Preprints of the
9th International Symposium on Advanced Control of Chemical Processes
The International Federation of Automatic Control
June 7-10, 2015, Whistler, British Columbia, Canada

# Methodology and Application of Pattern Mining in Multiple Alarm Flood Sequences [★]

**Shiqi Lai**, **Tongwen Chen**

*Department of Electrical and Computer Engineering, University of Alberta, Edmonton T6G 2V4, Canada, (e-mail: slai3@ualberta.ca, tchen@ualberta.ca)*

**Abstract:** Alarm floods have always been serious hazards in industrial process monitoring since they overwhelm operators with large amount of alarm messages raised within a short period of time. In this paper, we propose an algorithm to find an optimal alignment of multiple alarm flood sequences so that based on this alignment an alarm sequence pattern can be easily found. The pattern could reveal the correlation between alarm messages in the alarm floods, which cannot be obtained by applying other alarm management techniques such as delay timers and dead-bands. The pattern could also help find the root cause, locate badly designed part in alarm systems, and predict incoming alarm floods. A dataset from an actual chemical plant has been used to test the effectiveness of the proposed algorithm.

*Keywords:* Alarm flood analysis; Time-stamped sequences; Multiple sequence alignment; Smith-Waterman algorithm; Industrial alarm monitoring

## 1. INTRODUCTION

Alarm systems play a key role in industrial process monitoring. The advent of Distributed Control Systems (DCS) makes it much easier for engineers to configure alarms. However, without rational configurations, large amounts of alarm messages can be generated within short periods of time, resulting in alarm floods. During an alarm flood, the operator may be overwhelmed by numerous alarm messages, leading likely to improper handling of important alarms. In order to track alarm floods, EEMUA and ISA standards EEMUA (2007); ISA (2009) have suggested the threshold alarm rate to be 10 alarms per 10 minutes.

Chattering and consequence alarms are the two main components of alarm floods. Effective methods such as high density alarm plots, calculation of a chattering index, delay timers, and dead-bands have been proposed in Kondaveeti et al. (2010, 2013); Izadi et al. (2009) to handle chattering alarms. However, alarm floods usually cannot be totally suppressed by these methods because of the existence of consequence alarms. In this case, alarm flood analysis is useful for revealing correlations between alarm messages in alarm floods. Pattern mining is one of the key procedures in alarm flood analysis.

### *1.1 Current status of alarm flood pattern analysis*

Manual pattern mining based on expert consultation and process knowledge usually brings the most accurate result. But as the size of dataset grows, manual pattern mining becomes almost impossible because of its low efficiency. Moreover, process lags may influence the orders of the alarms, which further increases the difficulty of manual pattern mining in alarm floods.

Applying data mining techniques to analyze patterns in alarm floods has become a hot topic recently. In Kordic et al. (2008), the authors tried to find patterns in alarm floods by applying a context based segmentation, of which the start and termination points were determined by a target tag and a filter respectively. The authors in Ahmed et al. (2013) applied Dynamic Time Warping (DTW) to find the global alignment of a pair of alarm flood sequences in order to reveal the pattern. In Folmer and Vogel-Heuser (2012), a pattern mining method based on the pattern growth technique was applied to find pattern alarm flood sequences. An approach based on the Generalized Sequential Patterns (GSP) was proposed in Cisar et al. (2009) to mine the patterns in alarm floods; the algorithm was robust to disturbances and process lags. The authors in Cheng et al. (2013) modified the Smith-Waterman algorithm to align a pair of alarm floods; it tolerates to disturbances and process lags; but is limited to pairwise usage.

### *1.2 Background of sequence pattern analysis*

Frequent pattern mining and sequence alignment are two main approaches of sequence pattern analysis. Frequent pattern mining algorithms, which have been extensively used to analyze transactions in business area, search for the patterns that appear frequently in the database and the association rules between them. Foundamental methodologies of frequent pattern mining include Apriori-like algorithms such as Agrawal and Srikant (1995), FP-growth methods such as Han et al. (2000), and vertical pattern growth algorithms such as Zaki (1998). Techniques

[★] This work was supported by an NSERC CRD project.

involving hash tables and projections have also been developed to improve the efficiency of frequent pattern mining algorithms.

