Breaking Bad: robust Breakout detection based on E-Divisive with Medians (EDM) for modeling data quality control

Hao Zhang

Feburary 23, 2018

ABSTRACT

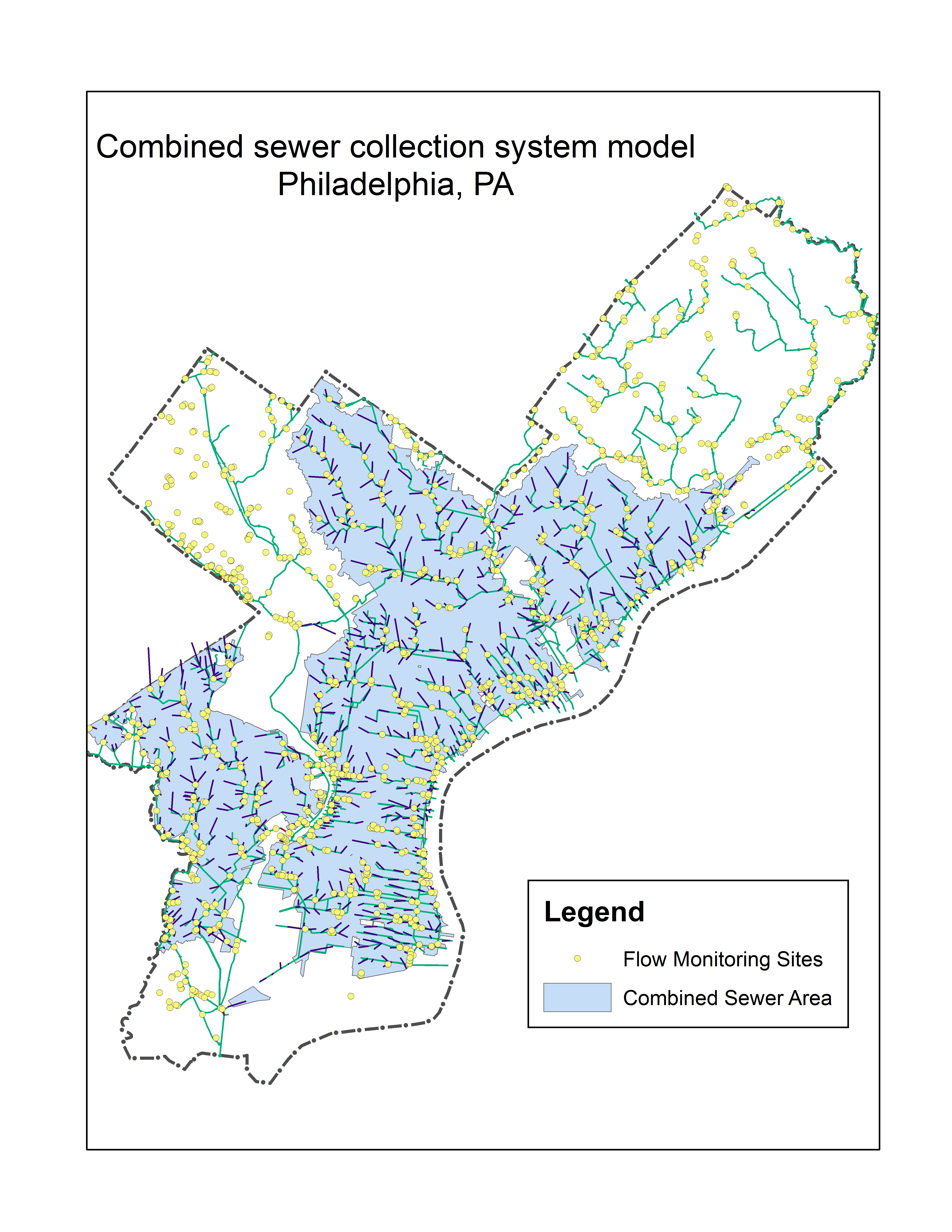
The Philadelphia water department (PWD) has been actively monitoring flow data at over 400 sites over Philadelphia since the 2000s. Data is collected twice a month through contractors. Due to the high solid content in sewage, flow data at sewer pipes (level, velocity) suffered from breakouts (mean shift, ramp up) over the time due to sensor ragging, pipe clogging, etc. A stringent Quality Control (QC) protocol is conducted before the data can be used for Hydrologic & Hydraulic modeling tasks. As one QC measure, the water level and velocity are examined to detect any potential breakout.

