15**(4), e0231866. doi:[10.1371/journal.pone.0231866](https://doi.org/10.1371/journal.pone.0231866).

See Also

[gb_get\(\)](#) downloads the boundaries described by the metadata. The [ADM wrappers](#) request a single administrative level. This metadata describes individual country boundaries, not the CGAZ layers returned by [gb_get_world\(\)](#).

Metadata and licensing functions: [gb_get_max_adm_lvl\(\)](#)

Examples

```
# Get boundary metadata for ADM4.

library(dplyr)

gb_get_metadata(adm_lvl = "ADM4") |>
  glimpse()
```

gb_get_world

*Download global composite boundaries from **geoBoundaries*****Description**

Returns global composite boundaries for the requested ADM level. Boundaries are clipped to international borders, with gaps between borders filled.

Always acknowledge **geoBoundaries** and follow the terms in the CGAZ archive.

Usage

```
gb_get_world(
  country = "all",
  adm_lvl = "adm0",
  quiet = TRUE,
  overwrite = FALSE,
  cache_dir = NULL
)
```

Arguments

country	A character vector of country names or ISO 3166-1 alpha-3 codes. Use "all" to return all countries. The complete global layer is downloaded and read before filtering, so country selection does not reduce the initial download size or the memory needed to read it.
adm_lvl	ADM level. Accepted values are levels 0, 1 and 2 ("adm0" is the country boundary, "adm1" is the first level of subnational boundaries and "adm2" is the second level). Uppercase versions ("ADM1") and level numbers (0, 1, 2) are also accepted, including numbers supplied as text (for example, "1").
quiet	A logical value. If TRUE, suppress informational messages.
overwrite	A logical value. If TRUE, force a fresh download of the source .zip archive.
cache_dir	A path to a cache directory. If NULL, use the configured directory, or a temporary directory when none is configured. See gb_set_cache_dir() .

Details

CGAZ uses extensive simplification and standardizes disputed areas. See `vignette("geobounds", package = "geobounds")` for product differences. Archives come from the repository's main branch and historical versions cannot be selected. See `vignette("metadata", package = "geobounds")` for provenance.

Value

An `sf` object from `sf` containing the requested boundaries. Returns `NULL` if no boundaries match the request or the downloads return no geometries.

Source

- **geoBoundaries** global downloads: <https://www.geoboundaries.org/globalDownloads.html>.
- CGAZ release files: <https://github.com/wmgeolab/geoBoundaries/tree/main/releaseData/CGAZ>.

References

Runfola et al. (2020) "geoBoundaries: A global database of political administrative boundaries." *PLOS ONE*, **15**(4), e0231866. doi:[10.1371/journal.pone.0231866](https://doi.org/10.1371/journal.pone.0231866).

See Also

[gb_get\(\)](#) downloads individual country boundaries instead of CGAZ layers. [gb_set_cache_dir\(\)](#) configures where downloaded archives are cached. See [vignette\("geobounds", package = "geobounds"\)](#) for a comparison of individual country boundaries and global composites.

Boundary download functions: [gb_get\(\)](#), [gb_get_adm](#)

Examples

```
# This download may take some time.
## Not run:
world <- gb_get_world()

library(ggplot2)

ggplot(world) +
  geom_sf() +
  coord_sf(expand = FALSE) +
  labs(caption = "Source: geoBoundaries (CGAZ)")

## End(Not run)
```

gb_set_cache_dir	Set the R href geobounds cache directory
------------------	---

Description

Sets the active cache directory and optionally saves it for future sessions. Use [gb_detect_cache_dir\(\)](#) to find the active cache directory.

Usage

```
gb_set_cache_dir(
  cache_dir = NULL,
  overwrite = FALSE,
  install = FALSE,
  quiet = FALSE
)
```

Arguments

cache_dir	A path to a cache directory. If NULL, the function stores cached archives in a temporary directory. See base::tempdir() .
overwrite	A logical value. If TRUE, replace a cache directory already saved in the configuration file.
install	A logical value. If TRUE, save the cache directory for use in future sessions. Defaults to FALSE. If cache_dir is NULL, this parameter is set to FALSE automatically.
quiet	A logical value. If TRUE, suppress informational messages.

