Article:

Ruddle, RA orcid.org/0000-0001-8662-8103, Bernard, J orcid.org/0000-0001-8741-9709, Lücke-Tieke, H orcid.org/0000-0002-0934-6820 et al. (2 more authors) (2021) The Effect of Alignment on Peoples Ability to Judge Event Sequence Similarity. IEEE Transactions on Visualization and Computer Graphics. ISSN 1077-2626

https://doi.org/10.1109/TVCG.2021.3050497

The Effect of Alignment on People's Ability to Judge Event Sequence Similarity

Roy A. Ruddle, Jürgen Bernard, *Member, IEEE,* Hendrik Lücke-Tieke, Thorsten May, and Jörn Kohlhammer, *Member, IEEE.*

Abstract—Event sequences are central to the analysis of data in domains that range from biology and health, to logfile analysis and people's everyday behavior. Many visualization tools have been created for such data, but people are error-prone when asked to judge the similarity of event sequences with basic presentation methods. This paper describes an experiment that investigates whether local and global alignment techniques improve people's performance when judging sequence similarity. Participants were divided into three groups (basic vs. local vs. global alignment), and each participant judged the similarity of 180 sets of pseudo-randomly generated sequences. Each set comprised a target, a correct choice and a wrong choice. After training, the global alignment group was more accurate than the local alignment group (98% vs. 93% correct), with the basic group getting 95% correct. Participants' response times were primarily affected by the number of event types, the similarity of sequences (measured by the Levenshtein distance) and the edit types (nine combinations of deletion, insertion and substitution). In summary, global alignment is superior and people's performance could be further improved by choosing alignment parameters that explicitly penalize sequence mismatches.

Index Terms—Event sequence visualization, sequence alignment, evaluation, user study.

1 Introduction

Comparing event sequences is a pivotal activity in data science [1], and analysis often requires both computation and visualization. For example, by interviewing researchers we have learned that biologists use heuristics to automatically align proteins, but the algorithms are difficult to parameterize so the biologists have to visualize the sequences to manually correct them, human experts need to provide guidance about patterns that are peculiar when analyzing telecommunications logfiles for fraud, and crime analysts consider their work is too dependent on human judgment to be automated. As a result there are many tools for visualizing sequences.

A key research question is how should sequences be presented in visualizations so that people can respond quickly and accurately to compare sequence similarity [2]? We hypothesize that the answer is to align sequences using local [3] or global methods [4]. These both use heuristics to increase the number of events that are aligned across sequences, with local alignment shifting sequences relative to each other, whereas global methods increase alignment by inserting gaps that also fragment sequences. By contrast, basic methods use a pivot event to align sequences and in the present research that was the first event (i.e., the sequences were left-justified). However, to date, there has not been any previous research that compares people's performance with such methods.

This paper investigates whether alignment techniques improve users' performance when judging sequence similarity, and makes three main contributions. First, we propose metrics that may be used in simulations to predict the difficulty of similarity judgments. Second, we compare basic, local, and global alignment methods in a controlled experiment, measuring participants' accuracy and response time. This is the first time that those methods have been investigated in such an experiment, and the results show statistically significant differences between the methods. Third, we show how participants' performance was affected by task complexity, because the experiment also included the number of event types, the similarity of sequences and the edit types as factors. Together, our findings pave the way for visualization tools to better exploit human perception in event sequence analysis.

2 Related Work

This section reviews related work from three perspectives: applications that use visualization to analyze event sequences, techniques for visually encoding event types, and sequence alignment methods. We initially identified 65 papers from visualization outlets (TVCG, Computer Graphics Forum, and Visual Analytics in Healthcare) and general literature searches. Lacking space for an exhaustive review, we chose a subset (see Table 1) that demonstrate diversity across applications, analysis goals/tasks, event types, sequence lengths, visual channels and alignment methods.

2.1 Sequence Analysis Applications

This section outlines the tasks users perform and the scale of data (number of event types and sequence lengths) in sequence analysis applications, from a literature review and interviews with two professors and a PhD student who research event sequence visualization in biology, public

- *R.A. Ruddle is with University of Leeds, Leeds, UK.*
- *J. Bernard is with the University of British Columbia, Canada and with the University of Zurich, Switzerland.*
- *H. Lücke-Tieke and T. May are with Fraunhofer IGD, Germany.*
- *J. Kohlhammer is with Fraunhofer IGD and TU Darmstadt, Germany.*

Manuscript received Month xx, 20xx; revised Month xx, 20xx.

TABLE 1
Characterization of sequence alignment methods used for application examples in previous research (MSA = multiple sequence alignment, MSLA = Multiple-Sequence Local Alignment, PW = pairwise sequence alignment.

