Package 'mapme.biodiversity'

May 8, 2026

Title Efficient Monitoring of Global Biodiversity Portfolios

Version 0.9.6

Description Biodiversity areas, especially primary forest, serve a multitude of functions for local economy, regional functionality of the ecosystems as well as the global health of our planet. Recently, adverse changes in human land use practices and climatic responses to increased greenhouse gas emissions, put these biodiversity areas under a variety of different threats. The present package helps to analyse a number of biodiversity indicators based on freely available geographical datasets. It supports computational efficient routines that allow the analysis of potentially global biodiversity portfolios. The primary use case of the package is to support evidence based reporting of an organization's effort to protect biodiversity areas under threat and to identify regions were intervention is most duly needed.

License GPL (>= 3)

URL https://mapme-initiative.github.io/mapme.biodiversity/, https://github.com/mapme-initiative/mapme.biodiversity/

BugReports https://github.com/mapme-initiative/mapme.biodiversity/issues

Depends R (>= 3.5.0)

SystemRequirements GDAL (>= 3.7.0), PROJ (>= 4.8.0)

Imports curl, dplyr, furrr, httr2, jsonlite, magrittr, purrr, sf, terra, tibble, tidyr, utils

Suggests exactextractr, future, knitr, landscapemetrics, progressr, rmarkdown, rstac, rvest, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/Needs/website DiagrammeR

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

Collate 'register.R' 'calc_biodiversity_intactness_index.R'
'calc_biome.R' 'calc_burned_area.R' 'calc_carbon.R'
'calc_deforestation_drivers.R' 'calc_drought_indicator.R'
'calc_ecoregion.R' 'calc_elevation.R'
'calc_exposed_population_acled.R'
'calc_exposed_population_ucdp.R' 'calc_fatalities_acled.R'
'calc_fatalities_ucdp.R' 'calc_gsw_change.R'
'calc_gsw_occurrence.R' 'calc_gsw_recurrence.R'
'calc_gsw_seasonality.R' 'calc_gsw_time_series.R'
'calc_gsw_transitions.R' 'calc_hfp.R' 'get_resources.R'
'calc_indicators.R' 'calc_ipbes_biomes.R' 'calc_iucn.R'
'calc_key_biodiversity_areas.R' 'calc_landcover.R'
'calc_mangroves_area.R' 'calc_population_count.R'
'calc_precipitation_chelsa.R' 'calc_precipitation_chirps.R'
'calc_precipitation_wc.R' 'calc_slope.R'
'calc_soilproperties.R' 'calc_temperature_max_wc.R'
'calc_temperature_min_wc.R' 'calc_traveltime.R'
'calc_traveltime_2000.R' 'calc_treecover_area.R'
'calc_treecover_area_and_emissions.R'
'calc_treecoverloss_emissions.R' 'calc_tri.R' 'chunking.R'
'engines.R' 'get_accessibility_2000.R' 'get_acled.R'
'get_biodiversity_intactness_index.R' 'get_carbon.R'
'get_chelsa.R' 'get_chirps.R' 'get_esalandcover.R'
'get_fritz_et_al.R' 'get_gfw_emissions.R' 'get_gfw_lossyear.R'
'get_gfw_treecover.R' 'get_gmw.R' 'get_gsw.R'
'get_gsw_time_series.R' 'get_hfp.R' 'get_ipbes_biomes.R'
'get_iucn.R' 'get_key_biodiversity_areas.R' 'get_mcd64A1.R'
'get_nasa_grace.R' 'get_nasa_srtm.R' 'get_nelson_et_al.R'
'get_soilgrids.R' 'get_teow.R' 'get_ucdp_ged.R'
'get_worldclim.R' 'get_worldpop.R' 'mapme.biodiversity-pkg.R'
'portfolio.R' 'spatial-utils.R' 'utils.R'

NeedsCompilation no

Author Darius A. Görgen [aut] (ORCID: <https://orcid.org/0009-0008-5503-7704>),
Om Prakash Bhandari [aut],
Andreas Petutschnig [ctb] (ORCID:
<https://orcid.org/0000-0001-5029-2425>),
Sven Bergtold [ctb],
Zivan Karaman [ctb, cre] (ORCID:
<https://orcid.org/0000-0002-8933-4589>),
MAPME-Initiative [cph, fnd]

Maintainer Zivan Karaman <zivan.karaman@gmail.com>

Repository CRAN

Date/Publication 2026-04-27 09:20:12 UTC

Contents

`accessibility_2000` *Accessibility to Cities in 2000*

Description

This resource provides global maps of travel time to cities of 50,000 or more people in year 2000. Accessibility refers to the ease with which larger cities can be reached from a certain location. This dataset represents travel time to major cities globally as of the year 2000, encoded in minutes. The data is essential for historical analyses, such as understanding the impact of accessibility on land use and socio-economic outcomes during this period.

Usage

```
get_accessibility_2000()
```

Value

A function that returns an `sf` footprint object.

Source

https://forobs.jrc.ec.europa.eu/gam

References

European Commission, Joint Research Centre (JRC), Global Accessibility Maps (GAM), 2000.

`acled` *Armed Conflict Location & Event Data (ACLED)*

Description

From ACLED's homepage: *The Armed Conflict Location & Event Data Project (ACLED) is a disaggregated data collection, analysis, and crisis mapping project. ACLED collects information on the dates, actors, locations, fatalities, and types of all reported political violence and protest events around the world. The ACLED team conducts analysis to describe, explore, and test conflict scenarios, and makes both data and analysis open for free use by the public.*

Usage

```
get_acled(
  years = 1997,
  email = Sys.getenv("ACLED_EMAIL"),
  password = Sys.getenv("ACLED_PASSWORD"),
  accept_terms = FALSE
)
```

Arguments

`years`	A numeric vector specifying the years for which to make ACLED data available (between 1997 and today).
`email`	Email addressed used to register with ACLED (see Details).
`password`	Password used to register with ACLED (see Details).
`accept_terms`	A logical indicating if you agree to abide by ACLED's terms of use. Defaults to FALSE, thus must be manually set to TRUE.

Details

In order to access data from the ACLED API, you first must register an account. Make sure you register with your institutional domain (e.g., organization, university, or company) email address rather than your personal email address, in order to be able to use the API, as explained here. Note that the ACLED API provides a *living database* with single events being altered or removed altogether over time.

Value

A function that returns an `sf` footprint object.

Source

Armed Conflict Location & Event Data Project (ACLED).

References

Raleigh, C., Kishi, R. & Linke, A. Political instability patterns are obscured by conflict dataset scope conditions, sources, and coding choices. Humanit Soc Sci Commun 10, 74 (2023). doi:10.1057/s41599023015594

`biodiversity_intactness_index_indicator`

Calculate Biodiversity Intactness Index

Description

This function calculates the mean biodiversity intactness index for a region.

Usage

```
calc_biodiversity_intactness_index()
```

Format

A function that returns an indicator tibble with variable biodiversity_intactness_index and corresponding values (unitless) as value.

Details

The required resources for this indicator are:

- biodiversity_intactness_index_resource

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

lbii <- system.file("res", "biodiversity_intactness_index", "lbii.asc",
                    package = "mapme.biodiversity")

aoi <- read_sf(
  system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
              package = "mapme.biodiversity"
))
aoi <- get_resources(aoi, get_biodiversity_intactness_index(lbii))
aoi <- calc_indicators(aoi, calc_biodiversity_intactness_index())
aoi <- portfolio_long(aoi)

aoi

## End(Not run)
```

---

biodiversity_intactness_index_resource

Biodiversity Intactness Index

Description

The variable is the modelled average abundance of originally-present species, relative to their abundance in an intact ecosystem. Please refer to Newbold et al. (2016) for all details, and please cite it when using these data.

Usage

```
get_biodiversity_intactness_index(path = NULL)
```

Arguments

path	A character vector to the biodiversity intactness index ASCII file.

Details

To use this data in mapme workflows, you will have to manually download the global data set and point towards the file path on your local machine. Please find the available data under the source link given below.

Value

A function that returns an `sf` footprints object.

References

Tim Newbold; Lawrence Hudson; Andy Arnell; Sara Contu et al. (2016). Global map of the Biodiversity Intactness Index, from Newbold et al. (2016) Science [Data set]. Natural History Museum. doi:10.5519/0009936

`biome` *Calculate biomes statistics (TEOW) based on WWF*

Description

This function allows to efficiently retrieve the name of the biomes and compute the corresponding area from Terrestrial Ecoregions of the World (TEOW) - World Wildlife Fund (WWF) for polygons. For each polygon, the name and area of the biomes (in hectare) is returned. The required resources for this indicator are:

- teow

Usage

```
calc_biome()
```

Value

A function that returns an indicator tibble with variable biome type and corresponding area (in ha) as values.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)
```

```
aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_teow()) %>%
  calc_indicators(calc_biome()) %>%
  portfolio_long()

aoi

## End(Not run)
```

`burned_area` *Calculate Monthly Burned Area based on MODIS (MCD64A1)*

Description

Calculates Monthly Burned Area based on the Terra and Aqua combined MCD64A1 Version 6.1. which s a monthly, global gridded 500 meter (m) product containing per-pixel burned-area information.

Usage

```
calc_burned_area(engine = "extract")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.

Details

The required resources for this indicator are:

- mcd64a1

Value

A function that returns an indicator tibble with variable burned area and corresponding area (in ha) as values.

References

Giglio, L., C. Justice, L. Boschetti, D. Roy. MODIS/Terra+Aqua Burned Area Monthly L3 Global 500m SIN Grid V061. 2021, distributed by NASA EOSDIS Land Processes Distributed Active Archive Center. doi:10.5067/MODIS/MCD64A1.061

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_mcd64a1(years = 2010)) %>%
  calc_indicators(calc_burned_area(engine = "extract")) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

calc_exposed_population_acled

Calculate population exposed to violent conflict from ACLED

Description

The indicator calculates the population exposed to conflict events within a specified buffer distance around events in ACLED. Per default, the first available WorldPop layer is used to estimate exposed populations for years before the respective year, while the most recent layer is used for years after.

Usage

```
calc_exposed_population_acled(
  distance = 5000,
  filter_category = c("event_type", "sub_event_type", "disorder_type"),
  filter_types = NULL,
  years = c(1997:2024),
  precision_location = 1,
  precision_time = 1
)
```

Arguments

`distance`	A numeric vector indicating the buffer radius in meters. If length is 1, the same buffer size around included conflict events is drawn. Otherwise, it must be equal to the length of included categories selected with `filter_types`.
`filter_category`	A character indicating the categories to be used to calculate the exposed population by. Defaults to `event_type` meaning one estimation per event type will be returned.
`filter_types`	A character vector of event types of the respective category specified in `filter_category` to retain. Defaults to NULL, meaning that no filter is applied and all types are retained.
`years`	A numeric vector indicating for which years to calculate the exposed population. Restricted to available years for ACLED. For years not intersecting with available WorldPop layers, the first layer is used for earlier years and the last layer to more recent years.
`precision_location`	A numeric indicating precision value for the geolocation up to which events are included. Defaults to 1.
`precision_time`	A numeric indicating the precision value of the temporal coding up to which events are included. Defaults to 1.

Details

The indicator is inspired by the Conflict Exposure tool from ACLED (see citation below), but differs in the regard that we simply flatten our buffered event layer instead of applying voronoi tessellation.

The required resources for this indicator are:

- acled
- worldpop

Events in ACLED are classified according to the schema described extensively in their codebook. You may filter for certain types of events. The categories for which a filter can be applied are either "event_type", "event_sub_type", or "disorder_type". These are translated into the following categories:

- event_type:
  - battles
  - protests
  - riots
  - explosions/remote_violence
  - violence_against_civilians
  - strategic_developments
- event_sub_type:
  - government_regains_territory
  - non-state_actor_overtakes_territory

  - armed_clash
  - excessive_force_against_protesters
  - protest_with_intervention
  - peaceful_protest
  - violent_demonstration
  - mob_violence
  - chemical_weapon
  - air/drone_strike
  - suicide_bomb
  - shelling/artillery/missile_attack
  - remote_explosive/landmine/ied
  - grenade
  - sexual_violence
  - attack
  - abduction/forced_disappearance
  - agreement
  - arrests
  - change_to_group/activity
  - disrupted_weapons_use
  - headquarters_or_base_established
  - looting/property_destruction
  - non-violent_transfer_of_territory
  - other
- disorder_type:
  - political_violence
  - political_violence;_demonstrations
  - demonstrations
  - political_violence
  - strategic_developments

You may supply buffer distances for each of the event categories. Custom buffers will then be drawn per category. Supply a single value if you do not wish do differentiate between categories. Otherwise, supply a vector of distances equal to the length of included categories.

You may apply quality filters based on the precision of the geolocation of events and the temporal precision. By default, these are set to only include events with the highest precision scores.

For geo-precision there are levels 1 to 3 with decreasing accuracy:

- value 1: the source reporting indicates a particular town, and coordinates are available for that town
- value 2: the source material indicates that activity took place in a small part of a region, and mentions a general area or if an activity occurs near a town or a city, the event is coded to a town with geo-referenced coordinates to represent that area

- value 3: a larger region is mentioned, the closest natural location noted in reporting (like "border area," "forest," or "sea," among others) – or a provincial capital is used if no other information at all is available

For temporal precision there are levels 1 to 3 with decreasing precision:

- value 1: the source material includes an actual date of an event
- value 2: the source material indicates that an event happened sometime during the week or within a similar period of time
- value 3: the source material only indicates that an event took place sometime during a month (i.e. in the past two or three weeks, or in January), without reference to the particular date, the month mid-point is chosen

Value

A function that returns an indicator tibble with conflict exposure as variable and percentage of the population as its value.

References

Raleigh, C; C Dowd; A Tatem; A Linke; N Tejedor-Garavito; M Bondarenko and K Kishi. 2023. Assessing and Mapping Global and Local Conflict Exposure. Working Paper.

Examples

