Package 'geobounds'

May 8, 2026

Title Download Map Data from 'geoBoundaries'

Version 0.1.1

Description Tools to download data from 'geoBoundaries' <https://www.geoboundaries.org/>. Several administration levels available. See Runfola, D. et al. (2020) geoBoundaries: A global database of political administrative boundaries. PLOS ONE 15(4): 1-9. <doi:10.1371/journal.pone.0231866>.

License CC BY 4.0

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

VignetteBuilder quarto

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

Config/testthat/edition 3

Config/testthat/parallel false

Copyright Attribution required. See file COPYRIGHTS for specific provisions.

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

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

NeedsCompilation no

Author Diego Hernangómez [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-8457-4658>), William and Mary geoLab [cph] (ROR: <https://ror.org/03hsf0573>)

Maintainer Diego Hernangómez <diego.hernangomezherrero@gmail.com>

Repository CRAN

Date/Publication 2026-03-24 18:10:02 UTC

Contents

`gb_clear_cache` *Clear your* **geobounds** *cache directory*

Description

Use this function with caution. This function will clear your cached data and configuration, specifically:

- Deletes the **geobounds** config directory (`tools::R_user_dir("geobounds", "config")`).
- Deletes the `cache_dir` directory.
- Deletes the values stored in `Sys.getenv("GEOBOUNDS_CACHE_DIR")`.

Usage

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

Arguments

`config`	Logical. If `TRUE`, will delete the configuration folder of **geobounds**.
`cached_data`	Logical. If `TRUE`, it will delete your `cache_dir` and all its content.
`quiet`	logical. If `TRUE` suppresses informational messages.

Details

This is a comprehensive reset function that is intended to reset your status as if you had never installed or used **geobounds**.

Value

`invisible()` This function is called for its side effects.

See Also

Other cache utilities: 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
ex <- file.path(tempdir(), "example", "cache")
gb_set_cache_dir(ex, quiet = TRUE)

gb_clear_cache(quiet = FALSE)

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

## End(Not run)
```

---

gb_detect_cache_dir *Detect cache directory for* **geobounds**

Description

Helper function to detect the current cache folder. See gb_set_cache_dir().

Usage

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

Arguments

x	Ignored.

Value

A character with the path to your cache_dir. The same path will appear also as a clickable message, see cli::inline-markup.

See Also

Other cache utilities: gb_clear_cache(), gb_set_cache_dir()

Examples

```
gb_detect_cache_dir()
```

gb_get *Get individual country files from geoBoundaries*

Description

Attribution is required for all uses of this dataset.

This function returns data of individual countries "as they would represent themselves", with no special identification of disputed areas.

If you would prefer data that explicitly includes disputed areas, please use `gb_get_world()`.

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 codes. It can be either `"all"` (which returns the data for all countries), a vector of country names, or ISO3 country codes. See also `countrycode::countrycode()`.
`adm_lvl`	Type of boundary. Accepted values are `"all"` (all available boundaries) or the ADM level (`"adm0"` is the country boundary, `"adm1"` is the first level of sub-national boundaries, `"adm2"` is the second level, and so on). Upper-case versions (`"ADM1"`) and the number of the level (`1, 2, 3, 4, 5`) are also accepted.
`simplified`	logical. Return the simplified boundary or not. The default `FALSE` uses the premier geoBoundaries release.
`release_type`	One of `"gbOpen"`, `"gbHumanitarian"`, `"gbAuthoritative"`. For most users, we suggest using `"gbOpen"` (the default), as it is CC-BY 4.0 compliant and can be used for most purposes so long as attribution is provided: • `"gbHumanitarian"` files are mirrored from UN OCHA, but may have more restrictive licensing. • `"gbAuthoritative"` files are mirrored from UN SALB, and cannot be used for commercial purposes, but are verified through in-country processes.
`quiet`	logical. If `TRUE` suppresses informational messages.
`overwrite`	logical. When set to `TRUE` it will force a fresh download of the source `.zip` file.
`cache_dir`	A path to a cache directory. If not set (the default `NULL`), the data will be stored in the default cache directory (see `gb_set_cache_dir()`). If no cache directory has been set, files will be stored in the temporary directory (see `base::tempdir()`). See caching strategies in `gb_set_cache_dir()`.

Details

Individual data files in the geoBoundaries database are governed by the license or licenses identified within the metadata for each respective boundary (see gb_get_metadata(). Users using individual boundary files from geoBoundaries should additionally ensure that they cite the sources provided in the metadata for each file. See **Examples**.

The following wrappers are also available:

- gb_get_adm0() returns the country boundary.
- gb_get_adm1() returns first-level administrative boundaries (e.g. States in the United States).
- gb_get_adm2() returns second-level administrative boundaries (e.g. Counties in the United States).
- gb_get_adm3() returns third-level administrative boundaries (e.g. towns or cities in some countries).
- gb_get_adm4() returns fourth-level administrative boundaries.
- gb_get_adm5() returns fifth-level administrative boundaries.

Value

A sf object.

Source

geoBoundaries API Service https://www.geoboundaries.org/api.html.

References

Runfola, D. et al. (2020) geoBoundaries: A global database of political administrative boundaries. *PLOS ONE 15*(4), 1-9. doi:10.1371/journal.pone.0231866.

See Also

Other API functions: gb_get_adm, gb_get_world()

Examples

