## Package ‘glottospace’

**Type** Package  
**Title** Language Mapping and Geospatial Analysis of Linguistic and Cultural Data  
**Version** 0.0.111  
**Author** Sietze Norder  
**Maintainer** Sietze Norder <mail@sietzenorder.nl>  
**Description** Streamlined workflows for geolinguistic analysis, including: accessing global linguistic and cultural databases, data import, data entry, data cleaning, data exploration, mapping, visualization and export.  
**License** GPL (>= 3)  
**URL** https://github.com/SietzeN/glottospace  
**BugReports** https://github.com/SietzeN/glottospace/issues  
**Depends** R (>= 4.1.0)  
**Imports** dplyr, ggplot2, magrittr, purrr, readxl, rlang, rnaturalearth, rnaturalearthdata, sf, tibble, tidyr, tmap, units, writexl  
**Suggests** cluster, grDevices, htmlwidgets, jsonlite, mapedit, plotly, RColorBrewer, s2, scales, vegan, viridisLite, xml2  
**Encoding** UTF-8  
**Language** en-US  
**LazyData** true  
**RoxygenNote** 7.1.2  
**NeedsCompilation** no  
**Repository** CRAN  
**Date/Publication** 2022-03-16 13:10:02 UTC
**R topics documented:**

- glottobooster .............................................. 2
- glottocheck .................................................. 3
- glottoclean .................................................. 4
- glottocode_exists .......................................... 5
- glottoconvert ................................................ 6
- glottocreate ................................................ 7
- glottocreate_addtable ..................................... 9
- glottodist ................................................... 9
- glottofilter ................................................. 10
- glottofiltermap ............................................. 11
- glottoget .................................................... 12
- glottojoin ................................................... 13
- glottomap ..................................................... 14
- glottonmds ................................................... 16
- glottoplot .................................................... 17
- glottosave ................................................... 18
- glottosearch ............................................... 19
- glottosimplify ............................................. 20
- glottospace .................................................. 21
- glottosplitmergemeta ..................................... 22
- glottospotlight ............................................. 22

**Index**

| glottobooster | Enhance glottolog data |

**Description**

This function restructures glottolog data, and optionally adds/removes data. If you want more flexibility in choosing which data to add/remove, you can use glottoboosterflex().

**Usage**

```r
glottobooster(
  glottologdata = NULL,
  space = TRUE,
  addfamname = TRUE,
  addisolates = TRUE,
  L1only = TRUE,
  addfamsizerank = TRUE
)
```
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glottologdata</td>
<td>data from glottolog, can be downloaded with glottoget(&quot;glottolog&quot;).</td>
</tr>
<tr>
<td>space</td>
<td>Return spatial object?</td>
</tr>
<tr>
<td>addfamname</td>
<td>Add column with familiy names?</td>
</tr>
<tr>
<td>addisolates</td>
<td>Add column to identify isolates?</td>
</tr>
<tr>
<td>L1only</td>
<td>Keep only L1 languages (remove bookkeeping, unclassifiable, sign languages, etc.).</td>
</tr>
<tr>
<td>addfamsize</td>
<td>Add column with family size?</td>
</tr>
<tr>
<td>addfamsizerank</td>
<td>Add column with family size rank?</td>
</tr>
</tbody>
</table>

Details

This function is used to generate 'glottobase' (the reference dataset used throughout the glottospace R package). The default options generate 'glottobase', which can be loaded directly using glottoget("glottobase").

Value

glottologdata object, either a spatial object (class: sf) or a data.frame.

See Also

Other <glottobooster>: glottoboosterflex()

Examples

```r
# Load glottolog
#glottologdata <- glottoget("glottolog")
#glottobase <- glottobooster(glottologdata)
```

Description

This function first checks whether a dataset is glottodata or glottosubdata, and depending on the result calls glottocheck_data or glottocheck_subdata.

