

# Duration-Aware Alignment of Process Traces

Sen Yang<sup>1</sup>, Moliang Zhou<sup>1</sup>, Rachel Webman<sup>2</sup>, JaeWon Yang<sup>2</sup>,  
Aleksandra Sarcevic<sup>3</sup>, Ivan Marsic<sup>1</sup> and Randall S. Burd<sup>2</sup>

<sup>1</sup> Department of Electrical and Computer Engineering, Rutgers University,  
Piscataway, New Jersey, United States

<sup>2</sup> Division of Trauma and Burn Surgery, Children’s Nat’l Medical Center,  
Washington, D.C., United States

<sup>3</sup> College of Information Science and Technology, Drexel University,  
Philadelphia, Pennsylvania, United States

**Abstract.** Objective: To develop an algorithm for aligning process traces that considers activity duration during alignment and helps derive data-driven insights from workflow data. Methods: We developed a duration-aware trace alignment algorithm as part of a Java application that provides visualization of the alignment. The relative weight of the activity type vs. activity duration during the alignment is an adjustable parameter. We evaluated proportional and logarithmic weights for activity duration. Results: We used duration-aware trace alignment on two real-world medical datasets. Compared with existing context-based alignment algorithm, our results show that duration-aware alignment algorithm achieves higher alignment accuracy and provides more intuitive insights for deviation detection and data visualization. Conclusion: Duration-aware trace alignment improves upon an existing trace alignment approach and offers better alignment accuracy and visualization.

## 1 Introduction

Many contemporary information systems record activity logs, including online shopping habits, personal calendars and electronic health records (EHR). Process mining techniques aim to extract knowledge and insights from these types of logs [1]. Most research in process mining has focused on workflow discovery using process model analysis (e.g., conformance checking [1]) and deviation discovery based on these models. Bose and Van der Aalst [2] derived a trace alignment from the multiple sequences alignment (MSA) algorithm in bioinformatics [3] and used it to gain insights from activity logs. Trace alignment places the same or similar activities in the same column of the alignment matrix. If a matching activity cannot be found, a gap symbol “-” is inserted. For example, given a trace with activities {a,c} and a trace with activities {a,b,a,c}, two possible alignments are:

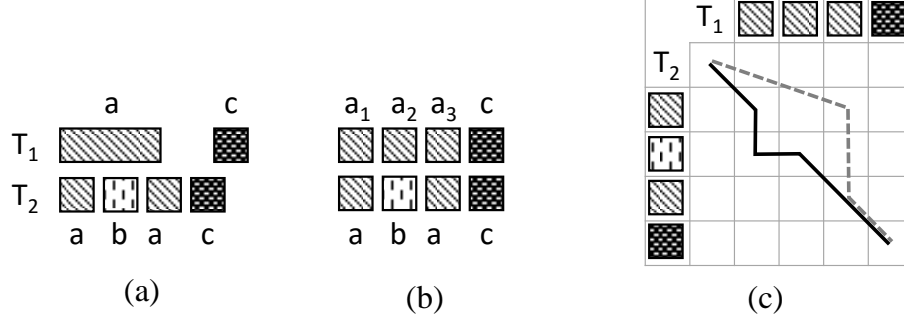
$$\begin{bmatrix} a & - & c \\ a & b & a & c \end{bmatrix} \text{ and } \begin{bmatrix} - & - & a & c \\ a & b & a & c \end{bmatrix}$$

Trace alignment can be used to find similarities within a group of traces and to determine how a given trace differs from the well-established work practice. Trace

alignment can also help identify conserved patterns of activities and deviations from the norm.

Existing trace alignment approaches consider only the sequential order of activities and ignore activity duration. Activity duration may, however, be an important parameter for some processes. For example, consideration of duration helped understand that nurses routinely switch between tasks, spending less time on interrupted tasks that are later resumed [21]. To consider activity duration during alignment when finding a matching activity in another trace, the duration may need to be part of similarity calculation. Activities of the same type and comparable durations should be assigned high similarity. Markedly different durations of the same activity type may indicate deviations and can be expected to have a lower similarity [23]. Activity duration can also help improve alignment accuracy by identifying deviations in the process execution. Activities with a typical duration suggest normal operation while activities with unusual durations indicate difficulties or atypical performance [24]. Additionally, activity duration may be linked with activity importance, i.e. task importance can vary based on situation and may require more time and effort when more integral to the process. Finally, visualizing the trace alignment result incorporating durations helps users detect duration-related deviations in activity performance that otherwise would not be evident.