```
# Map level 2 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 = "Source: www.geoboundaries.org")
```

```
# Metadata
library(dplyr)
gb_get_metadata(
  "Sri Lanka",
  adm_lvl = 2
) |>
  # Check individual license
  select(boundaryISO, boundaryType, licenseDetail, licenseSource) |>
  glimpse()
```

---

gb_get_adm *Get country files from geoBoundaries for a given administrative level*

Description

Attribution is required for all uses of this dataset.

These functions are wrappers of `gb_get()` for extracting any given administrative level:

- `gb_get_adm0()` returns the country boundary.
- `gb_get_adm1()` returns first-level administrative boundaries (e.g. States in the United States).
- `gb_get_adm2()` returns second-level administrative boundaries (e.g. Counties in the United States).
- `gb_get_adm3()` returns third-level administrative boundaries (e.g. towns or cities in some countries).
- `gb_get_adm4()` returns fourth-level administrative boundaries.
- `gb_get_adm5()` returns fifth-level administrative boundaries.

Note that not all countries have the same number of levels. Check gb_get_max_adm_lvl.

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 codes. It can be either "all" (which returns the data for all countries), a vector of country names, or ISO3 country codes. See also countrycode::countrycode().
simplified	logical. Return the simplified boundary or not. The default FALSE uses the premier geoBoundaries release.

release_type	One of "gbOpen", "gbHumanitarian", "gbAuthoritative". For most users, we suggest using "gbOpen" (the default), as it is CC-BY 4.0 compliant and can be used for most purposes so long as attribution is provided: • "gbHumanitarian" files are mirrored from UN OCHA, but may have more restrictive licensing. • "gbAuthoritative" files are mirrored from UN SALB, and cannot be used for commercial purposes, but are verified through in-country processes.
quiet	logical. If TRUE suppresses informational messages.
overwrite	logical. When set to TRUE it will force a fresh download of the source .zip file.
cache_dir	A path to a cache directory. If not set (the default NULL), the data will be stored in the default cache directory (see gb_set_cache_dir()). If no cache directory has been set, files will be stored in the temporary directory (see base::tempdir()). See caching strategies in gb_set_cache_dir().

Details

Individual data files in the geoBoundaries database are governed by the license or licenses identified within the metadata for each respective boundary (see gb_get_metadata(). Users using individual boundary files from geoBoundaries should additionally ensure that they are citing the sources provided in the metadata for each file.

Value

A sf object.

Source

geoBoundaries API Service https://www.geoboundaries.org/api.html.

References

Runfola, D. et al. (2020) geoBoundaries: A global database of political administrative boundaries. *PLOS ONE 15*(4), 1-9. doi:10.1371/journal.pone.0231866.

See Also

gb_get_max_adm_lvl().

Other API 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 administration",
    subtitle = "Selected countries",
    caption = "Source: www.geoboundaries.org"
  )
```

`gb_get_max_adm_lvl` *Get the highest administrative level available for a given country*

Description

Get a summary of selected or all countries and their highest administrative level available in geoBoundaries.