Since flow data fluctuates with rainfall-runoff events, the breakout detection algorithm must be robust to avoid the interference of runoff responses. Several breakout detection techniques were compared, and the E-Divisive with Medians (EDM) algorithm is adopted in this study. EDM recursively partitions a time series and uses a permutation test to determine change points. The EDM has following advantages: 1. EDM uses moving median as opposed to the mean, which is robust to the presence of anomalies; 2. EDM can detect both ‘mean shift’ (sudden change) and ‘ramping’ (gradual change) for multiple change points; 3. EDM takes a non-parametric approach, meaning the model will adapt to the data’s underlying distribution, and therefore can detect distribution change; 4. EDM is fast, as it uses interval trees to efficiently approximate the median.

The analysis is implemented in a program written in R, and the EDM algorithm is implemented via the ‘BreakoutDetection’ package developed by Twitter engineers. Non-trivial parameters of the EDM model are carefully tuned to best match the expected outcome. This analysis provides an additional assurance to the data quality. Also, field crews (monitoring, Operation & Maintenance, etc.) can quickly respond to the issue once a breakout has been detected. This analysis is also applicable for other monitored data, such as the trunk and outfall levels at drainage system regulators.

### 1 BACKGROUND

The Philadelphia Water Department (PWD) maintains hydrologic and hydraulic models of the combined sewer collection system for planning, management and compliance purposes. PWD relies on these models to evaluate the effectiveness of existing and proposed CSO control measures. Efforts are being made to refine the models and improve their accuracy as the program progresses from planning to implementation phases. Since the 2000s, PWD has been monitoring the sewerage level and velocity at over 400 manholes across the city for various model calibration/validation tasks. Data are measured at 15-minute interval, which are collected bi-weekly by contractor.



Due to the high solid content in sewage, level and velocity measurements at sewer pipes may suffered from breakouts (mean shift, ramp up) over the time due to sensor ragging, clogging, or pipe surcharging, etc. A stringent Quality Control (QC) protocol is conducted before the data can be used for Hydrologic & Hydraulic modeling tasks. As one QC measure, the water level and velocity time-series are examined to detect any potential breakout. Since a breakout isn’t always obvious due to the range of the observed values, visual detection of breakouts may not be sufficient, and thus a programmatic approach that can automatically detect breakouts is imperative.

Since flow data at sewer pipes fluctuates with rainfall-runoff events, the runoff response may interfere the breakout detection. Therefore, the algorithm must be robust against the presence of anomalies.

Also, field crews (monitoring, Operation & Maintenance, etc.) can quickly respond to the issue once a breakout has been detected. This analysis is also applicable for other monitored data, such as the trunk and outfall levels at drainage system regulators.

### 2 OBJECTIVES

Measurement Accuracy determines the overall model quality. This study aims to develop a workflow as a QA measure for detecting breakouts in flow monitoring data utilizing a sound change point detection algorithm. First, a sound breakout detection algorithm is selected from several state-of-art methods, where it must met the following requirements:

* able to detect multiple types of change
* able to detect mulitple breakouts
* weak or no assumption on data distribution
* robust against the presence of anomalies
* fast on large datasets, produce reliable results | low latency, high reliability

Next, the algorithm parameters are carefully tuned to optimize the outcome.

Finally, an application is developed in the R statistical programming language for analyzing the level and velocity time-series and generating a quarterly report, which is executed biweekly automatically.

### 3 METHODOLOGY

#### 3.1 Change point analysis

In statistics, the ‘breakout detection’ belongs to the change point analysis, which has been widely researched over the past 50 years in a wide variety of fields, such as finance (Edwards et al 2012), genetics (Chen and Gupta 2011), and signal processing (Basseville 1988). As we’ve entered the ‘Big Data’ era, it has gain it’s popularity in low latency, high reliability online analytics for cloud data (James, et al. 2016).