To consider activity duration as a parameter in trace alignment, we extended the classic Needleman Wunsch algorithm (NW) [4] used for aligning biological sequences, to include a cost for activity. Because a process execution can be considered as a sequential data on the timeline, one may expect that the alignment of traces with activity durations can be solved by Dynamic Time Warping (DTW) [5], an algorithm used in signal processing and pattern recognition to align time-series sequences. The standard DTW algorithm, however, does not explicitly penalize the difference in activity durations between aligned activities of the same type. A variation of DTW called Distortion Penalized DTW or Variable Penalty DTW [8][25], was developed to address this issue by introducing the time-distortion penalty to penalize the expansion and contraction of the original sequence. Our preliminary experimental results, however, showed that even the distortion-penalized DTW could not produce the alignment results as expected. To perform the distortion-penalized DTW, we discretized the time axis and displayed each activity as function of time over one or more time steps. This uniform slicing produces multiple contiguous segments for each activity, each segment being one unit of time long, that are treated as independent points during the DTW time-warping. Consider a trace  $T_1$  with activities  $\{a, c\}$  where  $a$  lasted three time units and  $c$  lasted one time unit, and another trace  $T_2$  with activities  $\{a, b, a, c\}$  where all lasted one time unit (Fig. 1(a)). After time discretization, activity  $a$  in  $T_1$  is split in three discrete-time segments  $\{a_1, a_2, a_3\}$  (Fig. 1(b)). Alignment of segments from one trace is attempted with segments from the other trace, without considering that adjacent segments may be part of the same activity. As a result, segments of one long-duration activity in one trace may be aligned with several short-duration activities in the other trace. In our example, after the alignment, the first and third segments of activity  $a$  in  $T_1$  will be separately aligned with the two different activities of type  $a$  in  $T_2$ . This observation occurs because the minimum warping distance results from warping around activity  $b$  in  $T_2$  rather than keeping  $a_1, a_2$ , and  $a_3$  together (Fig. 1(c)) [5][7]. Our modified trace alignment algorithm addresses this problem by introducing a time-scale



**Fig. 1.** Example of aligning two traces using distortion-penalized DTW. (a) Original traces with activity durations. (b) Representing activities on discrete timeline where activity  $a$  in  $T_1$  is three time units long. (c) Alignment of the discrete time-points of the two traces. An optimal path found by the distortion-penalized DTW is shown by the solid line, which does not keep together the segments of activity  $a$  in  $T_1$ . The expected alignment is shown by the dashed line.

distortion penalty into the alignment algorithm. In addition to modifying the trace alignment algorithm, we also customized several existing metrics for evaluating alignment algorithms, such as the sum-of-pairs score for evaluating alignment accuracy [17].

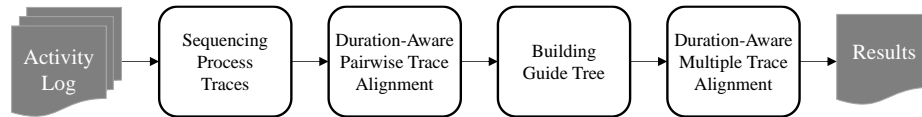
The rest of the paper is organized as follows. Section 2 introduces our duration-aware trace alignment algorithm. Section 3 evaluates the performance of this algorithm and compares the results to previous trace alignment work using two case studies. Section 4 concludes the paper and presents the limitations of our current work.

## 2 Methodology

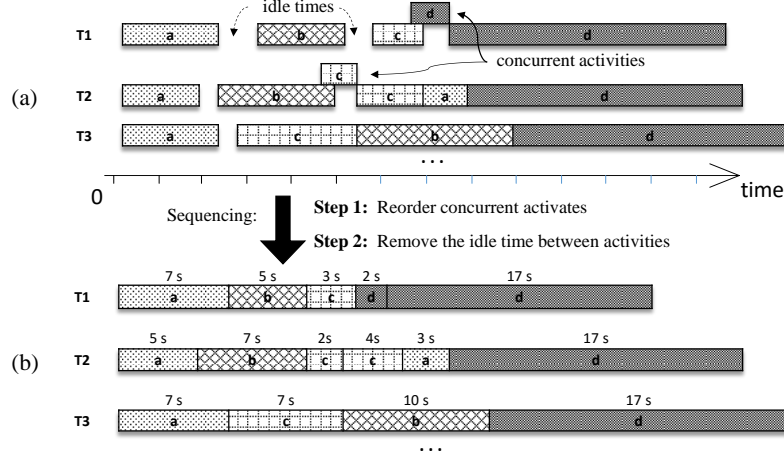
Our duration-aware trace alignment algorithm (Fig. 2) works by: (1) sequencing of traces with concurrent activities, (2) computing duration-aware pairwise alignment, (3) building a guide tree, and (4) performing progressive alignment of multiple traces.

### 2.1 Sequencing of Process Traces

Activities in a process mining dataset (or activity log) are usually coded with timestamps indicating the start time and end time for each activity. Idle time may exist



**Fig. 2.** Duration-aware trace alignment flowchart.



**Fig. 3.** Two steps of sequencing the traces with concurrent activities (such as  $d$  in  $T_1$  and  $c$  in  $T_2$ ) and idle times (white spaces in all three traces). (a) Example process traces before sequencing. (b) The same process traces after sequencing.

between activities and some activities may be executed concurrently with each other (Fig. 3(a)), e.g. a typist may be typing and reading concurrently. Because the trace alignment algorithm only works with linear sequences as input, the original traces need to be converted to linear sequences. In process mining, trace sequencing is performed by putting activities in ascending order of their start time (Fig. 3(b)).

## 2.2 Duration-Aware Pairwise Trace Alignment

Consider example traces  $T_1 = \{abcdd\}$  and  $T_2 = \{abccad\}$ . Example alignment is:

	1	2	3	4	5	6	7
$T_1$ :	a	b	c	-	-	d	d
$T_2$ :	a	b	c	c	a	d	-

The alignment includes three types of operations: substitution, insertion and deletion. The insertion and deletion operations are usually referred as indel operations because an insertion could be considered a deletion in another trace. For example, columns 1, 2, 3 and 6 show that substituting an activity in one trace with the same type of activity in the other trace; columns 4 and 5 show insertions of activities in  $T_2$  that do not exist in  $T_1$ ; column 7 shows deletion of a  $T_1$  activity in  $T_2$ . Substitution, insertion and deletion operations have associated costs, which are represented by a “scoring scheme” where the score is inversely proportional to the cost.