On the other hand, alignment algorithms have been widely used for pattern mining in gene and protein sequences. Foundamental approaches are pairwise alignment algorithms in Needleman and Wunsch (1970), Smith and Waterman (1981), Altschul et al. (1990), and Pearson and Lipman (1988). In order to align multiple sequences, the authors in Johnson and Doolittle (1986) modified a pairwise algorithm to align multiple sequences simultaneously. In addition, in Feng and Doolittle (1987) and Thompson et al. (1994), the authors first built up a phylogenetic tree based on pairwise similarity scores, then found the multiple sequence alignemnt by aligning the sequences progressively according to the tree. Hidden Markov Models (HMM) had also been applied to find homologies in gene and protein sequences Eddy (1995); but the states of HMM must be designed manually, as pointed out in Eddy (1998).

### 1.3 Contribution of our work

The algorithm proposed in this paper is an extension of the one in Cheng et al. (2013) to the case of aligning multiple alarm flood sequences. The technique involves the following new elements: a similarity scoring function, a dynamic programming equation, a back tracking procedure, and an alignment generation method. The common pattern of alarm flood sequences can be easily obtained thereafter based on the generated alignment. A dataset from an actual chemical plant has been used to test the effectiveness of the proposed algorithm.

### 1.4 Organization of the paper

The rest of the paper is organized as follows. Key procedures of alarm flood analysis are introduced in Section 2. Then in Section 3, we show the principle of the algorithm for aligning three alarm flood sequences. Datasets from an actual chemical plant will be used to test the effectiveness of the proposed algorithm for aligning three alarm flood sequences in Section 4. At last, conclusions are given in Section 5.

## 2. BACKGROUND OF ALARM FLOOD ANALYSIS

In this section, we briefly introduce the data format and several key procedures: chattering removing, alarm flood extraction, pattern matching, clustering, and pattern mining of alarm flood analysis.

### 2.1 Alarm message log

In a DCS (Distributed Control System) engaged plant, historical alarms with information such as tag name, tag identifier, time stamp, and priority are recorded in the database. To facilitate data analysis, we combine the tag name and the tag identifier of an alarm by adding a dot in between, e.g., Tag1.PVLO. Table 1 shows an example of alarm message log consists of only the tag name, tag identifier, and time stamp information; other attributes such as priority have been eliminated since they are not used in this study.

Table 1. An alarm message log example

| Time stamp | Tag name & identifier |
|---|---|
| 2013-07-31 18:33 | Tag1.PVLO |
| 2013-07-31 18:33 | Tag2.OFFNORM |
| 2013-07-31 18:34 | Tag8.BADPV |
| 2013-07-31 18:38 | Tag4.PVLO |

### 2.2 Alarm flood analysis procedures

Firstly, chattering alarms are removed from the alarm message log by applying delay timers to each tag. Then, alarm floods are extracted based on the threshold alarm rate suggested by the ISA standard, 10 alarms per 10 minutes. Figure 1 shows an example of removing chattering alarms and extracting alarm floods; respectively, the blue and red lines show the alarm rate before and after chattering alarms are removed; and the horizontal black line shows the alarm rate threshold for extracting alarm floods.

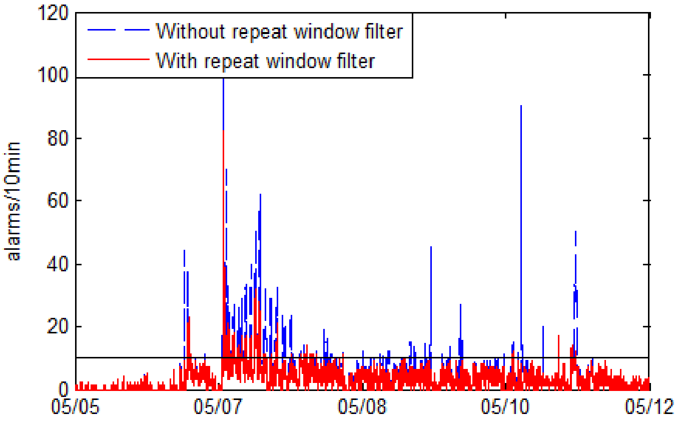

Fig. 1. Example of removing chattering alarms and extracting alarm floods

Next, the pattern matching algorithm proposed in Cheng et al. (2013) is used to align the extracted alarm floods pairwisely and obtain the similarity scores. Then, clustering is carried out based on these scores to group the similar alarm floods together. Figure 2 shows the result of applying the pattern matching and clustering algorithms on the extracted alarm floods from Figure 1.

