CANARY: A Water Quality Event Detection Algorithm Development Tool

David Hart[1], Sean A. McKenna[1], Katherine Klise[1], Victoria Cruz[1] and Mark Wilson[2]

[1]Geohydrology Department, Sandia National Laboratories, PO Box 5800 MS 0735, Albuquerque, NM 87185-0735, ph: 505 844 4674; fax: 505 844 7354; em: dbhart@sandia.gov

[2]GRAM Incorporated, 8500 Menaul Blvd., NE, Suite B-335, Albuquerque, New Mexico, 87112, em: mpwilso@graminc.com

Abstract

The detection of anomalous water quality events has become an increased priority for distribution systems, both for quality of service and security reasons. Because of the high cost associated with false detections, both missed events and false alarms, algorithms which aim to provide event detection aid need to be evaluated and configured properly. CANARY has been developed to provide both real-time, and off-line analysis tools to aid in the development of these algorithms, allowing algorithm developers to focus on the algorithms themselves, rather than on how to read in data and drive the algorithms. Among the features to be discussed and demonstrated are: 1) use of a standard data exchange format for input and output of water quality and operations data streams; 2) the ability to "plug in" various water quality change detection algorithms, both in MATLAB® and compiled library formats for testing and evaluation by using a well defined interface; 3) an "operations mode" to simulate what a utility operator will receive; 4) side-by-side comparison tools for different evaluation metrics, including ROC curves, time to detect, and false alarm rates. Results will be shown using three algorithms previously developed (Klise and McKenna, 2006; McKenna, et al., 2006) using test and real-life data sets.

Introduction

The CANARY program grew from a need to test multiple event detection algorithms simultaneously on multiple data sets. While each algorithm operated on the same types of data inputs, some included limited only data inputs, each algorithm used slightly different syntax and each produced slightly different result types. Rather than try to write drivers for each algorithm separately, CANARY was written to drive all three algorithms. Additionally, the algorithms were given a standardized input and output format so that new algorithms could be easily added to CANARY's testing capability. The goal of this work was to provide an extensible platform where event

detection algorithms can be tested without the need to re-write driver programs to read in water-quality data.

CANARY accepts any numeric time-series input in the appropriate format. It can also process data from multiple locations simultaneously, and can pass only a subset of information to a given algorithm if desired. For example, if a researcher wanted to test the performance of a "set-points" style algorithm operating only on chlorine data against a multivariate algorithm operating on a complete set of water-quality measurements (including the chlorine), CANARY could be set up to handle both algorithms and process them simultaneously.

The three algorithms CANARY was originally developed to use were: the time series increment (INC), linear filter (LPC) and multivariate nearest-neighbor (MV-NN) (Klise and McKenna, 2006; McKenna, et al., 2006). Since that time, the binomial event discriminator has been added to identify changes in baseline water quality and decrease false alarms (McKenna, et. al, 2007). This paper will outline the major capabilities of the CANARY software, describe the process to extend CANARY to additional algorithms, and provide information regarding how to obtain the CANARY program for research use.

Algorithm Testing

The primary input CANARY uses are water-quality measurements. These typically include chlorine residuals, total organic carbon, pH, specific conductivity, etc., but many event detection algorithms can use any numeric input that is time-series ordered. CANARY can be operated in one of four different modes. These are: on-line mode, off-line analysis mode, off-line testing mode, and off-line data conversion mode. Each of these different modes will be discussed separately.

On-line Testing Mode

CANARY's on-line mode most closely simulates what a water utility may experience in an operations center. Data are read from a file or database, and are updated at regular intervals. Every time new datum is input, CANARY runs the event detection algorithms to make a determination about event status. If one of the algorithms returns a result that exceeds the thresholds given, then CANARY will sound an "alarm" by printing a message to the screen. Because it is most likely that the water utility operators will not do analysis directly – this task falling to water quality or quality assurance staff – there is only limited information provided to the operator. Specifically, the location, the date and time, and the severity of the event are the only information printed in the operations window. An example of an event in both text and graphics modes is shown in Figures 1 and 2. In these examples, the MV-NN algorithm has been used with three different thresholds to find events.