```
## Not run:
if (FALSE) {
  library(sf)
  library(mapme.biodiversity)

  outdir <- file.path(tempdir(), "mapme-data")
  dir.create(outdir, showWarnings = FALSE)

  mapme_options(
    outdir = outdir,
    verbose = FALSE,
    chunk_size = 1e8
  )

  aoi <- system.file("extdata", "burundi.gpkg",
    package = "mapme.biodiversity"
  ) %>%
    read_sf() %>%
    get_resources(
      get_acled(year = 2000),
      get_worldpop(years = 2000)
    ) %>%
    calc_indicators(
      calc_exposed_population_acled(
        distance = 5000,
        years = 2000,
```

```
        precision_location = 1,
        precision_time = 1
      )
    ) %>%
    portfolio_long()

  aoi
}

## End(Not run)
```

---

carbon_indicators *Calculate carbon statistics*

Description

These functions allow to calculated statistics based on the harmonized carbon layers for 2010 and 2018 by Noon et al. (2022).

Usage

```
calc_irr_carbon(
  type = c("total", "soil", "biomass", "all"),
  engine = "extract",
  stats = "mean"
)

calc_man_carbon(
  type = c("total", "soil", "biomass", "all"),
  engine = "extract",
  stats = "mean"
)

calc_vul_carbon(
  type = c("total", "soil", "biomass", "all"),
  engine = "extract",
  stats = "mean"
)
```

Arguments

type	One of "total", "soil", "biomass", "all". Determines for which data layer the statistics are calculated.
engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
stats	Function to be applied to compute statistics for polygons either one or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum", and "var".

Details

The required resources for these indicators are:

- carbon_resources

Irrecoverable carbon is the amount of carbon that, if lost today, could not be recovered until 2050. It can be calculated for above- and below-ground carbon, the total amount of carbon, or for all layers.

Manageable carbon is the amount of carbon that, in principle, is manageable by human activities, e.g. its release to the atmosphere can be prevented. It can be calculated for above- and below-ground carbon, the total amount of carbon, or for all layers.

Vulnerable carbon is the amount of carbon that would be released in a typical land conversion activity. It can be calculated for above- and below-ground carbon, the total amount of carbon, or for all layers.

Value

A function that returns an indicator tibble with `(type)_carbon_(stat)` as variable and the respective statistic (in Mg) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(
    get_man_carbon(),
    get_vul_carbon(),
    get_irr_carbon()
  ) %>%
  calc_indicators(
    calc_man_carbon(stats = "sum"),
    calc_vul_carbon(stats = "sum"),
    calc_irr_carbon(stats = "sum")
  ) %>%
  portfolio_long()

aoi
```

```
## End(Not run)
```

`carbon_resources` *Carbon Layers*

Description

These resources are from the publication by Noon et al. (2022) "Mapping the irrecoverable carbon in Earth's ecosystems". This publication differentiates between 3 different kinds of carbon with varying degrees of manageability by humans. All three layers are available for above and below ground carbon, as well as a layer combining the two.

Usage

```
get_irr_carbon()

get_vul_carbon()

get_man_carbon()
```

Details

It may be required to increase the timeout option to successfully download theses layers from their source location via e.g. `options(timeout = 600)`.

Irrecoverable carbon is defined as the amount of carbon, that, if lost today, cannot be recovered until mid 21st century (so within 30 years, considering the publication date).

Vulnerable carbon is defined as the amount of carbon that would be lost in a hypothetical but typical conversion event (without including information of the probability of such an event to be actually occurring).

Manageable carbon is defined as all land areas, expect cyrosols, because carbon loss is driven by direct land-use conversion which could be halted or because climate change impacts affecting the area can potentially be directly mitigated through adaptive management.

Value

A function that returns an `sf` footprint object.

Source

https://zenodo.org/records/4091029

References

Noon, M.L., Goldstein, A., Ledezma, J.C. et al. Mapping the irrecoverable carbon in Earth's ecosystems. Nat Sustain 5, 37–46 (2022). doi:10.1038/s41893021008036

check_available_years *Helper to check yearly availability*

Description

Use this function to check if a specified vector of years intersects with the yearly availability of a resource.

Usage

```
check_available_years(target_years, available_years, indicator)
```

Arguments

target_years	Numeric indicating the target year.
available_years	Numeric indicating the available years.
indicator	A character vector with target resource/indicator name.

check_namespace *Checks if namespace is available*

Description

Use this function if your resource/indicator function requires the namespace of a certain package to be available. An informative error/warning message is printed if that is not the case.

Usage

```
check_namespace(pkg, error = TRUE)
```

Arguments

pkg	A character vector of length one indicating a package name for which the namespace is tested
error	A logical indicating whether or not to promote missing namespace to error. If FALSE, a warning is emitted.

Value

TRUE, invisible, if the namespace is available. An error message if error = TRUE, FALSE and a warning otherwise.

`chelsa` *Climatologies at High resolution for the Earth Land Surface Areas (CHELSA)*

Description

The CHELSA data (Karger et al. 2017) consists of downscaled model output temperature and precipitation estimates at a horizontal resolution of 30 arc sec. The precipitation algorithm incorporates orographic predictors including wind fields, valley exposition, and boundary layer height, with a subsequent bias correction. The spatial resolution is about 1-arc second (~1km at the equator). This resource makes V2 available.

Usage

```
get_chelsa(years = 1979:2019)
```

Arguments

`years`	A numeric vector of the years to make CHELSA monthly precipitation layers available for. Must be greater 1979, defaults to `c(1979:2019)`.

Value

A function that returns an `sf` footprint object.

Source

https://envicloud.wsl.ch/#/?prefix=chelsa/chelsa_V2/GLOBAL/

References

Karger, D.N., Conrad, O., Böhner, J., Kawohl, T., Kreft, H., Soria-Auza, R.W., Zimmermann, N.E., Linder, H.P. & Kessler, M. (2021) Climatologies at high resolution for the earth's land surface areas. EnviDat. doi:10.16904/envidat.228.v2.1

Karger, D.N., Conrad, O., Böhner, J., Kawohl, T., Kreft, H., Soria-Auza, R.W., Zimmermann, N.E., Linder, P., Kessler, M. (2017): Climatologies at high resolution for the Earth land surface areas. Scientific Data. 4 170122. doi:10.1038/sdata.2017.122

chirps *Climate Hazards Group InfraRed Precipitation with Station data (CHIRPS)*

Description

This resource is published by Funk et al. (2015) and represents a quasi-global (50°S-50°S) rainfall estimation at a monthly resolution starting with the year 1981 to the near-present. It has a spatial resolution of 0.05°. The data can be used to retrieve information on the amount of rainfall. Due to the availability of +30 years, anomaly detection and long-term average analysis is also possible. The routine will download the complete archive in order to support long-term average and anomaly calculations with respect to the 1981 - 2010 climate normal period. Thus no additional arguments need to be specified.

Usage

```
get_chirps(years = 1981:2020)
```

Arguments

`years`	A numeric vector of the years to download CHIRPS precipitation layers. Must be greater 1981, defaults to `c(1981:2020)`.

Value

A function that returns an `sf` footprint object.

Source

https://data.chc.ucsb.edu/products/CHIRPS-2.0/global_monthly/cogs/

References

Funk, C., Peterson, P., Landsfeld, M. et al. The climate hazards infrared precipitation with stations—a new environmental record for monitoring extremes. Sci Data 2, 150066 (2015). doi:10.1038/sdata.2015.66

deforestation_drivers *Calculate deforestation drivers*

Description

This function extracts areal statistics for the drivers of deforestation based on the data source produced by Fritz et al (2022).

Usage

```
calc_deforestation_drivers()
```

Details

The required resource for this indicator is:

- fritz_et_al

Value

A function that returns an indicator tibble with deforestation drivers as variable and corresponding area (in ha) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_fritz_et_al(resolution = 100)) %>%
  calc_indicators(calc_deforestation_drivers()) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

drought_indicator *Calculate drought indicator statistics*

Description

This function allows to efficiently calculate the relative wetness in the shallow groundwater section with regard to the the 1948-2012 reference period. The values represent the wetness percentile a given area achieves at a given point in time in regard to the reference period. For each polygon, the desired statistic/s (mean, median or sd) is/are returned.

Usage

```
calc_drought_indicator(engine = "extract", stats = "mean")
```

Arguments

engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
stats	Function to be applied to compute statistics for polygons either one or multiple inputs as character "mean", "median" or "sd".

Details

The required resources for this indicator are:

- nasa_grace

Value

A function that returns an indicator tibble with specified drought indicator statistics as variable and corresponding values as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nasa_grace(years = 2022)) %>%
  calc_indicators(
    calc_drought_indicator(
      engine = "extract",
      stats = c("mean", "median")
    )
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`ecoregion` *Calculate terrestrial ecoregions statistics (TEOW) based on WWF*

Description

This function allows to efficiently retrieve the name of the ecoregions and compute the corresponding area from Terrestrial Ecoregions of the World (TEOW) - World Wildlife Fund (WWF) for polygons. For each polygon, the name and area of the ecoregions (in hectare) is returned. The required resources for this indicator are:

- teow

Usage

```
calc_ecoregion()
```

Value

A function that returns an indicator tibble with ecoregion type as variable and corresponding area (in ha) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_teow()) %>%
  calc_indicators(calc_ecoregion()) %>%
  portfolio_long()

aoi

## End(Not run)
```

`elevation` *Calculate elevation statistics*

Description

This function allows to calculate elevation statistics for polygons. For each polygon, the desired statistic(s) are returned.

Usage

```
calc_elevation(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either one or multiple inputs as character "mean", "median" or "sd".

Details

The required resources for this indicator are:

- nasa_srtm

Value

A function that returns an indicator tibble with specified elevation statistics as variable and corresponding values (in meters) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nasa_srtm()) %>%
```

```
  calc_indicators(
    calc_elevation(engine = "extract", stats = c("mean", "median", "sd", "var"))
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

engine *Function to select processing engines*

Description

`check_engine()` checks if an extraction engine for zonal vector-raster operations is supported by the backend.

`check_stats` checks if one or multiple statistics are supported for zonal vector-raster extraction by the backend.

`select_engine` extracts zonal vector-raster statistics for supported engine and for one or more statistics. Columns are named according to the argument `name` plus the respective stat. Both `portfolio` and `asset` modes are supported.

Usage

```
check_engine(queried_engine)

check_stats(queried_stats)

select_engine(x, raster, stats, engine, name = NULL, mode = "asset")
```

Arguments

`queried_engine`	A character vector of length one indicating the engine to check for.
`queried_stats`	A character vector with statistic names to be checked if they are supported by the backend
`x`	An sf object representing a portfolio.
`raster`	An terra SpatRaster from which values are to be extracted.
`stats`	A character vector of statistics to aggregate the raster values with.
`engine`	A character vector of length one specifying the engine to be used for the extraction.
`name`	A character vector indicating the name to append to the columns names.
`mode`	A character vector indicating in which mode to conduct the extraction (e.g. `asset`-wise or for the whole `portfolio` at once).