Pairwise sequence alignment is usually solved using the Needleman Wunsch (NW) algorithm [4], and a variation of the classic NW was adopted for trace alignment [2]. We refer to this alignment as “context-based alignment” because the scores are derived from the sequential order of adjacent activities, i.e., the process context.

Both the NW alignment and context-based alignment algorithm align sequences without considering activity duration. To incorporate time information into alignment calculation, we introduce a time-scale distortion penalty into the trace alignment algorithm, which originates from Dynamic Time Warping [8]. The time distortion penalty is applied whenever a sequence is expanded or contracted during alignment.

The accumulated score matrix for our duration-aware trace alignment is defined as:

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + S(T_1(i), T_2(j)) * ddp(T_1(i), T_2(j)) & \text{Substitute} \\ F(i-1, j) + g * hdp(T_1(i)) & \text{Insert} \\ F(i, j-1) + g * vdp(T_2(j)) & \text{Delete} \end{cases} \quad (1)$$

where  $S(T_1(i), T_2(j))$  is the score for substituting element  $T_1(i)$  with  $T_2(j)$ , and  $g$  is the score for indel operations.  $ddp(x, y)$ ,  $hdp(x)$ , and  $vdp(x)$  are time-scale distortion-penalty functions. The initial conditions are:  $F(0, 0) = 0$ ,  $F(i, 0) = F(i-1, 0) + g * hdp(T_1(i))$ ,  $F(0, j) = F(0, j-1) + g * vdp(T_2(j))$ . The diagonal distortion penalty ( $ddp$ ) penalizes the time distortion generated in the substitution operation. The horizontal distortion penalty ( $hdp$ ) penalizes the time distortion generated in the horizontal direction, viz. deletion operation. The vertical distortion penalty ( $vdp$ ) penalizes the time distortion generated in the vertical direction, viz. insertion operation. These three distortion penalty functions are defined as:

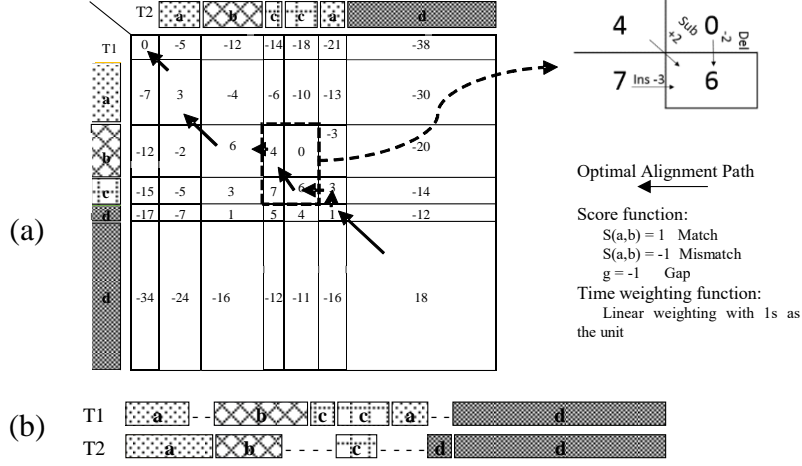
$$hdp(T_1(i)) = \varphi(d(T_1(i))) \quad (2)$$

$$vdp(T_2(j)) = \varphi(d(T_2(j))) \quad (3)$$

$$ddp(T_1(i), T_2(j)) = \begin{cases} \min(\varphi(d(T_1(i))), \varphi(d(T_2(j)))) - |\varphi(d(T_1(i))) - \varphi(d(T_2(j)))|, & S(T_1(i), T_2(j)) \geq 0 \\ \varphi(d(T_1(i))) + \varphi(d(T_2(j))), & S(T_1(i), T_2(j)) < 0 \end{cases} \quad (4)$$

where  $\varphi(t)$  is defined as the time-weighting function used to control the influence of activity duration, and  $d(activity)$  is the duration of an activity. The duration unit is the same as the unit used for recording the activity. For  $ddp$ , we need to consider two scenarios. First,  $S(T_1(i), T_2(j)) > 0$  means a “match” between activities  $T_1(i)$  and  $T_2(j)$ , i.e. activities are of the same type or substitutable. In this scenario,  $ddp$  is decided by rewarding the extent to which the durations overlap and penalizing the extent to which the durations differ. Second,  $S(T_1(i), T_2(j)) < 0$  means a “mismatch” between activities  $T_1(i)$  and  $T_2(j)$ , i.e.  $T_1(i)$  and  $T_2(j)$  are incompatible and should not be aligned. Instead, this substitution operation should be decomposed to a deletion and an insertion. In this scenario,  $ddp$  equals to the sum of  $hdp$  and  $vdp$ .