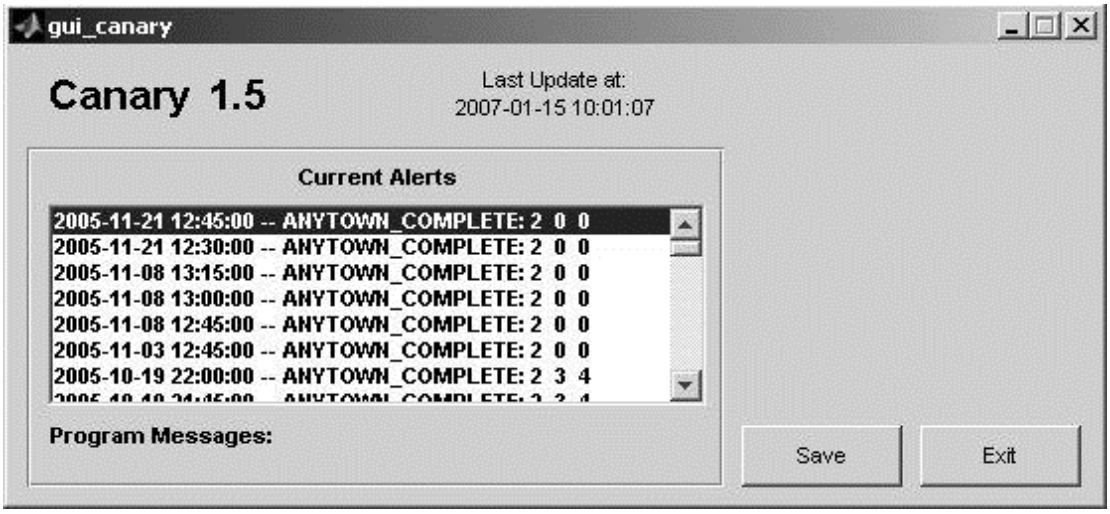

Figure 1 - Real-time alerts listed as they are processed.

```
     |      TS: 13776
     |      TS: 13777
-%^  | *** 2005-11-21 12:30:00 -- ANYTOWN_COMPLETE: 2  0  0
     |  *  2005-11-21 12:45:00 -- ANYTOWN_COMPLETE: 2  0  0
     |      TS: 13780
     |      TS: 13781
```

Figure 2 Console output showing an event at threshold level of 2, but not at any other thresholds.

Off-line Testing Mode

The off-line testing mode, called "batch" mode in the configuration, is nearly identical to the on-line mode in terms of output. However instead of updating at regular intervals, the file or database containing the water quality data is read only once, and the entire data set is then processed. This does not give CANARY or its algorithms any advantage in event detection, since the data is still processed in time-series order, one step at a time, but it does improve performance by limiting the number of times CANARY must read the data sources. Once the data has been input and processing is taking place, the screen output is identical to the on-line mode's output. A sample is shown in Figure 3.

Off-line Analysis Mode

In analysis mode CANARY provides a graphical interface that shows where events were detected, the raw data values at that time, and where any baseline changes may have occurred. This mode operates on data that have already been processed through the on- or off-line testing modes. This mode also can read a list of where "true" events took place. Whether these events are real water quality changes that have been identified by looking at the raw data, or whether they are simulated events where the raw data have been modified to create an event, they give CANARY additional information to use to calculate statistics regarding the algorithm's performance. These

include receiver-operating characteristic curves (ROC curves), false alarm rates, false-negative (missed event) rates, and average time-to-detect, all important metrics when evaluating an event detection algorithm's performance. Figure 4 shows an example of the analysis screen.

```
 -%^ | *** 12/11/2006 23:10:00 -- ANYTOWN: 0.5          0
0
     |      TS: 278
 -%^ | *** 12/11/2006 23:20:00 -- ANYTOWN: 0.5          0
0
     |   *  12/11/2006 23:25:00 -- ANYTOWN: 0.5          0
0
     |      TS: 281
 -%^ | *** 12/11/2006 23:35:00 -- ANYTOWN: 0.5          0
0
     |   *  12/11/2006 23:40:00 -- ANYTOWN: 0.5          0
0
     |   *  12/11/2006 23:45:00 -- ANYTOWN: 0.5          0
0
     |      TS: 285
 -%^ | *** 12/11/2006 23:55:00 -- ANYTOWN: 0.5          0
0
     |      TS: 287
```

Figure 3 Output showing the results of a test with the threshold set to 0.5.