Finally, the proposed algorithm is applied to align the multiple alarm flood sequences contained in each cluster and find the common patterns.

## 3. PRINCIPLE OF THE NEW ALGORITHM

The purpose of the proposed algorithm is to find an optimal alignment of multiple alarm flood sequences so that the common pattern can be obtained from this alignment. In the following part, we will introduce the algorithm that aligns three alarm flood sequences, the idea of which can be applied to the cases of aligning more sequences.

### 3.1 Problem description

Consider three sequences:

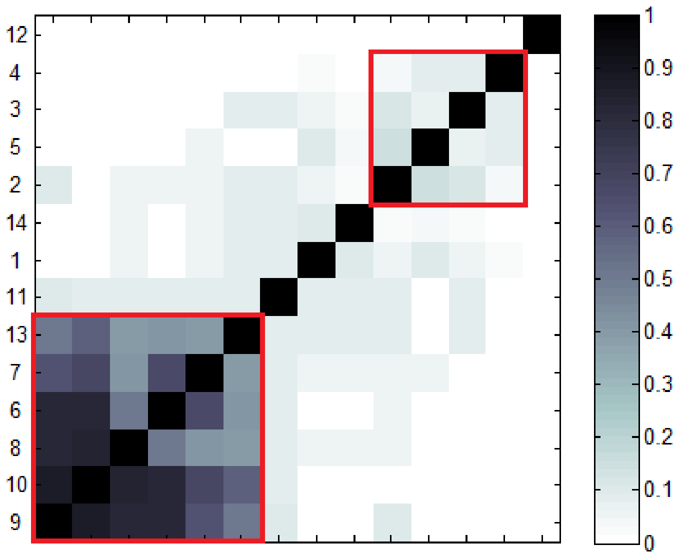

Fig. 2. Result of applying pattern matching and clustering algorithms on the extracted alarm floods

$$\begin{aligned}A &= \;<(e_{11},t_{11}),(e_{12},t_{12}),...,(e_{1m},t_{1m}),...,(e_{1M},t_{1M})>,\\ B &= \;<(e_{21},t_{21}),(e_{22},t_{22}),...,(e_{2n},t_{2n}),...,(e_{2N},t_{2N})>,\\ C &= \;<(e_{31},t_{31}),(e_{32},t_{32}),...,(e_{3o},t_{3o}),...,(e_{3O},t_{3O})>,\end{aligned}$$

where $e_{1m}$, $e_{2n}$, $e_{3o} \in \Sigma$, and $\Sigma = \{1, 2, ..., K\}$ is the set of different alarm types; $t_{1m}, t_{2n}$ and $t_{3o}$ are time stamps. The goal of the algorithm to find the optimal local alignment, e.g.,

$$\begin{array}{llllll} <(e_{16},t_{16}), & (e_{17},t_{17}), & (e_{18},t_{18}), & [\;] & ,(e_{19},t_{19})>, \\ <(e_{22},t_{22}), & [\;] & ,(e_{23},t_{23}), & (e_{24},t_{24}) & ,(e_{25},t_{25})>, \\ <\quad [\;] & ,(e_{31},t_{31}), & (e_{32},t_{32}), & [\;] & ,(e_{33},t_{33})>, \end{array}$$

where"[ ]" are the inserted gaps.