Usage

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

Arguments

`country`	A character vector of country codes. It can be either `"all"` (which returns the data for all countries), a vector of country names, or ISO3 country codes. See also `countrycode::countrycode()`.
`release_type`	One of `"gbOpen"`, `"gbHumanitarian"`, `"gbAuthoritative"`. For most users, we suggest using `"gbOpen"` (the default), as it is CC-BY 4.0 compliant and can be used for most purposes so long as attribution is provided: • `"gbHumanitarian"` files are mirrored from UN OCHA, but may have more restrictive licensing. • `"gbAuthoritative"` files are mirrored from UN SALB, and cannot be used for commercial purposes, but are verified through in-country processes.

Value

A tibble with the country names and corresponding highest administrative level.

Source

geoBoundaries API Service `https://www.geoboundaries.org/api.html`.

See Also

Other metadata functions: gb_get_metadata()

Examples

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

# Countries with only 1 level available
all |>
  filter(maxBoundaryType == 1)

# Countries with level 4 available
all |>
  filter(maxBoundaryType == 4)
```

---

gb_get_metadata *Get metadata of individual country files from geoBoundaries*

Description

This function returns metadata of 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 codes. It can be either "all" (which returns the data for all countries), a vector of country names, or ISO3 country codes. See also countrycode::countrycode().
adm_lvl	Type of boundary. Accepted values are "all" (all available boundaries) or the ADM level ("adm0" is the country boundary, "adm1" is the first level of sub-national boundaries, "adm2" is the second level, and so on). Upper-case versions ("ADM1") and the number of the level (1, 2, 3, 4, 5) are also accepted.
release_type	One of "gbOpen", "gbHumanitarian", "gbAuthoritative". For most users, we suggest using "gbOpen" (the default), as it is CC-BY 4.0 compliant and can be used for most purposes so long as attribution is provided: • "gbHumanitarian" files are mirrored from UN OCHA, but may have more restrictive licensing. • "gbAuthoritative" files are mirrored from UN SALB, and cannot be used for commercial purposes, but are verified through in-country processes.

Details

The result is a tibble with the following columns:

- `boundaryID`: The ID for this layer, which is a combination of the ISO code, the boundary type, and a unique identifier for the boundary generated based on the input metadata and geometry. This only changes if the underlying data changes.
- `boundaryName`: The name of the country the layer represents.
- `boundaryISO`: ISO-3166-1 (Alpha 3) code for the country.
- `boundaryYearRepresented`: The year, or range of years in `"START to END"` format, which the boundary layers represent.
- `boundaryType`: The type of boundary.
- `boundaryCanonical`: The canonical name of a given boundary.
- `boundarySource`: A comma-separated list of the primary sources for the boundary.
- `boundaryLicense`: The original license that the dataset was released under by the primary source.
- `licenseDetail`: Any notes regarding the license.
- `licenseSource`: The URL of the primary source.
- `sourceDataUpdateDate`: The date the source information was integrated into the geoBoundaries repository.
- `buildDate`: The date the source data was most recently standardized and built into a geoBoundaries release.
- `Continent`: The continent the country is associated with.
- `UNSDG-region`: The United Nations Sustainable Development Goals (SDG) region the country is associated with.
- `UNSDG-subregion`: The United Nations Sustainable Development Goals (SDG) subregion the country is associated with.
- `worldBankIncomeGroup`: The World Bank income group the country is associated with.
- `admUnitCount`: Count of administrative units in the file.
- `meanVertices`: Mean number of vertices defining the boundaries of each administrative unit in the layer.
- `minVertices`: Minimum number of vertices defining a boundary.
- `maxVertices`: Maximum number of vertices defining a boundary.
- `minPerimeterLengthKM`: The minimum perimeter length of an administrative unit in the layer, measured in kilometers (based on a World Equidistant Cylindrical projection).
- `meanPerimeterLengthKM`: The mean perimeter length of an administrative unit in the layer, measured in kilometers (based on a World Equidistant Cylindrical projection).
- `maxPerimeterLengthKM`: The maximum perimeter length of an administrative unit in the layer, measured in kilometers (based on a World Equidistant Cylindrical projection).
- `meanAreaSqKM`: The mean area of all administrative units in the layer, measured in square kilometers (based on a EASE-GRID 2 projection).

- `minAreaSqKM`: The minimum area of an administrative unit in the layer, measured in square kilometers (based on a EASE-GRID 2 projection).
- `maxAreaSqKM`: The maximum area of an administrative unit in the layer, measured in square kilometers (based on a EASE-GRID 2 projection).
- `staticDownloadLink`: The static download link for the aggregate zip file containing all boundary information.
- `gjDownloadURL`: The static download link for the `geoJSON`.
- `tjDownloadURL`: The static download link for the `topoJSON`.
- `imagePreview`: The static download link for the automatically rendered `PNG` of the layer.
- `simplifiedGeometryGeoJSON`: The static download link for the simplified `geoJSON`.

Value

A tibble.

Source

geoBoundaries API Service https://www.geoboundaries.org/api.html.

See Also

`gb_get()`

Other metadata functions: `gb_get_max_adm_lvl()`

Examples

