Package ‘hydflood’

November 20, 2023

Type Package

Title Flood Extents and Durations along the Rivers Elbe and Rhine

Version 0.5.7

Date 2023-11-17

Description Raster based flood modelling internally using 'hyd1d', an R package to interpolate 1d water level and gauging data. The package computes flood extent and durations through strategies originally developed for 'INFORM', an 'ArcGIS'-based hydro-ecological modelling framework. It does not provide a full, physical hydraulic modelling algorithm, but a simplified, near real time 'GIS' approach for flood extent and duration modelling. Computationally demanding annual flood durations have been computed already and data products were published by Weber (2022) <doi:10.1594/PANGAEA.948042>.

Depends R (>= 4.0.0), sf, terra, raster, hyd1d

Imports stats, Rdpack, grDevices

Suggests knitr, rmarkdown, devtools, pkgdown, roxygen2, testthat, plot3D, plotrix, shiny, shinyjs, shiny.i18n, leaflet, leafletextras, leaflet.esri, pangaear, rgrass, stringr

RdMacros Rdpack

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Collate 'classifyToPNV.R' 'createTiles.R' 'data.R' 'flood1.R' 'flood2.R' 'flood3.R' 'flood3Points.R' 'getDEM.R' 'hydflood.R' 'hydflood-internal.R' 'hydSpatRaster.R' 'w80ToSpatial.R' 'zzz.R'

VignetteBuilder knitr

BugReports https://github.com/bafg-bund/hydflood/issues/


NeedsCompilation no
**Function to reclassify flood durations to potential natural vegetation**

descrToPNV

topics documented:
classifyToPNV ........................................... 2
createTiles ............................................. 4
df.pnv .................................................... 4
flood1 ...................................................... 5
flood2 ...................................................... 8
flood3 ...................................................... 9
flood3Points ............................................. 11
getDEM ..................................................... 12
hydflood .................................................. 13
hydSpatRaster .......................................... 14
sf.af ....................................................... 16
sf.afe .................................................... 17
sf.afr ..................................................... 18
sf.tiles ................................................... 18
sf.tiles_elbe .......................................... 19
sf.tiles_rhine ......................................... 20
w80ToSFL ................................................ 21
w80ToSFP ................................................ 23

**Index**

classifyToPNV ........................................ 25

**Description**

This function is a wrapper to the function `classify` to convert flood durations computed with `flood3` into potential natural vegetation (PNV) distributions using reclassification rules supplied with `df.pnv`. Alternative reclassification rules may be applied, but they must match column names and types as given by `df.pnv`. `classify` is called with `include.lowest = TRUE` and `right = FALSE`.

**Usage**

```r
classifyToPNV(x, rcl = NULL, filename = "", ...)
```
classifyToPNV

Arguments

- **x**: argument of type `SpatRaster`.
- **rcl**: optional argument of type `data.frame` with columns and column types as specified in `df.pnv`.
- **filename**: supplies an optional output filename of type `character`.
- **...**: additional arguments as for `writeRaster`.

Value

`SpatRaster` object containing potential natural vegetation distribution as categorical raster.

References


See Also

`df.pnv`

Examples

```r
cache <- tempdir()
options("hyd1d.datadir" = cache)
options("hydflood.datadir" = cache)
options(timeout = 200)
library(hydflood)

# import the raster data and create a raster stack
c <- st_crs("EPSG:25833")
e <- ext(309000, 309300, 5749000, 5749300)
x <- hydSpatRaster(ext = e, crs = c)

# create a temporal sequence
seq <- seq(as.Date("2016-01-01"), as.Date("2016-12-31"), by = "day")

# compute a flood duration
fd <- flood3(x = x, seq = seq)

# reclassify to PNV
pnv <- classifyToPNV(fd)

# plot pnv map
plot(pnv)
```
createTiles  

Function to split large areas (sfc_POLYGON) into tiles

Description
To simplify and accelerate the computation of flood duration with flood3 in massive areas this function provides a simple tiling algorithm.

Usage
createTiles(x, size_x, size_y, subset = TRUE)

Arguments
- x: has to by type sf.
- size_x: tile size along the x-axis in the units of the current projection (numeric).
- size_y: tile size along the y-axis in the units of the current projection (numeric).
- subset: boolean determining whether all or only intersecting tiles are returned.

Value
sf object containing tiles covering x.

Examples
options("hydflood.datadir" = tempdir())
library(hydflood)
tiles <- createTiles(x = sf.af(name = "Elbe"),
                     size_x = 10000, size_y = 10000)
plot(tiles["tile_ID"])

df.pnv  

Reference data.frame used to classify flood duration into potential natural vegetation.