A breakout is typically characterized by two steady states and an intermediate transition period. Mathematically, for data z1,…,zn, if a changepoint exists at τ, then z1,…,zτ differ from zτ+1,…,zn in some way.

There are many different types of change, such as Mean shift, which is a sudden jump in the time series; Ramp up/down, which is a gradual change in the value of the metric from one steady state to another; distribution change, which is a change of the data distribution.

Change point analysis mainly answers the following questions:

* does a change point exists?
* when does the change point happen?
* what is the significance of the change point?
* how sure is the change point?

Numerous changepoint detection algorithms have been widely researched in various fields of industries (Rodionov 2005).

Depending on the data distribution assumption, a breakout detection algorithm generally falls into two categories:

* parametric: The parametric analysis assumes that the observed distributions belong to a family of distributions. Common parametric methods are PELT (Killick et al, 2012), …,
* non-parametric: do not make assumption on data distribution and use density estimation instead. non-parametric methods (Pohlert 2018), such as E-divisive (Matteson & James 2012), …

Based on its application, a breakout detection can be designed for

* online analytics: the data is streaming into the model;
* offline analytics: the data are processed in batches.

Several R packages already exists for breakout detection, e.g., [R packages introduction] — ecp, cpm, changepoint, reakoutDetection, etc. - ecp: e-divisive, e-agglometric

* cpm
* changepoint (Killick & Eckley, 2014)
* BreakoutDetection: EDM

Many of the existing techniques have limitation in analyzing flow data due to the fact that they are not robust in the presence of anomalies, and may not suitable for this study.

#### 3.2 E-divisive with medians (EDM)

EDM is a novel statistical technique that employs energy statistics (E-divisive) to detect divergence of means. To be robust against the presence of anomalies, EDM uses the rolling median as a local smoother to the raw data. In addition, EDM is capable of detecting multiple change points.

Energy statistics compares the distances of means of two random variables contained within a larger time series. The e-divisive method recursively partitions a time series and uses a permutation test to determine change points, but it is computationally intensive. To overcome this, EDM uses interval trees to efficiently approximate the median, and therefore is much faster than E-Divisive.

EDM can detect various types of change, including ‘mean shift’ (sudden change) , ‘ramping’ (gradual change), and change in distributions. since EDM is non-parametric, it doesn’t make any assumption about the distribution of the time-series, instead, it learn the current distribution as a reference. When the distribution suddenly change EDM can detect the variation; This is important since the distribution of production data seldom (if at all) follows the commonly assumed normal distribution or any other widely accepted model.

A comparsion (ref) shows that EDM outperformed the PELT in the majority of data sets. Due to the weaker assumption in EDM, The significance of the breakout is determined by permutation test, where data from the two time series are permutated a finite number of times, and hence the EDM takes longer to execute than the PELT. However, the EDM has shown a comparable efficacy than other change point analysis algorithms. According to xxxx (2016), EDM is 3.5x faster than the other popular algorithms.

#### 3.3 BreakoutDetection package in R

The EDM algorithm is implemented by the BreakoutDetection, which is an open-source R package developed by Twitter Engineers and has been used for analyzing network breakouts on a daily basis at Twitter.

The breakout() is the detector function, which contains the following parameters:

Z: The input time series. This is either a numeric vector or a data.frame which has ‘timestamp’ and ‘count’ components. min.size: The minimum number of observations between change points. method: either ‘amoc’ (At Most One Change) or ‘multi’ (Multiple Changes). For ‘amoc’ at most one change point location will be returned.

for single change point (method=‘amoc’):

alpha: The alpha parameter used to weight the distance between observations. This is a real number in the interval (0,2]. The default value is alpha=2.

exact: This flag is for selecting the use of true medians (TRUE) or approximate medians (FALSE) when determining change points. The default value is exact=TRUE.

sig.lvl: Once a change point is found its statistical significance is determined through a hypothesis test. sig.lvl specifies the significance for the hypothesis test. The default value is sig.lvl=0.05.

nperm: The number of permutations to perform in order to obtain an approximate p-value. If nperm=0 then then permutation test is not performed. The default value is nperm=0.