Usage

```r
glottocheck(glottodata, diagnostic = TRUE, checkmeta = FALSE)
```
Arguments

<table>
<thead>
<tr>
<th>glottodata</th>
<th>User-provided glottodata</th>
</tr>
</thead>
<tbody>
<tr>
<td>diagnostic</td>
<td>If TRUE (default) a data viewer will be opened to show the levels of each variable (including NAs), and a data coverage plot will be shown.</td>
</tr>
<tr>
<td>checkmeta</td>
<td>Should metadata be checked as well?</td>
</tr>
</tbody>
</table>

Details

It subsequently checks whether:

- one column exists with the name "glottocode"
- there are rows without a glottocode (missing IDs)
- there are rows with duplicated glottocodes (duplicate IDs)
- all variables have at least two levels
- all glottocodes are valid

Value

Diagnostic messages highlighting potential issues with glottodata or glottosubdata.

Examples

```r
glottodata <- glottoget("demodata")
glottocheck(glottodata, diagnostic = FALSE)
```

---

**glottoclean** 
Clean glottodata

Description

This function is a wrapper around glottorecode. This function has some built in default values that are being recoded. For example, if column type is 'symm' or 'asymm', values such as "No" and 0 are recoded to FALSE and "?" is recoded to NA. Use glottorecode directly if you don’t want to use these defaults.

Usage

```r
glottoclean(
  glottodata, 
  structure = NULL, 
  tona = NULL, 
  tofalse = NULL, 
  totrue = NULL
)
```
**glottocode_exists**

**Arguments**

- **glottodata**: glottodata (either a list or a data.frame)
- **structure**: Optional structure table (should be provided if glottodata doesn’t contain any)
- **tona**: Optional additional values to recode to NA (besides default)
- **tofalse**: Optional additional values to recode to FALSE (besides default)
- **totrue**: Optional additional values to recode to TRUE (besides default)

**Value**

A cleaned-up version of the original glottodata object (either a list or a data.frame, depending on the input)

**Examples**

```r
glottodata <- glottoget("demodata", meta = TRUE)
glottodata <- glottoclean(glottodata)
```

---

**glottocode_exists**  
*Check whether a set of glottocodes exist in glottolog*

**Description**

Checks whether a set of glottocodes exist in glottolog (checked at the level of L1 languages)

**Usage**

```r
glottocode_exists(glottocode)
```

**Arguments**

- **glottocode**: A glottocode or character vector of glottocodes

**Value**

A logical vector

**Examples**

```r
glottocode_exists(c("yucu1253"))
glottocode_exists(c("yucu1253", "abcd1234"))
```
glottoconvert

Convert a linguistic dataset into glottodata or glottosubdata

Description

Convert a linguistic dataset into glottodata or glottosubdata

Usage

```r
glottoconvert(
  data,
  var,
  glottocodes = NULL,
  table = NULL,
  glottocolumn = NULL,
  glottosubcolumn = NULL,
  ref = NULL,
  page = NULL,
  remark = NULL,
  contributor = NULL,
  varnamecol = NULL
)
```

Arguments

- **data**: A dataset that should be converted into glottodata/glottosubdata. This will generally be an excel file loaded with glottoget(). The dataset will be converted into glottodata if:
  - all data are stored in a single table, or
  - the dataset contains several tables of which one is called 'glottodata', or
  - a table argument is provided.
  Otherwise, glottospace will attempt to convert the dataset into glottosubdata. This works if:
  - table names are glottocodes, and
  - an argument is provided to glottocodes, or the dataset contains a sample table from which glottocodes can be obtained.

- **var**: Character string that distinguishes those columns which contain variable names.

- **glottocodes**: Optional character vector of glottocodes. If no glottocodes are supplied, glottoconvert will search for them in the sample table.

- **table**: In case dataset consists of multiple tables, indicate which table contains the data that should be converted.

- **glottocolumn**: Column name or column id with glottocodes (optional, provide if glottocodes are not stored in a column called 'glottocode')
glottocreate

**glottosubcolumn**
- Column name or column id with glottosubcodes (optional, provide if glottosubcodes are not stored in a column called 'glottosubcode')

**ref**
- Character string that distinguishes those columns which contain references.

**page**
- Character string that distinguishes those columns which contain page numbers.

**remark**
- Character string that distinguishes those columns which contain remarks.

**contributor**
- Character string that distinguishes those columns which contain contributors.

**varnamecol**
- In case the dataset contains a structure table, but the varnamecol is not called 'varname', its name should be specified.

**Value**

A glottodata or glottosubdata object (either a list or data.frame)

---

**Description**

Creates glottodata/glottosubdata and optionally save it as excel file.