Description
Reference data.frame used to classify flood duration into potential natural vegetation (PNV). It is an extended and more detailed table to reclassify flood duration into PNV based on Ochs et al. (2020).

Usage
df.pnv
Format

A data.frame containing 7 columns with attributes to reclassify flood duration into potential natural vegetation.

- **from**: lower limits of flood duration (included, type numeric).
- **to**: upper limits of flood duration (not included, type numeric).
- **id**: numeric replacements used to sort classes (type numeric).
- **vegtype**: names of the potential natural vegetation classes (type character).
- **r**: numeric coding for the r (red) of an rgb color code.
- **g**: numeric coding for the g (green) of an rgb color code.
- **b**: numeric coding for the b (blue) of an rgb color code.
- **html**: html color coding (type character).

References


---

**flood1**

*Function to compute flood extent or flood duration SpatRaster along the German federal waterways Elbe and Rhine using the 1d water level algorithm hyd1d::waterLevelFlood1()*

Description

Computes flood extent, if `length(seq)` equals 1, or flood duration for the active floodplains along the German federal waterways Elbe and Rhine based on 1d water levels computed by `waterLevelFlood1` provided by package *hyd1d* in analogy to the INFORM 3 module 'Flut1'.

Usage

```r
flood1(x, seq, gauging_station, uuid, filename = "", ...) 
```

Arguments

- **x**: has to be type SpatRaster and has to include both input layers csa (cross section areas) and dem (digital elevation model). To compute water levels along the River Elbe, `x` has to be in the coordinate reference system ETRS 1989 UTM 33N, for the River Rhine in ETRS 1989 UTM 32N. Other coordinate reference systems are not permitted.

- **seq**: has to be type c("POSIXct", "POSIXt") or Date and have a length larger than 0. Values of `seq` must be in the temporal range between 1960-01-01 and yesterday (Sys.Date() - 1). Internally `waterLevelFlood1` uses `getGaugingDataW` to obtain daily water level information from `df.gauging_data`.
gauging_station

has to be type character and has to have a length of one. Permitted values are:


uuid

has to be type character and has to have a length of one. Permitted values are:

'7cb7461b-3530-4c01-8978-7f676b871ed', '85d686f1-55b2-4d36-88a5-3207b50901a7', '70272185-b2b3-4178-96b8-43bea330dc9e', '24440872-5bd2-4bf3-8554-907b4f9816c9', 'b04b739d-9e14-9eb9-957a-4e05f508', '169b94e7-be14-41fd-946e-755597972c6', '38bbaed5-5d81-4bc6-9766-3bd70099c11f', '3fdcf807-c2bb-4b92-b00b-4e01935a2c6', 'c093b557-9454-4f05-85fc-6cd67f916c2d', '0701b4e-3872-4e07-b2e5-e25fd9251b93', '1ee53a59-33b9-4dc0-9eb7-3cd2a416407', 'ae9f3a56-1162-4514-b5fd-9bcaedc737c2', 'e97116a4-7d30-4671-8ba1-cdce0a1533a1', '1edc5fa4-88af-47f5-954a-0e7a06fe8b11', '094b96e5-cae6-e4d3-a8ee-e4d418add269', '939f828-15a9-49c8-888a-0c2f8a2d49e2', '90cb315-f0b0-41a8-a0ac-6122331bcb4f', 'b8567c1e-8610-4c2b-a240-65e8a74919f5', 'c0cc57f-2a2f-4193-8ae8-5710da3afaefdd', 'e30f2e83-8b08-46b6-f939-fa603170f7b7', '3ad88f18-d7a4-41d0-84f5-1143c98a6564', '133f06e-7ca1-4798-9360-5bf547dd383', '3e91b777-90f3-4a15-a320-641748e93c11', 'de4ecd05-1c6b-4eb2-b02e-29e753b4e544', 'f4c55f77-ab80-4e00-bed3-aaf6631a0b4', 'a3b02a8-8cd5-4053-8bc6-f9cc8469106', 'c0fbd349-91bd-49cc-8926-6cccc0e5a45', '48f26619-f9cb-4093-9d57-da2418ed656', '15060385-9ad1-4e55-bd52-34228b8d9988', 'e80a4f21-528c-4771-98d7-10cd519699a4', 'ac507f42-1593-49e9-865f-10b2523617c7', '6e3e4a79-48b1-408a-bc55-0986ce98ed5', 'c23367f4-259a-4304-b81f-dceu1f4d55', 'a26e579-1eb8-4bfa-ca80-902ab88df8', '67ed6882-b60c-40d3-975c-a667a2b4e0a', '6a41cde-5e52-4cb-bb8-e-705b6dce7da2', '33eb0ce-0-16df-4f8f-be9-d1a77e9851e', 'd9289367-8a8a-4d6ba-1a1-8577bec46bb', 'b3492e68-8373-4769-9b29-22f6635a478', '447e9555-07bd-45cf-9ed7-19460806db4', '069b7b8d-8cd4-48ac-a08e-2e16981ed281', '9da1d2b-88db-4cbb-8132-5edf98d75ba', '53898878-fad5-4f37-bb87-e6cb36b7078b', '787e5d63-6e1e-428c-acf0-633ee2ba1c32', '23a9fb0-2c82-4fba-acb8-92a065ed45a', 'b02e24b3-1364-4e97-8bb6-6757d7d843', '6b774a7b-586-49ae-8eeb-ecf1a278bi', 'b6c6d5c8-7e2-d446-8dd8-fa972e7f7eaa', '8e8e972e-80a-4eb9-847c-0925e5999a46', '2cb8ae5b-e5c9-4f4a-bac0-bb724f2754f4', '50790082-c51a-4d09-8340-b443cd50e1f5', '844a620f-f3b8-4b06-8e3c-783a2ae223a2', 'd2b8e7ed-1317-41c5-be6c-725369ed1171', 'a37a9a9a-345-4d90-9dfe-109fa28a5a', '666bfe0-5e38-436b-5804-509ad3b6eeb4', '03090d1-90e9-470e-99d4-2ee4b2c5f84', '1d26e504-79e-480a-b52c-5932be549ab', '550eb7a9-72e-48e4-a1e-1b761b42232', '2ff66739-d168-4022-8da0-16846d4e599b', 'd6dc44d1-
filename supplies an optional output filename and has to be type character.

