Package ‘transfR’

July 24, 2026

Type Package

Title Transfer of Hydrograph from Gauged to Ungauged Catchments

Version 1.1.18

Date 2026-07-23

Depends R (>= 4.1.0), stars (>= 0.4-0), sf (>= 0.8-0)

Imports doParallel, foreach, units, glmnet, Matrix, Rdpack

RdMacros Rdpack

VignetteBuilder knitr

Suggests knitr, rmarkdown, elevatr, progress, whitebox

Description Provides geomorphology-based hydrological modelling for transferring streamflow measurements from gauged to ungauged catchments. Inverse modelling makes it possible to estimate net rainfall from streamflow measurements following Boudhraâ et al. (2018) <doi:10.1080/02626667.2018.1425801>. The resulting net rainfall is then interpolated to ungauged catchments to simulate streamflow following de Lavenne et al. (2016) <doi:10.1002/2016WR018716>.

License GPL-2

URL `https://forge.inrae.fr/hydroGR/transfr`

BugReports `https://forge.inrae.fr/hydroGR/transfr/-/work_items`

NeedsCompilation yes

Encoding UTF-8

Author Alban de Lavenne [aut, cre] (ORCID: <https://orcid.org/0000-0002-9448-3490>),
Christophe Cudennec [ths] (ORCID: <https://orcid.org/0000-0002-1707-8926>),
Tom Loree [ctb],
Hervé Squividant [ctb]

Maintainer Alban de Lavenne <alban.delavenne@inrae.fr>

Config/roxygen2/version 8.0.0

Repository CRAN

Date/Publication 2026-07-24 10:00:02 UTC

Contents

`transfR-package` *Transfer of Hydrograph from Gauged to Ungauged Catchments*

Description

This R package proposes a geomorphology-based hydrological modelling approach to transfer streamflow measurements from gauged catchments to ungauged catchments, i.e. where there is no station monitoring the streamflow. It follows a runoff-runoff approach, i.e. it directly combines the observed streamflow series available at monitoring stations to estimate streamflow elsewhere in the surrounding rivers without the need to implement a full rainfall-runoff model. The package itself and the theoretical aspects of the approach are presented in detail and discussed by de Lavenne et al. (2023).

— Short description of the modelling approach

The hydrological modelling is based on a description of the hydro-geomorphometry of the river network, which can be easily observed for any given outlet. An inversion of this model for a gauged catchment allows the observed streamflow series to be deconvoluted in order to estimate an almost scale-independent signal, namely the net rainfall (Boudhraâ et al. 2018). Transferring this estimate of the net rainfall series to a targeted ungauged catchment then makes it possible to simulate the streamflow there. The use of streamflow observations from several gauged catchments in the neighbourhood increases the robustness of the simulation (de Lavenne et al. 2016). The methodology was first implemented on a few catchments in semiarid Tunisia at the event time scale (Boudhraâ et al. 2009), then in dense configurations of neighbouring and nested catchments in France, which mainly have a temperate oceanic climate (de Lavenne et al. 2015; de Lavenne et al. 2016; de

Lavenne and Cudennec 2019), and in snow-influenced Québec, Canada (Ecrepont et al. 2019).

— Functions and objects

To implement the method, it is advised to explore the following functions in this order:

- `as_transfr` creates a "transfR" database from a "stars" object and a morphometric description of the catchments (hydraulic lengths)
- `velocity` estimates the main model parameter, i.e. the streamflow velocity, from different regionalisation strategies
- `uh` estimates a simple linear model, i.e. the unit hydrograph, based on the analysis of catchment geomorphology and streamflow velocity
- `rapriori` provides an a priori estimate of net rainfall, as needed for model inversion
- `inversion` estimates net rainfall by inverse modelling
- `hdist` computes hydrological distances between catchments, such as the rescaled Ghosh distances
- `mixr` estimates the net rainfall of one catchment by averaging the net rainfall of neighbouring gauged catchments, weighted according to hydrological distances
- `convolution` computes the convolution of the net rainfall by the unit hydrograph to estimate streamflow

— How to get started

This package comes with two datasets (Blavet and Oudon) that contain all the necessary inputs to test the package and perform discharge prediction. Users are advised to check the 'Get started with transfR' vignette (`vignette("V01_get_started", package = "transfR")`) that provides a complete implementation of the method with the Oudon dataset. Two additional vignettes are proposed to help the preparation of input data: a spatiotemporal array of observed discharge (`vignette("V02_inputs_preparation_s`
`package = "transfR")`) and a morphometric description of the catchments (`vignette("V03_inputs_preparation_whiteb`
`package = "transfR")`). In addition, each function comes with different examples.

A detailed description of the modelling approach and the package has been published by de Lavenne et al. (2023): the theoretical aspects of each modelling step are described in more detail, arguments justifying the default values used in the functions are presented, and limitations of the approach are discussed for a consistent implementation of the approach.

For the French region of Brittany, a web service using this package was developed to facilitate the implementation of the method without the need for the user to have programming skills in R or to collect the necessary input data (Dallery et al. 2020).

References

Boudhraâ H, Cudennec C, Slimani M, Andrieu H (2009). "Hydrograph transposition between basins through a geomorphology-based deconvolution-reconvolution approach." *IAHS publication*, **333**, 76.

Boudhraâ H, Cudennec C, Andrieu H, Slimani M (2018). "Net rainfall estimation by the inversion of a geomorphology-based transfer function and discharge deconvolution." *Hydrological Sciences Journal*, **63**(2), 285–301. doi:10.1080/02626667.2018.1425801.

Ecrepont S, Cudennec C, Anctil F, Jaffrézic A (2019). "PUB in Québec: A robust geomorphology-based deconvolution-reconvolution framework for the spatial transposition of hydrographs." *Journal of Hydrology*, **570**, 378–392. doi:10.1016/j.jhydrol.2018.12.052.

Dallery D, Squividant H, de Lavenne A, Launay J, Cudennec C (2020). "An end-user-friendly hydrological Web Service for hydrograph prediction in ungauged basins." *Hydrological Sciences Journal*, 1–9. doi:10.1080/02626667.2020.1797045.

de Lavenne A, Boudhraâ H, Cudennec C (2015). "Streamflow prediction in ungauged basins through geomorphology-based hydrograph transposition." *Hydrology Research*, **46**(2), 291–302. doi:10.2166/nh.2013.099.

de Lavenne A, Skøien JO, Cudennec C, Curie F, Moatar F (2016). "Transferring measured discharge time series: Large-scale comparison of Top-kriging to geomorphology-based inverse modeling." *Water Resources Research*, **52**(7), 5555–5576. doi:10.1002/2016WR018716.

de Lavenne A, Cudennec C (2019). "Assessment of freshwater discharge into a coastal bay through multi-basin ensemble hydrological modelling." *Science of The Total Environment*, **669**, 812 - 820. ISSN 0048-9697. doi:10.1016/j.scitotenv.2019.02.387.

de Lavenne A, Loree T, Squividant H, Cudennec C (2023). "The transfR toolbox for transferring observed streamflow series to ungauged basins based on their hydrogeomorphology." *Environmental Modelling & Software*, **159**, 105562. ISSN 1364-8152. doi:10.1016/j.envsoft.2022.105562.