```
# Get metadata of ADM4 levels

library(dplyr)

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

---

gb_get_world *Get global composites data (CGAZ) from geoBoundaries*

Description

Attribution is required for all uses of this dataset.

This function returns a global composite of the required administrative level, clipped to international boundaries, with gaps filled between borders.

Usage

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

Arguments

country	A character vector of country codes. It can be either "all" (which returns the data for all countries), a vector of country names, or ISO3 country codes. See also countrycode::countrycode().
adm_lvl	Type of boundary. Accepted values are administrative levels 0, 1, and 2 ("adm0" is the country boundary, "adm1" is the first level of sub-national boundaries, "adm2" is the second level, and so on). Upper-case versions ("ADM1") and the number of the level (0, 1, 2) are also accepted.
quiet	logical. If TRUE suppresses informational messages.
overwrite	logical. When set to TRUE it will force a fresh download of the source .zip file.
cache_dir	A path to a cache directory. If not set (the default NULL), the data will be stored in the default cache directory (see gb_set_cache_dir()). If no cache directory has been set, files will be stored in the temporary directory (see base::tempdir()). See caching strategies in gb_set_cache_dir().

Details

Comprehensive Global Administrative Zones (CGAZ) are a set of global composites for administrative boundaries. There are two important distinctions between our global product and individual country downloads.

- Extensive simplification is performed to ensure that file sizes are small enough to be used in most traditional desktop software.
- Disputed areas are removed and replaced with polygons following US Department of State definitions.

Value

A sf object.

Source

geoBoundaries API Service https://www.geoboundaries.org/api.html.

References

Runfola, D. et al. (2020) geoBoundaries: A global database of political administrative boundaries. *PLOS ONE 15*(4), 1-9. doi:10.1371/journal.pone.0231866.

See Also

Other API 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: www.geoboundaries.org")

## End(Not run)
```

---

gb_set_cache_dir *Set your* **geobounds** *cache directory*

Description

This function stores your cache_dir path on your local machine and loads it for future sessions. Type gb_detect_cache_dir() to find your cache directory path.

Usage

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

Arguments

cache_dir	A path to a cache directory. If missing, the function will store the cache files in a temporary directory (see base::tempdir()).
overwrite	Logical. If this is set to TRUE, it will overwrite an existing cache_dir.
install	Logical. If TRUE, will install the key in your local machine for use in future sessions. Defaults to FALSE. If cache_dir is FALSE this parameter is set to FALSE automatically.
quiet	logical. If TRUE suppresses informational messages.

Details

By default, when no cache_dir is set the package uses a folder inside base::tempdir() (so files are temporary and are removed when the **R** session ends). To persist a cache across **R** sessions, use gb_set_cache_dir(path, install = TRUE), which writes the chosen path to a small configuration file under tools::R_user_dir("geobounds", "config").

Value

An (invisible()) character with the path to your cache_dir.

Caching strategies

- For occasional use, rely on the default tempdir()-based cache (no install).
- Modify the cache for a single session by setting gb_set_cache_dir(cache_dir = "a/path/here").
- For reproducible workflows, install a persistent cache with gb_set_cache_dir(cache_dir = "a/path/here", install = TRUE) that will be kept across **R** sessions.
- For caching specific files, use the cache_dir argument in the corresponding function. See gb_get().

See Also

tools::R_user_dir()

Other cache utilities: 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
ex <- file.path(tempdir(), "example", "cachenew")
gb_set_cache_dir(ex)

gb_detect_cache_dir()

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

## End(Not run)

gb_detect_cache_dir()
```

Index