Details  
For every time step provided in seq, flood1() computes a 1d water level using waterLevelFlood1 along the requested river section. This 1d water level is transferred to a wl (water level) raster layer, which is in fact a copy of the csa (cross section areas) layer, and then compared to the dem (digital elevation model) layer. Where the wl layer is higher than the dem layer, flood duration is increased by 1.

Value  
SpatRaster object with flood duration in the range of $[0, \text{length(seq)}]$.

References  

See Also  
df.gauging_data, getGaugingDataW, waterLevelFlood1, writeRaster, terraOptions

Examples  
```r  
options("hydflood.datadir" = tempdir())  
library(hydflood)  
# import the raster data and create a raster stack  
c <- st_crs("EPSG:25833")  
e <- ext(309000, 310000, 5749000, 5750000)  
x <- hydSpatRaster(ext = e, crs = c)  
# create a temporal sequence  
seq <- seq(as.Date("2016-12-01"), as.Date("2016-12-31"), by = "day")  
# compute a flood duration  
fd <- flood1(x = x, seq = seq, gauging_station = "ROSSLAU")```
flood2 Function to compute flood extent or flood duration SpatRaster along the German federal waterways Elbe and Rhine using the 1d water level algorithm hyd1d::waterLevelFlood2()

Description
Computes flood extent, if length(seq) equals 1, or flood duration for the active floodplains along the German federal waterways Elbe and Rhine based on 1d water levels computed by waterLevelFlood2 provided by package hyd1d in analogy to the INFORM 3 module 'Flut2'.

Usage
flood2(x, seq, filename = "", ...)

Arguments
- x has to by type SpatRaster and has to include both input layers csa (cross section areas) and dem (digital elevation model). To compute water levels along the River Elbe, x has to be in the coordinate reference system ETRS 1989 UTM 33N, for the River Rhine in ETRS 1989 UTM 32N. Other coordinate reference systems are not permitted.
- seq has to be type c("POSIXct", "POSIXt") or Date and have a length larger than 0. Values of seq must be in the temporal range between 1960-01-01 and yesterday (Sys.Date() - 1). Internally waterLevelFlood2() uses getGaugingDataW to obtain daily water level information from df.gauging_data.
- filename supplies an optional output filename and has to be type character.
- ... additional arguments as for writeRaster.

Details
For every time step provided in seq, flood2() computes a 1d water level using waterLevelFlood2 along the requested river section. This 1d water level is transferred to a wl (water level) raster layer, which is in fact a copy of the csa (cross section areas) layer, and then compared to the dem (digital elevation model) layer. Where the wl layer is higher than the dem, layer flood duration is increased by 1.

Value
SpatRaster object with flood duration in the range of [0, length(seq)].

References
flood3

See Also
df.gauging_data, getGaugingData, waterLevelFlood2, writeRaster, terraOptions

Examples

```r
options("hydflood.datadir" = tempdir())
library(hydflood)