Value

`check_engine()` returns the character of the queried engine, if supported. Throws an error otherwise.

`check_stats` returns a character vector of supported statistics. Throws an error if any of the queried statistics is not supported.

`select_engine` returns a tibble.

`esalandcover` *ESA Copernicus Global Land Cover layer*

Description

This 100 meter spatial resolution land cover resource is published by Buchhorn et al. (2020) "Copernicus Global Land Cover Layers—Collection 2". The resource represents the actual surface cover of ground available annually for the period 2015 to 2019. The cell values range from 0 to 200, representing total of 23 discrete classifications from ESA.

Usage

```
get_esalandcover(years = 2015L:2019L)
```

Arguments

`years`	A numeric vector indicating the years for which to make the resource available.

Details

This function mimics the behavior of the original `get_esalandcover()` function by downloading the parts of the global raster that correspond to the 20° x 20° tiles used previously. The files are named the same as before. This allows other code that uses this resource to remain unchanged.

Value

A function that returns an `sf` footprint object.

Source

https://zenodo.org/records/3939050

References

Buchhorn, M., Lesiv, M., Tsendbazar, N.-E., Herold, M., Bertels, L., & Smets, B. (2020). Copernicus Global Land Cover Layers - Collection 2. Remote Sensing, 12(6), 1044. doi:10.3390/rs12061044

`exposed_population_ucdp` *Calculate population exposed to violent conflict from UCDP GED*

Description

The indicator calculates the population exposed to conflict events within a specified buffer distance around violent events in UCDP GED. Per default, the first available WorldPop layer is used to estimate exposed populations for years before the respective year, while the most recent layer is used for years after.

Usage

```
calc_exposed_population_ucdp(
  distance = 5000,
  violence_types = 1:3,
  years = c(1989:2023),
  precision_location = 1,
  precision_time = 1
)
```

Arguments

`distance`	A numeric vector indicating the buffer size around included conflict events to calculate the exposed population. Either of length 1 to apply for all types of events, or discrete values for each category included in `violence_types`.
`violence_types`	A numeric vector indicating the types of violence to be included (see Details).
`years`	A numeric vector indicating for which years to calculate the exposed population. Restricted to available years for UCDP GED. For years not intersecting with available WorldPop layers, the first layer is used for earlier years and the last layer to more recent years.
`precision_location`	A numeric indicating precision value for the geolocation up to which events are included. Defaults to 1.
`precision_time`	A numeric indicating the precision value of the temporal coding up to which events are included. Defaults to 1.

Details

The indicator is inspired by the Conflict Exposure tool from ACLED (see citation below), but differs in the regard that we simply flatten our buffered event layer instead of applying voronoi tessellation.

The required resources for this indicator are:

- ucdp_ged
- worldpop

You may filter for certain types of violence. The coded types according to the UCDP codebook are: value 1: state-based conflict value 2: non-state conflict value 3: one-sided conflict

You may apply quality filters based on the precision of the geolocation of events and the temporal precision. By default, these are set to only include events with the highest precision scores.

For geo-precision there are levels 1 to 7 with decreasing accuracy:

- value 1: the location information corresponds exactly to the geographical coordinates available
- value 2: the location information refers to a limited area around a specified location
- value 3: the source refers to or can be specified to a larger location at the level of second order administrative divisions (ADM2), such as district or municipality, the GED uses centroid point coordinates for that ADM2.
- value 4: the location information refers to a first order administrative division, such as a province (ADM1), the GED uses the coordinates for the centroid point of ADM1
- value 5: is used in different cases if the source refers to parts of a country which are larger than ADM1, but smaller than the entire country; if two locations are mentioned a representative point in between is selected; if the location mentioned is an non-independent island; if the location is not very specifically mentioned or in relation to another location
- value 6: the location mentioned refers to an entire country and its centroid is used
- value 7: If the event takes place over water or in international airspace, the geographical coordinates in the dataset either represent the centroid point of a certain water area or estimated coordinates

For temporal precision there are levels 1 to 5 with decreasing precision:

- value 1: if the exact date of an event is known
- value 2: if start and end dates for events are of unspecified character, spanning more than one calendar day though no longer than six days
- value 3: if when start and end dates for events are specified to a certain week, but specific dates are not provided
- value 4: if start and end dates for events are specified to a certain month
- value 5: if start and end dates for events are specified to a certain year, but specific dates are not provided

Value

A function that returns an indicator tibble with conflict exposure as variable and percentage of the population as its value.

References

Raleigh, C; C Dowd; A Tatem; A Linke; N Tejedor-Garavito; M Bondarenko and K Kishi. 2023. Assessing and Mapping Global and Local Conflict Exposure. Working Paper.

Examples

```
## Not run:
if (FALSE) {
  library(sf)
  library(mapme.biodiversity)

  outdir <- file.path(tempdir(), "mapme-data")
  dir.create(outdir, showWarnings = FALSE)

  mapme_options(
    outdir = outdir,
    verbose = FALSE,
    chunk_size = 1e8
  )

  aoi <- system.file("extdata", "burundi.gpkg",
    package = "mapme.biodiversity"
  ) %>%
    read_sf() %>%
    get_resources(
      get_ucdp_ged(version = "22.1"),
      get_worldpop(years = 2000)
    ) %>%
    calc_indicators(
      conflict_exposure(
        distance = 5000,
        violence_types = 1:3,
        years = 2000,
        precision_location = 1,
        precision_time = 1
      )
    ) %>%
    portfolio_long()

  aoi
}

## End(Not run)
```

`fatalities_acled` *Calculate number of fatalities of conflict events from ACLED*

Description

The indicator aggregated the number of fatalities within a given asset on a monthly cadence stratified either by event type, sub-event type or disorder type. To learn about the different categorisation ACLED uses to encode events please consult ACLED's codebook.

Usage

```
calc_fatalities_acled(
  years = 2000,
  stratum = c("event_type", "sub_event_type", "disorder_type"),
  precision_location = 1,
  precision_time = 1
)
```

Arguments

years	A numeric vector indicating the years for which to summarize fatalities.
stratum	A character vector indicating the stratification to be applied. Should be one of "event_type", "sub_event_type", or "disorder_type". Defaults to "event_type".
precision_location	A numeric indicating precision value for the geolocation up to which events are included. Defaults to 1.
precision_time	A numeric indicating the precision value of the temporal coding up to which events are included. Defaults to 1.

Details

The required resources for this indicator are:

- acled

You may apply quality filters based on the precision of the geolocation of events and the temporal precision. By default, these are set to only include events with the highest precision scores.

For geo-precision there are levels 1 to 3 with decreasing accuracy:

- value 1: the source reporting indicates a particular town, and coordinates are available for that town
- value 2: the source material indicates that activity took place in a small part of a region, and mentions a general area or if an activity occurs near a town or a city, the event is coded to a town with geo-referenced coordinates to represent that area
- value 3: a larger region is mentioned, the closest natural location noted in reporting (like “border area,” “forest,” or “sea,” among others) – or a provincial capital is used if no other information at all is available

For temporal precision there are levels 1 to 3 with decreasing precision:

- value 1: the source material includes an actual date of an event
- value 2: the source material indicates that an event happened sometime during the week or within a similar period of time
- value 3: the source material only indicates that an event took place sometime during a month (i.e. in the past two or three weeks, or in January), without reference to the particular date, the month mid-point is chosen

Value

A function that returns an indicator tibble with the type of violence as variable and counts of civilian fatalities as value.

References

Raleigh, C., Kishi, R. & Linke, A. Political instability patterns are obscured by conflict dataset scope conditions, sources, and coding choices. Humanit Soc Sci Commun 10, 74 (2023). doi:10.1057/s41599023015594

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE,
  chunk_size = 1e8
)

aoi <- system.file("extdata", "burundi.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_acled(years = 2020)) %>%
  calc_indicators(
    calc_fatalities_acled(
      years = 2020,
      precision_location = 1,
      precision_time = 1
    )
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`fatalities_ucpd` *Calculate number of fatalities of violent conflict from UCDP GED*

Description

The indicator aggregated the number of fatalities within a given asset on a monthly cadence stratified by the type of conflict. The different types of conflicts encoded in the UCDP GED database are:

- state-based conflict
- non-state conflict
- one-sided violence

Usage

```
calc_fatalities_ucdp(
  years = 1989:2023,
  precision_location = 1,
  precision_time = 1
)
```

Arguments

`years`	A numeric vector indicating the years for which to summarize fatalities.
`precision_location`	A numeric indicating precision value for the geolocation up to which events are included. Defaults to 1.
`precision_time`	A numeric indicating the precision value of the temporal coding up to which events are included. Defaults to 1.

Details

The required resources for this indicator are:

- ucdp_ged

You may apply quality filters based on the precision of the geolocation of events and the temporal precision. By default, these are set to only include events with the highest precision scores.

For geo-precision there are levels 1 to 7 with decreasing accuracy:

- value 1: the location information corresponds exactly to the geographical coordinates available
- value 2: the location information refers to a limited area around a specified location
- value 3: the source refers to or can be specified to a larger location at the level of second order administrative divisions (ADM2), such as district or municipality, the GED uses centroid point coordinates for that ADM2.
- value 4: the location information refers to a first order administrative division, such as a province (ADM1), the GED uses the coordinates for the centroid point of ADM1
- value 5: is used in different cases if the source refers to parts of a country which are larger than ADM1, but smaller than the entire country; if two locations are mentioned a representative point in between is selected; if the location mentioned is an non-independent island; if the location is not very specifically mentioned or in relation to another location
- value 6: the location mentioned refers to an entire country and its centroid is used

- value 7: If the event takes place over water or in international airspace, the geographical coordinates in the dataset either represent the centroid point of a certain water area or estimated coordinates

For temporal precision there are levels 1 to 5 with decreasing precision:

- value 1: if the exact date of an event is known
- value 2: if start and end dates for events are of unspecified character, spanning more than one calendar day though no longer than six days
- value 3: if when start and end dates for events are specified to a certain week, but specific dates are not provided
- value 4: if start and end dates for events are specified to a certain month
- value 5: if start and end dates for events are specified to a certain year, but specific dates are not provided

Value

A function that returns an indicator tibble with the type of violence as variable and counts of civilian fatalities as value.

References

Sundberg, Ralph, and Erik Melander, 2013, "Introducing the UCDP Georeferenced Event Dataset", Journal of Peace Research, vol.50, no.4, 523-532

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE,
  chunk_size = 1e8
)

aoi <- system.file("extdata", "burundi.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_ucdp_ged(version = "22.1")) %>%
  calc_indicators(
    calc_fatalities(
      years = 1991:1992,
      precision_location = 1,
      precision_time = 1
    )
```

```
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`fritz_et_al` *Drivers of deforestation for tropical forests*

Description

This resource is produced by a nearest-neighbour matching of a crowd-sourced campaign to map dominant driver of forest loss based on visual interpretation of VHR images matched with Global Forest Loss data by Hansen (2013) version 1.7 The forest loss layer was re sampled to a resolution of 100 and 1.000 meters. Dominant drivers were determined for the period 2008 to 2009.

Usage

```
get_fritz_et_al(resolution = 100)
```

Arguments

`resolution`	An integer indicating the resolution to download. Defaults to 100.

Details

It indicates 9 different classes:

- commercial agriculture
- commercial oil palm plantations
- managed forests
- mining
- natural disturbances
- pasture
- roads
- wildfire
- other subsistence agriculture
- shifting cultivation

Value

A function that returns an `sf` footprint object.

Source

https://zenodo.org/record/7997885

References

Steffen, F., Carlos, J.C.L., See. L., Schepaschenko D., Hofhansl F., Jung M., Dürauer M., Georgieva I., Danylo O., Lesiv M., McCallum I. (2022) A Continental Assessment of the Drivers of Tropical Deforestation With a Focus on Protected Areas. F.Cos.Sc.(3) doi:10.3389/fcosc.2022.830248

`gfw_emissions` *Forest greenhouse gas emissions*

Description

This resource is part of the publication by Harris et al. (2021) "Global maps of twenty-first century forest carbon fluxes.". It represents "the greenhouse gas emissions arising from stand-replacing forest disturbances that occurred in each modelled year (megagrams CO2 emissions/ha, between 2001 and 2023). Emissions include all relevant ecosystem carbon pools (aboveground biomass, belowground biomass, dead wood, litter, soil) and greenhouse gases (CO2, CH4, N2O)." The area unit that is downloaded here corresponds to the "megagrams of CO2 emissions/pixel" layer, in order to support the calculation of area-wise emissions.