Reference	Application Context	Goal/ Task	Number of Event Types	Mean Sequence Length	Visual Channels	Alignment Method
[5]	Biology	Alignment Evaluation	3	18	Color & Shape	Global (*MSA*)
[5]	Biology	Alignment Evaluation	22	7	Color & Letter	Global (*MSA*)
[5]	Biology	Alignment Evaluation	22	18	Color & Shape & Texture	Global (*MSA*)
[6]	Biology	Clustering	21	86	Color & Letter	Global (*MSA*)
[7]	Biology	Event Sequence Search	20	475	Color & Letter	Global (*MSA*)
[8]	Biology	Sequence Alignment	∼20	97	Color & Letter	Global (*MSA*)
[9]	Biology	Sequence Alignment	∼20	35	Color & Letter	Global (*MSA*)
[10]	Biology	Sequence Alignment	23	102	Color & Letter	Global (*MSA*)
[11]	Biology	Sequence Alignment	23	10	Color & Direction	Basic: center
[12]	Biology	Sequence Alignment	(many)	6037	Color	Basic: left
[13]	Biology	Sequence Annotation	4	118	Letter	Global (*MSA*)
[13]	Biology	Sequence Annotation	18	24	Color & Letter	Basic: right
[6]	Biology	Sequence Annotation	21	86	Letter	Global (*MSA*)
[14]	Biology	Similarity Search	20	32	Color & Letter	Global (*PW*)
[15]	Biology	Similarity Search	20	14	Color	Global (*MSA*)
[16]	Biology	Sequence Alignment	4	60	Letter	Local (*MSLA*)
[17]	Health	Clustering	3	8	Color	Basic: left / time
[18]	Health	Clustering	13	∼7	Color	Basic: left
[19]	Health	Cohort Comparison	6	6	Color	Basic: left / time
[20]	Health	Event Sequence Search	6	15	Color	Basic: left
[21]	Health	Sequential Pattern Mining	8	15	Color & Text	Basic: left
[22]	Health	Visual Pattern Analysis	2	7	Shape	Basic: top / time
[23]	Health	Alignment Evaluation	2	5	Color	Basic: center
[24]	Health	Alignment Evaluation	2	7	Color	Basic: center
[25]	Health	Visual Pattern Analysis	3	60	Color & Size	Global (DTW)
[18]	User Log Analysis	Clustering	14	5	Color & Text	*MSA*
[26]	User Log Analysis	Interactive Grouping	10	16	Color	Basic: left
[27]	User Log Analysis	Process Mining	∼9	∼15	Shape & Text	Global
[28]	User Log Analysis	Sequential Pattern Mining	9	14	Color & Size	Basic: left / time
[29]	User Log Analysis	Visual Pattern Analysis	3	38	Color & Position	Basic: top
[30]	User Log Analysis	Visual Pattern Analysis	5	27	Color	Basic: left
[17]	Everyday Life	Clustering	6	4	Color	Sequence Length
[17]	Everyday Life	Clustering	10	n.a.	Color	Basic: left / time
[31]	Everyday Life	Sequential Pattern Mining	15	4	Color & Text	Basic: left
[32]	Everyday Life	Sequential Pattern Mining	74	32	Color	Basic: left
[33]	Neural Networks	Interactive Grouping	5	7	Color & Size	Basic: left
[34]	Synthetic Data	Visual Pattern Analysis	4	6	Color	Basic: left / time

safety and security.One thing that stands out is the variety of applications that involve event sequences, with biology dominant, and health, everyday life and user log analysis also common. There is also considerable variation in the complexity of the sequences (see Table 1).

Biology (including bio-informatics) is the most common application domain. The underlying goals are the analysis of genomes, proteins, and molecules, using tasks such as sequence alignment [5], [8], [9], [10], [11], [12], sequence search and retrieval [7], clustering [6], annotation [6], [13] and similarity search [14], [15]. One characteristic of biology applications that stands out is that they tend to involve a larger number of event types (typically about 20) and longer sequences than other applications, and this is why computation is combined with data visualization. For example, biologists cluster proteins into a small number (typically 4 or 5) of domains that have certain structural characteristics, and then analyze the sequences of domains rather than individual proteins [5], [15].

Health is the second largest application domain, and is characterized by a focus on patient cohorts or cohort comparison, accompanied by interactive clustering and grouping. Most of the examples from this domain involved six or fewer event types (our experiment used sequences with 2 or 6 types), which referred to diseases [22], therapies and drug prescriptions [19], patient transfer [20], or other relevant information in electronic health records [17]. Two approaches with more event types focused on diagnoses (8 types [21]) and hospital activities (13 types [18]).

User log analysis involves analysis tasks that range from blackbox clustering [18] and data mining [26], [29] to interactive grouping [27] and pattern analysis [28], [30]. The examples involve a similar number of event types to health, but longer sequences. However, some visualization tools allow sequences to be analyzed hierarchically to reduce complexity [18].

Everyday life is the fourth application domain, and the examples cover daily life [32], crime signature analysis [31], and career progression and car maintenance [17]. The main analysis tasks are sequence pattern mining and clustering, which are both exploratory in nature. The number of event types is very diverse, ranging from 6 to 74.

We identified two other pieces of research that do not fit into any of the above domains. One supports the interactive-visual grouping and comparison of neural network architectures, and has five event types and an average event length of seven [33]. The other is domain-agnostic, and uses synthetic data examples to outline conceptual aspects of event sequence comparison and visual analysis [34].

2.2 Sequence Visualizations

Color, *shape*, *letters*, *text* and *texture* are the main visual channels for encoding event types in sequence visualizations (see Table 1), and it is well-known that features such as color or shape allow pre-attentive processing in visual search [35]. This section reviews uses of those channels.

Color is the most frequently used channel, and works by color-coding a "canvas". Such a canvas can be either a particular type of area mark (e.g., rectangles [18]) or another visual channel that also varies (shape, letter or texture [5]). Interestingly, although guidelines recommend that no more than 12 categories are encoded with color [36], that is often exceeded by the examples that are listed in Table 1, and particularly by genomics and protein analysis applications. Sometimes similar events are grouped in an upstream step to keep the number of colors manageable [31], [32], [37]. Color is also sometimes used quantitatively in combination with similarity-preserving colormaps [38] to give similar event types similar colors [15]. Alternatively, events can be colored according to their relative position within a (long) sequence [12].

Shape is a visual channel that is suitable for data with a small number of categories [36], as is true for two of the four examples in Table 1 that use shape encoding. One of those uses diamonds and circles to differentiate between coarse and fine-grained events [22]). In general, geometrically regular primitives such as triangles and squares are often used, and guidance about maximizing the perceptual distance between shapes is provided by [39]. To further increase shape variety, one approach adds convexity to the outlines of graphics primitives, resulting in star-like shapes with 4, 6, or 8 points [5]. In other cases the visualization type pre-determines the set of shapes (e.g., states and operators in a process chart [27]).

Letters are frequently applied to encode event types, which is why we describe single letters as a separate visual channel. Referring to Table 1, 35 out of the 37 examples involve fewer event types than there are letters in the Latin alphabet. In genome analysis, a standard visualization approach is using letters to encode the adenine (A), thymine (T), cytosine (C) and guanine (G) base pairs [13]. A similar approach is taken for the amino acid alphabet [6], as one of nine examples that encode about 20 event types with letters. Rather than directly encoding event types, an interesting variant is to encode states that have been computed from event sequences, for example, using three letters to encode matches (M), deletions (D) and insertions (I) [6].

Text (i.e., labels that are made up of multiple letters) that is nested within an area mark is used as a visual channel in four examples [18], [21], [27], [31]. This can help users to make sense of varieties of event types, particularly when the event types are diverse or large in number.

Texture is used in one example. This adopted shape-like textures to increase the number of visually discernible event types [5].