`as_transfr` *Create transfR object*

Description

Create a transfR object or add new attributes to a transfR object.

Usage

```
as_transfr(
  object,
  st,
  uc,
  lagtime,
  surface,
  delineation,
  outlet,
  centroid,
  uh,
  hl
)
```

Arguments

`object`	object of class `transfR`

st	spatio-temporal array of class stars. Observed discharge must be identified by the column name 'Qobs'. Time should be the first dimension, space the second dimension. If no unit is provided, Qobs is assumed to be in [m^3/s] and RnInv is assumed to be in [mm/h] (or [mm/d] at daily time step).
uc	vector of the streamflow velocities of the catchments. If no unit is provided, uc is assumed to be in [m/s].
lagtime	vector of the lag times of the catchments. If no unit is provided, lagtime is assumed to be in [h].
surface	vector of the surfaces of the catchments. If no unit is provided, surface is assumed to be in [km^2].
delineation	spatial layer of the boundary of the catchments of class sfc_POLYGON.
outlet	spatial layer of the outlets of the catchments of class sfc_POINT.
centroid	spatial layer of the centroids of the catchments of class sfc_POINT.
uh	list of the unit hydrographs of the catchments.
hl	list of hydraulic length objects, one per catchment. Each element must be a stars object, a numeric matrix or a numeric vector. For a single catchment, one object can be supplied directly. If no unit is provided, hl is assumed to be in [m]. See details below.

Details

This function creates an object of class transfR or adds new attributes to an existing transfR object. It can be used to gather and organise most of the inputs and outputs of the other functions, like streamflow velocities, unit hydrographs, a priori estimates of net rainfall, inversions and simulations for every catchment.

This function can be used to organise the two user inputs required for a conventional use of the package, namely st and hl. The hydraulic lengths are defined as the flow path length from each pixel to the outlet within the river network (Cudennec et al. 2004; Aouissi et al. 2013). Catchment delineations and hydraulic lengths need to be prepared beforehand by the user. This package does not provide functions to create them. However, several GIS software packages can extract them from a digital elevation model, such as GRASS toolkits (Jasiewicz and Metz 2011), Whitebox GAT (see Lindsay (2016) or WhiteboxTools), TauDEM (D. Tarboton, Utah State University) or online services (see Squividant et al. (2015) for catchment delineation in the Brittany French region).

Value

An object of class transfR.

References

Aouissi J, Pouget J, Boudhraâ H, Storer G, Cudennec C (2013). "Joint spatial, topological and scaling analysis framework of river-network geomorphometry." *Géomorphologie : relief, processus, environnement*, **19**(1), 7–16. doi:10.4000/geomorphologie.10082.

Cudennec C, Fouad Y, Gatot IS, Duchesne J (2004). "A geomorphological explanation of the unit hydrograph concept." *Hydrological Processes*, **18**(4), 603–621. doi:10.1002/hyp.1368.

Jasiewicz J, Metz M (2011). "A new GRASS GIS toolkit for Hortonian analysis of drainage networks." *Computers & Geosciences*, **37**(8), 1162–1173. doi:10.1016/j.cageo.2011.03.003.

Lindsay JB (2016). "Whitebox GAT: A case study in geomorphometric analysis." *Computers & Geosciences*, **95**, 75–84. doi:10.1016/j.cageo.2016.07.003.

Squividant H, Bera R, Aurousseau P, Cudennec C (2015). "Online watershed boundary delineation: sharing models through Spatial Data Infrastructures." *Proceedings of the International Association of Hydrological Sciences*, **368**, 144–149. doi:10.5194/piahs3681442015.

Examples

```
data(Oudon)
object <- as_transfr(st = Oudon$obs, hl = Oudon$hl)
```

Blavet *Blavet French river dataset*

Description

'Blavet' is a dataset for the Blavet River on the Brittany peninsula in France and two neighbouring rivers (Claie and Coët-Organ). It contains all the inputs required to test the package and predict discharge at the outlets of six catchments:

- J5613010: Evel at Guénin (316 km²)
- J5618310: Fremeur et Guénin (15.1 km²)
- J5618320: Fremeur et Pluméliau (5.88 km²)
- J5704810: Coët-Organ at Quistinic (47.7 km²)
- J8433020: Claie at Saint-Jean-Brévelay (135 km²)
- AgrHys: Coët-Dan at Naizin (4.9 km²)

Hourly discharge observations for the six catchments are provided for one hydrological year, from 2013-10-01 to 2014-10-01. They were extracted from the French HYDRO database (http://www.hydro.eaufrance.fr), except for the Coët-Dan observations, which were provided by the AgrHys Environmental Research Observatory (Fovet et al. 2018), managed by INRAE (https://www6.inrae.fr/ore_agrhys_eng). Catchment delineations and the corresponding hydraulic-length maps were extracted from a digital elevation model with a resolution of 100 m.

Format

'Blavet' is a list of three objects:

- hl: A list of stars objects containing the six raster maps of hydraulic length.
- obs: A stars object with two dimensions (time and space, with catchment delineations as spatial support) and one attribute (discharge observations).
- network: An sf object representing the [French TOPAGE river network](https://bdtopage.eaufrance.fr/). It can be downloaded using the "Sandre - Eau France" Web Feature Service (WFS), as shown in the example below.

Source

http://www.hydro.eaufrance.fr

https://www6.inrae.fr/ore_agrhys_eng

http://bdtopage.eaufrance.fr

References

Fovet O, Ruiz L, Gruau G, Akkal N, Aquilina L, Busnot S, Dupas R, Durand P, Faucheux M, Fauvel Y, Fléchard C, Gilliet N, Grimaldi C, Hamon Y, Jaffrezic A, Jeanneau L, Labasque T, Henaff GL, Mérot P, Molénat J, Petitjean P, Pierson-Wickmann A, Squividant H, Viaud V, Walter C, Gascuel-Odoux C (2018). "AgrHyS: An Observatory of Response Times in Agro-Hydro Systems." *Vadose Zone Journal*, **17**(1), 180066. doi:10.2136/vzj2018.04.0066.