Our group and others (e.g. [21]) have observed that activity duration has a distribution where extremely long-duration activities occur rarely. These rare long duration activities, however, can have a significant effect on the distortion penalty. We use time weighting to control the influence of activity duration. Choosing a proper time-weighting method is critical for the performance of our algorithm. We analyzed two weighting methods: linear and logarithmic weighting:



**Fig. 4.** Pairwise alignment with linear duration-weighting function of traces  $T_1$ ,  $T_2$  from Fig. 3. (a) Scoring matrix for the alignment. (b) The alignment result.

$$\begin{cases} \varphi_{Linear}(d(event)) = c * d(event) \\ \varphi_{Log}(d(event)) = \log_b(d(event)) \end{cases} \quad (5)$$

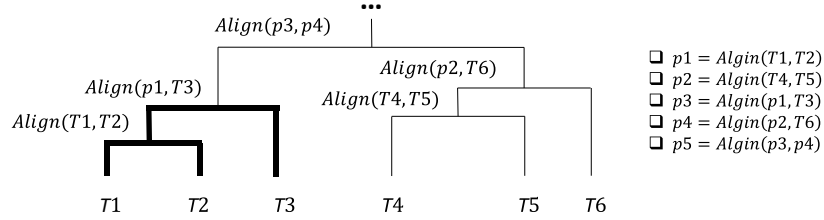
where  $c = 1$  and  $b = e$  (base of natural logarithm). Linear weighting preserves the original duration information, but a problem arises when the difference between activity durations is large. The duration becomes the driving factor for alignment and the trace context is mostly ignored. In other words, long-duration activities “overpower” short-duration activities and dominate the alignment. Logarithmic weighting can mitigate this problem.

The scoring scheme, or matrix that specifies the cost of the three operations during alignment, used in score matrix  $F$  can be customized based on domain knowledge. If domain knowledge is unavailable, a simple choice would be the unit score scheme (match = 1, mismatch = -1, indel = -1).

Once the score matrix  $F(i, j)$  is constructed, the optimal alignment can be deduced by tracing back from lower right corner of the score matrix to the upper left corner choosing the neighboring cell that leads to the maximum score at each step (Fig. 4).

### 2.3 Building the Guide Tree

A guide tree needs to be constructed to determine the order of trace pairs to be aligned in the progressive iteration of multiple traces. Pairwise alignment is then performed from guide tree’s leaves to the root. In our approach, the guide tree is generated based on the hierarchical clustering algorithm (AHC) [9] with Ward’s method [10][11] (Fig. 5). To measure the proximity of traces, we define a new distance named “Duration-Aware Edit Distance”, which is derived from Edit Distance [12] (also called Levenshtein Distance) and includes dissimilarity between activity durations. Given two traces  $\alpha$  and  $\beta$ , their duration-aware edit distance  $t_{\alpha,\beta}$  can be calculated progressively as:



**Fig. 5.** Guide tree that involves traces T1, T2 and T3 (thick lines) from Fig. 3.

$$t_{\alpha, \beta}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0 \\ \min \begin{cases} t_{\alpha, \beta}(i-1, j-1) + \begin{cases} d_{\alpha}(i) + d_{\beta}(j) & \text{if } \alpha_i \neq \beta_j \\ |d_{\alpha}(i) - d_{\beta}(j)| & \text{if } \alpha_i = \beta_j \end{cases} \\ t_{\alpha, \beta}(i, j-1) + d_{\beta}(j) \\ t_{\alpha, \beta}(i-1, j) + d_{\alpha}(i) \end{cases} & \text{otherwise} \end{cases} \quad (6)$$

where  $d_{\alpha}(i)$  and  $d_{\beta}(j)$  are the duration of  $i^{\text{th}}$  activity in  $\alpha$  and  $j^{\text{th}}$  activity in  $\beta$ .

## 2.4 Duration-Aware Multiple Trace Alignment

Multiple trace alignment is essentially a progressive pairwise alignment process that aligns pairs of individual traces, as well as a trace and a *profile* (a temporary alignment result at intermediate stage of the alignment process) (**Error! Reference source not found.**) or two profiles. Because a column in a profile contains more than one activity, the substitution score in multiple sequence alignment is redefined as:

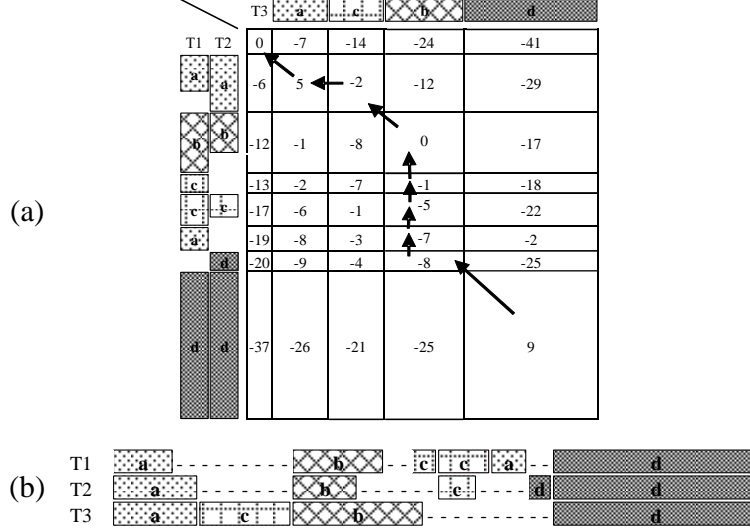
$$\mathcal{S}(C_A^i, C_B^j) = \sum_{a, b \in \mathbb{A}} n_A^i(a) \cdot n_B^j(b) \cdot S(a, b) \cdot \overline{ddp(a, b)} \quad (7)$$

where  $S(a, b)$  denotes the substitution score of activities  $a, b$ ;  $\mathbb{A}$  denotes the activity set of activity log;  $n_A^i(a)$  denotes the frequency of activity  $a$  in the column  $i$  of profile  $A$ ;  $C_A^i$  denotes the contents of  $i^{\text{th}}$  column of profile  $A$ .  $\overline{ddp(a, b)}$  denotes the

distortion penalty calculated based on the average duration of activity types  $a$  and  $b$ . Similarly, the indel score of a column is redefined as:

$$\begin{cases} \mathcal{I}(C_A^i) = \sum_{a \in \mathbb{A}} f_A^i(a) \cdot g \cdot \overline{hdp(a)} & \text{deletion} \\ \mathcal{I}(C_A^i) = \sum_{a \in \mathbb{A}} f_A^i(a) \cdot g \cdot \overline{vdp(a)} & \text{insertion} \end{cases} \quad (8)$$

where  $g$  is the indel score of activity  $a$ ;  $f_A^i(a)$  denotes the frequency of  $a$  in column  $i$  of profile  $A$ ;  $\overline{hdp(a)}$  and  $\overline{vdp(a)}$  are the horizontal or vertical distortion penalties calculated based on the average duration of a deleted or inserted activity.