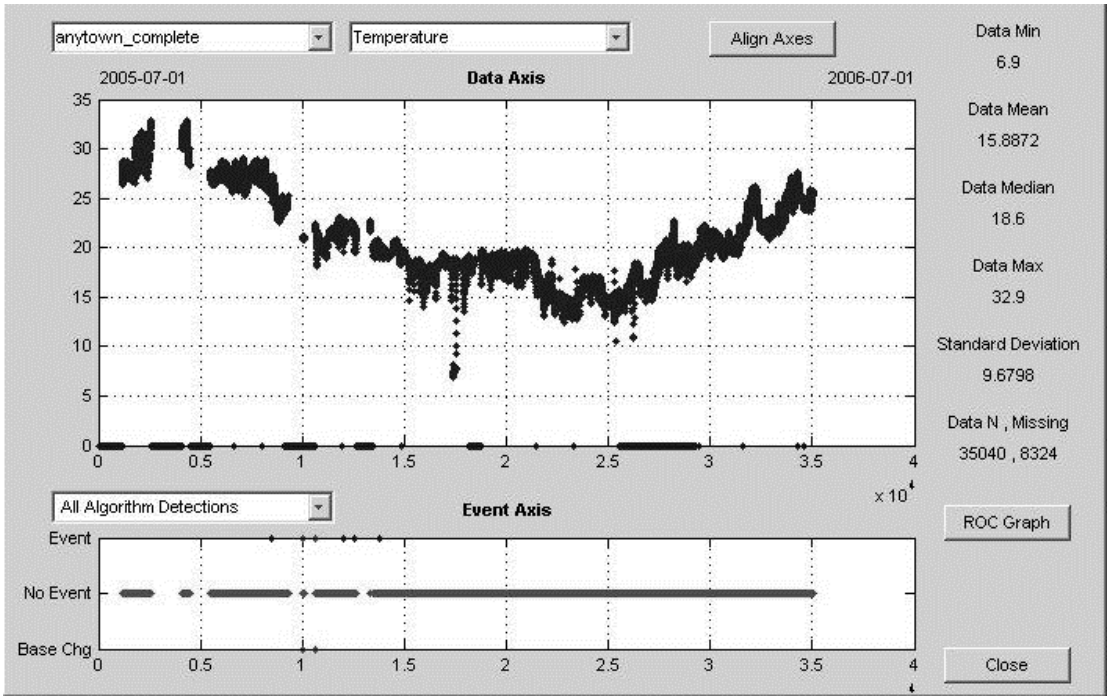

Figure 4 Analysis mode, showing the raw temperature data aligned with the events detected.

Off-line Data Conversion Mode

While this may seem like a rather routine mode compared to the others, it has the potential to be one of the most useful. File-based access and database access are fairly

time-intensive computing operations. Despite the fact that they are simple to do, accessing disk-drives is one of the slowest operations on a computer system. The data conversion mode, called "saveonly" mode, allows the user to read all the data one time, and have those data saved into a compact binary file that is both smaller than the original data and is much faster to read, as much as 500 times faster! Once this file has been written, CANARY can be run many times with different algorithm settings using the same data sets without the overhead that comes from reading the data from text-files or databases.

Inputs and Outputs

CANARY can use both files and databases as data sources; the program produces text- files as outputs. The advantage to using database access is that most utilities use databases to store their water quality data. Unfortunately, databases are difficult to use when creating test data or simulating events. Files, especially spreadsheet type files, are very easy to manipulate, save, and modify. The drawback with both is the nearly infinite different ways the data can be laid out within them. CANARY has defined two specific, but simple, layouts for data inputs. While they are discussed in detail in the CANARY User's Manual, a brief description will be provided here.

Input Data

The database format is very simple. There will be a more structured format available, but as the details have not been finalized by the US EPA. Currently, CANARY looks at a single table, with rows ordered by a field called "Datetime," and all water quality data contained in fields of an appropriate name. For a system with chlorine, pH and TOC sensors, the table might contain fields called "Datetime", "Cl2", "pH" and "TOC". The preferred file-based format is a comma-separated values (CSV) file. Each row is a single entry of raw water quality data. Table 1 shows a selection of data from a sample file. The file can contain information for multiple sensors, which would be identified by name in the first column, and even for multiple locations or sensor suites, indicated by name in the fourth column. A full description of the input fields is given in Appendix A.

