“HEDA” Package

Type Package

Title **H**ydropeaking **E**vents **D**etection **A**lgorithm

Version 1.0

Author Tingyu Li, Xiaotian Zou, Gregory B. Pasternack

Maintainer Tingyu Li (styli@ucdavis.edu)

Description HEDA is designed to identify hydropeaking events from raw time-series flow record, a rapid flow variantion induced by the hourly-adjusted electricity market. The novelty of HEDA is to use vector angle instead of the first-order derivative to detect change points which not only largely improves the computing efficiency but also accounts for the rate of change of the flow variation. More details: https://doi.org/10.1016/j.jhydrol.2021.126392 (https://doi.org/10.1016/j.jhydrol.2021.126392)

License

Encoding UTF-8

LazyData true

RoxygenNote

NeedsCompliation no

Repository CRAN

Date/Publication 2021-05-21

User Guide

Prior to using this library, four packages needs to install: **dplyr**, **lubridate**, **zoo**, **ggplot2**.

```
library(dplyr)
library(ggplot2)
library(zoo)
library(lubridate)
library(HEDA)
```

The input files should be .csv files, and are required to only contain 3 columns, which are site id, datetime and parameter value respectively. Site id is the identification number of gauge stations. Datetime is the date and time of the record. Parameter value refers to the discharge of flow record.

```
# import the flow data
HPK_Q <- read.table("D:/Ninalty/UCD_Hydropeaking/HPK_FlowData/HPK_WC/CA/11278400.csv", sep = ","
,header = TRUE, row.names = NULL)
head(HPK_Q)
```

```
##    site_no       dateTime X_00060_00000
## 1 11278400 10/1/1988 7:02           791
## 2 11278400 10/1/1988 7:17           667
## 3 11278400 10/1/1988 7:32           532
## 4 11278400 10/1/1988 7:47           507
## 5 11278400 10/1/1988 8:02           487
## 6 11278400 10/1/1988 8:17           461
```

Functions

```
HEDA_Tidy        Preprocess the data
```

Description

Format flow record into hourly record; Split record by designated season; Interpolate and smoothing the record.

Usage

HEDA_Tidy(dataframe, season)

Arguments

dataframe Name of the dataframe to be processed.
season Subset data by season.

Outputs

Output dataframe has 4 columns containing location id, datatime, parameter value, ann_thre. ann_thre is originally the mean annual discharge. If flow record is subset by season, ann_thre will be the mean discharge of that period.

Example

```
# format the time column to time format
HPK_Q$dateTime <- parse_date_time(HPK_Q$dateTime, "mdy HM")

# clean subset data by season
hpk_flow_cln = HEDA_Tidy(HPK_Q, season = c(6,7,8,9))
head(hpk_flow_cln)
```

```
## # A tibble: 6 x 4
##   location_id datetime         parameter_value ann_thre
##         <int> <chr>                      <dbl>    <dbl>
## 1    11278400 1989-6-1 0:00:00           1568.     723.
## 2    11278400 1989-6-1 1:00:00           1778.     723.
## 3    11278400 1989-6-1 2:00:00           1768.     723.
## 4    11278400 1989-6-1 3:00:00           1770      723.
## 5    11278400 1989-6-1 4:00:00           1762.     723.
## 6    11278400 1989-6-1 5:00:00           1768.     723.
```

```
ReversalCount        Detect hydropeaking events
```

Description

Detect change points of hydropeaking events and classified change points into four categories.

Usage

ReversalCount(dataframe, alpha1, theta)

Arguments

dataframe Name of the dataframe to be processed.
alpha1 Default value : 0.03
theta Default value : 60 degree for m^3/s, 85 degree for cfs. gamma default value: 1.1 m^3/s or 40 cfs.

Outputs

Output dataframe has six columns containing location id, datatime, parameter value, ann_thre, vt_degree and dgtag. ann_thre is originally the mean annual discharge. If flow record is subset by season, ann_thre will be the mean discharge of that period. vt_degree is the vector angle between two flow vectors. dgtag is the catogrized change points.

Example

```
hpk_flow_cg <- ReversalCount(hpk_flow_cln, alpha1 = 0.03, theta = 85, gamma = 40)
head(hpk_flow_cg)
```

```
##   location_id            datetime parameter_value ann_thre vt_degree dgtag
## 1    11278400 1989-06-01 00:00:00          1567.5 722.5849        NA    NA
## 2    11278400 1989-06-01 01:00:00          1752.5 722.5849   89.6903     2
## 3    11278400 1989-06-01 02:00:00          1752.5 722.5849    0.0000     0
## 4    11278400 1989-06-01 03:00:00          1752.5 722.5849    0.0000     0
## 5    11278400 1989-06-01 04:00:00          1752.5 722.5849    0.0000     0
## 6    11278400 1989-06-01 05:00:00          1752.5 722.5849    0.0000     0
```

```
Clean_position        Exclude change points in wrong position
```

Description

Change points are excluded if they are in the wrong position. For example, both point 3 and the peak pair represent the peaking discharge whose value (position) should be close to the daily maximum discharge. If the peaking discharge is close to the daily minimum discharge, change points will be removed since they are in the wrong positions.

Usage

Clean_position(dataframe, alpha2)

Arguments

dataframe Name of the dataframe to be processed.
alpha2 Default value : 0.3.

Outputs

The output is dataframe.

Example

```
hpk_flow_cg <- Clean_position(hpk_flow_cg, alpha2 = 0.3)
```

```
Clean_Spt        Clean repeated points
```

Description

Clean continuous points of the same type

Usage

clean_Spt(dataframe, alpha3, alpha4)

Arguments

dataframe	Name of the dataframe to be processed.
alpha3	The default value: 0.7
alpha4	The default value: 0.5

Outputs

The output file is dataframe.