**Fig. 6.** Duration-aware multiple trace alignment with linear weighting of traces  $T_1$ ,  $T_2$ ,  $T_3$  from Fig. 3. (a) Scoring matrix for the alignment. (b) The alignment result.

### 3 Experimental Results

We implemented our duration-aware trace alignment algorithm in the Java programming language. Due to the space limitations, we do not discuss the computational complexity of our algorithm, but we evaluated its performance on real logs and an artificially generated log. The results show that duration-aware trace alignment algorithm can compute and visualize an activity log of 50,000 activities ( $\sim 1,000$  traces and  $\sim 50$  activities for each trace) within  $25.5 \pm 1.5$  seconds (mean value and standard deviation over 20 different runs).

Although there are many quality-assessment approaches or metrics for biological sequence alignment (TCS [14], Heads-or-Tails [15], GUIDANCE [16], etc.), very few such metrics exist for trace alignment. We evaluated the performance of the duration-aware alignment using two case studies and compared the results with that of previous research on trace alignment.

#### 3.1 Evaluation Criteria

Our evaluation of alignment algorithm performance is based on the following criteria:

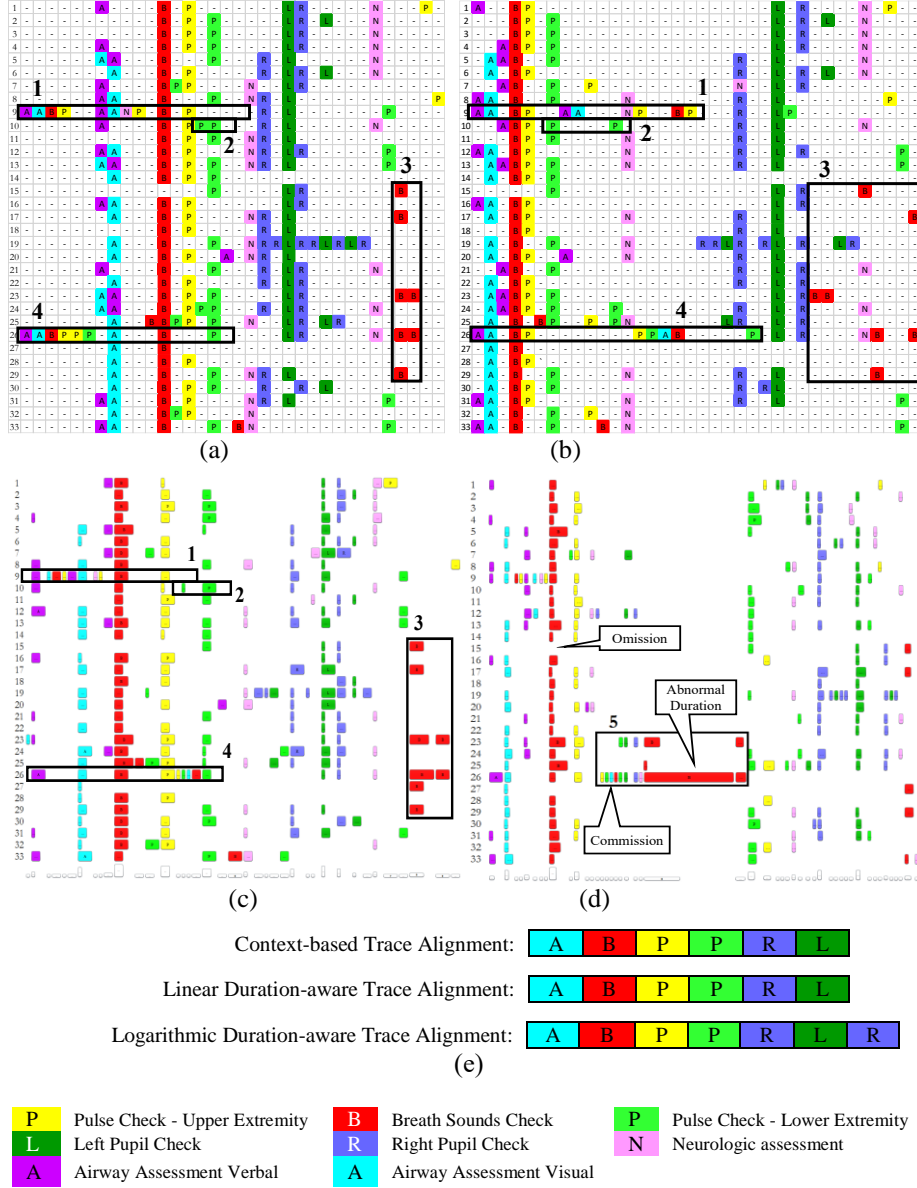
1. *Sum-of-pairs Score (SPS)*: This metric is widely used to measure the alignment accuracy of multiple sequences in bioinformatics [17]. To our knowledge, it has not been used in the context of process mining. SPS for biological sequences is defined as:  $SPS = n/N$  where  $n$  is the number of correctly aligned residue pairs found in the test alignment and  $N$  is the total number of aligned residue pairs in



the reference alignment. In the context of trace alignment, the elements are process activities instead of biological residues. Our ground-truth alignment (reference alignment) was generated by medical experts who aligned the traces manually.