About half of the examples in Table 1 (17 out of 37) employ multiple channels to encode characteristics of event sequences, and we note three different multi-channel strategies. One uses redundancy to encode the same information with multiple visual channels. In many cases, each letter or shape is colored differently [10], [13], or the letters themselves are black and their background (e.g., a rectangle) is colored differently [6], [8], [18], [40]. A second strategy visualizes event types with another source of information (metadata). Most of the examples in Table 1 use letters to encode the event types, in combination with color to link to metadata, such as a coarser level of granularity or other external characteristics [7], [10]. Rather different is the use of direction to differentiate binary characteristics in combination with color [11], or where the combination of colors and shapes results in a complex glyph-like structure [21]. The third strategy makes use of three channels (color, shape, and texture) to depict larger numbers of event types. The approach listed in Table 1 uses color and shape to differentiate between event sequences, and texture to differentiate sub-types within groups [5].

2.3 Sequence Alignment

We divide alignment methods into those that are *query-based* or *heuristic-based*. With all of the methods, the general aim is to either automatically compute or help users find sequences of events that are common or similar among two (*pairwise sequence alignment*) or more (*multiple sequence alignment*) sequences.

Query-based alignment requires pivot events to be selected for each sequence. Sometimes, this is the first event of a sequence (left-alignment) [9], the last event (right-alignment) or both [27]. We call all of these cases *basic alignment*. In other cases, the pivot is defined by a specific event type [41] or transition [42]. The most comprehensive approaches allow users to interactively specify multiple pivot events or transitions [25], [29], e.g., by combinations of event type. All of the query-based alignment approaches preserve the order of events and may preserve temporal differences. However, visual comparison is often limited to an area close to the pivot events.

Heuristic-based methods may be subdivided into methods for local and global alignment. *Local alignment* shifts whole sequences relative to each other to maximize the number of events that are aligned according to some criterion. Common criteria are to maximize the *number of matches* (i.e. count the number of identical events at corresponding positions [43]) or to maximize the number of contiguous matches (the *longest common subsequence* [3]). Our experiment used the latter criterion. *Global alignment* [4] calculates end-to-end correspondences between sequences and, in the process, may introduce gaps or even subsequence permutations. Global alignment quality can be determined by a number of criteria that are combined to calculate a score, i.e., the alignment of the sequences depends on the ratio of bonus and penalty parameters. The criteria are as follows:

- *Match bonuses* and *mismatch penalties* define similarity between corresponding elements and the probability of single-element change (e.g., see [18]). The match bonuses are added to the score and the mismatch penalties are subtracted.
- *Gap penalties* reflect insertions or deletions operating on sequences. Some approaches may further distinguish the number and length of gaps or between

insertion and deletion costs (see [20]), which are subtracted from the score.

- *Reordering penalties* account for the probability of elements or subsequences being rearranged, which is relevant in applications such as genetics [44].
- *Inversion penalties* model the probability of sequences being reversed [45].

Finally, Albers [12] et al. explore the design space for visualizing global alignments from a user-centered viewpoint, orthogonal to the alignment calculation. By changing the visual mapping, their approach allows for the preattentive perception of mismatches, gaps and rearrangements. This work further motivated us to conduct a study to investigate the use of (more complex) global alignment. To the best of our knowledge, such a study has not been conducted before.

3 Predicting the Effect of Alignments

This section describes simulations that were used to conduct a preliminary investigation that predicted the effect of different alignment methods, and choose parameters and factors for the user experiment that follows. Section 3.1 defines two metrics that quantify alignment quality. The other parts use those metrics to investigate how alignment quality varies with different bonus and penalty values in global alignment (Section 3.2) and compare basic, local and global alignment (Section 3.3).

3.1 Metrics

The time taken for serial search tasks increases with the number of items on a display [35], which in our paradigm corresponds to the number of positions in a set of sequences. This leads to a length-based metric (Qlength) that may be used to quantify alignment quality, and is normalized so that the minimum is 1.0 / number_of_sequences, and the maximum is 1.0:

$$Qlength = MAX(sequence\_length\_without\_gaps) \\ /Total\_length\_including\_gaps$$

It is quicker to determine that events match when they are aligned rather than offset, because the similarity is more salient and smaller eye saccades are required [35]. This leads to a second metric that is based on the number of matches (Qmatch) and also has a maximum value of 1.0:

$$Qmatch = Number\_of\_aligned\_matches \\ /MIN(sequence\_length\_without\_gaps)$$

3.2 Parameters for global alignment

The effect of the match bonus, mismatch penalty and gap penalty was investigated as follows. First, a coarse investigation of the effect was performed by randomly generating pairs of sequences for the variables that are listed in Table 2, and then a fine-grained investigation was performed using a narrower range of values for the variables (again, see Table 2). For both investigations, 100 random pairs were generated for each combination of values.

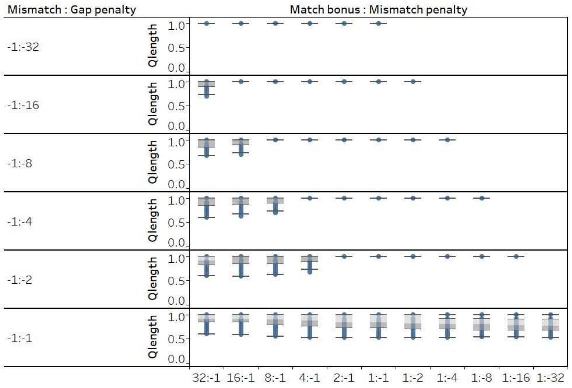


Fig. 1. Results for the fine-grained investigation of global alignment parameters. Qlength was always optimal (= 1.0) for half of the combinations of parameter values, but the other half produced a spread of Qlength values.

For each sequence, a target sequence that contained N different event types and was of length L was generated. The number of events of each type was chosen randomly, ensuring that there was at least one event of each type, and the position of each event was chosen randomly. Then a second sequence was generated by starting with the target, and randomly choosing a substitute event type for each of D positions. The Levenshtein distance [46] between the two sequences was checked, and if it did not equal D then the sequences were regenerated. Once a valid pair had been generated, they were aligned using the Needleman-Wunsch algorithm [4], and Qlength and Qmatch metrics were calculated. Only substitutions were used, because they are more difficult to accommodate in global alignment. Deletions and insertions were included when a comparison was made with basic and local alignment (see Section 3.3).