# import the raster data and create a raster stack
c <- st_crs("EPSG:25833")
e <- ext(309000, 310000, 5749000, 5750000)
x <- hydSpatRaster(ext = e, crs = c)

# create a temporal sequence
seq <- seq(as.Date("2016-12-01"), as.Date("2016-12-31"), by = "day")

# compute a flood duration
fd <- flood2(x = x, seq = seq)
```

Description

Computes flood extent, if length(seq) equals 1, or flood duration for the active floodplains along the German federal waterways Elbe and Rhine based on 1d water levels computed by waterLevel or waterLevelPegelonline provided by package hyd1d.

Usage

`flood3(x, seq, filename = "", ...)`

Arguments

`x` has to by type SpatRaster and has to include both input raster layers csa (cross section areas) and dem (digital elevation model). To compute water levels along the River Elbe `x` has to be in the coordinate reference system ETRS 1989 UTM 33N, for River Rhine in ETRS 1989 UTM 32N. Other coordinate reference systems are not permitted.
seq has to be type c("POSIXct", "POSIXt") or Date and have a length larger than 0. If seq is type c("POSIXct", "POSIXt"), values must be in the temporal range between 31 days ago (Sys.time() - 2678400) and now (Sys.time()). Then **waterLevelPegelonline** is used internally for the water level computations. If seq is type Date, values must be in the temporal range between 1960-01-01 and yesterday (Sys.Date() - 1) and **waterLevel** is used internally.

filename supplies an optional output filename and has to be type character.

... additional arguments as for **writeRaster**.

Details

For every time step provided in seq, flood3() computes a 1d water level along the requested river section. This 1d water level is transferred to a wl (water level) raster layer, which is in fact a copy of the csa (cross section areas) layer, and then compared to the dem (digital elevation model) layer. Where the wl layer is higher than the dem, layer flood duration is increased by 1.

Value

SpatRaster object with flood duration in the range of \([0, \text{length(seq)}]\).

References


See Also

**waterLevel**, **waterLevelPegelonline**, **writeRaster**, **terraOptions**

Examples

```r
options("hydflood.datadir" = tempdir())
library(hydflood)

# import the raster data and create a raster stack
c <- st_crs("EPSG:25833")
e <- ext(309000, 310000, 5749000, 5750000)
x <- hydSpatRaster(ext = e, crs = c)

# create a temporal sequence
seq <- seq(as.Date("2016-12-01"), as.Date("2016-12-31"), by = "day")

# compute a flood duration
fd <- flood3(x = x, seq = seq)
```
flood3Points

Function to compute flood duration for point coordinates along the German federal waterways Elbe and Rhine using the 1d water level algorithms hyd1d::waterLevel() and hyd1d::waterLevelPegelonline()

Description

Computes flood duration for points located in the active floodplains along the German federal waterways Elbe and Rhine based on 1d water levels computed by waterLevel or waterLevelPegelonline provided by package hyd1d.

Usage

flood3Points(x, seq)

Arguments

x has to be type sf possibly including columns csa (cross section areas) and dem (digital elevation model). To compute water levels along the River Elbe, x has to be in the coordinate reference system ETRS 1989 UTM 33N, for the River Rhine in ETRS 1989 UTM 32N. Other coordinate reference systems are not permitted.

seq has to be type c("POSIXct", "POSIXt") or Date and have a length larger than 0. If seq is type c("POSIXct", "POSIXt"), values must be in the temporal range between 31 days ago (Sys.time() - 2678400) and now (Sys.time()). Then waterLevelPegelonline is used internally for the water level computations. If seq is type Date, values must be in the temporal range between 1960-01-01 and yesterday (Sys.Date() - 1)

Details

For every time step provided in seq, flood3Points() computes a 1d water level along the requested river section. This 1d water level is transfered to a temporary wl (water level) column and then compared to the dem (digital elevation model) column. Where the wl is higher than the dem flood duration flood3 is increased by 1.

Since the underlying tiled digital elevation models (dem) are rather large datasets hydflood provides options to permanently cache these datasets. options("hydflood.datadir" = tempdir()) is the default. To modify the location of your raster cache to your needs set the respective options() prior to loading the package, e.g. options("hydflood.datadir" = "~/.hydflood");library(hydflood). The location can also be determined through the environmental variable hydflood_datadir.

Value

sf object with flood duration stored in column flood3 in the range of [0, length(seq)], elevation stored in column dem and cross section areas stored in column csa.
getDEM

Function to obtain the digital elevation models for the active flood-plains along the German federal waterways Elbe and Rhine

description

This function downloads and patches the tiled digital elevation models (dem) along the German federal waterways Elbe and Rhine that have been published on pangaea.de.

Usage

getDEM(filename = "", ext, crs, ...)

Arguments

filename supplies an optional in- and output filename and has to be type character.
ext argument of type SpatExtent.
crs argument of type crs or crs. It is used to select the respective river (Elbe: 'ETRS 1989 UTM 33N'; Rhine: 'ETRS 1989 UTM 32N')
... additional arguments as for writeRaster.
Details

Since the underlying tiled digital elevation models (DEM) are rather large datasets hydflood provides options to permanently cache these datasets. `options("hydflood.datadir" = tempdir())` is the default. To modify the location of your raster cache to your needs set the respective options() prior to loading the package, e.g. `options("hydflood.datadir" = "~/hydflood");library(hydflood)`. The location can also be determined through the environmental variable `hydflood_datadir`.

Since downloads of large individual datasets might cause timeouts, it is recommended to increase `options("timeout")`.

Value

SpatRaster object containing elevation data for the selected floodplain region.

References


Examples