### 3.2 Time distance and weight vectors

A "time distance vector" and a "time weight vector" are defined, same as in Cheng et al. (2013), to consider time information during the alignment. A time distance vector for an alarm message $(e_m, t_m)$ is defined as:

$$\mathbf{d}_m = [d_m^1, d_m^2, ..., d_m^k, ..., d_m^K],$$
$$d_m^k = \begin{cases} \min\limits_{1\le i\le M} \{|t_m - t_i| : e_i = k\}, & \text{if the set is not empty}\\ \infty, & \text{otherwise,}\end{cases} \tag{1}$$

which calculates the shortest time distance between $(e_m, t_m)$ and other types of alarm messages in the sequence. A time weight vector for $(e_m, e_m)$ is defined as:

$$\begin{aligned}\mathbf{w}_m &= [w_m^1, w_m^2, ..., w_m^k, ..., w_m^K]\\ &= [f(d_m^1), f(d_m^2), ..., f(d_m^k), ..., f(d_m^K)],\end{aligned} \tag{2}$$

where $f(\cdot) : \mathbb{R} \to \mathbb{R}$ is a time weighting function. Two weighting functions are chosen as follows:

$$f_1(x) = e^{-x^2/2\sigma^2}, \tag{3}$$

$$f_2(x) = \begin{cases} 1, & \text{if } x = 0\\ 0, & \text{if } x \ne 0,\end{cases} \tag{4}$$

where the first function normalizes the time distance to $[0, 1]$, and the second function converts the time distance to the binary values 0 and 1. Detailed reasons for choosing these two functions have been described in Cheng et al. (2013).

### 3.3 Calculation of similarity scores

Two kinds of scoring functions are defined to make the algorithm capable of aligning three sequences.

The *2-way similarity scoring function* is

$$\begin{aligned}&S((e_a,t_a),(e_b,t_b))\\ &= \max\{S_0((e_a,t_a),(e_b,t_b)),\\ &\qquad S_0((e_b,t_b),(e_a,t_a))\} \times (1-\mu) + \mu,\end{aligned} \tag{5}$$

where

$$S_0((e_a,t_a),(e_b,t_b)) = \max_{1\le k\le K} [w_a^k \times w_b^k]. \tag{6}$$

$(e_a, t_a)$ and $(e_b, t_b)$ are alarm messages from sequences $A$ and $B$. The time weight vectors of $(e_a, t_a)$ and $(e_b, t_b)$ are obtained by applying $f_1(\cdot)$ and $f_2(\cdot)$ respectively. The larger value between $S_0((e_a, t_a), (e_b, t_b))$ and $S_0((e_b, t_b), (e_a, t_a))$ is used to compute $S((e_a, t_a), (e_b, t_b))$. The negative parameter $\mu$ is the mismatch penalty.

The *3-way similarity scoring function* is

$$\begin{aligned}&S((e_a,t_a),(e_b,t_b),(e_c,t_c))\\ &= S_0((e_a,t_a),(e_b,t_b),(e_c,t_c)) \times (1-2\mu) + 2\mu,\end{aligned} \tag{7}$$

where

$$\begin{aligned}&S_0((e_a,t_a),(e_b,t_b),(e_c,t_c))\\ &= \max\left\{\begin{aligned}&\frac{S_0((e_b,t_b),(e_a,t_a)) + S_0((e_c,t_c),(e_a,t_a))}{2},\\ &\frac{S_0((e_a,t_a),(e_b,t_b)) + S_0((e_c,t_c),(e_b,t_b))}{2},\\ &\frac{S_0((e_a,t_a),(e_c,t_c)) + S_0((e_b,t_b),(e_c,t_c))}{2}\end{aligned}\right\}.\end{aligned} \tag{8}$$

$(e_a, t_a)$, $(e_b, t_b)$ and $(e_c, t_c)$ are alarm messages from sequences $A$, $B$, and $C$. The value of the 3-way similarity score is obtained by averaging 2-way similarity scores.

### 3.4 Dynamic programming

As shown in Figure 3, the score (red dot) is obtained from seven candidates (blue dots) in each step of dynamic programming. The following is the governing equation for calculating similarity index:

$$\begin{aligned}&H_{m+1,n+1,o+1}\\ &= \max_{1\le i\le M, 1\le j\le N, 1\le g\le O} (I(A_{i:m}, B_{j:n}, C_{g:o}), 0)\\ &=\\ &\max\{H_{m+1,n+1,o} + 2\delta, H_{m+1,n,o+1} + 2\delta, H_{m,n+1,o+1} + 2\delta,\\ &\qquad H_{m,n,o+1} + \delta + S((e_{m+1},t_{m+1}),(e_{n+1},t_{n+1})),\\ &\qquad H_{m,n+1,o} + \delta + S((e_{m+1},t_{m+1}),(e_{o+1},t_{o+1})),\\ &\qquad H_{m+1,n,o} + \delta + S((e_{n+1},t_{n+1}),(e_{o+1},t_{o+1})),\\ &\qquad H_{m,n,o} + S((e_{m+1},t_{m+1}),(e_{n+1},t_{n+1}),(e_{o+1},t_{o+1})),\\ &\qquad 0\},\end{aligned} \tag{9}$$

where $I(A_{i:m}, B_{j:n}, C_{g:o})$ is the similarity index for the segments $(A_{i:m}, B_{j:n}, C_{g:o})$, and $\delta$ is a negative parameter for gap penalty. Initial values of $H_{0,n,o}$, $H_{m,0,0}$ and $H_{0,0,0}$ are all set to be 0.

Apply Equation (9) iteratively until all the similarity indices are obtained. Then, start back tracking as shown in Figure 4. Starting from the position (red dot) of the maximum similarity index, following the arrows pointing

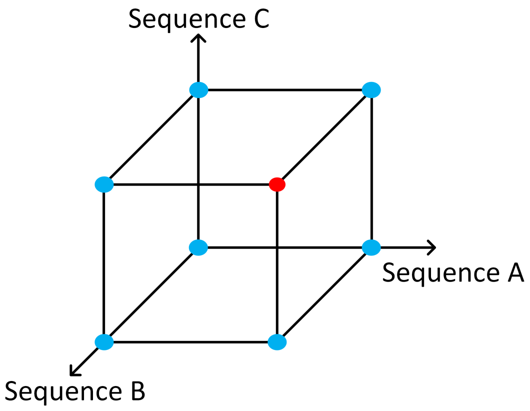

Fig. 3. Illustration of similarity score calculation for three sequences case

to the positions from which the current similarity index is obtained, the optimal alignment of the three sequences can be written out based on the changes of the position index along the path; insert a gap in the alignment if the position index remains the same on that dimension; write out the corresponding alarm message if the position index changes on that dimension.

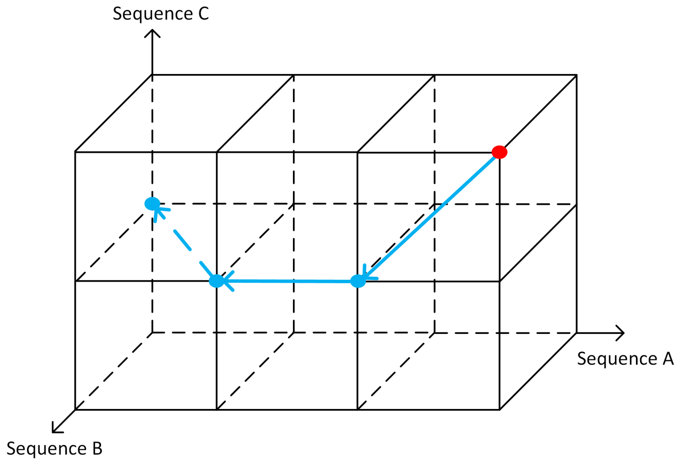

Fig. 4. An example of back tracking of a case with three alarm flood sequences

## 4. INDUSTRIAL CASE STUDY

The effectiveness of the proposed algorithm for aligning three alarm flood sequences has been tested on a dataset from an actual chemical plant. In order to eliminate the patterns formed by repeating alarms, off-delay timers of 300 seconds had been applied. Alarm floods were then extracted based on the ISA standard. General descriptions of the dataset can be found in Table 2.

Next, pairwise similarity scores between the extracted alarm floods were calculated using the algorithm in Cheng et al. (2013); and clustering was carried out based on these scores. Figure 5 shows the part of clustering result that had been used in this study; each square in the figure represents the similarity score between a corresponding pair of alarm flood sequences; the darker a square is, the higher the similarity score is.