Examples

```
## Not run:
# Working directory
wd <- tempdir(check = TRUE)
# Define a bbox that will encompass the catchments of the study area
blavet_bbox <- st_bbox(c(xmin = -3.3, xmax = -2.7, ymax = 48.11, ymin = 47.77),
                        crs = st_crs(4326))
# Download a French Topage river network within the bbox using the "Sandre - Eau France" WFS
download.file(url = paste0("https://services.sandre.eaufrance.fr/geo/topage2019",
                     "?request=GetFeature&service=WFS&version=2.0.0",
                     "&typeName=CoursEau_FXX_Topage2019",
                     "&outputFormat=application/json;
                     paste0(blavet_bbox[c("ymin","xmin","ymax","xmax")],
                            collapse=",")),
              destfile = file.path(wd,"CoursEau_FXX_Topage2019.geojson"))
CoursEau_Topage2019 <- st_read(dsn = file.path(wd,"CoursEau_FXX_Topage2019.geojson"),
                               drivers = "GeoJSON", stringsAsFactors = FALSE, quiet = FALSE,
                                  query = "SELECT gid FROM CoursEau_FXX_Topage2019")

## End(Not run)
```

---

convolution — *Convolution of net rainfall with unit hydrograph*

Description

Simulate the discharge by a convolution between the unit hydrograph and the net rainfall.

Usage

```
convolution(Rn, ...)

## Default S3 method:
convolution(Rn, uh, continuous = FALSE, ...)
```

```
## S3 method for class 'units'
convolution(Rn, uh, ...)

## S3 method for class 'transfR'
convolution(
  Rn,
  Rcol = "RnSim",
  Qcol = "Qsim",
  save_donor = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

Rn	net rainfall vector or an object of class transfR
...	further arguments passed to or from other methods
uh	unit hydrograph vector
continuous	logical indicating whether simulated values should be omitted when a time step might be influenced by past or future rainfall (according to the time span of the unit hydrograph)
Rcol	name of the space-time attribute for the net rainfall simulation in the transfR object
Qcol	name of the space-time attribute for the discharge in the transfR object
save_donor	logical indicating if additional discharge simulations should be computed using the net rainfall of each individual donor catchment instead of just the weighted average net rainfall. This requires save_donor to have been TRUE when using mixr
verbose	logical indicating if information messages should be written to the console

Value

An object of the same class as Rn. If Rn is a transfR object, it is returned with the computed results added as new attributes.

Examples

```
data(Oudon)
icatch <- 1
uc <- velocity(hl = Oudon$hl[[icatch]])
uh <- uh(hl = Oudon$hl[[icatch]], uc = uc, deltat = units::set_units(1,"h"))$prob
Rn <- units::set_units(c(1,5,2), "mm/h")
Qsim <- convolution(Rn = Rn, uh = uh)
```

hdist *Geographical distance between catchments*

Description

Calculate distances between two sets of catchments using their spatial support.

Usage

```
hdist(x, y, ...)

## S3 method for class 'sfc'
hdist(
  x,
  y,
  method = "rghosh",
  gres = 5,
  ditself = FALSE,
  maxsample = 25000,
  proj = NULL,
  parallel = FALSE,
  cores = NULL,
  verbose = TRUE,
  ...
)

## S3 method for class 'sf'
hdist(x, y, ...)

## S3 method for class 'stars'
hdist(x, y, ...)

## S3 method for class 'transfR'
hdist(x, y, method = "rghosh", weightO = 0.8, weightC = 0.2, ...)
```

Arguments

`x`	sf, stars or transfR object containing the first set of catchments
`y`	sf, stars or transfR object containing the second set of catchments
`...`	further arguments passed to or from other methods
`method`	the method to use for computing distance. This must be one of "ghosh", "rghosh", "points", "centroids", "combined"
`gres`	resolution of spatial discretisation (number of points by km^2) for Ghosh distance
`ditself`	logical value indicating if the distance to itself should be computed. It will add one row and one column in the distance matrix. Only used if method is "ghosh"

maxsample	maximum number of sampling points for each catchment during spatial discretisation
proj	logical indicating whether spatial layers use a projection. If TRUE, Euclidean distance is used. If FALSE, the great-circle distance is used
parallel	logical indicating whether the computation should be parallelised.
cores	the number of cores to use for parallel execution if `parallel` is TRUE. If not specified, the number of cores is set to the value of `parallel::detectCores()`.
verbose	logical indicating if information messages should be written to the console
weightO	weight given to the distance between outlets if method is "combined"
weightC	weight given to the distance between centroids if method is "combined"

Details

The `method` "ghosh" refers to a simplification of the distance defined by Ghosh (1951) as proposed by Gottschalk (1993); Gottschalk et al. (2011). The rescaled Ghosh distance (`method` "rghosh") is calculated following de Lavenne et al. (2016).

Value

A matrix of class units with the catchments of `x` organised in rows and the catchments of `y` organised in columns.

References

Ghosh B (1951). “Random distances within a rectangle and between two rectangles.” *Bull. Calcutta Math. Soc*, **43**(1), 17–24.

Gottschalk L (1993). “Interpolation of runoff applying objective methods.” *Stochastic Hydrology and Hydraulics*, **7**(4), 269–281. doi:10.1007/BF01581615.

Gottschalk L, Leblois E, Skøien JO (2011). “Distance measures for hydrological data having a support.” *J. Hydrol.*, **402**(3-4), 415–421. doi:10.1016/j.jhydrol.2011.03.020.

de Lavenne A, Skøien JO, Cudennec C, Curie F, Moatar F (2016). “Transferring measured discharge time series: Large-scale comparison of Top-kriging to geomorphology-based inverse modeling.” *Water Resources Research*, **52**(7), 5555–5576. doi:10.1002/2016WR018716.

Examples

```
data(Oudon)
catchments <- st_geometry(Oudon$obs)
hdist(x = catchments[1:2], y = catchments[3:5], gres = 5, method = "rghosh",
parallel= TRUE, cores=2)
```

`inversion` *Estimate net rainfall by inversion*

Description

Estimate net rainfall by inverse modelling, where the model is a convolution between net rainfall and a unit hydrograph in order to simulate discharge.

Usage

```
inversion(Qobs, ...)