Only the fine-grained investigation results are reported here. Qlength was optimal (see Figure 1) if both of the following conditions were satisfied:

gap penalty < mismatch penalty
match bonus ≤ ABS(gap penalty)

Qmatch changed more subtly, but the median score increased slightly as the mismatch:gap penalty ratio became closer to -1:-1, and the median also increased if match bonus > ABS(gap penalty).Therefore, for the next part of the investigation we chose parameter values of match bonus = 1, mismatch penalty = -1 and gap penalty = -2.

3.3 Comparing basic, local and global alignment

The next stage of the investigation was to compare basic, local and global alignment, using the same target sequence lengths, numbers of event types and Levenshtein distances as the fine-grained investigation of global alignment (see Table 2). Local alignment was performed using the longest common subsequence (LCS) methodology and basic, left-justified alignment [3].

The alignment methods were compared by randomly generating 1000 pairs of sequences for each combination of the variables values that are listed in Table 2. The target sequences were generated using the same method as Section 3.2, and the second sequence was generated by starting

TABLE 2
The variable values that were used in coarse and fine investigations of the effect of global alignment parameters (see Section 3.2), and to compare basic, local and global alignment (see Section 3.3).

Variable	Global alignment		Basic vs. local vs. global alignment
	Coarse investigation	Fine investigation	
Edit type	Substitute	Substitute	Delete, Insert or Substitute
Total number of sequence pairs	48,600	6,156,000	2,565,000
Match bonus	1, 10 or 100	1, 2, 4, 8, 16 or 32	1
Mismatch penalty	-1, -10 or -100	-1, -2, -4, -8, -16 or -32	-1
Gap penalty	-1, -10 or -100	-1, -2, -4, -8, -16 or -32	-2
Target sequence length (L)	5, 10 or 15	2 – 10, in steps of 1	2 – 10, in steps of 1
Number of event types (N)	2 or L (i.e., 5, 10 or 15)	2 – L, in steps of 1 (45 combinations of L and N)	2 – L, in steps of 1 (45 combinations of L and N)
Levenshtein distance (D)	1, (L + 1)/2 or (L - 1)	1 – (L-1), in steps of 1 (285 combinations of L, N and D)	1 – (L-1), in steps of 1 (285 combinations of L, N and D)

with the target, and randomly choosing a delete, insert or substitute event for each of D positions. Only one edit type was used for each pair.

For delete and insert events, Qmatch was optimal (= 1.0) for global alignment, and generally greater for local alignment than basic alignment (see Figures 2a and 2b). However, for substitute events there were not any clear differences between the alignment methods (see Figure 2c). Qlength was optimal (= 1.0) for every combination of alignment and edit type, except insert and substitute events with local alignment (see Figures 3b and 3c, respectively). Together, the metrics suggest that it is easier to judge the similarity of event sequences with global alignment than local or basic alignment. However, the difference between the alignment methods is affected by the ways in which sequences differ. The edit type produces the greatest differences, with additional differences caused by the number of event types, and the Levenshtein distance.

4 Experiment

This section describes an experiment that investigated the effect of alignment on participants' ability to judge the similarity of event sequences. The experiment used a mixed model design, with the alignment method (Basic, Global or Local) treated as a between participants factor, and three within participants factors. They were chosen from the Section 3.3 results and a four-person pilot study, and were:

- The number of event types in the target sequence (2 vs. 6).
- The Levenshtein distance between the target and two choices (small vs. large; for details, see Section 4.1.2).
- The combination of edit types that were used for the correct and wrong answers (9 combinations; delete vs. delete, delete vs. insert, delete vs. substitute, insert vs. delete, insert vs. insert, insert vs. substitute, substitute vs. delete, substitute vs. insert, and substitute vs. substitute).

From the results that are described in Section 3, we hypothesized that participants would perform faster and/or more accurately when sequences were visualized with global alignment. However it was not possible to make a hypothesis about the overall merits of the other types of alignment, because the Qmatch metric indicated that local alignment is superior to basic alignment, whereas the Qlength metric indicated the opposite.

4.1 Method

4.1.1 Participants

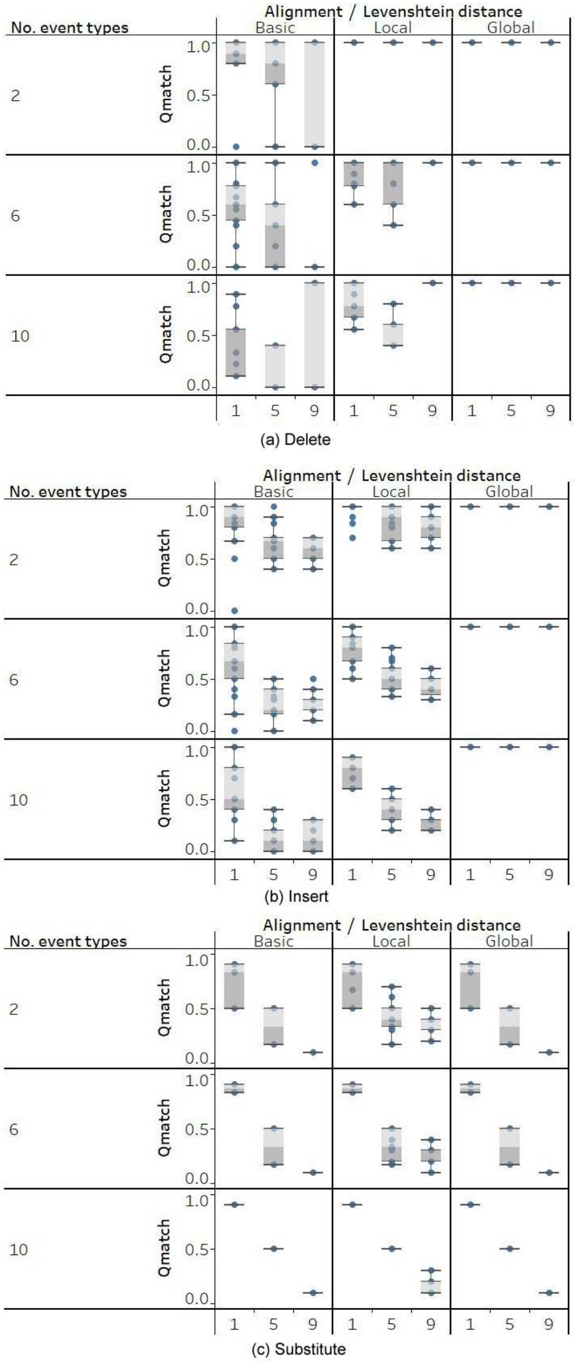
A total of 42 people (33 men, 7 women and 2 who declined to say) with a mean age of 27.4 years (SD = 6.0) participated. The participants were students and staff at the authors' institutions, 12 were studying for their first degree, 28 were graduates and two did not say. The participants took an average of 32 minutes (SD = 13) to complete the whole experiment, gave informed consent but did not receive any form of payment for their participation. The study was approved by the Ethics Committee at the first author's institution.