Usage

```
get_gfw_emissions()
```

Details

There are no arguments users need to specify. However, users should note that the spatial extent for this dataset does not totally cover the same extent as the `treecover2000` and `lossyear` resources by Hansen et al. (2013). A missing value (NA) will be inserted for greenhouse gas emissions for areas where no data is available.

Value

A function that returns an `sf` footprint object.

Source

https://data.globalforestwatch.org/datasets/gfw::forest-greenhouse-gas-emissions/about

References

Harris, N.L., Gibbs, D.A., Baccini, A. et al. Global maps of twenty-first century forest carbon fluxes. Nat. Clim. Chang. 11, 234–240 (2021). doi:10.1038/s41558020009766

`gfw_lossyear` *Year of forest loss occurrence*

Description

This resource is part of the publication by Hansen et al. (2013) "High-Resolution Global Maps of 21st-Century Forest Cover Change". It represents "Forest loss during the period 2000–2021, defined as a stand-replacement disturbance, or a change from a forest to non-forest state. Encoded as either 0 (no loss) or else a value in the range 1–20, representing loss detected primarily in the year 2001–2021, respectively." Due to changes in the satellites products used in the compilation of the tree loss product, results before the year 2011 and afterwards are not directly comparable until reprocessing has finished. Users should be aware of this limitation, especially when the timeframe of the analysis spans over the two periods delimited by the year 2011.

Usage

```
get_gfw_lossyear(version = "GFC-2024-v1.12")
```

Arguments

`version`	The version of the dataset to download. Defaults to "GFC-2024-v1.12". Check `mapme.biodiversity:::.available_gfw_versions()` to get a list of available versions

Value

A function that returns an `sf` footprint object.

Source

https://data.globalforestwatch.org/documents/tree-cover-loss/explore

References

Hansen, M. C., P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, and J. R. G. Townshend. 2013. “High-Resolution Global Maps of 21st-Century Forest Cover Change.” Science 342 (15 November): 850–53.

gfw_treecover *Treecover for the year 2000*

Description

This resource is part of the publication by Hansen et al. (2013) represents "tree cover in the year 2000, defined as canopy closure for all vegetation taller than 5m in height. Encoded as a percentage per output grid cell, in the range 0–100." Due to changes in the satellites products used in the compilation of the treecover product, results before the year 2011 and afterwards are not directly comparable until reprocessing has finished. Users should be aware of this limitation, especially when the timeframe of the analysis spans over the two periods delimited by the year 2011.

Usage

```
get_gfw_treecover(version = "GFC-2024-v1.12")
```

Arguments

version	The version of the dataset to download. Defaults to "GFC-2024-v1.12". Check mapme.biodiversity:::.available_gfw_versions() to get a list of available versions

Value

A function that returns an sf footprint object.

Source

https://data.globalforestwatch.org/documents/tree-cover-2000/explore

References

Hansen, M. C., P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, and J. R. G. Townshend. 2013. “High-Resolution Global Maps of 21st-Century Forest Cover Change.” Science 342 (15 November): 850–53.

global_surface_water_change *Global Surface Water Change*

Description

The Global Surface Water dataset was developed by the European Commission's Joint Research Centre in the framework of the Copernicus Programme. It maps the location and temporal distribution of water surfaces at the global scale over the past 3.8 decades and provides statistics on their extent and change. It is provisioned as a global tiled raster resource available for all land areas. The reported data represent aggregated observations between 1984 - 2021.

Usage

```
get_global_surface_water_change(version = "v1_4_2021")
```

Arguments

`version`	A character vector indicating the version of the GSW data set to make available.

Details

The change in water occurrence intensity between the two periods is derived from homologous pairs of months (i.e. same months containing valid observations in both periods). The difference in the occurrence of surface water was calculated for each homologous pair of months. The average of all of these differences constitutes the Surface Water Occurrence change intensity. The raster files have integer cell values between `[0, 200]` where 0 represents surface water loss and 200 represents surface water gain.

Value

A function that returns an `sf` footprint object.

Source

https://global-surface-water.appspot.com/

References

Pekel, JF., Cottam, A., Gorelick, N. et al. High-resolution mapping of global surface water and its long-term changes. Nature 540, 418–422 (2016). doi:10.1038/nature20584

---

global_surface_water_occurrence

Global Surface Water Occurrence

Description

The Global Surface Water dataset was developed by the European Commission's Joint Research Centre in the framework of the Copernicus Programme. It maps the location and temporal distribution of water surfaces at the global scale over the past 3.8 decades and provides statistics on their extent and change. It is provisioned as a global tiled raster resource available for all land areas. The reported data represent aggregated observations between 1984 - 2021.

Usage

```
get_global_surface_water_occurrence(version = "v1_4_2021")
```

Arguments

version	A character vector indicating the version of the GSW data set to make available.

Details

GSW occurrence raw data comes in raster files with integer cell values between `[0, 100]`. This value gives the percentage of the time that a given pixel was classified as water during the entire observation period. So a 0 denotes a pixel that was never classified as water, 100 denotes a pixel with permanent water.

Value

A character of file paths.

Source

https://global-surface-water.appspot.com/

References

Pekel, JF., Cottam, A., Gorelick, N. et al. High-resolution mapping of global surface water and its long-term changes. Nature 540, 418–422 (2016). doi:10.1038/nature20584

---

global_surface_water_recurrence

Global Surface Water Recurrence

Description

The Global Surface Water dataset was developed by the European Commission's Joint Research Centre in the framework of the Copernicus Programme. It maps the location and temporal distribution of water surfaces at the global scale over the past 3.8 decades and provides statistics on their extent and change. It is provisioned as a global tiled raster resource available for all land areas. The reported data represent aggregated observations between 1984 - 2021.

Usage

```
get_global_surface_water_recurrence(version = "v1_4_2021")
```

Arguments

version	A character vector indicating the version of the GSW data set to make available.

Details

Water Recurrence is a measurement of the degree of variability in the presence of water from year to year. It describes the frequency with which water returned to a particular location from one year to another, and is expressed as a percentage. The raster files have integer cell values between `[0, 100]`, where 100 represents that water reoccurs predictably every year, whereas lower values indicate that water only occurs episodically.

Value

A character of file paths.

Source

https://global-surface-water.appspot.com/

References

Pekel, JF., Cottam, A., Gorelick, N. et al. High-resolution mapping of global surface water and its long-term changes. Nature 540, 418–422 (2016). doi:10.1038/nature20584

`global_surface_water_seasonality` *Global Surface Water Seasonality*

Description

The Global Surface Water dataset was developed by the European Commission's Joint Research Centre in the framework of the Copernicus Programme. It maps the location and temporal distribution of water surfaces at the global scale over the past 3.8 decades and provides statistics on their extent and change. It is provisioned as a global tiled raster resource available for all land areas. The reported data represent aggregated observations between 1984 - 2021.

Usage

```
get_global_surface_water_seasonality(version = "v1_4_2021")
```

Arguments

`version`	A character vector indicating the version of the GSW data set to make available.

Details

GSW seasonality describes the intra-annual distribution of surface water for each pixel. The raster files have integer cell values between `[0, 12]`, indicating how many months per year the pixel was classified as water.

Value

A character of file paths.

Source

https://global-surface-water.appspot.com/

References

Pekel, JF., Cottam, A., Gorelick, N. et al. High-resolution mapping of global surface water and its long-term changes. Nature 540, 418–422 (2016). doi:10.1038/nature20584

`global_surface_water_transitions`
Global Surface Water Transitions

Description

The Global Surface Water dataset was developed by the European Commission's Joint Research Centre in the framework of the Copernicus Programme. It maps the location and temporal distribution of water surfaces at the global scale over the past 3.8 decades and provides statistics on their extent and change. It is provisioned as a global tiled raster resource available for all land areas. The reported data represent aggregated observations between 1984 - 2021.

Usage

```
get_global_surface_water_transitions(version = "v1_4_2021")
```

Arguments

`version`	A character vector indicating the version of the GSW data set to make available.

Details

GSW transition data contains information about the type of surface water change for each pixel. The raster files have integer cell values between `[0, 10]` that code for different transition classes:

Value	Transition Class
1	Permanent
2	New Permanent
3	Lost Permanent
4	Seasonal
5	New Seasonal
6	Lost Seasonal
7	Seasonal to Permanent
8	Permanent to Seasonal
9	Ephemeral Permanent

10 Ephemeral Seasonal

Value

A character of file paths.

Source

https://global-surface-water.appspot.com/

References

Pekel, JF., Cottam, A., Gorelick, N. et al. High-resolution mapping of global surface water and its long-term changes. Nature 540, 418–422 (2016). doi:10.1038/nature20584

---

gmw *Global Mangrove Extent Polygon*

Description

This resource is part of the publication by Bunting et al. (2018) "The Global Mangrove Watch—A New 2010 Global Baseline of Mangrove Extent". The polygons represent the mangrove, which is tropical coastal vegetation and considered the most significant part of the marine ecosystem. This resource is available for the selected years in the period 1996- 2020 from Global Mangrove Watch (GMW), providing geospatial information about global mangrove extent.

Usage

```
get_gmw(years = c(1996, 2007:2010, 2015:2020))
```

Arguments

`years`	A numeric vector of the years for which to make GMW available.

Value

A function that returns an `sf` footprint object.

Source

https://habitats.oceanplus.org/

References

Bunting P., Rosenqvist A., Lucas R., Rebelo L-M., Hilarides L., Thomas N., Hardy A., Itoh T., Shimada M. and Finlayson C.M. (2018). The Global Mangrove Watch – a New 2010 Global Baseline of Mangrove Extent. Remote Sensing 10(10): 1669. doi:10.3390/rs10101669.

gsw_change *Calculate Global Surface Water (GSW) Change*

Description

The change in water occurrence intensity between the two periods is derived from homologous pairs of months (i.e. same months containing valid observations in both periods). The difference in the occurrence of surface water was calculated for each homologous pair of months. The average of all of these differences constitutes the Surface Water Occurrence change intensity. The raster files have integer cell values between `[0, 200]` where 0 represents surface water loss and 200 represents surface water gain.

Usage

```
calc_gsw_change(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract". Default: "extract".
`stats`	Aggregation function with which the data are combined. Default: "mean".

Details

The pixel values are aggregated using method provided via the `stats` parameter using the specified `engine`.

The required resources for this indicator are:

- global_surface_water_change

Value

A function that returns an indicator tibble with change intensity as variable and corresponding (unitless) values as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)
```

```
aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_global_surface_water_change()) %>%
  calc_indicators(
    calc_gsw_change(engine = "extract", stats = "mean")
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`gsw_occurrence` *Calculate Global Surface Water (GSW) Occurrence*

Description

GSW occurrence raw data comes in raster files with integer cell values between `[0, 100]`. This value gives the percentage of the time that a given pixel was classified as water during the entire observation period. So a 0 denotes a pixel that was never classified as water, 100 denotes a pixel with permanent water.

Usage

```
calc_gsw_occurrence(engine = "extract", min_occurrence = NULL)
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract". Default: "extract".
`min_occurrence`	Threshold to define which pixels count towards the GSW occurrence area `[0, 100]`.

Details

The raw data values are aggregated based on a provided threshold parameter `min_occurrence`, the function returns the area covered by values greater or equal than this threshold.

The required resources for this indicator are:

- global_surface_water_occurrence

Value

A function that returns an indicator tibble with occurrence as variable and the corresponding area (in ha) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_global_surface_water_occurrence()) %>%
  calc_indicators(
    calc_gsw_occurrence(engine = "extract", min_occurrence = 10)
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`gsw_recurrence` *Calculate Global Surface Water (GSW) Recurrence*

Description

Water Recurrence is a measurement of the degree of variability in the presence of water from year to year. It describes the frequency with which water returned to a particular location from one year to another, and is expressed as a percentage. The raster files have integer cell values between `[0, 100]`, where 100 represents that water reoccurs predictably every year, whereas lower values indicate that water only occurs episodically.