## Default S3 method:
inversion(Qobs, uh, RnAp, deltat, ...)

## S3 method for class 'units'
inversion(
  Qobs,
  uh,
  RnAp,
  deltat,
  Bd = 0.01,
  Dd = 1,
  Bp = 0.001,
  Tp = 20,
  Ad = 0.01,
  Ap = 0.9,
  warmup = 10,
  cooldown = 8,
  dosplit = TRUE,
  split = 30,
  fixedpar = TRUE,
  parallel = FALSE,
  cores = NULL,
  cluster = NULL,
  ...
)

## S3 method for class 'transfR'
inversion(
  Qobs,
  verbose = TRUE,
  parallel = FALSE,
  cores = NULL,
  cluster = NULL,
  ...
)
```

Arguments

`Qobs`	discharge vector or object of class `transfR`. If no unit is provided, `Qobs` is assumed to be in [mm/h]
`...`	further arguments passed to or from other methods
`uh`	unit hydrograph vector
`RnAp`	a priori estimate of net rainfall. If no unit is provided, `RnAp` is assumed to be in [mm/h]
`deltat`	time step of the time series. If no unit is provided, `deltat` is assumed to be in [min]
`Bd`	parameter used to maintain a minimum value of standard deviation for low discharge values. If no unit is provided, `Bd` is assumed to be in [mm/h]
`Dd`	decorrelation time of discharge errors. If no unit is provided, `Dd` is assumed to be in [h]
`Bp`	parameter used to maintain a minimum value of standard deviation for low net rainfall values. If no unit is provided, `Bp` is assumed to be in [mm/h]
`Tp`	decorrelation time of net rainfall errors. If no unit is provided, `Tp` is assumed to be in [h]
`Ad`	parameter equivalent to the coefficient of variation of the discharge measurement error. If no unit is provided, `Ad` is assumed to be dimensionless
`Ap`	parameter equivalent to the coefficient of variation of the net rainfall error. If no unit is provided, `Ap` is assumed to be dimensionless
`warmup`	length of the warmup period. If no unit is provided, `warmup` is assumed to be in [days]
`cooldown`	length of the period removed at the end of the simulation. If no unit is provided, `cooldown` is assumed to be in [days]
`dosplit`	logical, if true the inversion is performed by subperiods of length defined by `split`
`split`	length of the subperiods if dosplit is true. If no unit is provided, `split` is assumed to be in [days]
`fixedpar`	logical; if FALSE, Ap and Ad are calibrated dynamically according to the coefficient of variation of RnAp and Qobs respectively (see details)
`parallel`	logical indicating whether the computation should be parallelised. Subperiods are parallelised for a single time series, whereas catchments are parallelised for a `transfR` object
`cores`	the number of cores to use for parallel execution if `parallel` is TRUE. If not specified, the number of cores is set to the value of `parallel::detectCores()`
`cluster`	an optional parallel cluster to reuse
`verbose`	logical indicating if information messages should be written to the console

Details

Given a convolution between the unit hydrograph (uh) and net rainfall that simulates streamflow at the outlet (Qobs), where net rainfall is the only unknown variable, this function estimates net rainfall by inversion (Tarantola and Valette 1982; Menke 1989; Boudhraâ et al. 2018). It requires an a priori estimate of net rainfall (which can be obtained with rapriori) and a description of errors in discharge (Ad, Bd, Dd) and net rainfall (Ap, Bp, Tp); these errors are assumed to be Gaussian and unbiased. Default values of these parameters are taken from de Lavenne et al. (2016). If `fixedpar` is deactivated, Ap is estimated at 20 of variation of Qobs.

It is recommended to use `warmup` and `cooldown` periods to reduce oscillations caused by inversion.

If Qobs is a `transfR` object, results are stored in a new space-time attribute called "RnInv".

Value

An object of the same class as Qobs. If Qobs is a transfR object, it is returned with the computed results added as new attributes.

References

Boudhraâ H, Cudennec C, Andrieu H, Slimani M (2018). "Net rainfall estimation by the inversion of a geomorphology-based transfer function and discharge deconvolution." *Hydrological Sciences Journal*, **63**(2), 285–301. doi:10.1080/02626667.2018.1425801.

de Lavenne A, Skøien JO, Cudennec C, Curie F, Moatar F (2016). "Transferring measured discharge time series: Large-scale comparison of Top-kriging to geomorphology-based inverse modeling." *Water Resources Research*, **52**(7), 5555–5576. doi:10.1002/2016WR018716.

Menke W (1989). *Geophysical data analysis: discrete inverse theory*, volume 45. Academic Press.

Tarantola A, Valette B (1982). "Inverse problems= quest for information." *Journal of Geophysics*, **50**(3), 150–170.

See Also

rapriori

Examples

```
data(Oudon)
icatch <- 1 # Catchment index
itime <- 1:1000 # Using the first values for a quicker example
Qobs <- Oudon$obs[["Qobs"]][itime,icatch]
Qspec <- units::set_units(Qobs/st_area(st_geometry(Oudon$obs)[icatch]), "mm/h")
deltat <- units::set_units(1, "h")
uc <- velocity(hl = Oudon$hl[[icatch]])
uh <- uh(hl = Oudon$hl[[icatch]], uc = uc, deltat = units::set_units(1,"h"))$prob
RnAp <- rapriori(Qobs = Qspec, lagtime = lagtime(hl = Oudon$hl[[icatch]], uc = uc),
deltat = deltat)
RnInv <- inversion(Qobs = Qspec, RnAp = RnAp, uh = uh, deltat = deltat, parallel = TRUE, cores=2)
```

`lagtime` *Lag time estimation*

Description

Estimate the lag time of the catchment.

Usage

```
lagtime(hl, ...)

## Default S3 method:
lagtime(hl, uc, ...)

## S3 method for class 'units'
lagtime(hl, uc, method = 1, ...)

## S3 method for class 'stars'
lagtime(hl, ...)

## S3 method for class 'transfR'
lagtime(hl, verbose = TRUE, ...)
```

Arguments

`hl`	hydraulic length of class `transfR`, `stars`, `matrix` or `vector`. If no unit is provided, `hl` is assumed to be in [m].
`...`	further arguments passed to or from other methods
`uc`	streamflow velocity. If no unit is provided, `uc` is assumed to be in [m/s].
`method`	integer describing the method to use for lag time estimation. Possible values: 1 (see details).
`verbose`	logical indicating if information messages should be written to the console.

Details

The function estimates the lag time of the catchment. It can be used to estimate one of the inputs of the function rapriori. If `method` is 1, the lag time is estimated from the ratio of the mean hydraulic length (`hl`) and the average streamflow velocity (`uc`).

Value

A numeric value of class units or, if `hl` is a transfR object, the `hl` object with the "lagtime" attribute added.

Examples

```
data(Oudon)
icatch <- 1
lagtime(Oudon$hl[[icatch]], uc = units::set_units(0.5, "m/s"))
```

---

mixr *Transfer of net rainfall to ungauged catchments*

Description

Combine the net rainfall of gauged catchments to simulate the net rainfall of an ungauged catchment.

Usage