4.1.2 Materials

The experiment was delivered via a web browser using custom-developed Java software [47], which generated sets of sequences (see Figure 4) using a method that was similar to the one that was used in Section 3.2. First, a target sequence that was of length six, and contained either two or six different event types, was generated. The number of events of each type was chosen randomly, ensuring that there was at least one event of each type, and the position of each event was chosen randomly.

Then two more sequences were generated (the correct choice and the wrong choice), for a given combination of edit types (e.g., insert for the correct choice and delete for the wrong choice) and either a small Levenshtein distance (distance = 1 vs. 2 for the correct and wrong choices, respectively) or a large Levenshtein distance (distance = 2 vs. 4). Each choice was generated by starting with the target, and randomly choosing the same number of positions as the Levenshtein distance. Depending on the edit type, the events in those positions were deleted, new events were inserted, or substitute event types were chosen. The type of each inserted or substituted event was chosen randomly from either two or six types, according to the number of types that were in the target. Finally, the Levenshtein distance between the sequence and the target was checked, and if it was not appropriate then the sequence was regenerated.

When a set of trials was displayed each type of event was shown in a different color, which is the visual channel that is most commonly used in applications of event sequence visualization (see Table 1), is highly ranked for categorical data [48], [49] and, in a previous experiment, allowed participants to judge sequence similarity slightly faster and

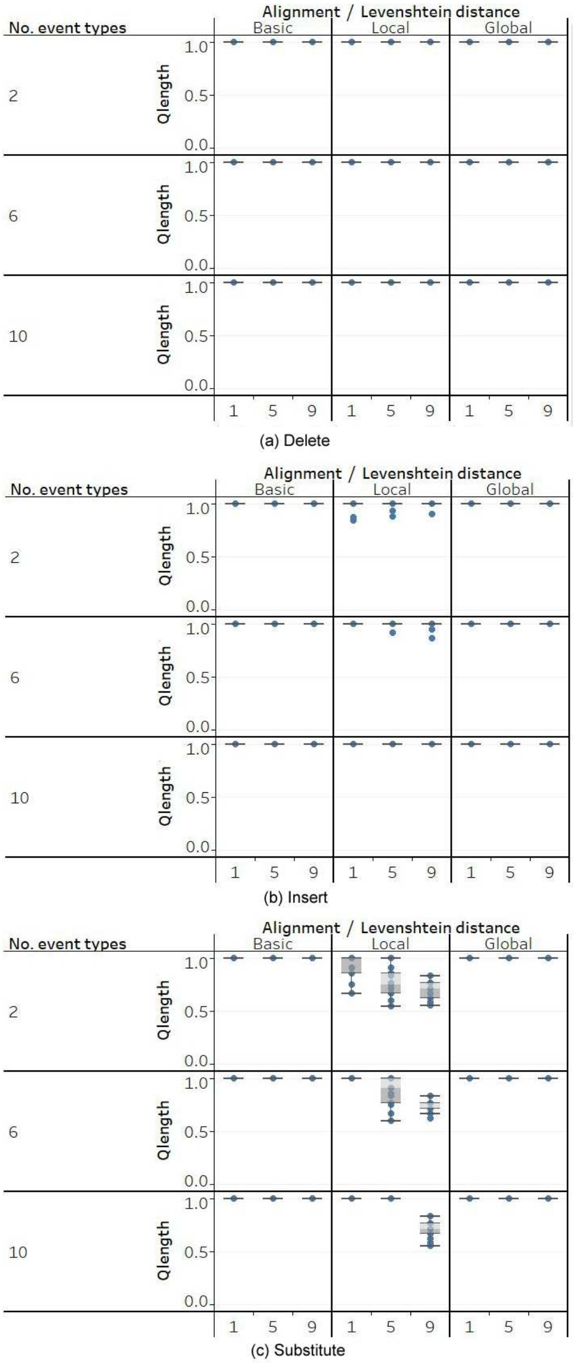


(a) Delete

(b) Insert

(c) Substitute

Fig. 2. Qmatch for combinations of alignment, number of event types, Levenshtein distance and edit type: (a) delete edits, (b) insert edits, and (b) substitute edits. For clarity, only the lowest middle and highest values of number of event types (2, 6 and 10) and Levenshtein distance are shown (1, 5 and 9). Each boxplot contains data for the range of values of target sequence length.



(a) Delete

(b) Insert

(c) Substitute

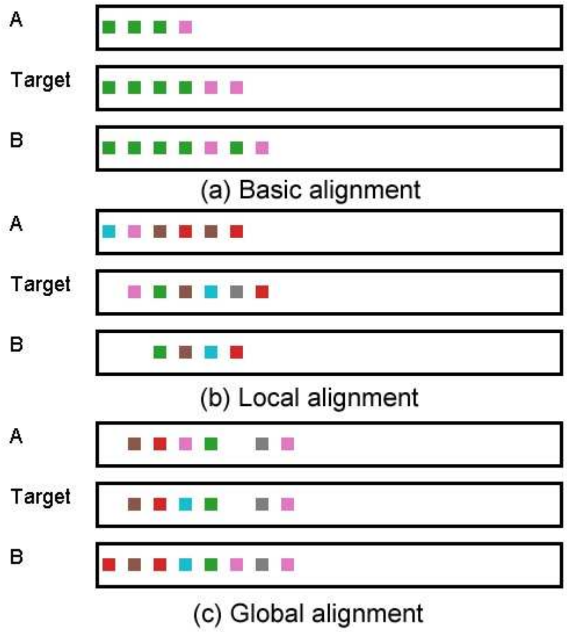
Fig. 3. Qlength for combinations of alignment, number of event types, Levenshtein distance and edit type: (a) delete edits, (b) insert edits, and (b) substitute edits.

more accurately than position encoding [50]. The colors were chosen from a ColorBrewer colorblind-safe palate [51].