```r
options("hydflood.datadir" = tempdir())
options("timeout" = 120)
library(hydflood)
dem <- getDEM(ext = ext(c(309000, 310000, 5749000, 5750000)),
               crs = st_crs("EPSG:25833"))
```

Description

Raster based flood modelling internally using `hyd1d`, an R package to interpolate 1d water level and gauging data. The package computes flood extent and durations through strategies originally developed for 'INFORM', an 'ArcGIS'-based hydro-ecological modelling framework. It does not provide a full, physical hydraulic modelling algorithm, but a simplified, near real time 'GIS' approach for flood extent and duration modelling. Computationally demanding annual flood durations have been computed already and data products were published by Weber (2022) doi: 10.1594/PANGAEA.948042.
**Description**

To initialize an object of class `SpatRaster` with layers dem and csa this function should be used. It checks all the required input data, downloads missing data automatically, clips and returns the final object, prepared for the `flood()` functions (`flood1`, `flood2` and `flood3`).

**Usage**

```r
hydSpatRaster(filename_dem = "", filename_csa = "", ext, crs, ...)
```

**Arguments**

- `filename_dem` an optional argument of length 1 with type character specifying a filename of a digital elevation model raster dataset.
  
  If the file exists it is imported via `rast` and used to build the SpatRaster, potentially cropped by argument `ext`. If the dem file does not exist, data are downloaded automatically and exported using `writeRaster` and can be reused to accelerate later computations.
  
  An existing dataset must be either in the coordinate reference system (crs) 'ETRS 1989 UTM 32N' (epsg: 25832) for the River Rhine or 'ETRS 1989 UTM 33N' (epsg: 25833) for the River Elbe. It must also overlap with the active floodplains (`sf.afe` or `sf.afr`) of the river selected through the crs.
  
  If argument `filename_csa` is specified and exists too, the coordinate reference system (`crs`), extent (`ext`) and resolution (`res`) of both raster datasets must match.
  
  Supported file types depend on available GDAL raster drivers.

- `filename_csa` an optional argument of length 1 with type character specifying a filename of a cross section area raster dataset.
  
  If the file exists it is imported via `rast` and used to build the SpatRaster, potentially cropped by argument `ext`. If the csa file does not exist, data are downloaded automatically and exported using `writeRaster` and can be reused to accelerate later computations.
  
  An existing dataset must be either in the coordinate reference system (crs) 'ETRS 1989 UTM 32N' (epsg: 25832) for the River Rhine or 'ETRS 1989 UTM 33N' (epsg: 25833) for the River Elbe. It must also overlap with the active floodplains (`sf.afe` or `sf.afr`) of the river selected through the crs and be in the possible range of `station_int` values: Elbe (m 0 - 585700), Rhine (m 336200 - 865700).
  
  If argument `filename_dem` is specified too, coordinate reference system (`crs`), extent (`ext`) and resolution (`res`) of both raster datasets must match.
  
  Supported file types depend on available GDAL raster drivers.
optional argument of type `SpatExtent`. If neither `filename_dem` nor `filename_csa` are specified, `ext` is required to download the respective data and generate temporary dem and csa datasets. If either `filename_dem` or `filename_csa` or both are specified, `ext` must be within the extent of provided raster layers. Then it is used to `crop` the supplied data.

optional argument of type `crs` or `crs`. If neither `filename_dem` nor `filename_csa` are specified, `crs` is used to select the respective river (Elbe: 'ETRS 1989 UTM 33N' (epsg: 25833); Rhine: 'ETRS 1989 UTM 32N' (epsg: 25832)) and `crop` downloaded dem and csa by the given `ext`. If either `filename_dem` or `filename_csa` or both are specified, `crs` must match their coordinate reference systems; otherwise an error is returned.

additional parameters passed to `writeRaster`.

Details

Since the underlying tiled digital elevation models (dem) are rather large datasets hydflood provides options to permanently cache these datasets. `options("hydflood.datadir" = tempdir())` is the default. To modify the location of your raster cache to your needs set the respective options() prior to loading the package, e.g. `options("hydflood.datadir" = "~/.hydflood");library(hydflood)`. The location can also be determined through the environmental variable `hydflood.datadir`.

Since downloads of large individual datasets might cause timeouts, it is recommended to increase options("timeout").

Value

SpatRaster object containing digital elevation (dem) and cross section area (csa) raster layers.

References


See Also

SpatRaster-class, rast, writeRaster, flood1, flood2, flood3, sf.afe, sf.afr

Examples