Table 1 - Sample data from a CSV formatted input file

Cl	Normal	User	Anytown	User	12/11/2006	0:05:00	0.958	mg/L	user
Cl	Normal	User	Anytown	User	12/11/2006	0:10:00	0.966	mg/L	user
Cl	Normal	User	Anytown	User	12/11/2006	0:15:00	0.972	mg/L	user
Cl	Event	User	Anytown	User	12/11/2006	0:20:00	0.988	mg/L	user
Cl	Event	User	Anytown	User	12/11/2006	0:25:00	0.973	mg/L	user
Cl	Normal	User	Anytown	User	12/11/2006	0:30:00	0.994	mg/L	user

While most of the columns will make sense on their own, column two deserves special note. The two entry types in this column, "Normal" and "Event", give CANARY additional information regarding the analysis portion. If an event is detected by an algorithm at a time step marked "Event", this determination is marked as a "true positive," or success. If an event is detected at a time step marked "Normal", the determination is considered a "false-positive." This information is

reported at the end of a CANARY run, as shown in Figure 5, and is also shown in the output files. It should be noted that marking places as events is not a requirement – however every detection will be considered a "false-alarm" if there are no "true" events indicated.

```
-%^ | *** For the 'ChangeDetect_MV_NN_144_3o45' algorithm at ANYTOWN
    |  *  Given thresholds of: 3.45
    |  *  number of false-pos: 475
    |  *  number of  true-pos: 25
    |  *  number of false-neg: 0
    |
-%^ | *** For the 'ChangeDetect_MV_NN_144_3o5' algorithm at ANYTOWN
    |  *  Given thresholds of: 3.5
    |  *  number of false-pos: 27
    |  *  number of  true-pos: 24
    |  *  number of false-neg: 1
    |
-%^ | *** ROC Area           : 0.9856
    |  *  Printing possible contamination events to file 'csvTest-out'
```

Figure 5 Termination of run information regarding algorithm performance.

Output Data

As CANARY is increasingly used by the development team, more outputs are added. Every CANARY invocation produces a log file that captures console output to a file. This is the same type of outputs shown in Figures 2 and 3. Each run also produces a MAT file, a binary file that can then be read in by CANARY to analyze, repeat, or modify a run with much faster performance than text files can provide.

Each CANARY run also produces a description file. This file contains an overview of the false-alarm rate, false-negative rate, total number of detections, overview statistics on each of the raw data streams, and information about other files that were produced. Each location analyzed produces a file containing raw data organized by time and date, the raw outputs of the algorithms (prior to comparison to thresholds), the detection results (after comparison to thresholds), and a ROC curve file. These files can be used to generate false-positive and other statistics if there were no "true" events indicated in the input. An example of the description file is shown in Appendix B.

Algorithm Extensions

One of the primary reasons CANARY was written was to aid in the development of algorithms in the future, which requires the ability to "plug in" a new algorithm. Rather than try to use linked libraries or call an external executable, a Java abstract class was defined that can be used as an interface by algorithm developers, and which would be directly accessible by CANARY. The abstract class "wqEventDetectAlgorithm" is defined in Figure 6.

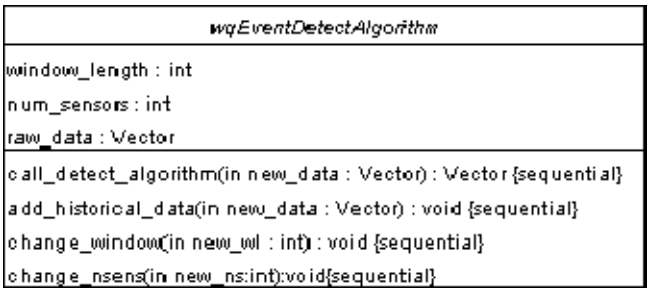

Figure 6 UML Class Diagram for wqEventDetectAlgorithm abstract class.

The Vector objects are vectors of doubles. The return vector contains one entry for each sensor, and one final entry for a combined value. The values returned are the raw results that are then compared to the thresholds set in the configuration to determine if an event has taken place.

Conclusion

CANARY is an extensible tool that will hopefully reduce the amount of redundant work needed to write and test event detection algorithms for use in water or other utilities. Large data sets can be processed with multiple algorithm settings simultaneously, and the data can be stored in a compact format for re-testing and analysis. It has been successful in the testing of the INC, LPC and MV-NN algorithms developed previously, and in testing the binomial event discriminator.

Acknowledgements

CANARY was written in the MATLAB® m-code language, with a smattering of Java. Because not every group that might be interested in CANARY has access to MATLAB, the MATLAB compiler was used to create a stand-alone version of the program that can be installed and run on nearly any computer. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by Sandia Corporation, the United States Government, or any agency thereof.
CANARY is available via the Lesser GNU Public License for research purposes; is it not intended to replace any commercial software, but rather to serve as an aid in developing software. To obtain a copy please contact one of the authors, or go to the software.sandia.gov website and choose “CANARY”.

This research was conducted as part of an Interagency Agreement between the U.S. Environmental Protection Agency and the Department of Energy. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000.