**Usage**

```r
glottocreate(
  glottocodes,
  variables,
  meta = TRUE,
  filename = NULL,
  simplify = TRUE,
  groups = NULL,
  n = NULL,
  levels = NULL,
  check = FALSE,
  maintainer = NULL,
  email = NULL,
  citation = NULL,
  url = NULL
)
```

**Arguments**

- **glottocodes**
  - Character vector of glottocodes

- **variables**
  - Either a vector with variable names, or a single number indicating the total number of variable columns to be generated

- **meta**
  - Should metatables be created?
filename  Optional name of excel file where to store glottodata
simplify  By default, if a glottodata table is created without metadata, the data will be returned as a data.frame (instead of placing the data inside a list of length 1)
groups    Character vector of group names (only for glottosubdata)
n         Optional, number of records to be assigned to each group (only for glottosubdata)
levels    Optional character vector with levels across all variables
check     Should glottocodes be checked? Default is FALSE because takes much time to run.
maintainer Name of the person/organization maintaining the data (optional)
email     Email address of maintainer/contact person (optional)
citation  How to cite the data (optional)
url       Optional url linking to a webpage.

Details

By default, glottodata will be created. In case a groups argument is provided, glottosubdata will be created.

glottodata has one table for all languages (and a number of metatables if meta = TRUE), with one row per glottocode. glottosubdata has one table for each language (and a number of metatables if meta = TRUE), with one row per glottosubcode.

Run glottoget("demodata") or glottoget("demosubdata") to see examples.

In case you already have your own dataset and want to convert it into glottodata, use: glottoconvert().

Value

A glottodata or glottosubdata object (either with or without metadata). The output can be a list or a data.frame.

Examples

# Creates glottodata table without metadata tables
glottocreate(glottocodes = c("yucu1253", "tani1257"),
variables = 3, meta = FALSE)

# Creates glottodata table with metadata tables (stored in a list):
glottocreate(glottocodes = c("yucu1253", "tani1257"), variables = 3)

# Creates glottosubdata table (stored in a list)
glottocreate(glottocodes = c("yucu1253", "tani1257"),
variables = 3, groups = c("a", "b") )
**glottocreate_addtable**  
*Add a table to glottodata*

---

**Description**

Add a table to glottodata

**Usage**

```r
glottocreate_addtable(glottodata, table, name)
```

**Arguments**

- `glottodata`: A glottodata table, or a list of glottodata tables
- `table`: A table to be added
- `name`: A name for the table

**Value**

a glottodata object with structure table added to it.

**Examples**

```r
glottodata <- glottoget("demodata", meta = FALSE)
structuretable <- glottocreate_structuretable(varnames = colnames(glottodata)[-1])
glottodata <- glottocreate_addtable(glottodata, table = structuretable, name = "structure")
```

---

**glottodist**  
*Calculate distances between languages*

---

**Description**

Calculate distances between languages

**Usage**

```r
glottodist(glottodata, structure = NULL, id = NULL)
```

**Arguments**

- `glottodata`: glottodata or glottosubdata, either with or without structure table.
- `structure`: If glottodata is a table without a structure table, you can add it separately. To create a structure table, you should run `glottocreate_structuretable()` and you can add it with `glottocreate_addtable()`
- `id`: By default, glottodist looks for a column named 'glottocode', if the id is in a different column, this should be specified.
Value

object of class dist

Examples

```r
  glottodata <- glottoget("demodata", meta = TRUE)
  glottodist <- glottodist(glottodata = glottodata)
```

Description

By default, the glottolog data will be used to filter from. But in case the user provides glottodata, this will be used.

Usage

```r
  glottofilter(
    glottodata = NULL,
    glottocode = NULL,
    location = NULL,
    name = NULL,
    family = NULL,
    family_id = NULL,
    continent = NULL,
    country = NULL,
    sovereignty = NULL,
    macroarea = NULL,
    expression = NULL,
    isocodes = NULL,
    colname = NULL,
    select = NULL,
    drop = NULL
  )
```

Arguments

- **glottodata**: A glottodata table
- **glottocode**: A character vector of glottocodes
- **location**: A character vector with a location (either a continent, country, macroarea, or sovereignty)
- **name**: A character vector of language names
- **family**: A character vector of language families
- **family_id**: A character vector of language family IDs
glottofiltermap

continent A character vector of continents
country A character vector of countries
sovereignty Sovereignty
macroarea Glottolog macroarea
expression A regular expression
isocodes A character vector of iso639p3codes
colname A column name
select Character vector of things to select (only if colname is provided)
drop Character vector of things to drop (only if colname is provided)

Value
A subset of the original glottodata table (data.frame or sf) containing only filtered languages.