```r
options("hydflood.datadir" = tempdir())
options("timeout" = 120)
library(hydflood)

e <- ext(436500, 438000, 5415000, 5416500)
c <- st_crs("EPSG:25832")

r <- hydSpatRaster(ext = e, crs = c)
r
```

---

**sf.af**

Obtain projected versions of sf.afe and sf.afr

### Description

Obtain projected versions of sf.afe and sf.afr

### Usage

```r
sf.af(name = NULL)
```

### Arguments

- **name**
  - either ‘Elbe’ or ‘Rhine’.

### Value

sf with the projected active floodplain

### See Also

sf.afe, sf.afr
Examples

```r
library(hydflood)
sf.af(name = "Elbe")
```

---

**sf.afe**  
*Active floodplain along the River Elbe*

**Description**

This dataset contains a polygon of the active floodplain along the German interior parts of the River Elbe from the Czech border to the weir in Geesthacht in the coordinate reference system ETRS 1989 UTM 33N.

Originally, this polygon was produced for the floodplain status report (Auenzustandsbericht; Brunotte et al. (2009), Bundesamt für Naturschutz (2009)) at a scale of 1:25,000. For hydflood it was updated with recent flood protection measures and manually improved with recent digital elevation models and aerial images at a scale of < 1:10,000.

**Usage**

`sf.afe`

**Format**

A `sf` containing 1 polygon

**References**


**See Also**

`saf`, `sf.afr`
sf.afr

Active floodplain along the River Rhine

Description

This dataset contains a polygon of the active floodplain along the German, freeflowing parts of the River Rhine from the weir Iffezheim to the Dutch border in the coordinate reference system ETRS 1989 UTM 32N.

Originally, this polygon was produced for the floodplain status report (Auenzustandsbericht; Brunotte et al. (2009), Bundesamt für Naturschutz (2009)) at a scale of 1:25,000. For hydflood it was updated with recent flood protection measures and manually improved with recent digital elevation models and aerial images at a scale of < 1:10,000.

Usage

sf.afr

Format

A sf containing 1 polygon

References


See Also

sf.af, sf.afe

sf.tiles

Obtain projected versions of sf.tiles_elbe and sf.tiles_rhine

Description

Obtain projected versions of sf.tiles_elbe and sf.tiles_rhine

Usage

sf.tiles(name = NULL)
Arguments

name either 'Elbe' or 'Rhine'.

Value

sf with projected tiles

See Also

sf.tiles_elbe, sf.tiles_rhine

Examples

library(hydflood)
sf.tiles(name = "Elbe")

sf.tiles_elbe  Tiling along the active floodplain of the River Elbe

Description

This dataset contains 49 rectangular polygons / tiles along the active floodplain along the German interior parts of the River Elbe from the Czech border to the weir in Geesthacht in the coordinate reference system ETRS 1989 UTM 33N.

The tiles represent the original tiling of the internally used digital elevation model (Weber 2020).

Usage

sf.tiles_elbe

Format

A sf containing 49 polygons with 18 attributes:

id of the tile (type integer).
name of the tile (type character).
river of the tile (type character) in this case 'ELBE'.
name_km of the tile (type character).
from_km river kilometer of the tiles upper limit (type numeric).
to_km river kilometer of the tiles lower limit (type numeric).
gs_upper name of the tiles upper gauging station (type character).
gs_lower name of the tiles lower gauging station (type character).
geometry sfc_POLYGON column storing the geometries.
