Package ‘hydflood’

January 17, 2023

Type Package

Title Flood Extents and Durations along the Rivers Elbe and Rhine

Version 0.5.2

Date 2023-01-13

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, leaflet.extras, leaflet.esri, pangaear, rgrass, tidyverse, 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/`

URL `https://hydflood.bafg.de`, `https://github.com/bafg-bund/hydflood`

NeedsCompilation no

Author Arnd Weber [aut, cre] (<https://orcid.org/0000-0002-5973-2770>), Stephan Rosenzweig [ctb]

Maintainer Arnd Weber <arnd.weber@bafg.de>

Repository CRAN

Date/Publication 2023-01-17 17:20:04 UTC

R topics documented:

`classifyToPNV` *Function to reclassify flood durations to potential natural vegetation*

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`, `right = FALSE` and `othersNA = TRUE`.

Usage

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

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

Ochs K, Egger G, Weber A, Ferreira T, Householder JE, Schneider M (2020). “The potential natural vegetation of large river floodplains – From dynamic to static equilibrium.” *Journal of Hydro-environment Research*, **30**, 71–81. doi:10.1016/j.jher.2020.01.005.

See Also

`df.pnv`

Examples

```
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, 310000, 5749000, 5750000)
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 detailled 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.

References

Ochs K, Egger G, Weber A, Ferreira T, Householder JE, Schneider M (2020). “The potential natural vegetation of large river floodplains – From dynamic to static equilibrium.” *Journal of Hydro-environment Research*, **30**, 71–81. doi:10.1016/j.jher.2020.01.005.

`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

```
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: 'SCHOENA', 'PIRNA', 'DRESDEN', 'MEISSEN', 'RIESA', 'MUEHLBERG', 'TORGAU', 'PRETZSCH-MAUKEN', 'ELSTER', 'WITTENBERG', 'COSWIG', 'VOCKERODE', 'ROSSLAU', 'DESSAU', 'AKEN', 'BARBY', 'SCHOENEBECK', 'MAGDEBURG-BUCKAU', 'MAGDEBURG-STROMBRUECKE', 'MAGDEBURG-ROTHENSEE', 'NIEGRIPP AP', 'ROGAETZ', 'TANGERMUENDE', 'STORKAU', 'SANDAU', 'SCHARLEUK', 'WITTENBERGE', 'MUEGGENDORF', 'SCHNACKENBURG', 'LENZEN', 'GORLEBEN', 'DOEMITZ', 'DAMNATZ', 'HITZACKER', 'NEU DARCHAU', 'BLECKEDE', 'BOIZENBURG', 'HOHNSTORF', 'ARTLENBURG', 'GEESTHACHT', 'RHEINWEILER', 'BREISACH', 'RUST', 'OTTENHEIM', 'KEHL-KRONENHOF', 'IFFEZHEIM', 'PLITTERSDORF', 'MAXAU', 'PHILIPPSBURG', 'SPEYER', 'MANNHEIM', 'WORMS', 'NIERSTEIN-OPPENHEIM', 'MAINZ', 'OESTRICH', 'BINGEN', 'KAUB', 'SANKT GOAR', 'BOPPARD', 'BRAUBACH', 'KOBLENZ', 'ANDERNACH', 'OBERWINTER', 'BONN', 'KOELN', 'DUESSELDORF', 'RUHRORT', 'WESEL', 'REES', 'EMMERICH'.