Usage

```
calc_gsw_recurrence(engine = "extract", min_recurrence = NULL)
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract". Default: "extract".
`min_recurrence`	Threshold to define which pixels count towards the GSW recurrence area `[0, 100]`.

Details

The raw data values are aggregated based on a provided threshold parameter `min_recurrence`, the function returns the area covered by values greater or equal than this threshold.

The required resources for this indicator are:

- global_surface_water_recurrence

Value

A function that returns an indicator tibble with recurrence as variable and the corresponding area (in ha) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_global_surface_water_recurrence()) %>%
  calc_indicators(
    calc_gsw_recurrence(engine = "extract", min_recurrence = 10)
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`gsw_seasonality` *Calculate Global Surface Water (GSW) Seasonality*

Description

GSW seasonality describes the intra-annual distribution of surface water for each pixel. The raster files have integer cell values between `[0, 12]`, indicating how many months per year the pixel was classified as water.

Usage

```
calc_gsw_seasonality()
```

Details

The pixel values are aggregated using method provided via the `stats` parameter.

The required resources for this indicator are:

- global_surface_water_seasonality

Value

A function that returns an indicator tibble with seasonality categories as variables and corresponding areas (in ha) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_global_surface_water_seasonality()) %>%
  calc_indicators(calc_gsw_seasonality()) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

gsw_time_series_indicator

Calculate Global Surface Water Time Series

Description

This function calculates the total area of the global surface water time series data, separated by the following classes:

Usage

```
calc_gsw_time_series()
```

Format

A function returning a tibble with time series of global surface water data classes.

Details

- No Observation: It was not possible to determine whether a pixel was water (this may be the case for frozen areas or during the polar night in extreme latitudes).
- Permanent Water: Water was detected in twelve months per year or in a combination of permanent and no observation.
- Seasonal Water: Water and no water was detected.
- No Water: No Water was detected.

The required resources for this indicator are:

- gsw_time_series_resource

Examples

```
## Not run:
library(mapme.biodiversity)
library(sf)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- read_sf(
  system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
              package = "mapme.biodiversity"
))
aoi <- get_resources(aoi, get_gsw_time_series (years = 2000:2001))
aoi <- calc_indicators(aoi, calc_gsw_time_series())
aoi <- portfolio_long(aoi)

aoi

## End(Not run)
```

`gsw_time_series_resource` *Helper function to download Global Surface Water (GSW) yearly time series data*

Description

This function constructs the necessary data URLs for a given data set, version and polygon and downloads them for further processing with the mapme.biodiversity package.

Usage

```
get_gsw_time_series(years, version = "LATEST")
```

Arguments

`years`	Numeric vector of years to process between 1984 and 2021. Default: `1984:2021`.
`version`	Version of the data set to process. Available options are (`VER1-0`, `VER2-0`, `VER3-0`, `VER4-0`, `VER5-0` and `LATEST`) Default: `LATEST`. Choosing `LATEST` will result in the latest available version.

Details

The available surface water classes for a given pixel are the following:

- No Observation: It was not possible to determine whether a pixel was water (this may be the case for frozen areas or during the polar night in extreme latitudes).
- Permanent Water: Water was detected in twelve months per year or in a combination of permanent and no observation.
- Seasonal Water: Water and no water was detected.
- No Water: No Water was detected.

Value

A function that returns a character vector of file paths.

Source

Raw Data: https://jeodpp.jrc.ec.europa.eu/ftp/jrc-opendata/GSWE/YearlyClassification/LATEST/tiles/

References

- Global Surface Water Explorer: https://global-surface-water.appspot.com/
- Data Users Guide: https://storage.cloud.google.com/global-surface-water/downloads_ancillary/DataUsersGuidev2021.pdf
- Research Article: https://www.nature.com/articles/nature20584

`gsw_transitions` *Calculate Global Surface Water (GSW) Transitions*

Description

GSW transition data contains information about the type of surface water change for each pixel. The raster files have integer cell values between `[0, 10]` that code for different transition classes:

Usage

```
calc_gsw_transitions()
```

Details

Value	Transition Class
1	Permanent
2	New Permanent
3	Lost Permanent
4	Seasonal
5	New Seasonal
6	Lost Seasonal
7	Seasonal to Permanent
8	Permanent to Seasonal
9	Ephemeral Permanent
10	Ephemeral Seasonal

To aggregate, we sum up the area of each transition class for a given region.

The required resources for this indicator are:

- global_surface_water_transitions

Value

A function that returns an indicator tibble with transition classes as variable and corresponding areas (in ha) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
```

```
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_global_surface_water_transitions()) %>%
  calc_indicators(calc_gsw_transitions()) %>%
  portfolio_long()

aoi

## End(Not run)
```

`humanfootprint_indicator`

Calculate human footprint statistics

Description

Human footprint data measures the pressure imposed on the natural environment by different dimensions of human actions. The theoretical maximum value, representing the highest level of human pressure, is 50. This routine allows to extract zonal statistics of the human footprint data.

Usage

```
calc_humanfootprint(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either one or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resources for this indicator are:

- humanfootprint_resource

Value

A function that returns an indicator tibble the `humanfootprint` as variable and the associated value (unitless) per year.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_humanfootprint(years = 2010)) %>%
  calc_indicators(calc_humanfootprint(stats = "median")) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

humanfootprint_resource

Terrestrial Human Foootprint

Description

This resource is part of the publication by Mu et al. (2022) "A global record of annual terrestrial Human Footprint dataset from 2000 to 2018". It is calculated based on 8 variables representing human pressures on natural ecosystems collected at a yearly cadence between 2000 and 2024 sampled at a 1km spatial resolution. The variables are used are the expansion of built environments (expressed as percentage of built-up areas within a grid cell), population density (aggregated at the grid cell), nighttime lights, crop and pasture lands, roads and railways (excluding trails and minor roads), and navigable waterways (compares waterways with nighttime lights dataset). The human footprint was then calculated based on a weighting scheme proposed by Venter et al. (2016), assigning each pixel a value between 0 and 50, with 50 representing the theoretical value of the highest human pressure.

Usage

```
get_humanfootprint(years = 2000:2024)
```

Arguments

`years`	A numeric vector indicating the years for which to download the human footprint data, defaults to `2000:2024`.

Value

A function that returns an `sf` footprint object.

Note

It may be required to increase the timeout option to successfully download theses layers from their source location via e.g. `options(timeout = 600)`. In case an 403 error occurs, you can create an account with Figshare and create an personal access token. If set as `FIGSHARE_PAT` environment variable, it will be used to authenticate.

Source

https://figshare.com/articles/figure/An_annual_global_terrestrial_Human_Footprint_dataset_from_2000_to_2018/16571064

References

Mu, H., Li, X., Wen, Y. et al. A global record of annual terrestrial Human Footprint dataset from 2000 to 2018. Sci Data 9, 176 (2022). doi:10.1038/s41597022012848

`indicators` *Register or list indicators in mapme.biodiversity*

Description

`register_indicator()` is used to register a new indicator function with base information to the package's internal environment used to inform users about available indicators. Note, registering a custom indicator will only have effect for the current R session.

`available_indicators()` returns a tibble of registered indicators with basic information such as the required resources.

Usage

```
register_indicator(name = NULL, description = NULL, resources = NULL)

available_indicators(indicators = NULL)
```

Arguments

`name`	A character vector indicating the name of the indicator.
`description`	A character vector with a basic description
`resources`	A character vector of the required resources that need to be available to calculate the indicator. The names must correspond with already registered resources.
`indicators`	If `NULL` returns a list of all registered indicators (default). Otherwise only the ones specified.

Value

`register_indicator()` is called for the side-effect of registering an indicator

`available_resources()` returns a tibble listing available indicators.

Examples

```
## Not run:
register_indicator(
  name = "treecover_area",
  description = "Area of forest cover by year",
  resources = c(
    "gfw_treecover",
    "gfw_lossyear"
  )
)

## End(Not run)
available_indicators()
```

---

ipbes_biomes *Terrestrial and Aquatic Biomes*

Description

This resource is part of the Global Assessment Report on Biodiversity and Ecosystem Services and represents a division of the Earth's surface into several subcategories. The classification differentiates between biomes and anthromes. Biomes are differentiated between terrestrial and aquatic biomes.

Usage

```
get_ipbes_biomes()
```

Details

Terrestrial biomes include:

- Tropical and subtropical dry and humid forests
- Temperate and boreal forests and woodlands
- Mediterranean forests, woodlands and scrub
- Tundra and High Mountain habitats
- Tropical and subtropical savannas and grasslands
- Temperate Grasslands
- Deserts and xeric shrublands
- Wetlands – peatlands, mires, bogs

Aquatic biomes include:

- Cryosphere
- Aquaculture areas
- Inland surface waters and water bodies/freshwater
- Shelf ecosystems (neritic and intertidal/littoral zone)
- Open ocean pelagic systems (euphotic zone)

Value

A function that returns an `sf` footprint object.

Source

https://zenodo.org/records/3975694

References

IPBES (2019): Summary for policymakers of the global assessment report on biodiversity and ecosystem services of the Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services. S. Díaz, J. Settele, E. S. Brondízio, H. T. Ngo, M. Guèze, J. Agard, A. Arneth, P. Balvanera, K. A. Brauman, S. H. M. Butchart, K. M. A. Chan, L. A. Garibaldi, K. Ichii, J. Liu, S. M. Subramanian, G. F. Midgley, P. Miloslavich, Z. Molnár, D. Obura, A. Pfaff, S. Polasky, A. Purvis, J. Razzaque, B. Reyers, R. Roy Chowdhury, Y. J. Shin, I. J. Visseren-Hamakers, K. J. Willis, and C. N. Zayas (eds.). IPBES secretariat, Bonn, Germany. 56 pages. doi:10.5281/zenodo.3553579

`ipbes_biome_stats` *Calculate areal statistics for IBPES Biomes*

Description

This indicator calculates the areal distribution of different biome classes within an asset based on the IBPES biomes dataset.

Usage

```
calc_ipbes_biomes()
```

Details

The required resources for this indicator are:

- ipbes_biomes

Value

A function that returns an indicator tibble with the biome class as variable and the respective area (in ha) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_ipbes_biomes()) %>%
  calc_indicators(calc_ipbes_biomes()) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

iucn *IUCN Red List of Threatened Species*

Description

This resource is part of the spatial data set Red List of Threatened Species released by IUCN. It is free to use under a non-commercial licence. For commercial uses, a request has to be sent to Integrated Biodiversity Assessment Tool (IBAT).

Usage

```
get_iucn(paths = NULL)
```

Arguments

paths	A character vector to the respective species range files in GTiff format. Note, that theses files have to be downloaded manually.

Details

To use this data in mapme workflows, you will have to manually download the global data set and point towards the file path on your local machine. Please find the available data under the source link given below.

Value

A function that returns an `sf` footprint object.

Source

https://www.iucnredlist.org/resources/other-spatial-downloads

References

IUCN (2024). The IUCN Red List of Threatened Species. https://www.iucnredlist.org

---

`key_biodiversity_areas_indicator`
Calculate Key Biodiversity Areas

Description

This function calculates the total area of key biodiversity areas for a given input polygon.

Usage

```
calc_key_biodiversity_area()
```

Format

A function returning an indicator tibble with `key_biodiversity_area` as variable and the total overlap area (in ha) as value.

Details

The required resources for this indicator are:

- key_biodiversity_areas_resource

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- read_sf(
```

```
  system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
              package = "mapme.biodiversity"
))
kbas <- system.file("res", "key_biodiversity_areas", "kbas.gpkg",
                    package = "mapme.biodiversity")
aoi <- get_resources(aoi, get_key_biodiversity_areas(kbas))
aoi <- calc_indicators(aoi, calc_key_biodiversity_area())
aoi <- portfolio_long(aoi)

aoi

## End(Not run)
```

`key_biodiversity_areas_resource` *Key Biodiversity Areas*

Description

This resource contains outlines of key biodiversity areas, which are areas representing sites with specific importance for nature conservation.

Usage

```
get_key_biodiversity_areas(path = NULL)
```

Arguments

`path`	A character vector to the key biodiversity areas GPKG file. Note, that the file has to be downloaded manually.

Details

To use this data in mapme workflows, you will have to manually download the global data set and point towards its file path on your local machine. Please find the available data under the source link given below.

Value

A function that returns an `sf` footprints object.

Source

https://www.keybiodiversityareas.org/request-gis-data

References

BirdLife International (2024). The World Database of Key Biodiversity Areas. Developed by the KBA Partnership: BirdLife International, International Union for the Conservation of Nature, Amphibian Survival Alliance, Conservation International, Critical Ecosystem Partnership Fund, Global Environment Facility, Re:wild, NatureServe, Rainforest Trust, Royal Society for the Protection of Birds, Wildlife Conservation Society and World Wildlife Fund. Available at www.keybiodiversityareas.org.