4.1.3 Procedure

The experiment was divided into two parts: an introduction and the test. In the introduction, a series of slides that were specific to the participant's group (Basic, Global or Local) were used to explain the task, and the three edit types (insert vs. delete vs. substitute). Then slides were used to present nine practice trials to a participant, one for each combination of correct answer edit type and wrong answer edit type (see Figure 4). For each slide, the participant's task was to "click on the sequence (A or B) that is most similar to the target sequence". Once the participant had made their choice the correct answer was displayed, together with the edits that needed to be made to change the answer into the target.

The test involved five blocks of trials, with 36 trials in each block (one trial for each combination of the within participants factors: number of event types, Levenshtein distance, and correct/wrong answer edit types). The trials



(a) Basic alignment

(b) Local alignment

(c) Global alignment

Fig. 4. Examples of the experiment's trials: (a) Basic alignment: Correct choice is B (1 insertion) whereas A has 2 deletions, (b) Local alignment: Correct choice is B (2 deletions) whereas A has 4 substitutions, and (c) Global alignment: Correct choice is A (1 substitution) whereas B has 2 insertions. Every target sequence contained six events, and either two or six event types.

TABLE 3
Summary of the effects that are reported in the main text, for the longitudinal and trained performance ANOVAs.

Longitudinal (Blocks 1-5)		
Independent variable(s)	% correct	Response time
Block	p = .04	p < .01
Block x Alignment	p = .24	p = .03
Alignment	p = .04	p = .62
Trained (Blocks 4 & 5)		
Independent variable(s)	% correct	Response time
Alignment	p < .01	p = .13
Alignment x Levenshtein distance	p < .01	p = .80
Alignment x Edit type	p = .01	p = .01
Alignment x Edit type x No. event types	p = .61	p = .04
No. event types	p = .68	p < .01
Levenshtein distance	p < .01	p < .01
Edit type	p = .15	p < .01
Levenshtein distance x No. event types	p = .86	p = .02
No. event types x Edit type	p = .07	p < .01
Levenshtein distance x Edit type x No. event types	p = .13	p < .01
Levenshtein distance x Edit type	p = .07	p < .01

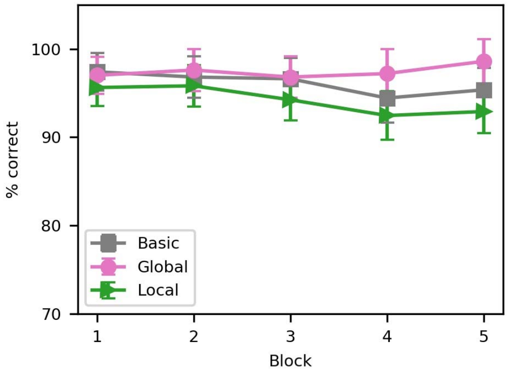


Fig. 5. The mean % correct trials in each block for the participants who used each type of alignment. Error bars show the 95% confidence intervals (CIs).

were presented in a random order. A participant's task was to choose whether sequence A or B was more similar to the target sequence, which they indicated by clicking inside the box that surrounded the relevant sequence. That caused the choice and the participant's response time to be recorded, and the screen to be blanked for 1 second before the next trial was displayed. To reduce fatigue, there was a 30 seconds pause between blocks.

4.2 Results and discussion

This section analyzes participants' performance in terms of the percentage of trials that they got correct and their response time. The response time data was normalized using a log10 transformation.

Analyses of variance (ANOVAs) were used to investigate participants' longitudinal performance and trained performance (see summary in Table 3, and details in Sections 4.2.1 and 4.2.2, respectively). Only statistically significant interactions are reported. A † after a p value indicates that the Greenhouse-Geisser sphericity correction was applied. Finally, in Section 4.2.3 the alignment quality metrics are used to explain some of participants' performance differences.

4.2.1 Longitudinal performance

Participants' performance across the five blocks of trials was analyzed using ANOVAs that treated the alignment group as a between participants factor and block as a within participants factor. For the percentage of trials that participants got correct an ANOVA showed that there were main effects of block (F(4,156) = 2.58, p = .04) and alignment (F(2, 39) = 3.43, p = .04) (see Figure 5). Bonferroni-adjusted pairwise comparisons showed that the Global group got more trials correct than the Local group (p = .04), but none of the other differences between the groups or between the blocks were statistically significant.

For response time, an ANOVA showed that there was a main effect of block (F(2, 86) = 52.84, p < .01†). Bonferroni-adjusted pairwise comparisons showed that participants responded faster from Blocks 1 to 2, 2 to 3 and 3 to 4 (p < .01 in each case), but there was no difference between Blocks 4 and 5 (p = 1.00). There was not a main effect of alignment (F(2, 39) = 0.48, p = .62), but there was a alignment x block interaction (F(8, 156) = 2.18, p = .03), with the Local group changing from responding slowest in Block 1 to quickest in Block 5 (see Figure 6).

4.2.2 Trained performance

Based on the longitudinal analyses, the data from Blocks 4 and 5 were combined to investigate differences that occurred between the experiment's factors once participants' performance had levelled off. This "trained" performance was analyzed using ANOVAs that treated the alignment group as a between participants factor, and the number of

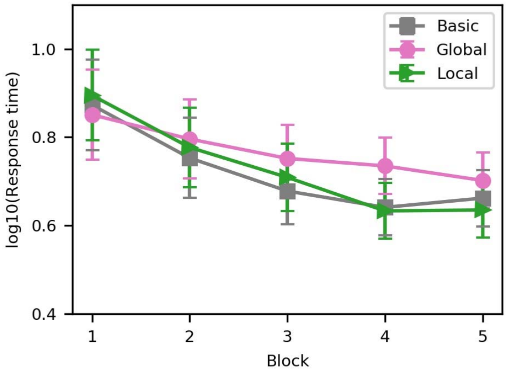


Fig. 6. The mean response time in each block for the participants who used each type of alignment. Error bars show the 95% CIs.