Details

By default, when no cache_dir is set, **geobounds** uses a directory inside [base::tempdir\(\)](#). Cached archives in this directory are removed when the R session ends. To reuse a cache directory across R sessions, use `gb_set_cache_dir(cache_dir = "a/path/here", install = TRUE)`. This saves the directory in the user configuration path returned by [tools::R_user_dir\(\)](#).

Cached archives are reused without checking for upstream changes. See `vignette("metadata", package = "geobounds")` for cache limitations and provenance.

Value

A [character](#) scalar containing the path to the cache directory, returned invisibly.

Cache strategies

- For occasional use, use the default temporary cache directory.
- Set the cache directory for the current session with `gb_set_cache_dir(cache_dir = "a/path/here")`.
- Save a persistent cache directory for future R sessions with `gb_set_cache_dir(cache_dir = "a/path/here", install = TRUE)`.
- Set the cache directory for an individual download with the cache_dir argument. See [gb_get\(\)](#).

See Also

[gb_get\(\)](#) and [gb_get_world\(\)](#) use the configured cache directory for downloaded boundary archives. [tools::R_user_dir\(\)](#) identifies standard locations for user-specific files.

Cache management functions: [gb_clear_cache\(\)](#), [gb_detect_cache_dir\(\)](#)

Examples

```
# Caution: this may modify your current state.

## Not run:
my_cache <- gb_detect_cache_dir()

# Set an example cache directory.
ex <- file.path(tempdir(), "example", "cachenew")
gb_set_cache_dir(ex)

gb_detect_cache_dir()

# Restore the initial cache.
gb_set_cache_dir(my_cache)
identical(my_cache, gb_detect_cache_dir())

## End(Not run)

gb_detect_cache_dir()
```

Index

- * **api**
 - gb_get, 4
 - gb_get_adm, 6
 - gb_get_world, 12
- * **cache**
 - gb_clear_cache, 2
 - gb_detect_cache_dir, 3
 - gb_set_cache_dir, 13
- * **metadata**
 - gb_get_max_adm_lvl, 9
 - gb_get_metadata, 10

ADM wrappers, 10, 11

base::Sys.getenv(), 2

base::tempdir(), 14

character, 4, 14

cli::inline-markup, 4

countrycode::countrycode(), 4, 8–10

Date, 11

gb_clear_cache, 2

gb_clear_cache(), 4, 14

gb_detect_cache_dir, 3

gb_detect_cache_dir(), 3, 13, 14

gb_get, 4

gb_get(), 6, 8, 10, 11, 13, 14

gb_get_adm, 5, 6, 13

gb_get_adm0 (gb_get_adm), 6

gb_get_adm0(), 4, 6

gb_get_adm1 (gb_get_adm), 6

gb_get_adm1(), 4, 6

gb_get_adm2 (gb_get_adm), 6

gb_get_adm2(), 4, 6

gb_get_adm3 (gb_get_adm), 6

gb_get_adm3(), 4, 6

gb_get_adm4 (gb_get_adm), 6

gb_get_adm4(), 4, 6

gb_get_adm5 (gb_get_adm), 6

gb_get_adm5(), 4, 6

gb_get_cgaz (gb_get_world), 12

gb_get_max_adm_lvl, 9

gb_get_max_adm_lvl(), 5, 8, 11

gb_get_meta (gb_get_metadata), 10

gb_get_metadata, 10

gb_get_metadata(), 5, 6, 8–10

gb_get_world, 12

gb_get_world(), 4, 5, 8, 11, 14

gb_set_cache_dir, 13

gb_set_cache_dir(), 3–5, 8, 12, 13

NULL, 3, 5, 8, 12

POSIXlt, 11

sf, 5, 8, 12

tibble, 9, 11

tools::R_user_dir(), 2, 14