`landcover` *Calculate area of different landcover classes*

Description

The land cover data shows us how much of the region is covered by forests, rivers, wetlands, barren land, or urban infrastructure thus allowing the observation of land cover dynamics over a period of time. This function allows to efficiently calculate area of different landcover classes for polygons. For each polygon, the area of the classes in hectare(ha) is returned.

Usage

```
calc_landcover()
```

Details

The required resources for this indicator are:

- esalandcover

Value

A function that returns an indicator tibble with landcover classes as variables and corresponding areas (in ha) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
```

```
  read_sf() %>%
  get_resources(get_esalandcover(years = 2016:2017)) %>%
  calc_indicators(calc_landcover()) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

make_footprints *Create footprints for vector or raster data sets*

Description

With this function you can create footprints for vector or raster datasets. Specify a character vector of GDAL readable sources of either vector or raster type. Internally, GDAL will be used to create an sf object with a single column indicating the source and the geometry indicating the bounding box of the respective source. Note, the performance for remote sources is dependent on your connection to the server. If you have other means to create footprints in your resource function (e.g. by using the output {`rstac::items_bbox()`}) you should prefer those means over this function for remote files.

Usage

```
make_footprints(
  srcs = NULL,
  filenames = if (inherits(srcs, "sf")) basename(srcs[["source"]]) else basename(srcs),
  what = c("vector", "raster"),
  oo = NULL,
  co = NULL,
  precision = 1e+05
)
```

Arguments

`srcs`	A character vector with GDAL readable paths to either vector or raster sources, then internal footprint functions are called, or an sf object which will be appended for filenames and potential options.
`filenames`	A character vector indicating the filenames of the source data sets if they were written to a destination. Defaults to `basename(srcs)` in case of character type or `basename(srcs[["source"]])` in case of an sf object.
`what`	A character vector indicating if the files are vector or raster files.
`oo`	Either a list or a character vector with opening options (-oo) of the respective GDAL driver. A list must have equal length of the input sources, a vector will be recycled.

co	Either a list or a character vector with creation options (-co) of the respective GDAL driver. A list must have equal length of the input sources, a vector will be recycled.
precision	A numeric indicating the precision of coordinates when a binary round-trip is done (see ?sf::st_as_binary()).

Value

An sf object with a the files sources and the geometry indicating their spatial footprint.

Examples

```
# a vector resource
# requires GDAL >= 3.7.0
if (FALSE) {
  vec <- system.file("shape/nc.shp", package = "sf")
  make_footprints(vec, what = "vector")
}

# a raster resource
ras <- system.file("ex/elev.tif", package = "terra")
make_footprints(ras, what = "raster")
```

---

make_global_grid *Helper to create a grid of regular resolution and CRS*

Description

Use this function to create a regular grid in a custom CRS. This is used e.g. to create the tile grid for Global Forest Watch in order to retrieve the intersecting tiles with a given portfolio.

Usage

```
make_global_grid(
  xmin = -180,
  xmax = 170,
  dx = 10,
  ymin = -50,
  ymax = 80,
  dy = 10,
  proj = NULL
)
```

Arguments

xmin	minimum longitude value (E/W)
xmax	maximum longitude value (E/W)
dx	difference in longitude value per grid

ymin	minimum latitude value (S/N)
ymax	maximum latitude value (E/W)
dy	difference in latitude value per grid
proj	projection system

Value

An sf object with a defined grid.

`mangroves_area` *Calculate mangrove extent based on Global Mangrove Watch (GMW)*

Description

This function allows to efficiently calculate area of mangrove from Global Mangrove Watch - World Conservation Monitoring Centre (WCMC) for polygons. For each polygon, the area of the mangrove (in hectare) for desired year is returned.

Usage

```
calc_mangroves_area()
```

Details

The required resources for this indicator are:

- gmw

Value

A function that returns an indicator tibble with mangroves as variable and corresponding areas (in ha) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
```

```
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_gmw(years = c(1996, 2016))) %>%
  calc_indicators(calc_mangroves_area()) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

mapme *Portfolio methods for mapme.biodiversity*

Description

`mapme_options()` sets default options for mapme.biodiversity to control the behaviour of downstream functions. Mainly, the output path as well as the chunk size (in ha), can be set. Additionally, the verbosity can be set and the path to a log directory can be controlled. It also allow to set the number of retries and the delay between the successive retries when retrieving the resources. Might be extended by other options in the future.

`get_resources()` data sets required for the calculation of indicators can be made available. The function supports the specification of several resource functions. To determine the output path, temporary directory and verbosity, the output of `mapme_options()` is used.

`calc_indicators()` calculates specific biodiversity indicators. A requirement is that the resources that are mandatory inputs for the requested indicators are available locally. Multiple indicators and their respective additional arguments can be supplied.

This function reads and crops available resources to the extent of a single asset. Specific resources can be queried. If not supplied (the default), all available resources will be prepared.

Usage

```
mapme_options(..., outdir, chunk_size, retries, delay, verbose, log_dir)

get_resources(x, ...)

calc_indicators(x, ...)

prep_resources(
  x,
  avail_resources = NULL,
  resources = NULL,
  mode = c("portfolio", "asset")
)
```

Arguments

...	One or more functions for resources/indicators
outdir	A length one character indicating the output path.
chunk_size	A numeric of length one giving the maximum chunk area in ha. Defaults to 100,000 ha. It refers to the area of an asset's bounding box. If it lies above the value chunk_size, splitting and chunking is considered. An asset will be processes as-is with an bounding box area below the specified value.
retries	An integer of length one indicating the number of retries the package should attempt to make a resource available. This commands the GDAL option GDAL_HTTP_MAX_RETRY (see here). Defaults to 3.
delay	An integer of length one indicating the number of seconds the package should wait between the successive retries. This commands the GDAL option GDAL_HTTP_RETRY_DELAY (see here). Defaults to 10.
verbose	A logical, indicating if informative messages should be printed.
log_dir	A character path pointing toward a GDAL-writable destination used to log erroneous assets. Defaults to NULL, meaning that erroneous assets will not be serialized to disk. If specified, a GPKG file named file.path(log_dir, paste0(Sys.Date(), "_mapme-error-assets.gpkg")) will be created and appended to in case of erroneous assets. In the case some of the resources could not be downloaded, their list will be written to a GPKG file named file.path(log_dir, paste0(Sys.Date(), "_", {resource_name}, "_mapme-error-resources.gpkg")).
x	An sf object with features of type "POLYGON"
avail_resources	A list object of available resources. If NULL (the default), the available resources will automatically be determined.
resources	A character vector with the resources to be prepared. If it it is NULL (the default) all available resources will be prepared.
mode	A character indicating the reading mode, e.g. either "portfolio" (the default) or "asset".

Value

mapme_options() returns a list of options if no arguments are specified. Otherwise sets matching arguments to new values in the package's internal environment.

get_resources() is called for its side effect of making resources available in the package environment. Returns x, invisibly.

calc_indicators() returns x, invisibly, with an additional nested list column per requested indicator.

prep_resources() returns a list with prepared vector and raster resources as sf and SpatRaster-objects.

Examples

```
library(mapme.biodiversity)
mapme_options()
```

`mcd64a1` *MODIS Burned Area Monthly (MCD64A1)*

Description

The Terra and Aqua combined MCD64A1 Version 6.1 Burned Area data product is a monthly, global gridded 500 meter (m) product containing per-pixel burned-area and quality information. The MCD64A1 burned-area mapping approach employs 500 m Moderate Resolution Imaging Spectroradiometer (MODIS) Surface Reflectance imagery coupled with 1 kilometre (km) MODIS active fire observations.

Usage

```
get_mcd64a1(years = 2000:2022)
```

Arguments

`years`	Numeric vector of years to make the MCD64A1 product available for. Must be greater than the year 2000.

Details

The algorithm uses a burn sensitive Vegetation Index (VI) to create dynamic thresholds that are applied to the composite data. The VI is derived from MODIS shortwave infrared atmospherically corrected surface reflectance bands 5 and 7 with a measure of temporal texture. The algorithm identifies the date of burn for the 500 m grid cells within each individual MODIS tile. The date is encoded in a single data layer as the ordinal day of the calendar year on which the burn occurred with values assigned to unburned land pixels and additional special values reserved for missing data and water grid cells.

Value

A function that returns an `sf` footprint object.

Source

https://planetarycomputer.microsoft.com/dataset/modis-64A1-061

References

Giglio, L., C. Justice, L. Boschetti, D. Roy. MODIS/Terra+Aqua Burned Area Monthly L3 Global 500m SIN Grid V061. 2021, distributed by NASA EOSDIS Land Processes Distributed Active Archive Center. doi:10.5067/MODIS/MCD64A1.061

nasa_grace *NASA GRACE-based Drought Indicator layer*

Description

The resource is published by NASA GRACE Tellus. This data set reflects on potential drought conditions in the shallow groundwater section relative to a reference period spanning from 1948 to 2012. It is available as a global raster with a weekly temporal resolution starting with the year 2003. The value indicates the wetness percentile of a given pixel with regard to the reference period.

Usage

```
get_nasa_grace(years = 2003:2022)
```

Arguments

years	A numeric vector indicating the years for which to make the resource available.

Value

A function that returns an `sf` footprint object.

nasa_srtm *NASADEM HGT v001*

Description

This resource is processed by the Land Processes Distributed Active Archive Center (LP DAAC) and made available at the Microsoft Planetery Computer. NASADEM are distributed in 1 degree latitude by 1 degree longitude tiles and consist of all land between 60° N and 56° S latitude. This accounts for about 80% of Earth's total landmass.

Usage

```
get_nasa_srtm()
```

Value

A function that returns an `sf` footprint object.

Source

https://planetarycomputer.microsoft.com/dataset/nasadem

References

NASA JPL (2020). NASADEM Merged DEM Global 1 arc second V001. NASA EOSDIS Land Processes DAAC. Accessed 2023-07-01 from doi:10.5067/MEaSUREs/NASADEM/NASADEM_HGT.001

`nelson_et_al` *Accessibility to Cities layer*

Description

This resource is published by Weiss et al. (2018) "A global map of travel time to cities to assess inequalities in accessibility in 2015" on journal nature. Accessibility is the ease with which larger cities can be reached from a certain location. This resource represents the travel time to major cities in the year 2015. Encoded as minutes, representing the time needed to reach that particular cell from nearby city of target population range. The following ranges to nearby cities are available:

- "5k_10k"
- "10k_20k"
- "20k_50k"
- "50k_100k"
- "100k_200k"
- "200k_500k"
- "500k_1mio"
- "1mio_5mio"
- "50k_50mio"
- "5k_110mio"
- "20k_110mio"
- "5mio_50mio"

Usage

```
get_nelson_et_al(ranges = "20k_50k")
```

Arguments

`ranges`	A character vector indicating one or more ranges to download.

Value

A function that returns an `sf` footprint object.

Note

Note, that the 'figshare' server applies a rather restrictive rate limit thus frequently resulting in opaque error codes (see https://github.com/mapme-initiative/mapme.biodiversity/issues/308). Please set GDAL configuration options to sensible values in case you are running into this issue, e.g.: `Sys.setenv("GDAL_HTTP_MAX_RETRY" = "5", "GDAL_HTTP_RETRY_DELAY" = "15")`.

Source

https://figshare.com/articles/dataset/Travel_time_to_cities_and_ports_in_the_year_2015/7638134/3

References

Weiss, D. J., Nelson, A., Gibson, H. S., Temperley, W., Peedell, S., Lieber, A., ... & Gething, P. W. (2018). A global map of travel time to cities to assess inequalities in accessibility in 2015. Nature, 553(7688), 333-336.

`population_count` *Calculate population count statistics*

Description

WorldPop, which was initiated in 2013, offers easy access to spatial demographic datasets, claiming to use peer-reviewed and fully transparent methods to create global mosaics for the years 2000 to 2020. This function allows to efficiently calculate population count statistics (e.g. total number of population) for polygons. For each polygon, the desired statistic/s (min, max, sum, mean, median, sd or var) is/are returned.