`uuid`
has to be type `character` and has to have a length of one. Permitted values are: '7cb7461b-3530-4c01-8978-7f676b8f71ed', '85d686f1-55b2-4d36-8dba-3207b50901a7', '70272185-b2b3-4178-96b8-43bea330dcae', '24440872-5bd2-4fb3-8554-907b49816c49', 'b04b739d-7ffa-41ee-9eb9-95cb1b4ef508', '16b9b4e7-be14-41fd-941e-6755c97276cc', '83bbaedb-5d81-4bc6-9f66-3bd700c99c1f', 'f3dc8f07-c2bb-4b92-b0b0-4e01a395a2c6', 'c093b557-4954-4f05-8f5c-6c6d7916c62d', '070b1eb4-3872-4e07-b2e5-e25fd9251b93', '1ce53a59-33b9-40dc-9b17-3cd2a2414607', 'ae93f2a5-612e-4514-b5fd-9c8aecdd73c7', 'e97116a4-7d30-4671-8ba1-cdce0a153d1d', '1edc5fa4-88af-47f5-95a4-0e77a06fe8b1', '094b96e5-caeb-46d3-a8ee-d44182add069', '939f82ec-15a9-49c8-8828-dc2f8a2d49e2', '90bcb315-f080-41a8-a0ac-6122331bb4cf', 'b8567c1e-8610-4c2b-a240-65e8a74919fa', 'ccccb57f-a2f9-4183-ae88-5710d3afaefd', 'e30f2e83-b80b-4b96-8f39-fa60317afcc7', '3adf88fd-fd7a-41d0-84f5-1143c98a6564', '133f0f6c-2ca1-4798-9360-5b5f417dd839', '13e91b77-90f3-41a5-a320-641748e9c311', 'de4cc1db-51cb-4b62-bee2-9750cbe4f5c4', 'f4c55f77-ab80-4e00-bed3-aa6631aba074', 'e32b0a28-8cd5-4053-bc86-fff9c6469106', 'cbf3cd49-91bd-49cc-8926-ccc6c0e7eca4', '48f2661f-f9cb-4093-9d57-da2418ed656e', '550e3885-a9d1-4e55-bd25-34228bd6d988', 'c80a4f21-528c-4771-98d7-10cd591699a4', 'ac507f42-1593-49ea-865f-10b2523617c7', '6e3ea719-48b1-408a-bc55-0986c1e94cd5', 'c233674f-259a-4304-b81f-dce1f415d85b', 'a26e57c9-1cb8-4fca-ba80-9e02abc81df8', '67d6e882-b60c-40d3-975c-a6d7a2b4e40a', '6aa1cd8e-e528-4bcb-ba8e-705b6dcb7da2', '33e0bce0-13df-4ffc-be9d-f1a79e795e1c', 'd9289367-c8aa-4b6a-b1ad-857fec94c6bb', 'b3492c68-8373-4769-9b29-22f66635a478', '44f7e955-c97d-45c8-9ed7-19406806fb4c', '06b978dd-8c4d-48ac-a0c8-2c16681ed281', '9da1ad2b-88db-4cbb-8132-eddfab07d5ba', '5389b878-fad5-4f37-bb87-e6cb36b7078b', '787e5d63-61e2-48cc-acf0-633e2bf923f2', '23af9b02-5c82-4f6e-acb8-f92a06e5e4da', 'b02be240-1364-4c97-8bb6-675d7d842332', '6b774802-fcb5-49ae-8ecb-ecaf1a278b1c', 'b6c6d5c8-e2d5-4469-8dd8-fa972ef7eaea', '88e972e1-88a0-4eb9-847c-0925e5999a46', '2cb8ae5b-c5c9-4fa8-bac0-bb724f2754f4', '57090802-c51a-4d09-8340-b4453cd0e1f5', '844a620f-f3b8-4b6b-8e3c-783ae2aa232a', 'd28e7ed1-3317-41c5-bec6-725369ed1171', 'a37a9aa3-45e9-4d90-9df6-109f3a28a5af', '665be0fe-5e38-43f6-8b04-02a93bdbeeb4', '0309cd61-90c9-470e-99d4-2ee4fb2c5f84', '1d26e504-7f9e-480a-b52c-5932be6549ab', '550eb7e9-172e-48e4-ae1e-d1b761b42223', '2ff6379d-d168-4022-8da0-16846d45ef9b', 'd6dc44d1-

63ac-4871-b175-60ac4040069a', '4c7d796a-39f2-4f26-97a9-3aad01713e29', '5735892a-ec65-4b29-97c5-50939aa9584e', 'b45359df-c020-4314-adb1-d1921db642da', '593647aa-9fea-43ec-a7d6-6476a76ae868', 'a6ee8177-107b-47dd-bcfd-30960ccc6e9c', '8f7e5f92-1153-4f93-acba-ca48670c8ca9', 'c0f51e35-d0e8-4318-afaf-c5fcbc29f4c1', 'f33c3cc9-dc4b-4b77-baa9-5a5f10704398', '2f025389-fac8-4557-94d3-7d0428878c86', '9598e4cb-0849-401e-bba0-689234b27644'.