Table 2. General descriptions of the dataset

| Description | Number |
|---|---|
| Total time period | 336 days |
| Total number of tags | 1502 |
| Total number of alarms | 109393 |
| Average alarm rate | 14/h |
| Highest peak alarm rate | 334/10 min |
| Number of alarm floods | 359 |
| Average length of alarm floods | 39 |

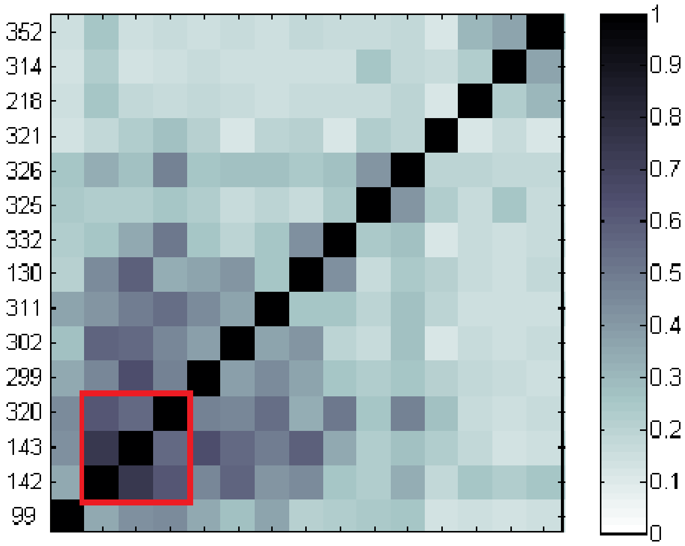

Fig. 5. Part of the clustering result of extracted alarm floods

As shown by the red box in Figure 5, a cluster of three alarm floods, with indices 142, 143, and 320, were selected to be the testing sequences; the lengths of the three sequences were 13, 16, and 12 respectively. The proposed algorithm for pattern mining in three sequences was applied with parameters set as: $\sigma = 0.2$, $\mu = -1$, and $\delta = -1$. On a 64 bit Windows PC with Intel(R) Core(TM) i7-4770 3.40GHz CPU and 24.0 GB memory, the algorithm took only 0.9 seconds to finish and the result is shown in Table 3.

One can notice that the orders of the alarm messages in the alignment are not the same; that is because the algorithm allowed some extent of swaps between alarms messages in the alignment when they were raised closely with each other. This type of complicated alignment is hard to be achieved manually even by an expert; however, the proposed algorithm was able to provide the result accurately and almost immediately. In addition, as revealed by the time stamps, the three alarm floods occured during Oct, Dec, and Feb respectively; and the priorities (not listed here in order to save space) of most of the alarms in the three sequences had been configured as “High”. Thus, the pattern obtained from the alignments could be valuable for predictive alarming and operator training.

Table 3. Alignment result of the three sequences in the cluster

| Flood 142 | Flood 143 | Flood 320 |
|---|---|---|
| Tag593.PVHH<br>27-Oct-2013 16:58:28 | Tag662.PVHH<br>26-Feb-2014 00:18:12 | Tag593.PVHH<br>16-Dec-2013 21:30:16 |
| Tag662.PVHH<br>27-Oct-2013 16:58:29 | Tag593.PVHH<br>26-Feb-2014 00:18:14 | Tag662.PVHH<br>16-Dec-2013 21:30:18 |
| Tag598.PVHH<br>27-Oct-2013 16:58:31 | Tag598.PVHH<br>26-Feb-2014 00:18:18 | Tag598.PVHH<br>16-Dec-2013 21:30:21 |
| []<br>[] | []<br>[] | Tag163.OFFLINE<br>16-Dec-2013 21:30:27 |
| Tag71.PVLO<br>27-Oct-2013 16:58:45 | Tag71.PVLO<br>26-Feb-2014 00:18:30 | Tag71.PVLO<br>16-Dec-2013 21:30:33 |
| Tag403.NORM<br>27-Oct-2013 16:58:56 | []<br>[] | Tag403.NORM<br>16-Dec-2013 21:30:36 |
| Tag407.NORM<br>27-Oct-2013 16:59:00 | Tag407.NORM<br>26-Feb-2014 00:19:08 | Tag407.NORM<br>16-Dec-2013 21:30:42 |
| Tag408.NORM<br>27-Oct-2013 16:59:02 | []<br>[] | Tag408.NORM<br>16-Dec-2013 21:30:49 |
| Tag427.OFFLINE<br>27-Oct-2013 16:59:09 | Tag427.OFFLINE<br>26-Feb-2014 00:19:17 | Tag427.OFFLINE<br>16-Dec-2013 21:30:51 |
| Tag1457.OFFNORM<br>27-Oct-2013 17:01:09 | Tag1457.OFFNORM<br>26-Feb-2014 00:21:48 | []<br>[] |

