

Package ‘hydropeak’

June 24, 2026

Type Package

Title Detect and Characterize Sub-Daily Flow Fluctuations

Version 0.1.3

Description An important environmental impact on running water ecosystems is caused by hydropeaking - the discontinuous release of turbine water because of peaks of energy demand. An event-based algorithm is implemented to detect flow fluctuations referring to increase events (IC) and decrease events (DC). For each event, a set of parameters related to the fluctuation intensity is calculated. The framework is introduced in Greimel et al. (2016) `"A method to detect and characterize sub-daily flow fluctuations"` [doi:10.1002/hyp.10773](https://doi.org/10.1002/hyp.10773) and can be used to identify different fluctuation types according to the potential source: e.g., sub-daily flow fluctuations caused by hydropeaking, rainfall, or snow and glacier melt. This is a companion to the package 'hydroroute', which is used to detect and follow hydropower plant-specific hydropeaking waves at the sub-catchment scale and to describe how hydropeaking flow parameters change along the longitudinal flow path as proposed and validated in Greimel et al. (2022) [doi:10.1002/trr.3978](https://doi.org/10.1002/trr.3978).

License GPL-2

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Suggests rmarkdown, knitr, testthat (>= 3.0.0), data.table

Config/testthat/edition 3

Imports utils, parallel

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Bettina Grün [cre, ctb] (ORCID:

<https://orcid.org/0000-0001-7265-4773>),

Julia Haider [aut],

Benjamin Altmann [ctb] (ORCID: <https://orcid.org/0009-0000-2451-6018>),

Franz Greimel [ctb] (ORCID: <https://orcid.org/0000-0002-8000-1227>)

Maintainer Bettina Grün <Bettina.Gruen@R-project.org>

Repository CRAN

Date/Publication 2026-06-24 06:00:02 UTC

Contents

all_metrics	2
amp	3
dur	4
durmid	5
Events	5
event_type	6
flow	7
get_events	8
get_events_dir	10
get_events_file	11
mafr	12
maxfr	13
mefr	13
mefr1	14
mefr2	15
minfr	16
Q	16
qfrom	17
qmid	18
qto	18
ratio	19
Index	20

all_metrics	<i>Event Based Fluctuation Flow Parameters</i>
-------------	--

Description

Given an event with equal flow trend, all metrics implemented in the package **hydropeak** are computed and combined to a data frame. It includes also an identifier of the gauging station provided by the user and the starting time of the event.

Usage

```
all_metrics(x, id, event_type)
```

Arguments

x	Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format (see flow()).
id	Character string which refers to the identifier of the gauging station (in Austria: HZBCODE).
event_type	Numeric value which indicates the event type. By using get_events() , this is internally computed. If metrics for a single event have to be computed individually, the event type has to be provided as an argument or can be computed with event_type() .

Value

A data frame including all computed metrics, IDs of gauging stations, event type (see [event_type\(\)](#) for an overview of possible event types), and starting time of an event x. Included metrics are [amp\(\)](#), [mafr\(\)](#), [mefr\(\)](#), [dur\(\)](#), [ratio\(\)](#).

Examples

```
data(Q)
# decreasing event:
Q4 <- flow(Q[3:4, ])
all_metrics(Q4, id = Q$ID[1], event_type = 4)
all_metrics(Q4, id = Q$ID[1], event_type = event_type(Q4))

# increasing event:
Q2 <- flow(Q[486:487, ])
all_metrics(Q2, id = Q$ID[1], event_type = 2)
all_metrics(Q2, id = Q$ID[1], event_type = event_type(Q2))

# constant event (at beginning or after NA event):
Q0 <- flow(Q[1:3, ])
all_metrics(Q0, id = Q$ID[1], event_type = 0)
all_metrics(Q0, id = Q$ID[1], event_type = event_type(Q0))
```

amp

AMP - Amplitude (Metric 1)

Description

The amplitude (AMP, unit: m^3/s) of an event is defined as the difference between the flow maximum (Q_{max}) and the flow minimum (Q_{min}). Given an event with equal flow trend, the amplitude is computed and returned.

Usage

```
amp(x)
```

Arguments

x Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format (see [flow\(\)](#)).