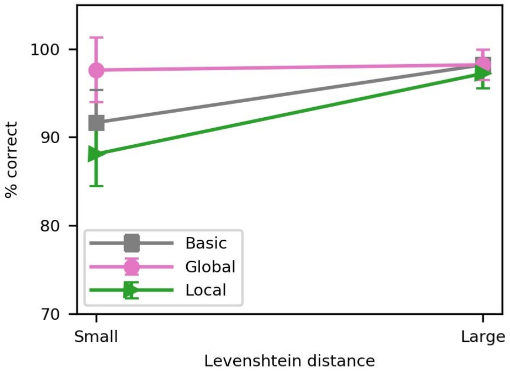


Fig. 7. The percentage of trials that each alignment group got correct with the two Levenshtein distances, in Blocks 4 & 5. Error bars show the 95% CIs.

event types (2 vs. 6), the Levenshtein distance (small vs. large) and the edit types (9 combinations of types for the correct and wrong answers) as within participants factors.

Percentage correct: An ANOVA produced two main effects and two interactions (see Table 3). There was a main effect of alignment ($F(2, 39) = 5.86$, $p < .01$), and Bonferroni-adjusted pairwise comparisons showed that the Global group got more trials correct than the Local group ($p < .01$), but none of the other differences between the groups were statistically significant. There was also a main effect of Levenshtein distance ($F(1, 39) = 27.10$, $p < .01$) and a alignment x Levenshtein distance interaction ($F(2, 39) = 5.88$, $p < .01$). The Basic and Local groups got more trials correct when the distance was large rather than small, whereas the Global group's performance was not affected by distance and appeared to have reached a ceiling (see Figure 7).

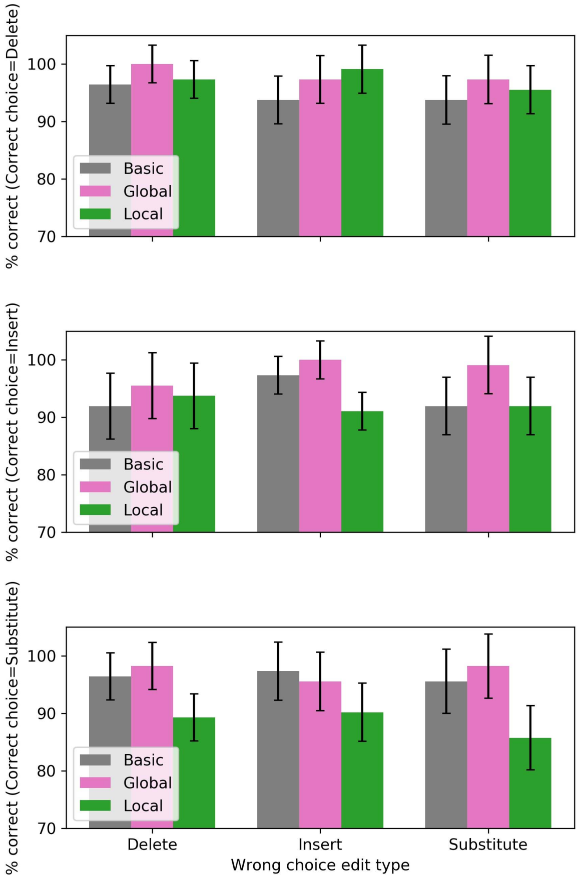


Fig. 8. The percentage of trials that each alignment group got correct for each combination of correct and wrong choice edit type, in Blocks 4 & 5. Error bars show the 95% CIs.

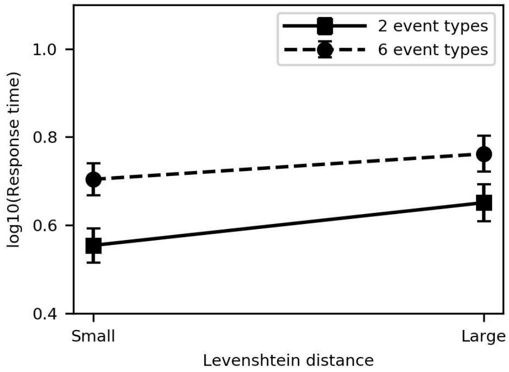


Fig. 9. Participants' mean response times for each combination of Levenshtein distances and the number of event types, in Blocks 4 & 5. Error bars show the 95% CIs.

For the percentage correct data, there was no effect of the number of event types ($F(1, 39) = 0.17$, $p = .68$) and no effect of edit type ($F(6, 234) = 1.59$, $p = .15$†). However, there was a alignment x edit type interaction ($F(16, 312) = 2.05$, $p = .01$). The Global group was largely unaffected by the edit types, whereas the other groups got substantially fewer trials correct for certain combinations of edit type (see Figure 8). The worst performance was by the local alignment group when both the correct and wrong choices involved the substitute edit type.

Response time: An ANOVA produced three main effects and significant interactions (see Table 3). Participants responded faster when there were two rather than six event types ($F(1, 39) = 134.62$, $p < .01$) or a small Levenshtein distance ($F(1, 39) = 58.89$, $p < .01$), and there was a Levenshtein distance x number of event types interaction ($F(1, 39) = 6.35$, $p = .02$) (see Figure 9).