```
mixr(
  obs,
  sim,
  mdist,
  msim = NULL,
  distance = "rghosh",
  gres = 5,
  weightO = 0.8,
  weightC = 0.2,
  similarity,
  similarity_obs,
  mdist_obs,
  FUN = "invKGE",
  symmetrize = mean,
  parallel = FALSE,
  cores = NULL,
  power = c(1, 0.5),
  ndonors = 5,
  donors = NULL,
  maxdist = 50000,
  min_ndonors = 3,
  flexible_donor = TRUE,
  two_steps = FALSE,
  cv = FALSE,
  save_donor = FALSE,
  save_model = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

obs	"transfR" object of the gauged catchments
sim	"transfR" object of the ungauged catchments
mdist	the geographical distance matrix between gauged and ungauged catchments, as computed by the function hdist, used for donor selection
msim	the similarity matrix of net rainfall time series for gauged catchments, as computed by rsimilarity. It serves as the target variable in a multivariate model that predicts this similarity at ungauged locations using several predictors (see the `similarity` argument) and optimizes the donors' weighting strategy. Thus, `msim` is required only if `similarity` and `similarity_obs` are lists containing several predictors
distance	the method to use for computing distance matrix if `mdist` is not provided. Possible values are "ghosh", "rghosh", "points", "centroids", "combined" as available in the function hdist
gres	resolution of spatial discretisation (number of points by km^2) for Ghosh distance (see the function hdist)
weightO	weight given to the distance between outlets if `distance` is "combined" (see the function hdist)
weightC	weight given to the distance between centroids if `distance` is "combined" (see the function hdist)
similarity	a hydrological similarity matrix between gauged and ungauged catchments used for donor weighting (`1/mdist^power[1]` is used if not provided). Gauged catchments should be in rows, ungauged catchments in columns. Several predictors of hydrological similarity can be provided in a list of matrices in order to be used by a multivariate model for predicting similarity between a gauged and an ungauged catchment (see `predictors` of the rsimilarity_model function).
similarity_obs	list of square matrices of hydrological similarity predictors between gauged catchments in order to train a multivariate model for predicting similarity (`msim`). Similar to `similarity` but between gauged catchments only. `similarity` will be used instead of `similarity_obs` if `cv` is TRUE
mdist_obs	the geographical distance matrix among gauged catchments, computed by the function hdist, used for training a first model of hydrological similarity
FUN	either a function or a character string specifying the name of a predefined function to quantify the similarity of two net rainfall time series. See rsimilarity for more details
symmetrize	a function to combine `FUN(Rn[,i],Rn[,j])` and `FUN(Rn[,j],Rn[,i])` into one similarity value (typically if `FUN=KGE`, `KGE(Rn[,i],Rn[,j])!=KGE(Rn[,j],Rn[,i])`). Default is mean
parallel	logical indicating if the computation should be parallelised
cores	the number of cores to use for parallel execution if `parallel` is TRUE. If not specified, the number of cores is set to the value of `parallel::detectCores()`
power	exponents. The first vector value is used in the inverse distance weighting strategy (as defined by the function weightr) and the second vector value is used during computation of Rn similarities (as defined by the function rsimilarity_model)

`ndonors`	maximum number of catchments to be used to simulate discharge of an ungauged catchment as defined by the function weightr
`donors`	list of vectors indicating the catchment IDs from which donors are selected for each ungauged catchment. If empty, the `ndonors` closest catchments are used
`maxdist`	maximum distance between a gauged and an ungauged catchment to allow the net rainfall to be transferred. This threshold is applied to the `mdist` distance matrix. If no unit is provided, `maxdist` is assumed to be in [m].
`min_ndonors`	minimum number of gauged catchments to start using rsimilarity_model to build a similarity model of Rn time series for the weighting strategy. So this argument is only used when `similarity` is a list of matrices. It cannot be less than 3. If less than `min_ndonors` donors are found inverse distance weighting is applied using `mdist`
`flexible_donor`	logical indicating if the donor catchments can change during the simulation period according to the availability of discharge observations. See weightr for more details
`two_steps`	logical indicating if a first model of hydrological similarity should be built first on geographical distances. Residuals of this first model will be predicted from predictors provided by `similarity` and `similarity_obs` in a second model (see `first_model` of rsimilarity_model)
`cv`	logical indicating whether cross-validation evaluation should be performed. If TRUE, the function estimates the net rainfall of every gauged catchment (`obs`) as if it were ungauged (leave-one-out evaluation)
`save_donor`	logical indicating if the net rainfall of each of the `ndonors` catchments should be stored in the sim object for further analysis. If TRUE, the function adds three new space-time attributes to the `sim` object called "RnDonor", "Idonor" and "Wdonor", describing the net rainfall, the ID and the weight of the donor catchments respectively
`save_model`	logical indicating whether additional results of the local similarity model should be saved. If true, then a list of data frames of observed and simulated similarities among gauged catchments neighbouring each ungauged catchment will be available in the "model_training" attribute of the output
`verbose`	logical indicating if information messages should be written to the console
`...`	other arguments to be passed to rsimilarity_model if `similarity` is a list of matrices

Details

This function is a wrapper function for hdist and weightr to directly estimate the net rainfall on a set of ungauged catchments (`sim`) from a set of gauged catchments (`obs`). It returns the simulated net rainfall as a new space-time attribute in the `sim` object called "RnSim". The simulated net rainfall of a given ungauged catchment i is a weighted average of the net rainfalls of `ndonors` gauged catchments j:

$$R_n^i = \Sigma_{j=1}^{ndonors} R_n^j \cdot \lambda_j$$

where λ_j are defined by an inverse distance weighting function (see weightr). If `similarity` is a list of predictors of hydrological similarity, then λ_j are provided by rsimilarity_model.

Value

The `sim` object with the computed results added as new attributes.

See Also

hdist, weightr, rsimilarity_model, rsimilarity

Examples

```
data(Oudon)
object <- as_transfr(st = Oudon$obs, hl = Oudon$hl)
object <- velocity(object)
object <- uh(object)
object <- lagtime(object)
object <- rapriori(object)
object <- inversion(object, parallel = TRUE, cores = 2)
mdist  <- hdist(x = object, y = object, method = "rghosh")
object <- mixr(obs = object, mdist = mdist, parallel = TRUE, cores=2,
cv = TRUE, flexible_donor = TRUE, save_donor = FALSE)
object <- convolution(object, save_donor = FALSE)
plot(object, i = 1, attribute = c("Qobs", "Qsim"))
```

`Oudon` *Oudon French river dataset*

Description

'Oudon' is a dataset for the Oudon River in France, part of the wider Loire catchment. It contains all the inputs required to test the package and predict discharge at the outlets of six sub-catchments:

- M3771810: Oudon at Châtelais (734 km²)
- M3774010: Chéran at la Boissière (85 km²)
- M3823010: Verzée at Bourg-d'Iré (205 km²)
- M3834030: Argos at Sainte-Gemmes-d'Andigné (153 km²)
- M3851810: Oudon at Segré (1310 km²)
- M3711810: Oudon at Cossé-le-Vivien (133 km²)

Hourly discharge observations for the six sub-catchments are provided from 2019-12-01 to 2020-03-01 and were extracted from the French HYDRO database (http://www.hydro.eaufrance.fr). Catchment delineations and the corresponding hydraulic-length maps were extracted from a digital elevation model with a resolution of 100 m.