See Also
glottofiltermap()

Examples

points <- glottofilter(location = "Australia")
points <- glottofilter(glottocode = "wari1268")
points <- glottofilter(family = "Indo-European")
points <- glottofilter(continent = "South America")
points <- glottofilter(family = "Indo-European", continent = "South America")
points <- glottofilter(country = c("Colombia", "Venezuela"))
points <- glottofilter(expression = family %in% c("Arawakan", "Tucanoan"))
points <- glottofilter(expression = family_size > 2)
points <- glottofilter(colname = "family", drop = "Indo-European")
Arguments

- `glottodata`: Spatial glottodata object
- `mode`: Either "draw" or "click"
- `...`: Additional arguments to pass to glottofilter

Value

A set of languages selected from the original glottodata object

Examples

```r
## Not run:
selected <- glottofiltermap()
glottomap(selected)

glottofiltermap(continent = "South America")
glottofiltermap(country = "Netherlands")

## End(Not run)
```

---

**glottoget**

*Get glottodata from local path or online global databases*

**Description**

Load locally stored glottodata, download databases from online sources, or load built-in demo data

**Usage**

```r
glottoget(
  glottodata = NULL,
  meta = FALSE,
  download = FALSE,
  dirpath = NULL,
  url = NULL
)
```

**Arguments**

- `glottodata`: options are:
  - A filepath to locally stored glottodata with file extension (.xlsx .xls .gpkg .shp). See also: options meta and simplify.
  - "glottobase" - Default option, an spatially enhanced version of glottolog. See [glottoboster](https://example.com) for details.
  - "wals" - This is a spatially enhanced version of WALS.
  - "dplace" - Not yet supported. This is a spatially enhanced version of D-PLACE.
• "glottolog" - This is a restructured (non-spatial) version of glottolog.
• "glottospace" - A simple dataset with glottocodes and a geometry column. This is a subset of all languages in glottolog with spatial coordinates.
• "demodata" - Built-in artificial glottodata (included for demonstration and testing)

meta
In case 'glottodata' is a path to locally stored data (or demodata/demosubdata): by default, meta sheets are not loaded. Use meta=TRUE if you want to include them.

download
By default internally stored versions of global databases are used. Specify download = TRUE in case you want to download the latest version from a remote server.

dirpath
Optional, if you want to store a global CLDF dataset in a specific directory, or load it from a specific directory.

url
Zenodo url, something like this: "https://zenodo.org/api/records/3260727"

Value
A glottodata or glottosubdata object (a data.frame or list, depending on which glottodata is requested)

See Also
Other <glottodata>: glottosave()

Examples

glottoget("glottolog")

---

glottojoin

Join glottodata with other objects, datasets, or databases.

Description
Join glottodata with other objects, datasets, or databases.

Usage

glottojoin(glottodata, with = NULL, id = NULL, rm.na = FALSE, type = "left")
Arguments

- `glottodata`:
  - `glottodata` or `glottosubdata` Optional: `glottodata` (class `data.frame`), a dist object (class `dist`), or the name of a glottodatabase ("glottobase" or "glottospace")

- `with`:
  - By default, data is joined by a column named "glottocode" or "glottosubcode".
  - In case you want to join using another column, the column name should be specified.

- `id`:
  - Only used when joining with a dist object. By default NAs are kept.

- `rm.na`:
  - In case two glottodata objects are joined, you can specify the type of join: "left" (default), "right", "full", or "inner"

Value

- `glottodata` or `glottosubdata`, either with or without metatables. Object is returned as a data.frame or list, depending on the input.