## 5. CONCLUSIONS

In this paper we proposed an algorithm to find the optimal alignment of multiple alarm flood sequences so that based on this alignment a common pattern can be obtained thereafter. The algorithm is an extension of the one in Cheng et al. (2013) to the case of aligning multiple sequences. It is one of the key techniques in alarm flood analysis, together with delay timers, pattern matching algorithms, and clustering methods. In the future, pruning techniques to reduce the algorithm complexity and the impact of parameter tuning on the alignment result may be studied.

## ACKNOWLEDGEMENTS

The authors benefited from discussions with John MacGowan and Aris Espejo of Syncrude Canada Limited.

## REFERENCES

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering, 1995*, 3–14.

Ahmed, K., Izadi, I., Chen, T., Joe, D., and Burton, T. (2013). Similarity analysis of industrial alarm flood data. *IEEE Transactions on Automation Science and Engineering*, 10(2), 452–457.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D.J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3), 403–410.

Cheng, Y., Izadi, I., and Chen, T. (2013). Pattern matching of alarm flood sequences by a modified smith-waterman algorithm. *Chemical Engineering Research and Design*, 91(6), 1085–1094.

Cisar, P., Hostalkova, E., and Stluka, P. (2009). Data mining techniques for alarm rationalization. In *19th European Symposium on Computer Aided Process Engineering, Cracow, Poland*, 1457–1462.

Eddy, S.R. (1995). Multiple alignment using hidden markov models. In *ISMB-95 Proceedings*, volume 3, 114–120.

Eddy, S.R. (1998). Profile hidden markov models. *Bioinformatics*, 14(9), 755–763.

EEMUA (2007). *Alarm Systems - A Guide to Design, Management and Procurement.*

Feng, D.F. and Doolittle, R.F. (1987). Progressive sequence alignment as a prerequisitet to correct phylogenetic trees. *Journal of Molecular Evolution*, 25(4), 351–360.

Folmer, J. and Vogel-Heuser, B. (2012). Computing dependent industrial alarms for alarm flood reduction. In *9th International Multi-Conference on Systems, Signals and Devices (SSD)*, 1–6.

Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. In *ACM SIGMOD Record*, volume 29, 1–12.

ISA (2009). *Management of Alarm Systems for the Process Industries.*

Izadi, I., Shah, S.L., Shook, D.S., Kondaveeti, S.R., and Chen, T. (2009). A framework for optimal design of alarm systems. In *Proceedings of 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 651–656.

Johnson, M.S. and Doolittle, R.F. (1986). A method for the simultaneous alignment of three or more amino acid sequences. *Journal of Molecular Evolution*, 23(3), 267–278.

Kondaveeti, S.R., Izadi, I., Shah, S.L., and Black, T. (2010). Graphical representation of industrial alarm data. In *Analysis, Design, and Evaluation of Human-Machine Systems*, volume 11, 181–186.

Kondaveeti, S.R., Izadi, I., Shah, S.L., Shook, D.S., Kadali, R., and Chen, T. (2013). Quantification of alarm chatter based on run length distributions. *Chemical Engineering Research and Design*, 91(12), 2550–2558.

Kordic, S., Lam, P., Xiao, J., and Li, H. (2008). *Analysis of Alarm Sequences in a Chemical Plant.*

Needleman, S.B. and Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3), 443–453.

Pearson, W.R. and Lipman, D.J. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8), 2444–2448.

Smith, T.F. and Waterman, M.S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1), 195–197.

Thompson, J.D., Higgins, D.G., and Gibson, T.J. (1994). Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22), 4673–4680.

Zaki, M.J. (1998). Efficient enumeration of frequent sequences. In *Proceedings of the 7th International Conference on Information and Knowledge Management*, 68–75.