Format

'Oudon' is a list of two objects:

- hl: A list of stars objects containing the six raster maps of hydraulic length.
- obs: A stars object with two dimensions (time and space, with catchment delineations as spatial support) and one attribute (discharge observations).

Source

http://www.hydro.eaufrance.fr

---

plot *Plot transfR object*

Description

Plot transfR object.

Usage

```
## S3 method for class 'transfR'
plot(
  x,
  y,
  i,
  attribute,
  main = sprintf("Catchment %i", i),
  xlab,
  ylab,
  format,
  at,
  nticks = 5,
  type = "l",
  lwd = 2,
  las = 1,
  cex.names = 1,
  col = c("#045a8d", "#fb8072", "#bebada", "#ffffb3", "#8dd3c7"),
  keeplocal = TRUE,
  ...
)
```

Arguments

x	transfR object
y	ignored
i	spatial index to plot
attribute	attribute of the transfR object to plot
main	a main title for the plot, see also title
xlab	a label for the x axis, defaults to a description of x
ylab	a label for the y axis, defaults to a description of y
format	format for labels of time series on x axis
at	a date-time or date object for ticks on x axis

nticks	number of ticks on x axis
type	1-character string giving the type of plot desired (for details, see plot)
lwd	the line width (for details, see par)
las	the style of axis labels (for details, see par)
cex.names	expansion factor for axis names (for details, see barplot)
col	a specification for the default plotting color (for details, see par)
keeplocal	logical indicating whether to preserve local graphical parameters
...	further specifications, see plot

Examples

```
data(Oudon)
object <- as_transfr(st = Oudon$obs, hl = Oudon$hl)
plot(object, attribute = "Qobs")
```

---

quick_transfr *Transfer of observed discharge from gauged catchments to ungauged catchments*

Description

Wrap up all the modelling steps into one function for a quick implementation of this R package.

Usage

```
quick_transfr(
  obs,
  sim,
  velocity = "loire2016",
  distance = "rghosh",
  gres = 5,
  weightO = 0.8,
  weightC = 0.2,
  power = 1,
  ndonors = 5,
  maxdist = 50000,
  flexible_donor = TRUE,
  cv = FALSE,
  save_donor = FALSE,
  warmup = 10,
  cooldown = 8,
  dosplit = TRUE,
  split = 30,
  parallel = FALSE,
  cores = NULL,
  verbose = TRUE
)
```

Arguments

obs	"transfR" object of the gauged catchments
sim	"transfR" object of the ungauged catchments
velocity	character string describing the method to estimate the streamflow velocity. See velocity for the available options (method argument)
distance	character string describing the method to compute the distance between catchments. See hdist for the available options (method argument)
gres	resolution of spatial discretisation (number of points by km²) for Ghosh distance. See hdist for more details
weightO	weight given to the distance between outlets if distance method is "combined". See hdist for more details
weightC	weight given to the distance between centroids if distance method is "combined". See hdist for more details
power	exponent applied in the inverse distance weighting strategy. See weightr for more details
ndonors	maximum number of catchments to be used to simulate discharge of an ungauged catchment. See weightr for more details
maxdist	maximum distance between a gauged and an ungauged catchment to allow the net rainfall to be transferred. This threshold is applied to the mdist distance matrix. If no unit is provided, maxdist is assumed to be in [m]. See mixr for more details
flexible_donor	logical indicating if the donor catchments can change during the simulation period according to the availability of discharge observations. See weightr for more details
cv	logical indicating whether cross-validation evaluation should be performed. If TRUE, the function estimates the net rainfall of every gauged catchment (obs) as if it were ungauged (leave-one-out evaluation)
save_donor	logical indicating if the net rainfall of each of the ndonors catchments should be stored in the sim object for further analysis. If TRUE, the function adds three new space-time attributes to the sim object called "RnDonor", "Idonor" and "Wdonor", describing the net rainfall, the ID and the weight of the donor catchments respectively. See mixr for more details
warmup	length of the warmup period. If no unit is provided, warmup is assumed to be in [days]. See inversion for more details
cooldown	length of the period removed at the end of the simulation. If no unit is provided, cooldown is assumed to be in [days]. See inversion for more details
dosplit	logical, if true the inversion is performed by subperiods of length defined by split. See inversion for more details
split	length of the subperiods if dosplit is true. If no unit is provided, split is assumed to be in [days]. See inversion for more details
parallel	logical indicating if the computation should be parallelised
cores	the number of cores to use for parallel execution if parallel is TRUE. If not specified, the number of cores is set to the value of parallel::detectCores()
verbose	logical indicating if information messages should be written to the console

Details

The function applies sequentially the following functions: velocity, uh, lagtime, rapriori, inversion, hdist, mixr and convolution. Please refer to the help of each of these functions and to transfR-package for a general description of the modelling approach.

Value

The `sim` object with the computed results added as new attributes.

See Also

velocity, uh, lagtime, rapriori, inversion, hdist, mixr, convolution

Examples

```
data(Oudon)
obs <- as_transfr(st = Oudon$obs[,,1:3], hl = Oudon$hl[1:3]) #gauged catchments
sim <- as_transfr(st = Oudon$obs[,,4:6], hl = Oudon$hl[4:6]) #catchments considered as ungauged
sim <- quick_transfr(obs, sim)
```

`rapriori` *A priori net rainfall estimation*

Description

A priori estimate of net rainfall as required for the inversion.

Usage

```
rapriori(Qobs, ...)

## Default S3 method:
rapriori(Qobs, area, lagtime, deltat, ...)

## S3 method for class 'units'
rapriori(Qobs, area, lagtime, deltat, ...)

## S3 method for class 'transfR'
rapriori(Qobs, verbose = TRUE, ...)
```

Arguments

`Qobs`	vector of discharge values or object of class `transfR`. If no unit is provided, `Qobs` is assumed to be in [m^3/s].
`...`	further arguments passed to or from other methods
`area`	drainage area of the catchment. If no unit is provided, `area` is assumed to be in [km^2].

lagtime — lag time value of the catchment. If no unit is provided, `lagtime` is assumed to be in [h].

deltat — time step of the time series. If no unit is provided, `deltat` is assumed to be in [min].

verbose — logical indicating if information messages should be written to the console

Details

The function estimates an a priori value for net rainfall from Qobs. It converts Qobs to specific discharge and removes the delay caused by transfer time in the river network (given by `lagtime` and which can be estimated with lagtime). If an object of class `transfR` is provided, `area` is estimated from its `st` attribute. Results are stored as a new space-time attribute, called "RnAp", in the `transfR` object.