Usage

```
calc_population_count(engine = "extract", stats = "sum")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either one or multiple inputs as character "min", "max", "sum", "mean", "median" "sd" or "var".

Details

The required resources for this indicator are:

- worldpop

Value

A function that returns an indicator tibble with the specified populations statistics as variable and the corresponding values as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_worldpop(years = 2010:2020)) %>%
  calc_indicators(
    calc_population_count(engine = "extract", stats = c("sum", "median"))
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`portfolio` *Portfolio methods*

Description

`write_portfolio()` writes a processed biodiversity portfolio to disk. Portfolio data will only be serialized to disk as a GeoPackage including two tables: `metadata` and `indicators`. The `metadata` tables includes, among other simple variables the geometries and a primary key called `assetid`. The 'indicators' tables includes the foreign key `assetid`, a column called `indicator` giving the name of the original indicator as well as the standard indicator columns `datetime`, `variable`, `unit`, and `value`. For convenience, use `read_portfolio()` to read such a portfolio GeoPackage back into R.

`portfolio_long()` transforms a portfolio to long-format, potentially dropping geometries in the process.

`portfolio_wide()` transforms a portfolio to wide-format, potentially dropping geometries in the process.

Usage

```
write_portfolio(x, dsn, ...)

read_portfolio(src, ...)
```

```
portfolio_long(x, indicators = NULL, drop_geoms = FALSE)

portfolio_wide(x, indicators = NULL, drop_geoms = FALSE)
```

Arguments

x	A portfolio object processed with mapme.biodiversity.
dsn	A file path for the output file (must end with gpkg).
...	Additional arguments supplied to write_sf() or read_sf()
src	A character vector pointing to a GeoPackage that has been previously written to disk via write_portfolio()
indicators	If NULL (the default), all indicator columns will be detected and transformed automatically. If a character vector is supplied, only those indicators will be transformed.
drop_geoms	A logical, indicating if geometries should be dropped.

Value

write_portfolio() returns dsn, invisibly.

read_portfolio() returns an sf object object with nested list columns for every indicator found in the GeoPackage source file.

portfolio_long() returns the portfolio object in long-format.

portfolio_wide() returns the portfolio object in wide-format.

---

precipitation_chelsa *Calculate precipitation average based on CHELSA*

Description

This functions allows to calculate averaged precipitation from the CHELSA downscaled precipitation layers. Based on user-selected years, monthly averages of precipitation are calculated.

Usage

```
calc_precipitation_chelsa(years = 1979:2018, engine = "extract")
```

Arguments

years	A numeric vector indicating the years for which to calculate precipitation statistics.
engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.

Details

The required resources for this indicator are:

- chelsa

Value

A function that returns an indicator tibble with variable precipitation and sum of precipitation (in mm/m^2) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_chelsa(years = 2010)) %>%
  calc_indicators(
    calc_precipitation_chelsa(
      years = 2010,
      engine = "extract"
    )
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

precipitation_chirps *Calculate precipitation sums based on CHIRPS*

Description

This functions allows to calculate precipitation sums based on the CHIRPS rainfall estimates. Corresponding to the time-frame of the analysis of the portfolio, monthly precipitation sums are calculated.

Usage

```
calc_precipitation_chirps(years = 1981:2020, engine = "extract")
```

Arguments

years	A numeric vector indicating the years for which to calculate precipitation statistics.
engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.

Details

The required resources for this indicator are:

- chirps

Value

A function that returns an indicator tibble with variable precipitation and sum of precipitation (in mm) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_chirps(years = 2010)) %>%
  calc_indicators(
    calc_precipitation_chirps(
      years = 2010,
      engine = "extract"
    )
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

precipitation_wc *Calculate precipitation statistics*

Description

This function allows to efficiently calculate precipitation statistics from Worldclim for polygons. For each polygon, the desired statistic/s (min, max, sum, mean, median, sd or var) is/are returned.

Usage

```
calc_precipitation_wc(engine = "extract", stats = "mean")
```

Arguments

engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
stats	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resources for this indicator are:

- precipitation layer from worldclim_precipitation

Value

A function that returns an indicator tibble with precipitation statistics as variable and corresponding values as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
```

```
  get_resources(get_worldclim_precipitation(years = 2018)) %>%
  calc_indicators(
    calc_precipitation_wc(
      engine = "extract",
      stats = c("mean", "median")
    )
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`resources` *Register or list resources in mapme.biodiversity*

Description

`register_resource()` is used to register a new resource function with base information to the package's internal environment used to inform users about available resources. Note, registering a custom resource will only have effect for the current R session.

`available_resources()` returns a tibble of registered resources with basic information such as the source and the licence.

Usage

```
register_resource(
  name = NULL,
  description = NULL,
  licence = NULL,
  source = NULL,
  type = NULL
)

available_resources(resources = NULL)
```

Arguments

`name`	A character vector indicating the name of the resource.
`description`	A character vector with a basic description
`licence`	A character vector indicating the licence of the resource. In case it is a custom licence, put a link to the licence text.
`source`	Optional, preferably a URL where the data is found.
`type`	A character vector indicating the type of the resource. Either 'vector' or 'raster'.
`resources`	If `NULL` returns a list of all resources (default). Otherwise only the ones specified.

Value

`register_resource()` is called for the side-effect of registering a resource.

`available_resources()` returns a tibble listing available resources.

Examples

```
## Not run:
register_resource(
  name = "gfw_treecover",
  description = "Global Forest Watch - Percentage of canopy closure in 2000",
  licence = "CC-BY 4.0",
  source = "https://data.globalforestwatch.org/documents/tree-cover-2000/explore",
  type = "raster"
)

## End(Not run)
available_resources()
```

---

slope *Calculate slope statistics*

Description

This function allows to calculate slope statistics for polygons. For each polygon, the desired statistic(s) are returned.

Usage

```
calc_slope(engine = "exactextract", stats = "mean")
```

Arguments

`engine`	The preferred processing function from either one of "zonal", "extract" or "exactextract" as a character string.
`stats`	Function to be applied to compute statistics for polygons. Accepts either a single string or a vector of strings, such as "mean", "median", "sd", "min", "max", "sum", or "var".

Details

The required resource for this indicator is:

- nasa_srtm

Value

A function that returns an indicator tibble with specified slope statistics as variables and corresponding values (in degrees).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nasa_srtm()) %>%
  calc_indicators(
    calc_slope(stats = c("mean", "median", "sd", "var"), engine = "extract")
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`soilgrids` *SoilGrids data layers*

Description

SoilGrids is a project combining global observation data with machine learning to map the spatial distribution of soil properties across the globe. It is produced at a spatial resolution of 250 meters and each parameter is mapped at different depths. In order to be able to assess prediction uncertainty, besides the mean and median prediction, the 0.05 and 0.95 percentile predictions are available. The following parameters are available:

bdod Bulk density of the fine earth fraction (kg/dm3)

cec Cation Exchange Capacity of the soil (cmol(c)/kg)

cfvo Volumetric fraction of coarse fragments > 2 mm (cm3/100cm3 (volPerc))

clay Proportion of clay particles < 0.002 mm in the fine earth fraction (g/100g)

nitrogen Total nitrogen (g/kg)

phh2o Soil pH (pH)

sand Proportion of sand particles > 0.05 mm in the fine earth fraction (g/100g)

silt Proportion of silt particles >= 0.002 mm and <= 0.05 mm in the fine earth fraction (g/100g)

soc Soil organic carbon content in the fine earth fraction (g/kg)

ocd Organic carbon density (kg/m3)

ocs Organic carbon stocks (kg/m²)

Usage

```
get_soilgrids(layers, depths, stats)
```

Arguments

`layers`	A character vector indicating the layers to download from soilgrids
`depths`	A character vector indicating the depths to download
`stats`	A character vector indicating the statistics to download.

Details

Except for `ocs`, which is only available for a depth of `"0-30cm"`, all other parameters are available at the following depths:

- "0-5cm"
- "5-15cm"
- "15-30cm"
- "30-60cm"
- "60-100cm"
- "100-200cm"

Each parameter and depth is available for the following statistics:

- "Q0.05"
- "Q0.50"
- "mean"
- "Q0.95"

Value

A function that returns an `sf` footprint object.

Source

https://isric.org/explore/soilgrids

References

Poggio, L., de Sousa, L. M., Batjes, N. H., Heuvelink, G. B. M., Kempen, B., Ribeiro, E., and Rossiter, D.: SoilGrids 2.0: producing soil information for the globe with quantified spatial uncertainty, SOIL, 7, 217–240, 2021. doi:10.5194/soil72172021

`soilproperties` *Calculate Zonal Soil Properties*

Description

This indicator allows the extraction of zonal statistics for resource layers previously downloaded from SoilGrids, thus in total supporting the calculation of zonal statistics for 10 different soil properties at 6 different depths for a total of 4 different model outputs (stat). Zonal statistics will be calculated for all SoilGrid layers that have been previously made available vie `get_resources()`.

Usage

```
calc_soilproperties(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resource for this indicator is:

- soilgrids

Value

A function that returns an indicator tibble with soilgrid layers and statistics as variables and the corresponding statistics as value.

Examples

```
if (FALSE) {
  library(sf)
  library(mapme.biodiversity)

  mapme_options(
    outdir = NULL,
    verbose = FALSE
  )

  aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
    package = "mapme.biodiversity"
  ) %>%
    read_sf() %>%
    get_resources(
```

```
      get_soilgrids(
        layers = "clay",
        depths = "0-5cm",
        stats = "mean"
      )
    ) %>%
    calc_indicators(
      calc_soilproperties(engine = "extract", stats = c("mean", "median"))
    ) %>%
    portfolio_long()

  aoi
}
```

---

spds_exists *Check if a spatial data sets exists*

Description

This function uses a file path readable by GDAL to check if it can query it for information. Note, this should also work for remote files, e.g. in an S3 bucket. You can use this function in your custom resource function to query if a file is already present at the destination. Note, that performance will be dependent on your connection to the server. It can also be used for files on the local file system.

Usage

```
spds_exists(path, oo = character(0), what = c("vector", "raster"))
```

Arguments

path	A length 1 character vector with a GDAL readable file path.
oo	Either a list or a character vector with opening options (-oo) of the respective GDAL driver. A list must have equal length of the input sources, a vector will be recycled.
what	A character vector indicating if the resource is a vector or raster file.

Value

A logical, TRUE if the file exists, FALSE if it does not.

Examples

```
# a vector resource
vec <- system.file("shape/nc.shp", package = "sf")
spds_exists(vec, what = "vector")

# a raster resource
ras <- system.file("ex/elev.tif", package = "terra")
spds_exists(ras, what = "raster")
```

```
# a non existing file
spds_exists("not-here.gpkg", what = "vector")
```

`species_richness` *Species richness based on IUCN raster data*

Description

Species richness counts the number of potential species intersecting with a polygon grouped by the IUCN threat categorization. Note, that this indicator function requires the manual download of the respective raster files.

Usage

```
calc_species_richness(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either one or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The specific meaning of the species richness indicator depends on the supplied raster file.

The required resources for this indicator are:

- iucn

Value

A function that returns an indicator tibble with IUCN layers with specified statistics as variable and respective species richness (count) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
```

```
  verbose = FALSE
)

iucn_dir <- system.file("res", "iucn", package = "mapme.biodiversity")
sr_rasters <- list.files(iucn_dir, pattern = "*_SR_*", full.names = TRUE)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_iucn(sr_rasters)) %>%
  calc_indicators(calc_species_richness(stats = "median")) %>%
  portfolio_long()

aoi

## End(Not run)
```

`temperature_max_wc` *Calculate maximum temperature statistics*

Description

This function allows to efficiently calculate maximum temperature statistics from Worldclim for polygons. For each polygon, the desired statistic/s (min, max, sum, mean, median, sd or var) is/are returned.

Usage

```
calc_temperature_max_wc(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resources for this indicator are:

- maximum temperature layer from worldclim_max_temperature

Value

A function that returns an indicator tibble with maximum temperature statistics as variables and corresponding values as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_worldclim_max_temperature(years = 2018)) %>%
  calc_indicators(
    calc_temperature_max_wc(
      engine = "extract",
      stats = c("mean", "median")
    )
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`temperature_min_wc` *Calculate minimum temperature statistics based on WorldClim*

Description

This function allows to efficiently calculate minimum temperature statistics from Worldclim for polygons. For each polygon, the desired statistic/s (min, max, sum, mean, median, sd or var) is/are returned.

Usage

```
calc_temperature_min_wc(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resources for this indicator are:

- minimum temperature layer from worldclim_min_temperature

Value

A function that returns an indicator tibble with minimum temperature statistics as variables and corresponding values as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_worldclim_min_temperature(years = 2018)) %>%
  calc_indicators(
    calc_temperature_min_wc(
      engine = "extract",
      stats = c("mean", "median")
    )
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

teow *Terrestrial Ecoregions of the World (TEOW) Polygon*

Description

This resource is part of the publication by Olson et al. (2004) "Terrestrial Ecosystems of the World (TEOW) from WWF-US (Olson)". It depicts 867 terrestrial ecoregions around the world classified into 14 different terrestrial biomes such as forests, grasslands, or deserts. The polygons represent the ecoregions, defined as relatively large units of land or inland water sharing a large majority of biodiversity. The datasets is made available from World Wildlife Fund (WWF) for the year 2001.