Appendix A CSV Formatted Input

Comma-separated-values (CSV) files are a common spreadsheet format that is compact and readable. CANARY supports reading from a CSV formatted file that has no header rows, and which has data organized into the following columns:

- 1st Column Sensor or instrument identifier – this identifier must start with a letter and be unique among sensors in a given location / monitoring station grouping. The identifier should not include any non-alphanumeric characters other than spaces, '-' signs, and '_' characters. These characters will be converted to '_' characters inside CANARY.
- 2nd Column Event Indicator – either "Normal" or "Event"; "Event" signifies that a known event took place, or was continuing, at this time-step at the appropriate location. These provided events are what CANARY uses for comparison against the events detected by each to obtain false-alarms, ROC curves, and detection likelihoods. Use "0" or "Normal" for normal and "1" or "Event" for events. It is OK to run CANARY without knowledge of any actual events in which case this column is all zeros (all "normal").
- 3rd Column Not currently used – but the column must be present with any value.
- 4th Column Location Identifier – used to group together a set of sensors that are all at the same location or monitoring station. Because algorithms may fuse data from different sensors at the same location, these location identifiers should be unique for each monitoring station; again, these identifiers must start with a letter and have the same restrictions as the sensor identifier.
- 5th Column Not currently used – but the column must be present with any value.
- 6th Column Measurement Date – while the date format can be re-configured, the preferred format is "yyyy-mm-dd".
- 7th Column Measurement Time – the time format can be re-configured, but "HH:MM:SS" is preferred.
- 8th Column Measurement Value – the important information! If blank, the sensor will be assumed to have dropped out (i.e., no data reported for that time step at that sensor and location), although "NaN" is an acceptable value to indicate sensor failure as well.
- 9th Column Measurement Units – although the units are not currently used by any of CANARY's algorithms, they can be provided.
- 10th Column User Defined – typically used to provide some text information or notes within the CSV file – please don't use commas, but any other values are acceptable.

Appendix B Sample Description File

```
--- Description file "csvTest-out.descr" ---
   Input from CSVFILE "csvTest.csv"
   Output to files: "csvTest-out.*"
   Run-log in file: "install-tests.log"

--- Data Summaries -- Statistics based only on non-empty values present in data
  Location     Sensor  Min     Mean    Median  Max     Std.Dev N-total N-missing
  Anytown      cl      0.950   1.268   1.250   2.602   0.297   1000    0
  anytown      ph      6.841   6.999   7.001   7.151   0.055   1000    0

   ROC-Curve Output to file: csvTest-out.anytown.roc    ROC area = 0.985600

   Step-by-Step detection output (DET) to file: csvTest-out.anytown.det

--- DET Files ------------------------
 Values of 1 indicate an event, -1 indicates a baseline shift,
     NaN indicates no data available, 0 indicates no event

 First row: header line indicating the threshold used in each column
 Remaining rows: event information starting at 12/11/2006 0:05:00
     and continuing every 00:05:00 hours until 12/14/2006 11:25:00

Column 1 : Provided events
Column 2 : Events detected by algorithm "ChangeDetect_MV_NN", win= 144, threshold= 0.5
Column 3 : Events detected by algorithm "ChangeDetect_MV_NN", win= 144, threshold= 1.5
Column 4 : Events detected by algorithm "ChangeDetect_MV_NN", win= 144, threshold= 2.5
Column 5 : Events detected by algorithm "ChangeDetect_MV_NN", win= 144, threshold= 3.5
Column 6 : Events detected by algorithm "ChangeDetect_MV_NN", win= 144, threshold= 3.5
```

References

McKenna, S.A., K.A. Klise and M.P. Wilson, 2006, Testing Water Quality Change Detection Algorithms, in Proceedings of the 8th Annual Water Distribution System Analysis Symposium, Cincinnati, OH, August 27-30, 2006.

Klise, K.A. and S.A. McKenna, 2006, Multivariate Applications for Detecting Anomalous Water Quality, in Proceedings of the 8th Annual Water Distribution System Analysis Symposium, Cincinnati, OH, August 27-30, 2006.

Klise, K.A. and S.A. McKenna, 2006, Water quality change detection: multivariate algorithms, in Proceedings of SPIE (International Society for Optical Engineering), Defense and Security Symposium 2006, April 18-20, Orlando. Florida, 9pp.

McKenna, S.A, D.B. Hart, K.A. Klise, V.A. Cruz, and M.P. Wilson, 2007, Event Detection from Water Quality Time Series, in Proceedings of World Environmental & Water Resources Congress 2007, May 15-19, Tampa, Florida.