Value

An object of the same class as `Qobs`. If `Qobs` is a transfR object, it is returned with the "RnAp" attribute added.

Examples

```
data(Oudon)
icatch <- 1
Qobs <- Oudon$obs[["Qobs"]][,icatch]
Qspec <- units::set_units(Qobs/st_area(st_geometry(Oudon$obs)[icatch]), "mm/h")
deltat <- units::set_units(1,"h")
uc <- velocity(hl = Oudon$hl[[icatch]])
uh <- uh(hl = Oudon$hl[[icatch]], uc = uc, deltat = deltat)$prob
RnAp <- rapriori(Qobs = Qspec, lagtime = lagtime(hl = Oudon$hl[[icatch]], uc = uc),
deltat = deltat)
```

---

rsimilarity — *Similarity of net rainfall time series*

Description

Calculate the similarity of net rainfall time series

Usage

```
rsimilarity(Rn, FUN = "invKGE", parallel = FALSE, cores = NULL)
```

Arguments

Rn	numeric matrix or data frame of net rainfall for gauged catchments (rows for time indices and columns for catchment indices)
FUN	either a function or a character string specifying the name of a predefined function to quantify the similarity of two net rainfall time series. Higher values should indicate a higher probability of similarity. Predefined functions include: "KGE" (Kling–Gupta efficiency), "invRMSE" (inverse of root mean square error), "invKGE" (1 / (1 - KGE)) and "RMSE" (root mean square error, should not be directly used as a similarity metric). The default is "invKGE"
parallel	logical indicating whether the computation should be parallelised
cores	the number of cores to use for parallel execution if `parallel` is TRUE. If not specified, the number of cores is set to the value of `parallel::detectCores()`

Value

A square matrix of the similarity metric between each pair of catchments

See Also

rsimilarity_model

Examples

```
data(Oudon)
obs <- as_transfr(st = Oudon$obs, hl = Oudon$hl)
obs <- velocity(obs, method = "loire2016")
obs <- uh(obs)
obs <- lagtime(obs)
obs <- rapriori(obs)
obs <- inversion(obs)
msim <- rsimilarity(Rn = obs$st$RnInv, FUN="KGE", parallel = TRUE, cores=2)
```

---

rsimilarity_model — *Prediction of hydrological similarity to drive transfer of hydrograph to ungauged catchments*

Description

Quantify the similarity of net rainfall and model this hydrological similarity from catchment similarity predictors

Usage

```
rsimilarity_model(
  Rn,
  msim = NULL,
  predictors,
  newpredictors,
  FUN = "invKGE",
  power = 0.5,
  symmetrize = mean,
  model = "glmnet",
  args_glmnet = list(s = "lambda.min", lower.limits = 0),
  standardisation = FALSE,
  linearisation = FALSE,
  first_model = NULL,
  parallel = FALSE,
  cores = NULL,
  verbose = TRUE,
  seed = NULL
)
```

Arguments

`Rn`	net rainfall matrix of gauged catchments (rows for time index, and columns for catchment index)
`msim`	similarity matrix of net rainfall time series (as produced by rsimilarity)
`predictors`	a list of at least two square matrices that could be used to predict similarity of net rainfall. The matrices should be square to describe the similarity of each pair of gauged catchments. High values should indicate a high probability of similarity in net rainfall
`newpredictors`	a matrix or data frame of predictors to be used as new input to the model being built and from which net rainfall similarity will be predicted. Typically, its rows describe the similarities between an ungauged catchment and several gauged catchments that may be weighted as donors. Each column should correspond to one element of `predictors`, and each row should correspond to one pair of catchments analysed.
`FUN`	either a function or a character string specifying the name of a predefined function to quantify the similarity of two net rainfall time series. See rsimilarity for more details
`power`	exponent applied on `Rn` time series when computing `Rn` similarity (generally between -1 and 1, a lower value emphasising the similarity of low flows). Default value is 0.5
`symmetrize`	a function to combine `FUN(Rn[,i],Rn[,j])` and `FUN(Rn[,j],Rn[,i])` into one similarity value (typically if `FUN=KGE`, `KGE(Rn[,i],Rn[,j])!=KGE(Rn[,j],Rn[,i])`). Default is mean
`model`	the method to estimate similarity of `Rn` from `descriptors`. This must be one of "lm", "glmnet"

`args_glmnet` list of arguments to be passed to `glmnet::cv.glmnet()`

`standardisation` logical indicating whether the predictors should be standardised (in particular to make their coefficient comparable)

`linearisation` logical indicating whether a power transformation should be applied to the predictors in order to maximise linearity with the target variable, based on the Pearson correlation coefficient

`first_model` a first model to use for predicting similarity (generally based on distance, see `two_steps` of mixr). The actual model will be built on residuals of the first one

`parallel` logical indicating whether the computation of Rn similarities should be parallelised

`cores` the number of cores to use for parallel execution if `parallel` is TRUE. If not specified, the number of cores is set to the value of `parallel::detectCores()`

`verbose` logical indicating if warning messages should be written to the console

`seed` integer value to be used by `set.seed()` for reproducible results. Used only if `model="glmnet"`

See Also

rsimilarity, mixr

Examples

```
data(Oudon)
obs <- as_transfr(st = Oudon$obs, hl = Oudon$hl)
obs <- velocity(obs, method = "loire2016")
obs <- uh(obs)
obs <- lagtime(obs)
obs <- rapriori(obs)
obs <- inversion(obs)
mdist1 <- hdist(x = obs, y = obs, method = "rghosh", parallel = TRUE, cores=2)
mdist2 <- mdist1^2
rghosh1 <- seq(1000, 5000, by=100)
rghosh2 <- rghosh1^2
res <- rsimilarity_model(Rn = obs$st$RnInv,
                   predictors = list(rghosh1=1/mdist1, rghosh2=1/mdist2),
                   newpredictors = data.frame(rghosh1=1/rghosh1, rghosh2=1/rghosh2),
                   seed=1234)
plot(rghosh1, res$similarity, ylab = "Predicted Rn similarity")
plot(rghosh2, res$similarity, ylab = "Predicted Rn similarity")
# rsimilarity_model() is automatically called by mixr() if mdist is a list
obs <- mixr(obs = obs, mdist = mdist1,
            similarity = list(rghosh1=1/mdist1, rghosh2=1/mdist2),
            parallel = TRUE, cores=2, cv = TRUE, save_donor = TRUE)
obs$similarity_models
obs <- convolution(obs)
plot(obs, i = 1, attribute = c("Qobs", "Qsim"))
```

uh *Unit hydrograph estimation*

Description

Estimate the unit hydrograph from a sample of hydraulic lengths and a streamflow velocity.