See Also

- `glottosplit` 

Examples

```r
# Join with a dist object
glottodata <- glottoget("demodata", meta = TRUE)
dist <- glottodist(glottodata)
merged_data <- glottojoin(glottodata, with = dist)
``` 

# Join glottosubdata tables:
```r
glottosubdata <- glottocreate(glottocodes = c("yucu1253", "tani1257"),
variables = 3, groups = c("a", "b"), n = 2, meta = FALSE)
merged_data <- glottojoin(glottodata = glottosubdata)
``` 

---

**glottomap**  
Create static and dynamic maps from glottodata, or select languages from a map

Description

With this function you can easily create static and dynamic maps from glottodata (by setting type to 'static' or 'dynamic'). Alternatively, by specifying type = "filter", you can select languages by drawing/clicking on a map.
Usage

glottomap(
  glottodata = NULL,
  color = NULL,
  label = NULL,
  type = NULL,
  ptsize = NULL,
  alpha = NULL,
  lbsize = NULL,
  palette = NULL,
  rivers = FALSE,
  nclass = NULL,
  numcat = FALSE,
  filename = NULL,
  projection = NULL,
  mode = NULL,
  ...
)

Arguments

- **glottodata**: Optional, user-provided glottodata. In case no glottodata is provided, you can pass arguments directly to glottofilter.
- **color**: glottovar, column name, or column index to be used to color features (optional). Run glottovars() to see glottovars
- **label**: glottovar, column name, or column index to be used to label features (optional). Run glottovars() to see glottovars
- **type**: One of: "static", "dynamic", or "filter". Default is "static".
- **ptsize**: Size of points between 0 and 1
- **alpha**: Transparency of points between 0 (very transparent) and 1 (not transparent)
- **lbsize**: Size of labels between 0 and 1
- **palette**: Color palette, see glottocolpal("all") for possible options, and run glottocolpal("turbo") to see what it looks like (replace it with palette name). Alternatively, you could also run tmaptools::palette_explorer(), RColorBrewer::display.brewer.all(), ?viridisLite::viridis, or scales::show_col(viridisLite::viridis(n=20))
- **rivers**: Do you want to plot rivers (only for static maps)?
- **nclass**: Preferred number of classes (default is 5)
- **numcat**: Do numbers represent categories? For example, if your dataset consists of 0 and 1, you might want to set this to TRUE.
- **filename**: Optional filename if you want to save resulting map
- **projection**: For static maps, you can choose one of the following: 'eqarea' (equal-area Eckert IV, default), 'pacific' (Pacific-centered), or any other Coordinate Reference System, specified using an EPSG code (https://epsg.io/).
- **mode**: In case type = "filter", you can set mode to either "draw" or "click".
- **...**: Additional parameters to glottofilter
Value

a map created from a glotto(sub)data object and can be saved with glottosave()

Examples

```r

# map created from netherlands
glottomap(country = "Netherlands")

# from latin America
glottopoints <- glottofilter(continent = "South America")
glottopols <- glottospace(glottopoints, method = "voronoi")
glottomap(glottodata = glottopols, color = "family_size_rank")

glottomap(glottodata = glottopols, color = "family", palette = "turbo",
type = "dynamic", label = "name")

# from large families
families <- dplyr::count(glottodata, family, sort = TRUE)

# highlight 10 largest families:
# or, place 10 largest families in background
```

### glottonmds

*Nonmetric Multidimensional Scaling for a glottodist object*

**Description**

Nonmetric Multidimensional Scaling for a glottodist object

**Usage**

```r

glottonmds(glottodist = NULL, k = NULL, rm.na = FALSE, row2id = NULL)
```

**Arguments**

- `glottodist` A glottodist object
- `k` Number of dimensions. Either 2 or 3 for nmds.
- `rm.na` Whether na’s should be removed (default is FALSE)
- `row2id` In case of nmds, specify what each row contains (either 'glottocode' or 'glotto-subcode')

**Value**

a glottonmds object
Description

This function offers different types of visualizations for linguistic data and linguistic distances.

Usage

```r
glottoplot(
  glottodata = NULL,
  glottodist = NULL,
  type = NULL,
  glottonmds = NULL,
  color = NULL,
  ptsize = NULL,
  label = NULL,
  filename = NULL,
  palette = NULL,
  k = NULL,
  rm.na = FALSE,
  row2id = NULL,
  preventoverlap = FALSE,
  alpha = NULL
)
```