Usage

```
get_teow(path = NULL)
```

Arguments

path	A character vector to the Terrestrial Ecoregions of the World (TEOW) zip file. Note, that the file has to be downloaded manually.

Value

A function that returns an sf footprint object.

Source

https://files.worldwildlife.org/wwfcmsprod/files/Publication/file/6kcchn7e3u_official_teow.zip

References

Olson, D. M., Dinerstein, E., Wikramanayake, E. D., Burgess, N. D., Powell, G. V. N., Underwood, E. C., D'Amico, J. A., Itoua, I., Strand, H. E., Morrison, J. C., Loucks, C. J., Allnutt, T. F., Ricketts, T. H., Kura, Y., Lamoreux, J. F., Wettengel, W. W., Hedao, P., Kassem, K. R. 2001. Terrestrial ecoregions of the world: a new map of life on Earth. Bioscience 51(11):933-938. doi:10.1641/00063568(2001)051[0933:TEOTWA]2.0.CO;2

---

traveltime *Calculate accessibility statistics*

Description

Accessibility is the ease with which larger cities can be reached from a certain location. This function allows to efficiently calculate accessibility statistics (i.e. travel time to nearby major cities) for polygons. For each polygon, the desired statistic/s (mean, median or sd) is/are returned.

Usage

```
calc_traveltime(engine = "extract", stats = "mean")
```

Arguments

engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
stats	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resources for this indicator are:

- nelson_et_al

Value

A function that returns an indicator tibble with city ranges and statistics as variable and corresponding values (in minutes) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nelson_et_al(ranges = "100k_200k")) %>%
  calc_indicators(
    calc_traveltime(engine = "extract", stats = c("min", "max"))
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`traveltime_2000` *Calculate accessibility statistics for the year 2000*

Description

Accessibility refers to the ease with which cities can be reached from a certain location. This function allows efficient calculation of accessibility statistics (i.e., travel time to the nearest city) for polygons. For each polygon, the desired statistic/s (mean, median or sd) is/are returned.

Usage

```
calc_traveltime_2000(engine = "extract", stats = "mean")
```

Arguments

engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
stats	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum", "var".

Details

The required resource for this indicator is:

- accessibility_2000

Value

A function that returns an indicator tibble with accessibility statistics for the year 2000 as variables and corresponding values (in minutes) as values.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_accessibility_2000()) %>%
  calc_indicators(
    calc_traveltime_2000(stats = c("mean", "median", "sd"), engine = "extract")
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`treecoverloss_emissions` *Calculate emission statistics*

Description

This functions allows to efficiently calculate emission statistics for areas of interest. For each year in the analysis timeframe, the forest losses from Hansen et al. (2013) are overlayed with the respective emission layer from Harris et al. (2021) and area-wise emission statistics are calculated for each year.

Usage

```
calc_treecoverloss_emissions(years = 2000:2024, min_size = 10, min_cover = 35)
```

Arguments

`years`	A numeric vector with the years for which to calculate emissions caused by treecover loss.
`min_size`	The minimum size of a forest patch in ha.
`min_cover`	The minimum threshold of stand density for a pixel to be considered forest in the year 2000.

Details

The required resources for this indicator are:

- gfw_treecover
- gfw_lossyear
- gfw_emissions

Value

A function that returns an indicator tibble with emissions as variable and emitted CO2 equivalent (in Mg) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
```

```
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(
    get_gfw_treecover(version = "GFC-2024-v1.12"),
    get_gfw_lossyear(version = "GFC-2024-v1.12"),
    get_gfw_emissions()
  ) %>%
  calc_indicators(
    calc_treecoverloss_emissions(years = 2016:2017, min_size = 1, min_cover = 30)
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`treecover_area` *Calculate treecover statistics*

Description

This functions allows to efficiently calculate treecover statistics for polygons. For each year in the analysis timeframe, the forest losses in preceding and the current years are subtracted from the treecover in the year 2000 and actual treecover figures within the polygon are returned.

Usage

```
calc_treecover_area(years = 2000:2024, min_size = 10, min_cover = 35)
```

Arguments

`years`	A numeric vector with the years for which to calculate treecover area.
`min_size`	The minimum size of a forest patch to be considered as forest in ha.
`min_cover`	The minimum cover percentage per pixel to be considered as forest.

Details

The required resources for this indicator are:

- gfw_treecover
- gfw_lossyear

Value

A function that returns an indicator tibble with variable treecover and corresponding area (in ha) as value.

If the gfw_treecover resource for an asset contains only zeros (no trees at all, e.g. big water body like an ocean or a sea), the NULL value is returned instead of the indicator tibble. If it doesn't contain any pixels with value >= min_cover, indicator tibble reports area of zero hectares.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(
    get_gfw_treecover(version = "GFC-2024-v1.12"),
    get_gfw_lossyear(version = "GFC-2024-v1.12")
  ) %>%
 calc_indicators(calc_treecover_area(years = 2016:2017, min_size = 1, min_cover = 30)) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

treecover_area_and_emissions

Calculate treeloss statistics

Description

This functions allows to efficiently calculate the treecover and emissions indicators in a single function call together. Since most of the pre-processing operations for treecover and emissions are the same, it is more efficient to calculate them in one run if users are actually interested in both statistics. Otherwise users are advised to use the respective single indicator functions.

Usage

```
calc_treecover_area_and_emissions(
  years = 2000:2024,
  min_size = 10,
  min_cover = 35
)
```

Arguments

years	A numeric vector with the years for which to calculate treecover area and emissions.
min_size	The minimum size of a forest patch in ha.
min_cover	The minimum threshold of stand density for a pixel to be considered forest in the year 2000.

Details

The required resources for this indicator are:

- gfw_treecover
- gfw_lossyear
- gfw_emissions

Value

A function that returns an indicator tibble with variables treecover and emissions and corresponding values (in ha and Mg) as value.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(
    get_gfw_treecover(version = "GFC-2024-v1.12"),
    get_gfw_lossyear(version = "GFC-2024-v1.12"),
    get_gfw_emissions()
```

```
  ) %>%
  calc_indicators(
    calc_treecover_area_and_emissions(years = 2016:2017, min_size = 1, min_cover = 30)
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

`tri` *Calculate Terrain Ruggedness Index (TRI) statistics*

Description

Terrain Ruggedness Index is a measurement developed by Riley, et al. (1999). The elevation difference between the centre pixel and its eight immediate pixels are squared and then averaged and its square root is taken to get the TRI value. This function allows to calculate terrain ruggedness index (tri) statistics for polygons. For each polygon, the desired statistic(s) are returned.

Usage

```
calc_tri(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The range of index values and corresponding meaning:

- 0-80 m - level surface
- 81-116 m - nearly level surface
- 117-161 m - slightly rugged surface
- 162-239 m - intermediately rugged surface
- 240-497 m - moderately rugged surface
- 498-958 m - highly rugged surface
- 959-4367 m extremely rugged surface

The required resources for this indicator are:

- nasa_srtm

Value

A function that returns an indicator tibble with tri as variable and the respective statistic as value.

References

Riley, S. J., DeGloria, S. D., & Elliot, R. (1999). Index that quantifies topographic heterogeneity. Intermountain Journal of Sciences, 5(1-4), 23-27.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nasa_srtm()) %>%
  calc_indicators(
    calc_tri(stats = c("mean", "median", "sd", "var"), engine = "extract")
  ) %>%
  portfolio_long()

aoi

## End(Not run)
```

---

ucdp_ged *UCDP Georeferenced Event Dataset (UCDP GED)*

Description

This resource distributed by the Uppsala Conflict Data Program (UCDP) constitutes its most disaggregated dataset on individual events of organized violence. It encodes the different actors involved, is spatially disaggregated down to village levels anc currently covers the time period of 1989 to 2021. Older versions of the data set can be downloaded, but users are recommended to download the latest data set.

Usage

```
get_ucdp_ged(version = "latest")
```

Arguments

`version`	A character vector specifying the version to download. Defaults to "latest".

Details

The following versions are available:

- 5.0
- 17.1
- 17.2
- 18.1
- 19.1
- 20.1
- 21.1
- 22.1
- 23.1
- 24.1
- latest

Value

A function that returns an `sf` footprint object.

Source

https://ucdp.uu.se/downloads/

References

Davies, Shawn, Therese Pettersson & Magnus Öberg (2022). Organized violence 1989-2021 and drone warfare. Journal of Peace Research 59(4). doi:10.1177/00223433221108428

`worldclim_max_temperature` *Downloads WorldClim Maximum Temperature layer*

Description

This resource is published by Fick et al. (2017) "WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas" and represents multiple climatic variables from which we will be requiring minimum temperature, maximum temperature, and mean precipitation layers. The layers are available to download for the period 1960 - 2024 on monthly basis from WorldClim.

Usage

```
get_worldclim_max_temperature(
  years = 2000:2024,
  resolution = c("2.5m", "5m", "10m")
)
```

Arguments

years	A numeric vector indicating for which years to make the resource available. Defaults to 2000:2024.
resolution	A character vector indicating the desired resolution: "2.5m", "5m", or "10m".

Details

This resource represents the maximum temperature, layers available to download for the period 1960 - 2024 on monthly basis from WorldClim. Encoded as (°C), representing the maximum temperature per output grid cell.

Value

A function that returns an `sf` footprint object.

Source

https://www.worldclim.org/data/index.html

---

worldclim_min_temperature

Downloads WorldClim Minimum Temperature layer

Description

This resource is published by Fick et al. (2017) "WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas" and represents multiple climatic variables from which we will be requiring minimum temperature, maximum temperature, and mean precipitation layers. The layers are available to download for the period 1960 - 2024 on monthly basis from WorldClim.

Usage

```
get_worldclim_min_temperature(
  years = 2000:2024,
  resolution = c("2.5m", "5m", "10m")
)
```

Arguments

years	A numeric vector indicating for which years to make the resource available. Defaults to 2000:2024.
resolution	A character vector indicating the desired resolution: "2.5m", "5m", or "10m".

Details

This resource represents the minimum temperature, layers available to download for the period 1960 - 2024 on monthly basis from WorldClim. Encoded as (°C), representing the minimum temperature per output grid cell.

Value

A function that returns an `sf` footprint object.

Source

https://www.worldclim.org/data/index.html

---

worldclim_precipitation

Downloads WorldClim Mean Precipitation layer

Description

This resource is published by Fick et al. (2017) "WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas" and represents multiple climatic variables from which we will be requiring minimum temperature, maximum temperature, and mean precipitation layers. The layers are available to download for the period 1960 - 2024 on monthly basis from WorldClim.

Usage

```
get_worldclim_precipitation(
  years = 1960:2024,
  resolution = c("2.5m", "5m", "10m")
)
```

Arguments

years	A numeric vector indicating for which years to make the resource available.
resolution	A character vector indicating the desired resolution.

Details

This resource represents the average precipitation, layers available to download for the period 1960 - 2024 on monthly basis from WorldClim. Encoded as (mm), representing the mean precipitation per output grid cell.

Value

A function that returns an `sf` footprint object.

Source

https://www.worldclim.org/data/index.html

`worldpop` *Population Count layer for year 2000-2020*

Description

This resource is published by open spatial demographic data and research organization called WorldPop. This resource represents the population count, 1 km spatial resolution layers available to download from the year 2000 to 2020. The dataset is called as WorldPop Unconstrained Global Mosaics. The encoded cell value represents the total number of people in that particular grid cell.

Usage

```
get_worldpop(years = 2000)
```

Arguments

`years`	A numeric vector indicating the years for which to make the resource available.

Details

It may be required to increase the timeout option to successfully download theses WorldPop layers from their source location via e.g. `options(timeout = 600)`.

Value

A function that returns an `sf` footprint object.

Source

https://www.worldpop.org/

Index