Usage

```
uh(hl, ...)

## Default S3 method:
uh(hl, uc, deltat, ...)

## S3 method for class 'units'
uh(hl, uc, deltat, ...)

## S3 method for class 'stars'
uh(hl, ...)

## S3 method for class 'transfR'
uh(hl, verbose = TRUE, ...)
```

Arguments

`hl`	hydraulic length of class `stars`, `matrix`, `vector` or `transfR`. If no unit is provided, `hl` is assumed to be in [m].
`...`	further arguments passed to or from other methods
`uc`	streamflow velocity. If no unit is provided, `uc` is assumed to be in [m/s].
`deltat`	time step of the time series. If no unit is provided, `deltat` is assumed to be in [min].
`verbose`	logical indicating if information messages should be written to the console

Details

The function estimates the unit hydrograph from geomorphometric information. A travel time to the outlet is estimated by assuming an average streamflow velocity (`uc`) within the river network and by applying `uc` to the sample of hydraulic lengths (`hl`). The unit hydrograph is the probability distribution of this travel time to the outlet given at each time step (`deltat`).

Value

A data frame with vectors of class units or, if `hl` is a transfR object, the `hl` object with the "uh" attribute added.

Examples

```
data(Oudon)
uh1 <- uh(hl=Oudon$hl[[1]], uc=units::set_units(0.5,"m/s"),
deltat=units::set_units(1,"h"))
plot(units::set_units(uh1$max_time,"h"), cumsum(uh1$prob), type = "b",
xlab = "Travel~time", ylab = "Probability~of~non-exceedance")

object <- as_transfr(st = Oudon$obs, hl = Oudon$hl)
object <- velocity(object)
object <- uh(object)
plot(object, i = 1, attribute = c("uh"))
```

---

velocity *Streamflow velocity estimation*

Description

Estimate the average streamflow velocity over the catchment.

Usage

```
velocity(hl, ...)

## Default S3 method:
velocity(hl, lagtime, method = "loire2016", ...)

## S3 method for class 'units'
velocity(hl, lagtime = NULL, method = "loire2016", ...)

## S3 method for class 'stars'
velocity(hl, ...)

## S3 method for class 'transfR'
velocity(hl, ...)
```

Arguments

`hl`	hydraulic length of class `stars`, `matrix`, `vector` or `transfR`. If no unit is provided, `hl` is assumed to be in [m].
`...`	further arguments passed to or from other methods
`lagtime`	lag time of the catchment. If no unit is provided, `lagtime` is assumed to be in [h].
`method`	character string describing the method to estimate the velocity. One of "loire2016" (default), "brittany2013" or "lagtime" (see details).

Details

Estimate the average streamflow velocity of the catchment from three different approaches. Method "lagtime" estimates the velocity from the ratio between the mean hydraulic length and the lag time of the catchment. Method "loire2016" estimates the velocity from a regression based on hydraulic length only:

$$a \cdot hl^b$$

where $a = 4.38e-4$ and $b = 0.69$ have been calibrated over the Loire river basin (de Lavenne et al. 2016). Method "brittany2013" used a similar regression calibrated for the French Brittany region where $a = 8.59e-4$ and $b = 0.61$ (de Lavenne 2013).

Value

A numeric value of class units or, if `hl` is a transfR object, the `hl` object with the "uc" attribute added.

References

de Lavenne A (2013). *Modélisation hydrologique à base géomorphologique de bassins versants non jaugés par régionalisation et transposition d'hydrogramme*. Ph.D. thesis, Agrocampus-Ouest Rennes. https://hal.inrae.fr/tel-02810356.

de Lavenne A, Skøien JO, Cudennec C, Curie F, Moatar F (2016). "Transferring measured discharge time series: Large-scale comparison of Top-kriging to geomorphology-based inverse modeling." *Water Resources Research*, **52**(7), 5555–5576. doi:10.1002/2016WR018716.

Examples

```
data(Oudon)
velocity(Oudon$hl[[1]], method = "loire2016")

object <- as_transfr(st = Oudon$obs, hl = Oudon$hl)
object <- velocity(object)
object$uc
```

---

weightr *Weights of donor catchments*

Description

Estimate the weighting applied at each time step and to each gauged catchment (donors) for the calculation of the average net rainfall of an ungauged catchment

Usage

```
weightr(
  Rn,
  distances,
  similarities,
  ndonors = 5,
  donors = NULL,
  power = 1,
  flexible_donor = TRUE
)
```

Arguments

`Rn`	net rainfall matrix of donor catchments (rows for time indices and columns for donor indices)
`distances`	vector of the distances to each donor catchment, used for donor selection (see hdist)
`similarities`	vector of the hydrological similarities to each donor catchment, used for donor weighting (`1/distances^power` is used if not provided)
`ndonors`	maximum number of donor catchments to use
`donors`	vector of catchment IDs from which donors are selected. If empty, the `ndonors` closest catchments are used
`power`	exponent applied in the inverse distance weighting function (see details)
`flexible_donor`	logical indicating if the donor catchments can change during the simulation period according to the availability of discharge observations (see details)

Details

This function returns a matrix of weights for each time step (rows) and each gauged catchment (columns) for the calculation of the average net rainfall of an ungauged catchment (see mixr). The weights λ are estimated by an inverse distance weighting function (de Lavenne et al. 2016):

$$\lambda_i = \frac{1}{d_i^p}$$

$$\Sigma_{i=1}^{ndonors} \lambda_i = 1$$

where d is the `distances` argument and p is the `power` argument. The weights are rescaled so the sum is equal to 1.

Two strategies to handle missing data in the `Rn` matrix are possible. If `flexible_donor` is TRUE, donor catchments are redefined at each time step and are chosen from those that are effectively gauged at that time step. This aims to keep a constant number of donor catchments throughout the simulation period. If `flexible_donor` is FALSE, the donor catchments are chosen once within all the gauged catchments, regardless of missing data, and remain the same throughout the entire simulation period. This stability of donor catchments might, however, lead to a reduced number of donors (below `ndonors`) during periods of missing data.

Value

A matrix with the same dimensions as `Rn`.

References

de Lavenne A, Skøien JO, Cudennec C, Curie F, Moatar F (2016). "Transferring measured discharge time series: Large-scale comparison of Top-kriging to geomorphology-based inverse modeling." *Water Resources Research*, **52**(7), 5555–5576. doi:10.1002/2016WR018716.

See Also

hdist, mixr

Index