Value

Returns a positive numeric value which is the difference of $\max(x\$Q)$ and $\min(x\$Q)$ of an event. If a data frame containing NA flow rates (Q) is given, NA is returned.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
amp(Q)
```

dur	<i>DUR - Duration (Metric 4)</i>
-----	----------------------------------

Description

The duration of an event is specified as the number of consecutive time steps with equal flow trend.

Usage

```
dur(x)
```

Arguments

x Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format. (see [flow\(\)](#)).

Value

Returns an integer value which is the number of consecutive time steps.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
dur(Q)
```

durmid	<i>DUR_mid - Duration until midpoint (Metric 4.5)</i>
--------	---

Description

The duration until the midpoint of an event is specified as the number of consecutive time steps with equal flow trend until the average value of that event is reached.

Usage

```
durmid(x)
```

Arguments

x	Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format. (see flow()).
---	---

Value

Returns an integer value which is the number of consecutive time steps.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
durmid(Q)
```

Events	<i>Events</i>
--------	---------------

Description

A complementary dataset to the flow fluctuation dataset [Q](#). It contains the events and metrics such as computed by [get_events\(\)](#).

Usage

```
Events
```

Format

A data frame with 165 rows and 8 variables:

ID Character string which refers to the identifier of the gauging station (in Austria: HZBCODE)

EVENT_TYPE Event types are defined as follows:

- 0: Constant event after NA event or constant event as first event in time series
- 1: Constant event after DC
- 2: Increasing event (IC)
- 3: Constant event after IC
- 4: Decreasing event (DC)
- 5: NA event

Time Date-time of event starting point

AMP Amplitude ([amp\(\)](#))

MAFR Maximum flow fluctuation rate ([mafr\(\)](#))

MEFR Mean flow fluctuation rate ([mefr\(\)](#))

DUR Duration ([dur\(\)](#))

RATIO Flow ratio ([ratio\(\)](#))

event_type

Event Type

Description

Given a data frame (time series) of equal flow (Q) trend, it is determined whether the flow is increasing or decreasing, constant or if missing values occur. It returns a numeric value which indicates the event type. As the event type is already determined when the change points are computed, this function is mainly used for demonstration purpose or if metrics should be computed individually.

Usage

```
event_type(x)
```

Arguments

x Data frame (time series) from an event with equal flow trend. Data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format. (see [flow\(\)](#)).

Value

Returns an numeric value which indicates the event type. Event types are defined as follows:

- 0: Constant event after NA event or constant event as first event in time series
- 1: Constant event after DC
- 2: Increasing event (IC)
- 3: Constant event after IC
- 4: Decreasing event (DC)
- 5: NA event

Examples

```
data(Q)
# decreasing event
Q4 <- flow(Q[3:4, ])
event_type(Q4)

# increasing event
Q2 <- flow(Q[486:487, ])
event_type(Q2)
```

flow

'flow' S3 Class for Flow Rate Time Series

Description

The function `flow()` creates a flow rate time series object which is formatted to be compatible with the functions in the **hydropeak** package.

Usage

```
flow(
  x,
  format = "%d.%m.%Y %H:%M",
  tz = "Etc/GMT-1",
  cols = c(1, 2, 3),
  steplength = 15,
  full = TRUE
)

validate_flow(x)
```

Arguments

x	Data frame which contains at least a column with an ID of the gauging station, a column with date-time values in character representation and a column with flow rates
format	Character string giving the date-time format of the date-time column in the input data frame (default: dd.mm.YYYY HH:MM)
tz	Character string specifying the time zone to be used for the conversion (default: Etc/GMT-1).
cols	Integer vector specifying column indices in the input data frame which contain gauging station ID, date-time and flow rate to be renamed. The default indices are 1 (ID), 2 (date-time) and 3 (flow rate, Q).
steplength	Numeric value which specifies the distance between (equispaced) time steps in minutes. (default: 15, which refers to 15 minutes). Non-equispaced time steps are not supported and missing time steps are imputed if argument full is set to TRUE (default), Q values are assumed to be NA.
full	A logical. If TRUE (default) imputes missing time step values so the time series is complete. Imputed Q values are set to NA. It should only be set to FALSE, if it is known, that the time series is complete.

Value

Returns a flow object which inherits from data frame (time series). It contains at least a gauging station ID column (ID) converted to character values, a date-time column (Time) converted to class "POSIXct" (after conversion using `base::strptime()`) and a flow rate column (Q), which is converted to numeric values. The `flow()` object ensures that input flow fluctuation time series data can be processed with the functions in the **hydropeak** package. Therefore, it is mandatory to provide the correct indices (see argument `cols`) and the correct date-time format (see argument `format`) of the input data frame.

Examples