`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`, `flood1()` computes a 1d water level using `waterLevelFlood1` along the requested river section. This 1d water level is transfered 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

Rosenzweig S, Giebel H, Schleuter M (2011). "Ökologische Modellierungen für die Wasser- und Schifffahrtsverwaltung – Das integrierte Flussauenmodell INFORM in seiner neuesten Fassung (Version 3). Bundesanstalt für Gewässerkunde, Koblenz, Germany." doi:10.5675/bfg1667.

See Also

`df.gauging_data`, `getGaugingDataW`, `waterLevelFlood1`, `writeRaster`, `terraOptions`

Examples

```
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 transfered 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

Rosenzweig S, Giebel H, Schleuter M (2011). "Ökologische Modellierungen für die Wasser- und Schifffahrtsverwaltung – Das integrierte Flussauenmodell INFORM in seiner neuesten Fassung (Version 3). Bundesanstalt für Gewässerkunde, Koblenz, Germany." doi:10.5675/bfg1667.

See Also

df.gauging_data, getGaugingDataW, waterLevelFlood2, writeRaster, terraOptions

Examples

```
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)
```

---

flood3 — *Function to compute flood extent or flood duration* `SpatRaster` *along the German federal waterways Elbe and Rhine using the 1d water level algorithms* `hyd1d::waterLevel()` *and* `hyd1d::waterLevelPegelonline()`

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 transfered 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)].

See Also

waterLevel, waterLevelPegelonline, writeRaster, terraOptions

Examples

```
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 by 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 permanentely 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`.

See Also

waterLevel, waterLevelPegelonline

Examples

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

# create a random points object
c <- st_crs(25833)
e <- st_as_sfc(st_bbox(c(xmin = 309000, xmax = 310000,
                         ymin = 5749000, ymax = 5750000)))
st_crs(e) <- c
set.seed(123)
points <- st_sample(e, size = 10, "random")
p <- data.frame(id = 1:10)
st_geometry(p) <- points

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

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

---

getDEM — *Function to obtain the digital elevation models for the active floodplains 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 permanentely 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

`SpatRaster` object containing elevation data for the selected floodplain region.

References

Weber A (2020). "Digital elevation models of German waterway and navigation authorities - Version 0.1.0." https://doi.org/10.5675/BfG-2011.

Weber A (2020). "Digital elevation model (DEM1) of the River Elbe floodplain between Schmilka and Geesthacht, Germany." https://doi.org/10.1594/PANGAEA.919293.

Weber A (2020). "Digital elevation model (DEM1) of the River Rhine floodplain between Iffezheim and Kleve, Germany." https://doi.org/10.1594/PANGAEA.919308.

Examples

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

`hydflood` *hydflood: A package to compute flood extent and duration along the German federal waterways Elbe and Rhine*

Description

To be extended ...

hydSpatRaster *Initialize a* `SpatRaster` *for the flood-functions*

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

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

Arguments

`filename_dem` an optional argument of length 1 with type `character` specifying a filename of a **d**igital **e**levation **m**odel 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 **c**ross **s**ection **a**rea 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.

ext	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.
crs	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 permanentely 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

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

References

Wasserstraßen- und Schifffahrtsverwaltung des Bundes (WSV) (2016). “Digitales Geländemodell des Wasserlaufs (DGM-W).” https://www.govdata.de/daten/-/details/1c669080-c804-11e4-8731-1681e6b88ec1.

Brockmann H, Großkordt U, Schumann L (2008). “Auswertung digitaler Fernerkundungsaufnahmen des Elbe-Wasserlaufes (FE-Datenauswertung Elbe).”

Brockmann H, Schumann L (2012). “Produktblatt: DGM-W Elbe-Lenzen, 2003-2011.”

Brockmann H, Großkordt U, Schumann L (2008). “Digitales Geländemodell des Rhein-Wasserlaufes von Iffezheim bis Bonn (DGM-W Rhein).”

smile consult GmbH & Inphoris GmbH (2011). “DGM-W Oberrhein 1.”

FUGRO-HGN GmbH (2011). “Aufbau eines Digitalen Geländemodells des Oberrheinwasserverlaufes (DGM-W Oberrhein-2, Basel bis Iffezheim).”

ARGE Vermessung Schmid - Inphoris (2012). “Aufbau eines Digitalen Geländemodells des Niederrheinwasserlaufes (DGM-W Niederrhein).”