Arguments

- `glottodata`: glottodata table
- `glottodist`: A dist object created with `glottodist`
- `type`: The type of plot: "heatmap", "nmds", or "missing". Default is heatmap if nothing is provided.
- `glottonmds`: A glottonmds object created with `glottonmds`
- `color`: glottovar to be used to color features (optional). Run `glottovars()` to see the options.
- `ptsizesize`: Size of points between 0 and 1 (optional)
- `label`: glottovar to be used to label features (optional). Run `glottovars()` to see the options.
- `filename`: Optional filename if output should be saved.
- `palette`: Name of color palette, use `glottocolpal("all")` to see the options
- `k`: Number of dimensions. Either 2 or 3 for nmds.
- `rm.na`: Whether na’s should be removed (default is FALSE)
- `row2id`: In case of nmds, specify what each row contains (either 'glottocode' or 'glottosubcode')
For nmds with 2 dimensions, should overlap between data points be prevented?

alpha

For nmds with 2 dimensions: Transparency of points between 0 (very transparent) and 1 (not transparent)

Value

a visualization of a glotto(sub)data, glottodist or glottonmds object, which can be saved with glottosave()

Examples

# Plot glottodist as nmds:
glottodata <- glottoget("demodata", meta = TRUE)
glottodist <- glottodist(glottodata = glottodata)
glottoplot(glottodist = glottodist, type = "nmds",
          k = 3, color = "family", label = "name", row2id = "glottocode")

# To create a stress/scree plot, you can run:
# goeveg::dimcheckMDS(matrix = as.matrix(glottodist), k = k)

# Plot missing data:
glottodata <- glottoget("demodata", meta = TRUE)
glottodata <- glottosimplify(glottodata)
glottoplot(glottodata = glottodata, type = "missing")

Description

If no filename is provided, the name of the glottodata object will be used.

Usage

glottosave(glottodata, filename = NULL)

Arguments

  glottodata User-provided glottodata
  filename Filename either with or without file extension

Details

If no file extension is provided, a sensible default file extension is chosen. Dynamic maps (tmap) are saved in .html format, static maps (tmap) are saved as .png. Spatial data (sf) are saved as geopackage (.GPKG) by default, but .shp is also possible.
glottosearch

Value

No object is returned, it will be save locally at the specified location

See Also

glottoget, glottodata

Other <glottodata>: glottoget()

Examples

glotodata <- glottoget("demodata", meta = FALSE)
# Saves as .xlsx
glottosave(glotodata, filename = file.path(tempdir(), "glottodata") )

glottospacedata <- glottospace(glotodata)
# Saves as .GPKG
glottosave(glotodata, filename = file.path(tempdir(), "glottodata") )

glottomap <- glottomap(glotodata)
# Saves as .png
glottosave(glottomap, filename = file.path(tempdir(), "glottomap") )

# Saves as .html
glottomap <- glottomap(glotodata, type = "dynamic",
    filename = file.path(tempdir(), "glottomap") )

---

glottosearch  
Search within glottodata for languages, glottocodes, etc.

Description

Search within glottodata for languages, glottocodes, etc.

Usage

glottosearch(
    search,
    glottodata = NULL,
    partialmatch = TRUE,
    columns = NULL,
    tolerance = NULL
)
Arguments

- **search**: Character string to search for, this can be the name of a language, a family, a glottocode, isocode.
- **glottodata**: Any linguistic or cultural dataset. Default is to search within glottobase.
- **partialmatch**: By default, partial matches will be returned as well. In case you only want exact matches, this argument should be set to FALSE.
- **columns**: By default, the entire dataset is searched, but optionally the search can be limited to specific columns.
- **tolerance**: In case partialmatch is TRUE: what is the maximum difference between search term and match? Default is 0.1

Value

A subset of glottodata that matches search conditions (object returned as a data.frame/tibble)

Examples

```r
glottosearch(search = "Yucuni")
glottosearch(search = "Yucuni", columns = "name")
glottosearch(search = "Yucuni", columns = c("name", "family"))
```

---

**glottosimplify**  
Simplify glottodata structures

**Description**

With glottosimplify, the structure of a glottodata object is simplified by removing tables and properties

**Usage**

```r
glottosimplify(  
  glottodata,  
  droplist = TRUE,  
  dropmeta = TRUE,  
  dropspatial = TRUE,  
  submerge = TRUE,  
  dropunits = FALSE  
)
```
## glottospace

### Arguments

- `glottodata`: glottodata or glottosubdata.
- `droplist`: By default if glottodata is a list of tables, only the glottodata table is returned.
- `dropmeta`: By default all metadata is removed.
- `dropspatial`: By default spatial properties are removed.
- `submerge`: By default, glottosubdata tables are merged into a single glottodata table.
- `dropunits`: By default units are kept.