xmin of the tile extent (type integer). Minimum of UTM Easting (m).
**xmax** of the tile extent (type integer). Maximum of UTM Easting (m).

**ymin** of the tile extent (type integer). Minimum of UTM Northing (m).

**ymax** of the tile extent (type integer). Maximum of UTM Northing (m).

**lon_min** of the tile extent (type numeric). Minimum of Longitude (decimal °).

**lon_max** of the tile extent (type numeric). Maximum of Longitude (decimal °).

**lat_min** of the tile extent (type numeric). Minimum of Latitude (decimal °).

**lat_max** of the tile extent (type numeric). Maximum of Latitude (decimal °).

**url** of the tile (type character).

**References**


**See Also**

* sf.tiles, sf.tiles_rhine *

**sf.tiles_rhine**

Tiling along the active floodplain of the River Rhine

**Description**

This dataset contains 40 rectangular polygons / tiles along the active floodplain along the German, freeflowing parts of the River Rhine from the weir Iffezheim to the Dutch border near Kleve in the coordinate reference system **ETRS 1989 UTM 32N**.

The tiles represent the original tiling of the internally used digital elevation model (Weber 2020).

**Usage**

* sf.tiles_rhine *

**Format**

A sf containing 40 polygons with 18 attributes:

**id** of the tile (type integer).

**name** of the tile (type character).

**river** of the tile (type character) in this case RHINE’.

**name_km** of the tile (type character).

**from_km** river kilometer of the tiles upper limit (type numeric).

**to_km** river kilometer of the tiles lower limit (type numeric).
gs_upper  name of the tiles upper gauging station (type character).
gs_lower  name of the tiles lower gauging station (type character).
geometry  sfc_POLYGON column storing the geometries.
xmin  of the tile extent (type integer). Minimum of UTM Easting (m).
xmax  of the tile extent (type integer). Maximum of UTM Easting (m).
ymin  of the tile extent (type integer). Minimum of UTM Northing (m).
ymax  of the tile extent (type integer). Maximum of UTM Northing (m).
lon_min  of the tile extent (type numeric). Minimum of Longitude (decimal °).
lon_max  of the tile extent (type numeric). Maximum of Longitude (decimal °).
lat_min  of the tile extent (type numeric). Minimum of Latitude (decimal °).
lat_max  of the tile extent (type numeric). Maximum of Latitude (decimal °).
url  of the tile (type character).

References

See Also
sf.tiles_elbe, sf.tiles_rhine

---

w80ToSFL  Function to convert w80-files to sfc_LINESTRING.

Description
This function converts w80-files, an ascii-format with 80 characters per line for spatial point data used by the German Waterways and Shipping Administration (WSV). Every single row codes for one point:

|_1_|2_|3______|4|____________5______________|_______6_______|_____7___|_8_|

W0701 55 2594611 1330938065557502425901108035 5795591108035 Bu.15 01

W0701 57 2594611 733093296145750248401108035 5538181108035 Bu.15 01

Within each row very specific sections code for specific attributes:

<table>
<thead>
<tr>
<th>section</th>
<th>column(s)</th>
<th>attribute</th>
<th>column name in result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>state id, here W=WSV</td>
<td>sid</td>
</tr>
</tbody>
</table>
In a second step these points are aggregated to a sfc_LINESTRING using the grouping column id.

Usage