```
data(Q)
Q <- flow(Q)
```

get_events

Flow Fluctuation Events and Metrics

Description

Given a data frame (time series) of stage measurements, all increase (IC) and decrease (DC) events are determined and all metrics implemented in the package **hydropeak** (see `all_metrics()`) are computed and combined to a data frame. Optionally, NA events and constant events can be included. NA events occur due to missing stage measurement values. The beginning of NA events refers to the last measurement with a non-missing Q value. Constant events are events where the Q values stay constant over time. An event is uniquely identifiable through the combination of the event starting time (Time) and the gauging station identifier (ID).

Usage

```
get_events(x, mc.cores = 2L, omit.constant = TRUE, omit.na = TRUE, ...)
```

Arguments

<code>x</code>	Data frame (time series) of stage measurements which contains at least a column with ID of the gauging station (default: column index 1) column with date-time values (default: columns index 2) in character representation and a column with flow rates (default: column index 3). If the column indices differ from <code>c(1, 2, 3)</code> , they have to be specified as <code>...</code> argument in the format <code>c(i, j, k)</code> .
<code>mc.cores</code>	Number of cores to use with <code>parallel::mclapply()</code> . On Windows, this will be set to 1.
<code>omit.constant</code>	A logical. If <code>FALSE</code> (default) it does not return events with constant measurements. Otherwise these events are included.
<code>omit.na</code>	A logical. If <code>FALSE</code> (default) it does not return missing value events. Otherwise these events are included.
<code>...</code>	Arguments to be passed to <code>flow()</code> to specify the date-time format in the input data (default: <code>dd.mm.YYYY HH:MM</code>), the time zone used for the conversion (default: <code>Etc/GMT-1</code>) and the column indices in the input data, which contain date-time values and flow rate values. The default indices are 1 (ID), 2 (date-time) and 3 (flow rate, Q), i.e. <code>cols = c(1, 2, 3)</code> .

Value

A data frame which contains for every event in a given time series all metrics (`all_metrics()`), gauging station ID, event type, and starting time of an event. Included metrics are `amp()`, `mafr()`, `mefr()`, `dur()`, `ratio()`. These metrics are only computed for increasing (IC) and decreasing (DC) events. For all other events the values are set to 0 except for flow ratio that is set to 1. Event types are defined as follows:

- 0: Constant event after NA event or constant event as first event in time series
- 1: Constant event after DC
- 2: Increasing event (IC)
- 3: Constant event after IC
- 4: Decreasing event (DC)
- 5: NA event

Examples

```
# Data with multiple events and different stations
data(Q)
get_events(Q)

# including constant events
get_events(Q, omit.constant = FALSE)
```

Description

Given a directory path it calls `get_events_file()` for each file in the directory, recursively. The resulting events are split into separate files for each gauging station ID (ID) and Event_Type and are written to the given output directory.

Usage

```
get_events_dir(  
  Q_dir,  
  inputsep = ";",  
  inputdec = ".",  
  outdir = file.path(tempdir(), "Events"),  
  mc.cores = 2L,  
  ...  
)
```

Arguments

<code>Q_dir</code>	A character string containing the path name where the input data are located.
<code>inputsep</code>	Field separator character string for input data.
<code>inputdec</code>	Character string for decimal points in input data.
<code>outdir</code>	A character string naming a directory where the output file(s) should be written to.
<code>mc.cores</code>	Number of cores to use with <code>parallel::mclapply()</code> . On Windows, this will be set to 1.
<code>...</code>	Arguments to be passed to <code>get_events_file()</code> and further to <code>get_events()</code> and <code>flow()</code> .

Value

No return value, called for side effects.

Examples

```
Q_dir <- "../inst/extdata"  
get_events_dir(Q_dir, inputsep = ",", inputdec = ".")
```

Description

Given a file path it reads a data frame (time series) of stage measurements and calls `get_events()`. The resulting events can be optionally written to a single file or to separate files for each gauging station ID (ID) and Event_Type. Files which produce errors return NULL.

Usage