Example

```
hpk_flow_cg <- Clean_Spt(hpk_flow_cg, alpha3 = 0.7, alpha4 = 0.5)
```

```
Clean_conectD        Evaluate difference between peaking and off-peaking discharge
```

Description

Evaluate whether the difference in discharge between peaking and off-peaking points is qualified to be identified as hydropeaking events.

Usage

clean_conectD(dataframe, alpha3, alpha4)

Arguments

dataframe	Name of the input dataframe to be processed.
alpha3	The default value: 0.7.
alpha4	The default value: 0.5.

Outputs

The output will be a dataframe in the same form with the input.

Example

```
hpk_flow_cg <- Clean_conectD(hpk_flow_cg, alpha3 = 0.7, alpha4 = 0.5)
```

```
HPK_plot        Plot hydrograph
```

Description

Plot the hydrograph of the processed data with change points marked by different colors.

Usage

HPK_plot(dataframe)

Arguments

dataframe   Name of the input dataframe to be processed.

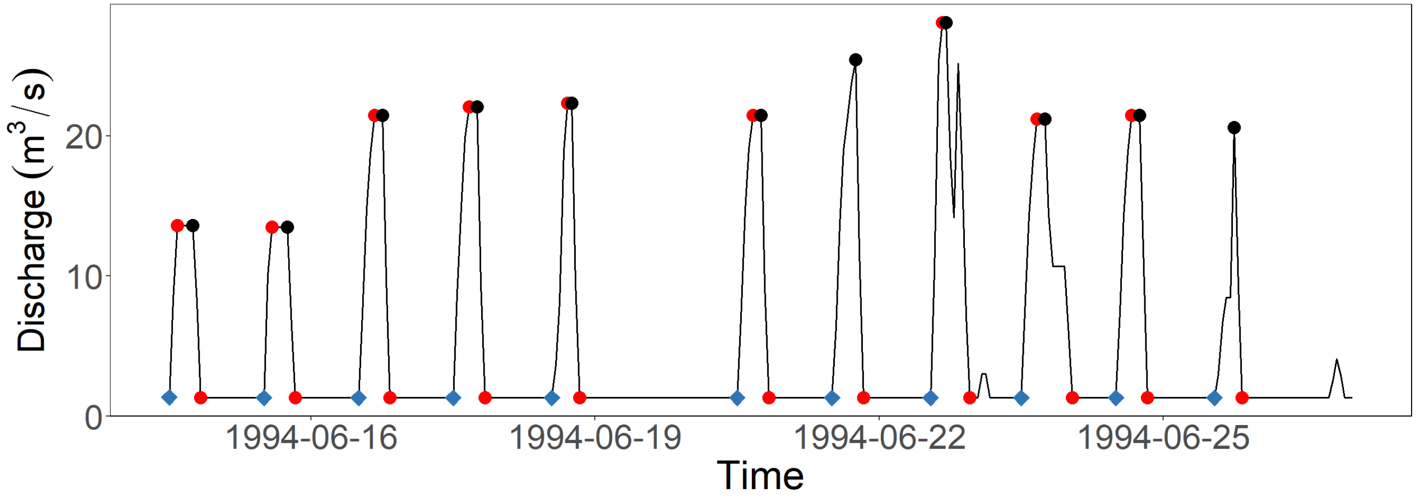
Outputs

The plot will be presented under the “Plots” tab in RStudio. Users can determine whether to save out the diagram by themselves. The plot shows flow in m^3/s.

Example

```
## subset the data
tt = hpk_flow_cg[13100:13400,]

# plot the hydrograph of the subset data
HPK_plot(tt)
```



***

```
HpkFrqMgt        Extract frequency and magnitude metrics
```

Description

Extract Qpeak, offQpeak, pk_no and pkraio from the identified hydropeaking events.

Usage

HPK_frq_mgt(dataframe)

Arguments

dataframe Name of the dataframe to be processed.

Outputs

Output dataframe includes time-series values of four metrics. Qpeak is the hydropeaking discharge, offQpeak is base flow, pk_no is the daily number of rise and fall process, pkratio is the ratio of days with hydropeaking to the total number of days analyzed

Example

```
HpkFrqMgt <- HPK_frq_mgt(hpk_flow_cg)
head(HpkFrqMgt)
```

```
HpkRtDur        Extract rate of change and duration related metrics
```

Description

Extract rate of chane and duration related metrics: pk_rtn is the retention of peaking process. offpk_rtn is the retention of base flow. D_rampup/D_ramp is the duration of rise/fall process. RB_Indx_up/RB_Index_dw is the flashness of rise/fall process. Ramp_up/Ramp_dw is the rate of change of rise/fall process. Strange_up/Strange_dw is the standardized rise/fall amplitude.

Usage

HPK_frq_mgt(dataframe)

Arguments

dataframe Name of the dataframe to be processed.

Outputs

Output dataframe includes time-series values of four metrics. Qpeak is the hydropeaking discharge, offQpeak is base flow, pk_no is the daily number of rise and fall process, pkratio is the ratio of days with hydropeaking to the total number of days analyzed.

Example

```
HpkRtDur <- HPK_rt_dur(hpk_flow_cg)

# to extract the time-series of one metric withouth missing value
pk_rtn <- HpkRtDur[na.omit(HpkRtDur$pk_rtn),c("location_id", "datetime","pk_rtn")]
```