### Value

A simplified version of the original dataset, either a data.frame/tibble or a list (depending on the selected options)

### Examples

```r
glottodata <- glottoget("demodata", meta = TRUE)
glottosimplify(glottodata)
```

### Description

This function takes glottodata (either with or without metadata) and turns it into spatial points or polygons.

### Usage

```r
glottospace(glottodata, method = NULL, radius = NULL)
```

### Arguments

- `glottodata`: A glottodata table, or list of a glottodata table and metadata table(s)
- `method`: Interpolation method, either "buffer" or "voronoi" (synonymous with "thiessen")
- `radius`: In case interpolation method "buffer", the radius in km.

### Value

A spatial version of glottodata. In case glottodata has metadata, only glottodata will be converted to spatial (but all metadata tables are kept). Object returned as sf object, or a list of which the first element is an sf object, depending on the input.

### Examples

```r
glottodata <- glottoget("demodata", meta = TRUE)
glottospacedata <- glottospace(glottodata, method = "voronoi")
```
glottosplitmergemeta  
_Split or merge metadata from glottodata (or glottosubdata)_

**Description**

Usually, you will run this function twice, once to split metadata from glottodata, and a second time to join it again.

**Usage**

`glottosplitmergemeta(glottodata, splitted = NULL)`

**Arguments**

- `glottodata`: glottodata
- `splitted`: if provided, the second element of the list will be joined with glottodata

**Value**

A list of length 2 in case only glottodata is provided, and a merged glottodata object otherwise.

**See Also**

- `glottojoin`
- `glottosimplify`

**Examples**

```r
glottodata <- glottoget("demodata", meta = TRUE)
splitted <- glottosplitmergemeta(glottodata)
merged <- glottosplitmergemeta(glottodata = glottodata, splitted = splitted)
```

---

**glottospotlight**  
_Highlight certain data points in visualizations_

**Description**

This function creates two separate color scales: one for points to highlight, and a second for the remaining background points. It also creates a legend. This is useful for preparing the data for visualizations such as maps or other plots.
Usage

```r
glottospotlight(
    glottodata,
    spotcol,
    spotlight,
    spotcontrast = NULL,
    bgcontrast = NULL,
    palette = NULL
)
```

Arguments

- `glottodata`: User-provided glottodata
- `spotcol`: Name of the column that contains the data to put in the spotlights, as well as remaining background data.
- `spotlight`: Selection of data to put in the spotlights.
- `spotcontrast`: Optional column to contrast between data points in the spotlight.
- `bgcontrast`: Optional column to contrast between background data points
- `palette`: Color palette. Use `glottocolpal()` or `glottocolpal_options()` to see them.

Value

A glottodata object with columns added to be used in visualization.

Examples

```r
glottodata <- glottofilter(country = c("Netherlands", "Germany", "Belgium") )
glottodata <- glottospotlight(glottodata = glottodata, spotcol = "country",
    spotlight = "Netherlands", spotcontrast = "name")
glottomap(glottodata, color = "color")
```
Index

* <glottobooster>
  glottobooster, 2
* <glottocheck><glottosearch>
  glottocode_exists, 5
* <glottocheck>
  glottocheck, 3
* <glottodata>
  glottoget, 12
  glottosave, 18
* <glottofilter><glottosearch>
  glottofilter, 10
* <glottoget><glottocreate>
  glottocreate, 7
* <glottomap>
  glottomap, 14
* <glottosearch><glottofilter>
  glottosearch, 19

  glottobooster, 2, 12
  glottoboosterflex, 3
  glottocheck, 3
  glottoclean, 4
  glottocode_exists, 5
  glottoconvert, 6
  glottocreate, 7
  glottocreate_addtable, 9
  glottodist, 9, 17
  glottofilter, 10
  glottofiltermap, 11
  glottoget, 12, 19
  glottojoin, 13
  glottomap, 14
  glottonmmds, 16, 17
  glottoplot, 17
  glottosave, 13, 18
  glottosearch, 19
  glottosimplify, 20
  glottospace, 21
  glottosplitmerge, 22
  glottospotlight, 22