For multiple change analysis (method=’multi“):

degree: The degree of the penalization polynomial. degree can take the values 0, 1, and 2. The default value is degree=1.

beta: A real numbered constant used to further control the amount of penalization. This is the default form of penalization, if neither (or both) beta or (and) percent are supplied this argument will be used. The default value is beta=0.008.

percent: A real numbered constant used to further control the amount of penalization. This value specifies the minimum percent change in the goodness of fit statistic to consider adding an additional change point. A value of 0.25 corresponds to a 25% increase. percent doesn’t have a default value.

### RESULTS

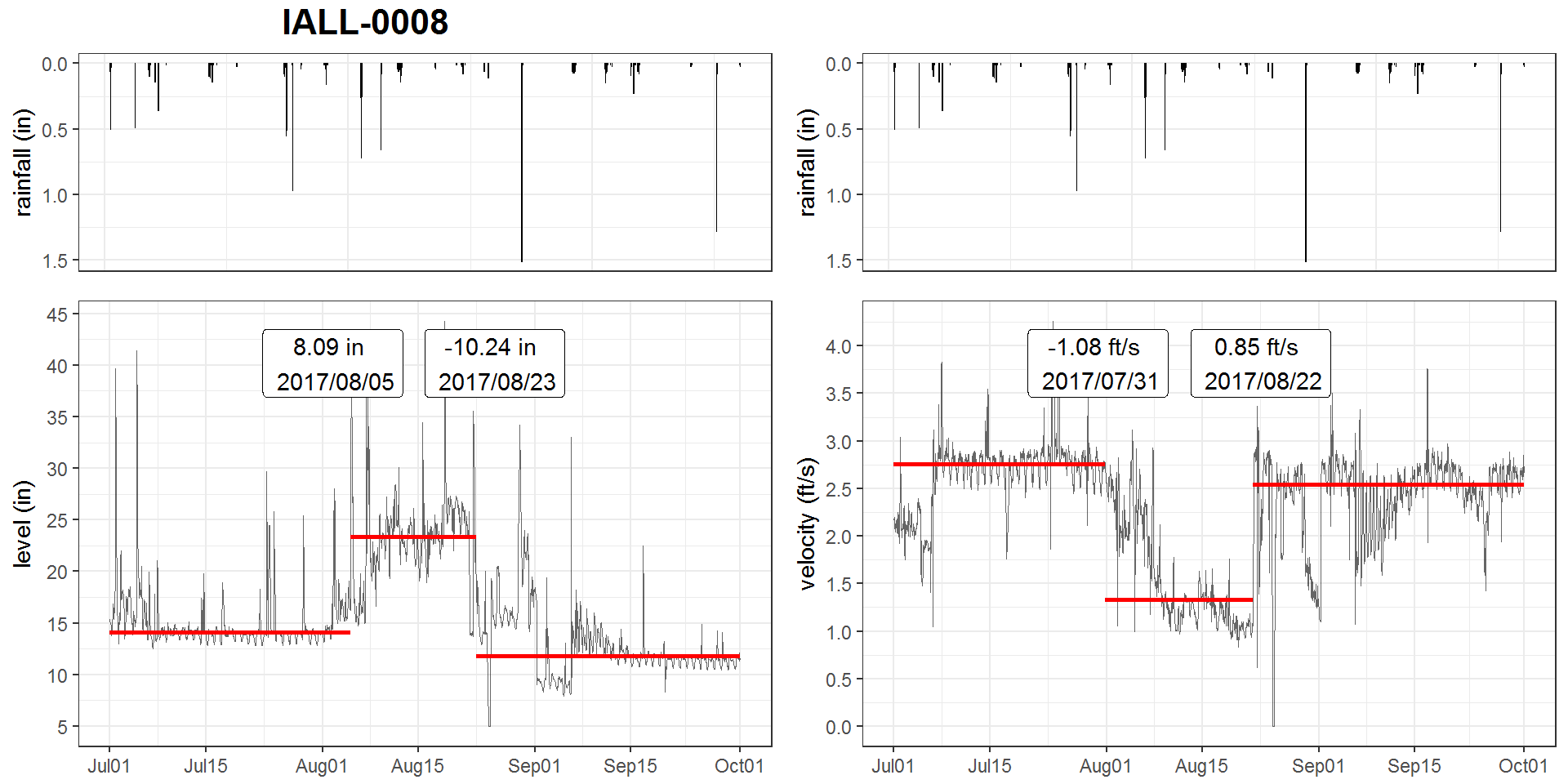
After a thorough literature review and testing with sample data, the E-Divisive with Medians (EDM) algorithm is adopted in this study.