2. *Average Information Score*: The information score is defined for each column of the alignment matrix as [2]:  $1 - E/E_{max}$  where  $E_{max}$  is the maximum entropy of a column, equal to  $\log_2(|\mathbb{A}| + 1)$ .  $E$  is the entropy of activities in the column, defined as:  $E = -\sum_{a \in \mathbb{A} \cup \{-\}} p_a \log_2(p_a)$  where  $a$  is an activity;  $\mathbb{A}$  is the set of activity types in the activity log; “-” denotes the gap symbol and  $p_a$  is the probability of  $a$ ’s occurrence in this column. Lower information score indicates sparser distribution and higher diversity of activities in a column. Because the purpose of alignment is to find significant information with strong confidence, high diversity of activities in one column is not expected. For this reason, higher information score indicates higher alignment quality. To reflect the quality of the whole alignment, we used the mean value of information scores of all columns.
3. *Consensus Sequence (CS)*: The concept of consensus sequence comes from bioinformatics, where it denotes a sequence of most frequent residues found in each column of the alignment. In process mining, the consensus sequence captures the most frequent activity in each column [2]. A gap could also be included in the consensus sequence if the corresponding column is mostly filled with gaps, but we are more interested in non-gap activities in the consensus sequence. The consensus sequence measures the alignment quality because good alignment algorithms should be able to discover the common activity sequences in a process.
4. *Alignment Matrix Length*: Alignment matrix length could also reflect the quality of alignment. Longer alignment matrix indicates that more gaps are introduced into the alignment, which tends to be sparse. Good alignment is expected to be dense with only necessary gaps included and unnecessary gaps avoided.
5. *Deviation Detection Ability*: A major objective of trace alignment is to help diagnose the process executions, which includes the ability to identify deviations from common practice. These deviations were previously classified into two main categories, viz. omission and commission [2]. “Omission” denotes an activity that should exist at certain position in a trace but is missing. “Commission” denotes an activity that should not exist but is inserted. In addition, in duration-aware trace alignment we are able to observe one more type of deviation, viz. “abnormal duration”. Abnormal duration is present when an activity is either much shorter or much longer when compared to other activities in the same column. We did not quantify this metric but rather illustrate it by examples.

Among these metrics, the sum-of-pairs score reflects alignment accuracy because it performs direct comparison to the reference alignment (ground truth) [17]. The average information score is associated with the alignment matrix length. Including more gaps into the alignment will not only increase the alignment matrix length, but may also increase the average information score. The reason is that the average information score increases for every column mostly filled with gaps. Based on the definition of information score, the information score of such columns will be high.



**Fig. 7.** (a) Reference alignment created by medical experts. (b) Context-based trace alignment. (c) Duration-aware trace alignment with logarithmic weights. (d) Duration-aware trace alignment with linear weights. Rows of the alignment were originally ordered according to the guide tree but we reordered them by trace-ID for easier comparison. (e) Consensus sequences of different alignments. Legend at the bottom shows the color-coding of activities. The boxes inside the alignment matrices labeled with numbers are discussed in text.

### 3.2 Case Study 1: Trauma Resuscitation Process

**Input Dataset Information.** This dataset was obtained from trauma resuscitations performed at Children’s National Medical Center in Washington, DC. We videotaped and coded the start and end times of 8 different activity types during the initial evaluation of 33 injured children [18]. Resuscitation traces contained on average 7.4 activities and lasted on average of 3.9 minutes (between 0.5 and 13 minutes). Our ground-truth reference alignment of activities was created manually by medical experts (Fig. 7(a)).

**Generating Alignment Matrix.** Context-based trace alignment (Fig. 7(b)) was generated in ProM (<http://www.promtools.org/>) using the Trace Alignment Plugin with the recommended settings. The guide tree was constructed using maximal repeats as the feature set with Euclidean distance as the proximity measure and minimum variance as the join criteria [2]. The scoring scheme was derived based on the trace context [19]. The duration-aware trace alignment results (Fig. 7(c), (d)) were computed with the following settings. The guide tree was built using duration-aware edit distance to measure the trace proximity (Equation 6) and minimum variance as the join criteria. We assume that users may not have domain knowledge for generating scoring metrics, so we adopted the unit score function (Section 2.2) as our scoring scheme for computing alignment. The alignment results (Table 1) were compared based on the criteria described in Section 3.1.

**Table 1.** Performance comparison of context-based alignments and duration-aware alignments on trauma resuscitation dataset. The best results in each row are shown in bold font.