```r
w80ToSFL(
  filename,
  crs,
  id = c("sid", "fwid", "wsvpt", "station", "bank", "id", "x", "y", "date_coor",
         "acc_coor", "z", "date_z", "acc_z", "tom", "comment", "status", "lat", "lon",
         "station_int", "station_c")
)
```

Arguments

- `filename` argument of length 1 and type character specifying an existing w80-file.
- `crs` argument of type `crs` or `crs`.
- `id` argument of type character specifying a grouping column.

Value

A sfc_LINESTRING.

Examples

```r
options("hydflood.datadir" = tempdir())
library(hydflood)
c <- st_crs("EPSG:25833")
filename <- tempfile(fileext = ".w80")

# write temporary w80 file
cat("W0701 55 2594611 1330938065557502425901108035 5795591108035 Bu.15 01\n", file = filename, sep = "\n")
```
ToSFP

file = filename)
cat("W0701 57 2594611 7330932961457502484041108035 5538181108035 Bu.15 01\n", 
file = filename, append = TRUE)

# import temporary w80 file as sf LINESTRING
sl <- w80ToSFL(filename, c, "station_int")

---

w80ToSFP  
Function to convert w80-files to sfc_POINT.

Description

This function converts w80-files, an asci-format with 80 characters per line for spatial point data used by the German Waterways and Shipping Administration (WSV). Every single row codes for one point:

<table>
<thead>
<tr>
<th><em>1</em></th>
<th>2_</th>
<th><em>3</em>_____</th>
<th>4</th>
<th><strong><strong><strong><strong><strong><strong>5</strong></strong></strong></strong></strong></strong>__</th>
<th><em><strong><strong><strong>6</strong></strong></strong></em></th>
<th>__<em><strong>7</strong></em></th>
<th><em>8</em></th>
</tr>
</thead>
</table>

W0701 55 2594611 13309380655557502425901108035 5795591108035 Bu.15 01
W0701 57 2594611 7330932961457502484041108035 5538181108035 Bu.15 01

Within each row very specific sections code for specific attributes:

<table>
<thead>
<tr>
<th>section</th>
<th>column(s)</th>
<th>attribute</th>
<th>column name in result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>state id, here W=WSV</td>
<td>sid</td>
</tr>
<tr>
<td>1</td>
<td>2-5</td>
<td>Federal Waterway ID</td>
<td>fwid</td>
</tr>
<tr>
<td>2</td>
<td>6-8</td>
<td>WSV point type</td>
<td>wsvpt</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>blank</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10-15</td>
<td>river station (km)</td>
<td>station</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>bank: 1 left, 2 right</td>
<td>bank</td>
</tr>
<tr>
<td>4</td>
<td>17-20</td>
<td>continuous id</td>
<td>id</td>
</tr>
<tr>
<td>5</td>
<td>21-30</td>
<td>easting in GK-coordinates</td>
<td>x</td>
</tr>
<tr>
<td>5</td>
<td>31-40</td>
<td>northing in GK-coordinates</td>
<td>y</td>
</tr>
<tr>
<td>6</td>
<td>41-46</td>
<td>datum of measurement</td>
<td>date_coor</td>
</tr>
<tr>
<td>6</td>
<td>47</td>
<td>accuracy</td>
<td>acc_coor</td>
</tr>
<tr>
<td>6</td>
<td>48-54</td>
<td>elevation</td>
<td>z</td>
</tr>
<tr>
<td>6</td>
<td>55-60</td>
<td>date of the elevation measurement</td>
<td>date_z</td>
</tr>
<tr>
<td>6</td>
<td>61</td>
<td>accuracy of the elevation measurement</td>
<td>acc_z</td>
</tr>
<tr>
<td>6</td>
<td>62-64</td>
<td>type of measurement</td>
<td>tom</td>
</tr>
<tr>
<td>7</td>
<td>65-84</td>
<td>comment</td>
<td>comment</td>
</tr>
<tr>
<td>8</td>
<td>85-86</td>
<td>point status</td>
<td>status</td>
</tr>
</tbody>
</table>
Usage

`w80ToSFP(filename, crs)`

Arguments

- `filename` argument of length 1 and type character specifying an existing w80-file.
- `crs` argument of type `crs` or `crs`.

Value

`sfc_POINT`.

Examples

```r
options("hydflood.datadir" = tempdir())
library(hydflood)
c <- st_crs("EPSG:25833")
filename <- tempfile(fileext = ".w80")

# write temporary w80 file
cat("W0701 55 2594611 1330938065557502425901108035 5795591108035 Bu.15 01\n", file = filename)
cat("W0701 57 2594611 7330932961457502484041008035 5538181108035 Bu.15 01\n", file = filename, append = TRUE)

# import temporary w80 file as sf POINT
sf <- w80ToSFP(filename, c)
```
Index

* datasets
  df.pnv, 4
  sf.afe, 17
  sf.afr, 18
  sf.tiles_elbe, 19
  sf.tiles_rhine, 20
  classify, 2
  classifyToPNV, 2
  createTiles, 4
  crop, 15
  crs, 12, 14, 15, 22, 24

  df.gauging_data, 5, 7–9
  df.pnv, 2, 3, 4

  ext, 14

  flood1, 5, 14, 16
  flood2, 8, 14, 16
  flood3, 2, 4, 9, 14, 16
  flood3Points, 11

  getDEM, 12
  getGaugingDataW, 5, 7–9

  hydflood, 13
  hydflood-package (hydflood), 13
  hydSpatRaster, 14

  rast, 14, 16
  res, 14

  sf.af, 16, 17, 18
  sf.afe, 14, 16, 17, 18
  sf.afr, 14, 16, 17, 18
  sf.tiles, 18, 20
  sf.tiles_elbe, 18, 19, 19, 21
  sf.tiles_rhine, 18–20, 20, 21
  SpatExtent, 12, 15
  SpatRaster, 3, 14

  terraOptions, 7, 9, 10
  w80ToFSL, 21
  w80ToFSP, 23
  waterLevel, 9–12
  waterLevelFlood1, 5, 7
  waterLevelFlood2, 8, 9
  waterLevelPegelonline, 9–12
  writeRaster, 3, 7–10, 12, 14–16