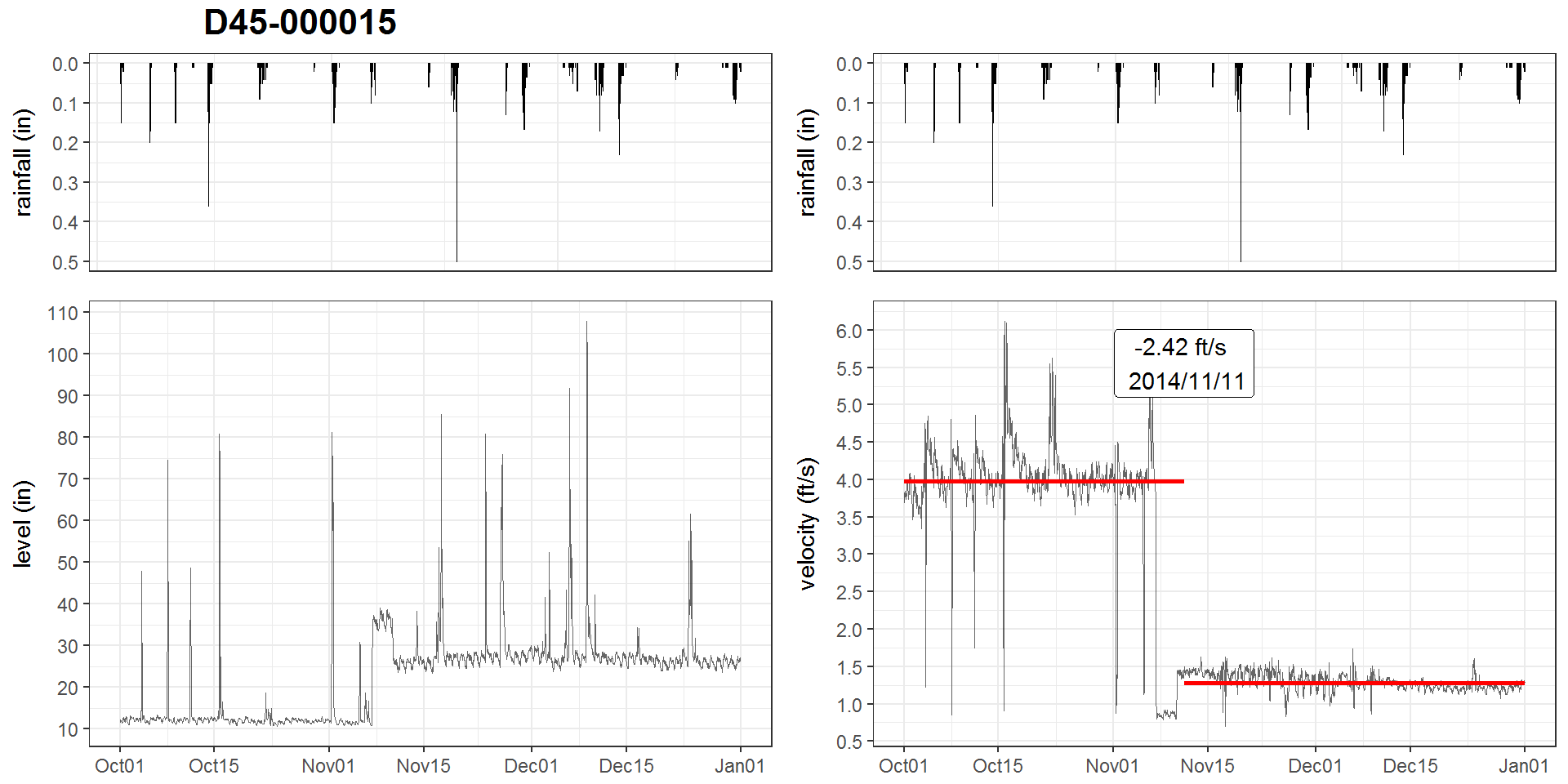
Based on a series of tests using sample data, the breakout() function arguments are determined:

* Z: The quarterly time-series averaged by the hour is used for better efficiency
* min.size: set to 240, which means the min distance between two change points must be at least 10 days (10 days x 24 sample/day = 240 samples)
* method: set to ‘multi’ as detecting multiple breakouts is desired
* degree: set to 1, leave as default.
* beta: set to 0.008, determined by elbow plot
* percent: set to 0.10

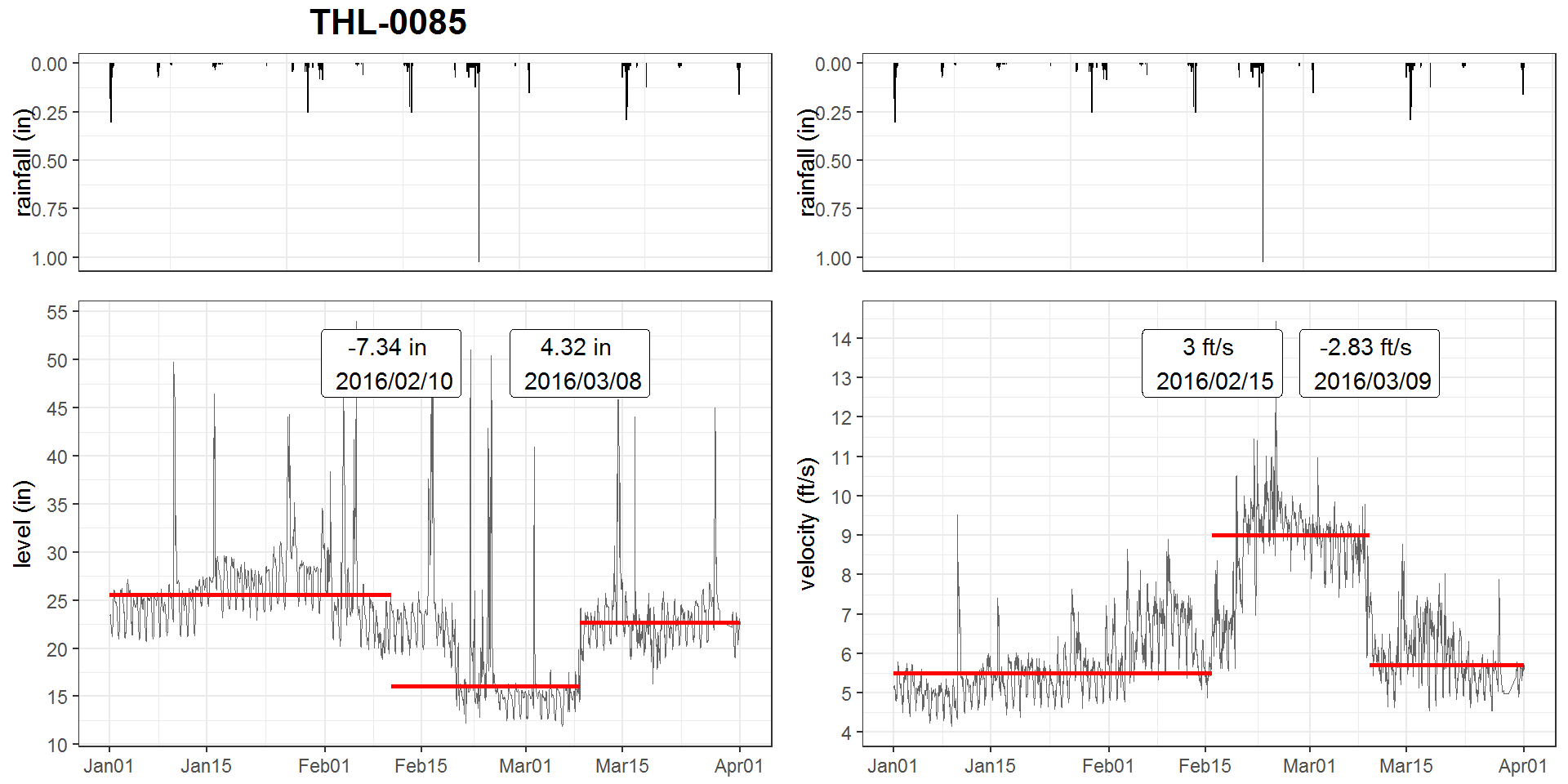
A R markdown document is developed that includes scripts for breakout detection, summarize the results, and plot breakouts with time-series for multiple sites. To improve the performance, parallel computation is utilized. The output is a quarterly report that includes a summary table of breakouts, and hydrographs for all sites with hyetograph overlaid. The report is automatically updated bi-weekly when data is updated. In the future, it’s expected to be updated more frequently when real-time data becomes available.