```
get_events_file(
  Q_file,
  inputsep = ";",
  inputdec = ".",
  save = FALSE,
  split = TRUE,
  outdir = file.path(tempdir(), "Events"),
  mc.cores = 2L,
  return = TRUE,
  ...
)
```

Arguments

<code>Q_file</code>	A character string containing the name of the file which the data are to be read from with <code>utils::read.csv()</code> .
<code>inputsep</code>	Field separator character string for input data.
<code>inputdec</code>	Character string for decimal points in input data.
<code>save</code>	A logical. If FALSE (default) events (results from <code>get_events()</code>) are not written to file(s), otherwise events are written to <code>outdir</code> .
<code>split</code>	A logical. If TRUE (default) output files are separated by their gauging station ID (ID) and by Event_Type, otherwise all events are written to a single file.
<code>outdir</code>	A character string naming a directory where the output file(s) should be written to.
<code>mc.cores</code>	Number of cores to use with <code>parallel::mclapply()</code> . On Windows, this will be set to 1.
<code>return</code>	A logical. If TRUE (default) it returns the resulting data frame or list of data frames. Otherwise it returns NULL.
<code>...</code>	Arguments to be passed to <code>get_events()</code> and further to <code>flow()</code> .

Value

A data frame which contains for every increase or decrease event in a given time series all metrics (`all_metrics()`), gauging station ID, event type, and starting time of an event. Included metrics are `amp()`, `mafr()`, `mefr()`, `dur()`, `ratio()`. The returned data frame is not split. Returns NULL, if argument `return` is set to FALSE.

Examples

```
Q_file <- system.file("extdata", "Q.csv", package = "hydropeak")
# save to tempdir()
events <- get_events_file(Q_file, inputsep = ",", inputdec = ".",
save = TRUE, split = TRUE, return = TRUE)
```

mafr

MAFR - Maximum Flow Fluctuation Rate (Metric 2)

Description

The maximum flow fluctuation rate (MAFR, unit: m^3/s) represents the highest absolute flow change of two consecutive time steps within an event. Given an event with equal flow trend, the maximum flow fluctuation rate is computed and returned.

Usage

```
mafr(x)
```

Arguments

`x` Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (`Time`) and a flow rate column (`Q`) and must be in a compatible format (see `flow()`).

Value

Returns a numeric value which is the maximum (absolute) flow fluctuation rate. If a data frame containing NA flow rates (`Q`) is given, NA is returned.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
mafr(Q)
```

maxfr	<i>MAX - Maximum Flow Rate (Metric 6)</i>
-------	---

Description

The maximum flow rate (MAX, unit: m^3/s) represents the highest absolute flow rate of one time step within an event. Given an event with equal flow trend, the maximum flow rate is computed and returned.

Usage

```
maxfr(x)
```

Arguments

x	Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format (see flow()).
---	--

Value

Returns a numeric value which is the maximum flow rate. If a data frame containing NA flow rates (Q) is given, NA is returned.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
maxfr(Q)
```

mefr	<i>MEFR - Mean Flow Fluctuation Rate (Metric 3)</i>
------	---

Description

The mean flow fluctuation rate (MEFR, unit: m^3/s^2) is calculated by the event amplitude divided by the number of time steps (duration) within an event. Given an event with equal flow trend, amplitude and duration are computed. From these metrics the mean flow fluctuation rate is calculated and returned.

Usage

```
mefr(x)
```

Arguments

x Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format. (see [flow\(\)](#)).

Value

Returns a numeric value which is the mean flow fluctuation rate computed by the event amplitude [amp\(\)](#) divided by the number of time steps [dur\(\)](#). If a data frame containing NA flow rates (Q) is given, NA is returned.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
mefr(Q)
```

mefr1	<i>MEFR_half1 - Mean Flow Fluctuation Rate of the first half (Metric 11)</i>
-------	--

Description

The mean flow fluctuation rate (MEFR, unit: m^3/s^2) is calculated by the event amplitude divided by the number of time steps (duration) within an event. Given an event with equal flow trend, amplitude and duration are computed. From these metrics the mean flow fluctuation rate is calculated and returned. Here the MEFR for the first half is calculated. The first half of an event is defined as all time steps with a flow rate below the average flow rate of an event.

Usage

```
mefr1(x)
```

Arguments

x Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format. (see [flow\(\)](#)).