Algorithm:	Duration-Aware	Duration-Aware	Context-
Metrics:	(linear)	(logarithmic)	based
Sum-of-pairs Score	0.617	<b>0.807</b>	0.731
Avg. Information Score	<b>0.870</b>	0.863	0.848
No. Non-gap Activities in CS	6	<b>7</b>	6
Alignment Matrix Length	49	39	<b>36</b>

**Discussion of Performance Comparison Metrics.** Our results (Table 1) show that the duration-aware trace alignment with logarithmic weights has the highest alignment accuracy with the sum-of-pairs score (0.807) based on reference alignment (Fig. 7(a)). This finding shows that activity duration can influence the alignment results and proper weighting of the duration can improve the alignment accuracy. We believe this improvement occurs for the following reasons:

1. *Repeated activities aligned based on typical duration for the column:* When faced with multiple options (because of a repeated activity), duration-aware trace alignment chooses to align the activities with similar durations. We observed that most resuscitation activities had a “typical” duration. When an activity is repeated within a trace, the instance with the typical duration was likely performed similarly as the same activity in other traces. For example, several activities in trace 9 were repeated, so different alignments are possible. Context-based alignment aligns the first set of repeated activities in trace 9 with the same activities in other traces (Fig. 7(b), box “1”), while duration-aware alignment aligns the second set of activities (Fig. 7(c), box “1”). *Medical explanation:* The second set of repeated

activities in trace 9 is more similar to the typical performance as it was done by the physician assigned to this portion of the evaluation, rather than the first set which was done by a substituting team member. Another example is the two instances of “Pulse Check-Lower Extremity” in trace 10 (Fig. 7(b) & (c), box 2). Again, context-based alignment aligns the first set, while duration-aware alignment aligns the second set. *Medical explanation:* the task “Pulse Check-Lower Extremity” was repeated when the resident prematurely advanced to the secondary survey. A senior physician intervened and instructed the resident to return to the initial assessment where the “Pulse Check-Lower Extremity” task was repeated.

2. *Better guide tree:* Another factor that influences alignment accuracy is misalignment, the incorrect positioning of activities. The possibility of misalignments increases if dissimilar traces are aligned early in the progressive alignment. These misalignments cannot be corrected later and can propagate into more alignment errors [20][22]. Because the guide tree algorithm determines the order of traces being aligned, the quality of guide tree is directly associated with the occurrence of misalignment. Compared with the feature (maximal repeats [2]) based distance, duration-aware edit distance, which is based on both activity type and activity duration, can do better in capturing the similarity of traces and producing a guide tree. As a result, misalignment is reduced and alignment accuracy improves. For example, the activity “Breath Sounds Check” in traces 15, 17, 23, 26, 29 is well aligned in duration-aware trace alignment, but it is poorly aligned in the context-based alignment (Fig. 7 (a), (b), (c), block “3”). Because of similarity, repeat performance of “Breath Sounds Check” is captured and properly weighted by our duration-aware edit distance and these five traces are aligned early in the algorithm, with a lower risk of being misaligned according to the guide tree.

In some scenarios duration-aware trace alignment may not perform well:

1. *Activity duration does not always accurately predict an anomaly:* From an activity duration perspective, the first performance of “Airway Assessment Visual and Breath Sounds Check” in 26 (Fig. 7(b) & (c), box “4”) has a more typical duration than the second performance. The reference alignment, however shows that the second performance rather than the first should be aligned. The context-based alignment also made a mistake in this case (Fig. 7(b), box “4”). *Medical explanation:* Video review showed these tasks initially being performed by a substituting clinician and the second performed by the regular team member.
2. *Long-duration activities can dominate:* A single long-duration activity might dominate over several short-duration activities which should be aligned if only activity type were considered (Fig. 7(d), box “5”). This situation explains why linear duration-aware trace alignment performs worse in some cases. This problem is mitigated by using logarithmic time weighting function. The logarithmic duration-aware alignment achieves higher accuracy than the context-based one.

Linear duration-aware alignment had the highest average information score, but it also had the longest matrix (49 columns). This matrix length is much larger than the other two alignment matrices (39 and 36 columns). The large matrix indicates that many unnecessary gaps are included into the alignment and can make visual analysis more difficult.

The consensus sequences for context-based trace alignment and the two duration-aware alignment results were similar (Fig. 7(e)). Context-based alignment had a

consensus sequence with six non-gap activities. Linear and logarithmic duration-aware trace alignment had six non-gap activities and seven non-gap activities respectively in the consensus sequence. The additional activity discovered by logarithmic duration-aware alignment is the second check of “Right Pupil”. *Medical explanation:* The reason for this interesting finding is unclear, but it may be because the examining clinicians stand on the right side of the patient and leading them to check the right pupil, then the left pupil, then the right pupil again on their way back.

**Deviation Detection.** Like the context-based trace alignment, our duration-aware algorithms can discover commission and omission deviations (Fig. 7(d)). Duration-aware alignment can also provide additional insights based on activity duration and can identify duration anomalies. For example, the third “Breath Sounds Check” in trace 26 had an abnormally long duration (Fig. 7(d)). This fact could indicate patient disease or clinician’s error. *Medical explanation:* In trace 26 the patient had an injury to the lungs and the physician spent extra time examining the chest to be sure the breath sounds were normal.

### 3.3 Case Study 2: Endotracheal Intubation Process

**Input Dataset Information.** The endotracheal intubation (breathing tube insertion) process was also reviewed using videos from Children’s National Medical. This dataset contained 31 cases with a total of 602 activities of 21 different types.

**Table 2.** Performance comparison between context-based alignment and duration-aware alignment for endotracheal intubation process. The best results in each row are shown in bold.