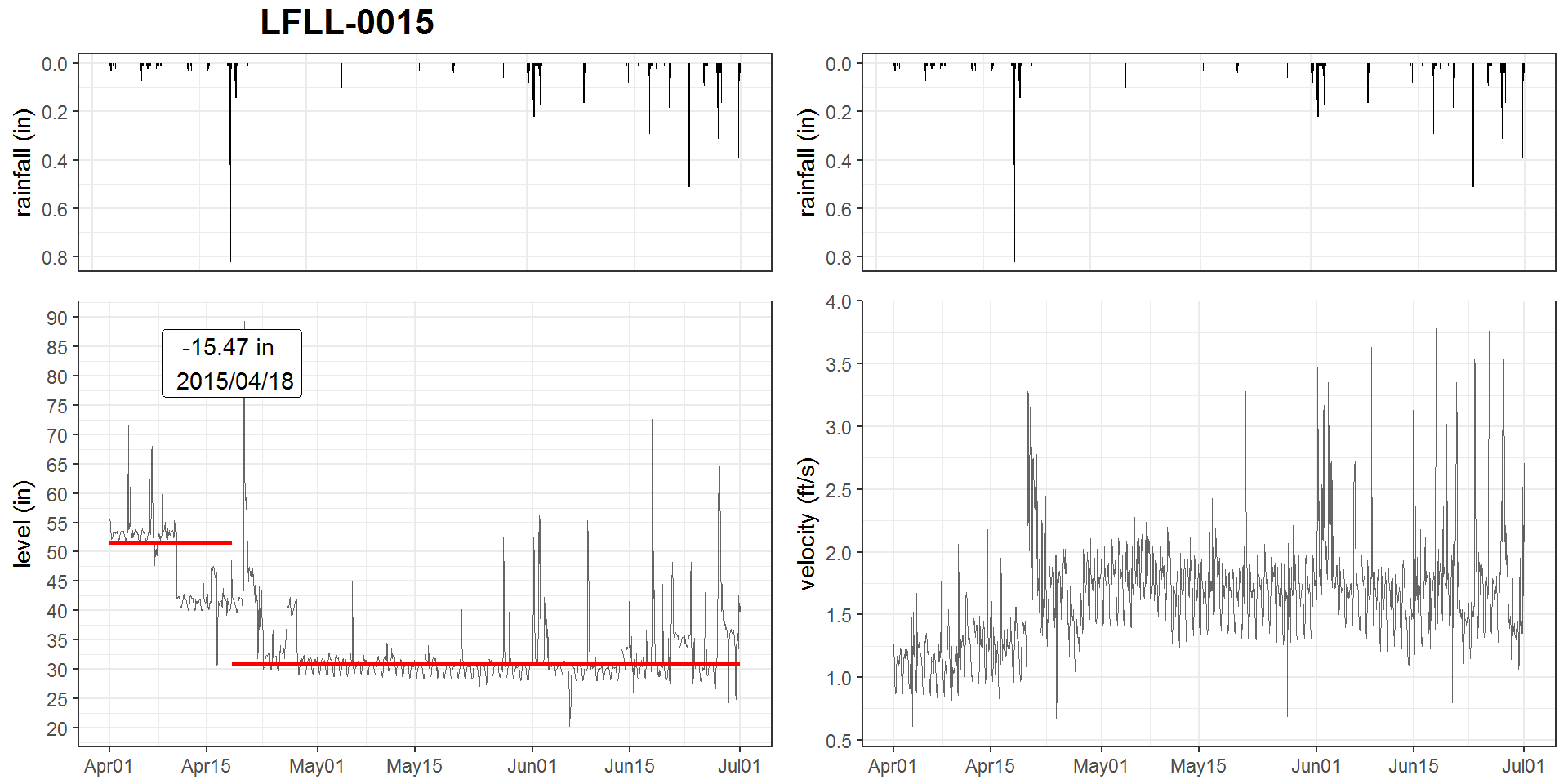
A few breakout examples based on real data are shown in Fig.xxx. a) shows an positive change in level and a negative change in velocity near August 1, 2017, and it resets to its original values near August 22, 2017. This is typically caused by pipe surcharging.