Value

Returns a numeric value which is the mean flow fluctuation rate computed by the event amplitude [amp\(\)](#) divided by the number of time steps [dur\(\)](#) in the first half. If a data frame containing NA flow rates (Q) is given, NA is returned.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
mefr1(Q)
```

mefr2	<i>MEFR_half2 - Mean Flow Fluctuation Rate of the second half (Metric 12)</i>
-------	---

Description

The mean flow fluctuation rate (MEFR, unit: m^3/s^2) is calculated by the event amplitude divided by the number of time steps (duration) within an event. Given an event with equal flow trend, amplitude and duration are computed. From these metrics the mean flow fluctuation rate is calculated and returned. Here the MEFR for the second half is calculated. The second half of an event is defined as all time steps with a flow rate above the average flow rate of an event.

Usage

```
mefr2(x)
```

Arguments

x Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format. (see [flow\(\)](#)).

Value

Returns a numeric value which is the mean flow fluctuation rate computed by the event amplitude [amp\(\)](#) divided by the number of time steps [dur\(\)](#) in the second half. If a data frame containing NA flow rates (Q) is given, NA is returned.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
mefr2(Q)
```

 minfr

MIN - Minimum Flow Rate (Metric 7)

Description

The minimum flow rate (MIN, unit: m^3/s) represents the lowest absolute flow rate of one time step within an event. Given an event with equal flow trend, the minimum flow rate is computed and returned.

Usage

```
minfr(x)
```

Arguments

x Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format (see [flow\(\)](#)).

Value

Returns a numeric value which is the minimum flow rate. If a data frame containing NA flow rates (Q) is given, NA is returned.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
minfr(Q)
```

 Q

Flow Fluctuations Q

Description

A dataset containing sub-daily flow fluctuations of five consecutive days from two different gauging stations. One time step is 15 minutes.

Usage

```
Q
```

Format

A data frame with 960 rows and 3 variables:

ID Character string which refers to the identifier of the gauging station (in Austria: HZBCODE)

Time Character string with date-time information of stage measurements which needs to be converted to a compatible format (see [flow\(\)](#))

Q Flow, stage measurements in m^3/s

qfrom	<i>Q_from - Starting flow rate (Metric 8)</i>
-------	---

Description

The starting flow rate represents the flow rate at the beginning of an event.

Usage

```
qfrom(x)
```

Arguments

x Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format. (see [flow\(\)](#)).

Value

Returns a numeric value which is the flow rate at the start of the event.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
qfrom(Q)
```

qmid	<i>Q_mid - Average flow rate (Metric 10)</i>
------	--

Description

The average flow rate represents the mean flow rate of an event.

Usage

```
qmid(x)
```

Arguments

x Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format. (see [flow\(\)](#)).

Value

Returns a numeric value which is the average flow rate of the event.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
qmid(Q)
```

qto	<i>Q_to - Final flow rate (Metric 9)</i>
-----	--

Description

The final flow rate represents the flow rate at the end of an event.

Usage

```
qto(x)
```

Arguments

x Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format. (see [flow\(\)](#)).

Value

Returns a numeric value which is the flow rate at the end of the event.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
qto(Q)
```

ratio	<i>RATIO - Flow Ratio (Metric 5)</i>
-------	--------------------------------------

Description

The metric flow ratio (RATIO) is defined as the flow maximum divided by the flow minimum, $\frac{Q_{max}}{Q_{min}}$. Given an event with equal flow trend, the flow ratio is computed and returned.

Usage

```
ratio(x, event_type)
```

Arguments

x	Data frame (time series) from an event with equal flow trend. The data frame must contain a date-time column (Time) and a flow rate column (Q) and must be in a compatible format. (see flow()).
event_type	Numeric value which specifies the event type. See get_events() for an overview of the event types.

Value

Returns a numeric value which is the flow ratio computed by $\max(x\$Q)$ divided by $\min(x\$Q)$. If a data frame containing NA flow rates (Q) is given, NA is returned.

Examples

```
data(Q)
Q <- flow(Q[3:4, ])
ratio(Q, event_type(Q))
```

Index

* datasets

Events, 5
Q, 16

all_metrics, 2
all_metrics(), 8, 9, 12
amp, 3
amp(), 3, 6, 9, 12, 14, 15

base::strptime(), 8

dur, 4
dur(), 3, 6, 9, 12, 14, 15
durmid, 5

event_type, 6
event_type(), 3
Events, 5

flow, 7
flow(), 3–6, 9–19

get_events, 8
get_events(), 3, 5, 10, 11, 19
get_events_dir, 10
get_events_file, 11
get_events_file(), 10

mafr, 12
mafr(), 3, 6, 9, 12
maxfr, 13
mefr, 13
mefr(), 3, 6, 9, 12
mefr1, 14
mefr2, 15
minfr, 16

parallel::mclapply(), 9–11

Q, 5, 16
qfrom, 17

qmid, 18
qto, 18

ratio, 19
ratio(), 3, 6, 9, 12

utils::read.csv(), 11

validate_flow(flow), 7