Algorithm:	Duration-Aware	Duration-Aware	Context-
Metrics:	(linear)	(logarithmic)	based
Sum-of-pairs Score	0.731	<b>0.843</b>	0.721
Avg. Information Score	0.918	<b>0.919</b>	0.899
No. Non-gap Activities in CS	10	12	<b>13</b>
Alignment Matrix Length	134	119	<b>115</b>

**Discussion of Performance Comparison Metrics.** The results show that logarithmic duration-aware trace alignment achieved the highest sum-of-pairs score on this dataset (Table 2). The duration-aware alignment with a logarithmic time weight function had similar alignment matrix length but higher average information score compared to context-based alignment. The context-based alignment had one more non-gap activity “Passive Oxygen Placement” in the consensus sequence than the logarithmic duration-aware alignment. The linear duration-aware alignment performed the worst. The difference in performance between linear duration-aware and logarithmic duration-aware alignments shows that the logarithmic weighting strategy is better than linear weighting strategy in this context.

## 4 Conclusions

We implemented a novel trace alignment approach using activity duration to improve trace alignment accuracy. We also introduced a set of criteria to quantify trace alignment performance. Based on these criteria and case studies, we compared our algorithms with an existing trace alignment algorithm. The results showed that our duration-aware trace alignment achieved better alignment accuracy and provided more insights into deviations. Our algorithm has important limitations. First, this algorithm cannot handle concurrent activities. When two or more activities occur simultaneously, the algorithm rearranges them in a chronological order based on their start times. Alignment of concurrent activities is still an open research. Second, our duration-aware trace alignment algorithm needs activity durations as the input and cannot be applied on a dataset without activity duration information. Our future work will extend the concepts presented in this paper to improve process trace alignment by considering more information from the time dimension, e.g. activity start/end time, idle time.

**Acknowledgments.** This paper is based on research supported by National Institutes of Health under grant number 1R01LM011834-01A1.

## References

1. Van Der Aalst, Wil. Process mining: discovery, conformance and enhancement of business processes. Springer Science & Business Media, 2011
2. Bose, RP Jagadeesh Chandra, and Wil MP van der Aalst. "Process diagnostics using trace alignment: opportunities, issues, and challenges." *Information Systems* 37.2 (2012): 117-141
3. Waterman, Michael S. Introduction to computational biology: maps, sequences and genomes. CRC Press, 1995.
4. Needleman, Saul B., and Christian D. Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins." *Journal of molecular biology* 48.3 (1970): 443-453.
5. Rakthanmanon, Thanawin, et al. "Searching and mining trillions of time series subsequences under dynamic time warping." *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.
6. Forestier, Germain, et al. "Classification of surgical processes using dynamic time warping." *Journal of biomedical informatics* 45.2 (2012): 255-264.
7. Müller, Meinard. "Dynamic Time Warping." *Information retrieval for music and motion* (2007): 69-84.
8. Holmes, Wendy. Speech synthesis and recognition. CRC press, 2001.
9. Jain, Anil K., and Richard C. Dubes. Algorithms for clustering data. Prentice-Hall, Inc., 1988.
10. Ward Jr, Joe H. "Hierarchical grouping to optimize an objective function." *Journal of the American statistical association* 58.301 (1963): 236-244.
11. Murtagh, Fionn, and Pierre Legendre. "Ward's hierarchical clustering method: Clustering criterion and agglomerative algorithm." *arXiv preprint arXiv: 1111.6285* (2011).
12. Levenshtein, Vladimir I. "Binary codes capable of correcting deletions, insertions, and reversals." *Soviet physics doklady*. Vol. 10. No. 8. 1966.

13. Edgar, Robert C. "MUSCLE: a multiple sequence alignment method with reduced time and space complexity." *BMC bioinformatics* 5.1 (2004): 113.
14. Chang J M, Di Tommaso P, Notredame C. TCS: a new multiple sequences alignment reliability measure to estimate alignment accuracy and improve phylogenetic tree reconstruction [J]. *Molecular Biology and Evolution*, 2014: msu117.
15. Landan, Giddy, and Dan Graur. "Heads or tails: a simple reliability check for multiple sequence alignments." *Molecular biology and evolution* 24.6 (2007): 1380-1383.
16. Penn, Osnat, et al. "GUIDANCE: a web server for assessing alignment confidence scores." *Nucleic acids research* 38.suppl 2 (2010): W23-W28.
17. Thompson, Julie D., Frédéric Plewniak, and Olivier Poch. "A comprehensive comparison of multiple sequence alignment programs." *Nucleic acids research* 27.13 (1999): 2682-2690.
18. Kelleher, Deirdre C., et al. "Effect of a checklist on advanced trauma life support task performance during pediatric trauma resuscitation." *Academic emergency medicine* 21.10 (2014): 1129-1134.
19. Bose, RP Jagadeesh Chandra, and Wil MP van der Aalst. "Context aware trace clustering: towards improving process mining results." *SDM*. 2009.
20. Feng, Da-Fei, and Russell F. Doolittle. "Progressive sequence alignment as a prerequisite to correct phylogenetic trees." *Journal of molecular evolution* 25.4 (1987): 351-360.
21. Cornell, Paul, et al. "Transforming nursing workflow, part 1: the chaotic nature of nurse activities." *Journal of Nursing Administration* 40.9 (2010): 366-373.
22. Chakrabarti, Saikat, et al. "Refining multiple sequence alignments with conserved core regions." *Nucleic acids research* 34.9 (2006): 2598-2606.
23. Bashford, H., et al. "Workflow analysis in production homebuilding." *Associated Schools of Construction 43rd Annual International Conference*. 2007.
24. Rogge-Solti, Andreas, and Gjergji Kasneci. "Temporal anomaly detection in business processes." *Business Process Management*. Springer International Publishing, 2014. 234-249.
25. Clifford, David, et al. "Alignment using variable penalty dynamic time warping." *Analytical chemistry* 81.3 (2009): 1000-1007.