 b) shows a sudden downward shift on November 11, 2014. The opposite trend is observed in the level data, but the breakout was not detected somehow.



1. flushing effect?





limitation:

* cannot detect breakouts at both ends of the time-series
* large runoff response may be identified as breakouts

### CONCLUSIONS

EDM is proven to be a reliable, effective, and efficient breakouts detection technique for flow monitoring data (level & velocity) with properly tuned parameters. This method is expected to be applicable for other monitiored time-series data, such as outfall levels at CSO regulators.

In this study, A R application that utilizes EDM for breakout detection is developed. This analysis provides a quality control to the modeling data, which would be beneficial for improving the model quality, and quickly respond to field issues. This analysis is also applicable for other monitored data, such as the trunk and outfall levels at drainage system regulators.

### REFERENCE

James, Nicholas A., Arun Kejariwal, and David S. Matteson. “Leveraging cloud data to mitigate user experience from ‘Breaking Bad’: The Twitter Approach.” In Big Data (Big Data), 2016 IEEE International Conference on, pp. 3499-3508. IEEE, 2016.

Matteson, David S., and James, Nicholas A. “A nonparametric approach for multiple change point analysis of multivariate data.” Journal of the American Statistical Association 109, no. 505 (2014): 334-345.

James, Nicholas A., and David S. Matteson. “ecp: An R package for nonparametric multiple change point analysis of multivariate data.” arXiv preprint arXiv:1309.3295 (2013).

Edwards, Robert D., Magee, John, and Bassetti, W.H.C.. “Technical analysis of stock trends”. CRC Press, 2012

Jie Chen and Arjun K Gupta. Parametric Statistical Change Point Analysis: With Applications to Genetics, Medicine, and Finance. Springer, 2011.

Michèle Basseville. Detecting changes in signals and systemsa survey. Automatica, 24(3):309–326, 1988

Killick, Rebecca, and Idris Eckley. “changepoint: An R package for changepoint analysis.” Journal of statistical software 58, no. 3 (2014): 1-19

Rebecca Killick, Paul Fearnhead, and IA Eckley. Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500):1590–1598, 2012

Rodionov, S. N. “A brief overview of the regime shift detection methods.” Large-scale disturbances (regime shifts) and recovery in aquatic ecosystems: challenges for management toward sustainability (2005): 17-24.

Pohlert, Thorsten. “Non-parametric trend tests and change-point detection.” CC BY-ND 4 (2018).