Weber A (2020). “Digital elevation models of German waterway and navigation authorities - Version 0.1.0.” https://doi.org/10.5675/BfG-2011.

Weber A (2020). “Digital elevation model (DEM1) of the River Elbe floodplain between Schmilka and Geesthacht, Germany.” https://doi.org/10.1594/PANGAEA.919293.

Weber A (2020). “Digital elevation model (DEM1) of the River Rhine floodplain between Iffezheim and Kleve, Germany.” https://doi.org/10.1594/PANGAEA.919308.

Bundesanstalt für Gewässerkunde (2016). "FLYS – Flusshydrologischer Webdienst." http://www.bafg.de/DE/08_Ref/M2/03_Fliessgewmod/01_FLYS/flys_node.html.

Brunotte E, Dister E, Günther-Diringer D, Koenzen U, Mehl D (2009). "Zustand der rezenten Flussauen in Deutschland - Geodaten."

See Also

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

Examples

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

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

Arguments

name	either 'Elbe' or 'Rhine'.

Value

sf with the projected active floodplain

See Also

sf.afe, sf.afr

Examples

```
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

Brunotte E, Dister E, Günther-Diringer D, Koenzen U, Mehl D (2009). "Flussauen in Deutschland - Erfassung und Beurteilung des Auenzustandes."

Brunotte E, Dister E, Günther-Diringer D, Koenzen U, Mehl D (2009). "Zustand der rezenten Flussauen in Deutschland - Geodaten."

Bundesamt für Naturschutz (2009). "Zustand der rezenten Flussauen in Deutschland."

See Also

`sf.af`, `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

Brunotte E, Dister E, Günther-Diringer D, Koenzen U, Mehl D (2009). "Flussauen in Deutschland - Erfassung und Beurteilung des Auenzustandes."

Brunotte E, Dister E, Günther-Diringer D, Koenzen U, Mehl D (2009). "Zustand der rezenten Flussauen in Deutschland - Geodaten."

Bundesamt für Naturschutz (2009). "Zustand der rezenten Flussauen in Deutschland."

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

Weber A (2020). "Digital elevation models of German waterway and navigation authorities - Version 0.1.0." https://doi.org/10.5675/BfG-2011.

Weber A (2020). "Digital elevation model (DEM1) of the River Elbe floodplain between Schmilka and Geesthacht, Germany." https://doi.org/10.1594/PANGAEA.919293.

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

Weber A (2020). "Digital elevation models of German waterway and navigation authorities - Version 0.1.0." https://doi.org/10.5675/BfG-2011.

Weber A (2020). "Digital elevation model (DEM1) of the River Rhine floodplain between Iffezheim and Kleve, Germany." https://doi.org/10.1594/PANGAEA.919308.

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 7330932961457502484041108035 5538181108035 Bu.15 01
```

Within each row very specific sections code for specific attributes:

section	column(s)	attribute	column name in result
1	1	state id, here W=WSV	sid

1	2-5	Federal Waterway ID	fwid
2	6-8	WSV point type	wsvpt
3	9	blank	-
3	10-15	river station (km)	station
4	16	bank: 1 left, 2 right	bank
4	17-20	continuous id	id
5	21-30	easting in GK-coordinates	x
5	31-40	northing in GK-coordinates	y
6	41-46	datum of measurement	date_coor
6	47	accuracy	acc_coor
6	48-54	elevation	z
6	55-60	date of the elevation measurement	date_z
6	61	accuracy of the elevation measurement	acc_z
6	62-64	type of measurement	tom
7	65-84	comment	comment
8	85-86	point status	status

In a second step these points are aggregated to a `sfc_LINESTRING` using the grouping column `id`.

Usage

```
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

`sfc_LINESTRING`.

Examples

```
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   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:

`|_1_|2_|_3______|4|____________5______________|_______6_______|_____7___|_8_|`

`W0701 55 2594611 1330938065557502425901108035 5795591108035 Bu.15 01`

`W0701 57 2594611 7330932961457502484041108035 5538181108035 Bu.15 01`

Within each row very specific sections code for specific attributes:

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

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

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

# write temporary w80 file
cat("W0701 55 2594611   13309380655575024259011O8035 5795591108035        Bu.15   01\n",
    file = filename)
cat("W0701 57 2594611   73309329614575024840411O8035 5538181108035        Bu.15   01\n",
    file = filename, append = TRUE)

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

Index