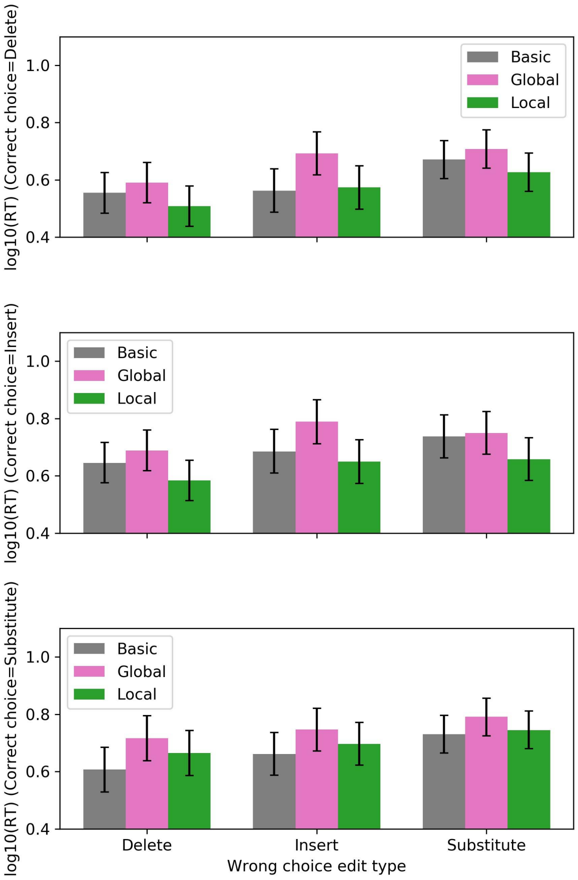


Fig. 10. The alignment groups' mean response times for each combination of correct and wrong choice edit type. The data are for Blocks 4 & 5 and the error bars show the 95% CIs.

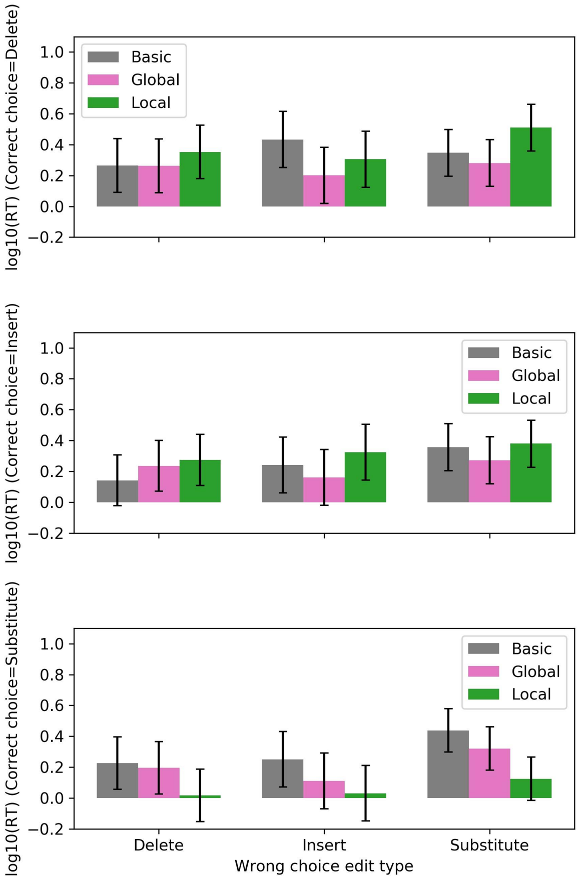


Fig. 11. The difference in the alignment groups' mean response times with 6 event types – 2 event types, for each combination of correct and wrong choice edit type. The data are for Blocks 4 & 5 and the error bars show the 95% CIs.

There was a main effect of edit type (F(8, 312) = 27.93, $p < .01$), with participants responding faster when the correct and/or wrong choice involved deletions, and slower when those choices involved substitutions. There were interactions of Levenshtein distance x edit type (F(8, 312) = 8.85, $p < .01$), number of event types x edit type (F(8, 312) = 3.60, $p < .01$), and Levenshtein distance x number of event types x edit type (F(8, 312) = 2.88, $p < .01$) (see Table 4). These interactions shared two general trends, which were that the response time increased with the number of event types and with the Levenshtein distance. A notable exception occurred when the correct and wrong choices both involved deletions because, with that, participants responded faster when the Levenshtein distance was large.

Finally, although there was no main effect of alignment (F(2, 39) = 2.19, p = .13), there was an alignment x edit type interaction (F(16, 312) = 2.00, p = .01). The Global group were slowest for every combination of edit type, but the magnitude of the difference between the groups varied across the edit types (see Figure 10). There was also an alignment x edit type x number of event types interaction (F(16, 312) = 1.74, p = .04), because participants were typically slower with six event types than two event types, but there were some notable exceptions (e.g., when the local group's correct choice involved substitutions; see Figure 11).

4.2.3 Performance and metrics of alignment quality

In the trained performance data, participants' accuracy was affected by the type of alignment. The local and basic groups were worst, but the global group's better accuracy came at the expense of slower response times. This section investigates errors that the local and basic groups made, and then why the global group may have responded slower. Both investigations started with manual inspection of sets of sequences that were presented to participants, to identify patterns, which were then investigated with alignment quality metrics from Section 3. One of the basic group's participants was omitted because a software error meant that their sets of event sequences were not stored when the experiment took place.

Most of the errors occurred with the small Levenshtein distance. Manual inspection of the trials showed that 62% of the local group's errors occurred when the wrong choice had either the same number or more events aligned with the target than the correct choice. This problem is inherent to local alignment, and made it more likely that participants would make a mistake. Manual inspection also showed that 47% of the basic group's errors occurred when both the correct and wrong choices had the same number of events aligned with the target, which is also a problem that is inherent with the alignment method.

TABLE 4
Mean [95% confidence intervals] of the log10(Response time) for each combination of the correct/wrong choice edit type, the number of event types, and the small and large Levenshtein distance.

Correct choice	Wrong choice	2 event types		6 event types	
		Small	Large	Small	Large
Delete	Delete	0.53 [0.46, 0.60]	0.43 [0.38, 0.47]	0.63 [0.58, 0.68]	0.62 [0.57, 0.67]
	Insert	0.48 [0.44, 0.53]	0.58 [0.52, 0.64]	0.65 [0.59, 0.71]	0.72 [0.66, 0.79]
	Substitute	0.51 [0.46, 0.56]	0.64 [0.58, 0.70]	0.69 [0.64, 0.74]	0.84 [0.79, 0.89]
Insert	Delete	0.55 [0.50, 0.60]	0.62 [0.56, 0.69]	0.72 [0.67, 0.77]	0.67 [0.61, 0.73]
	Insert	0.57 [0.51, 0.63]	0.73 [0.67, 0.79]	0.72 [0.67, 0.78]	0.81 [0.75, 0.87]
	Substitute	0.53 [0.48, 0.59]	0.73 [0.67, 0.78]	0.76 [0.71, 0.81]	0.84 [0.77, 0.91]
Substitute	Delete	0.60 [0.54, 0.67]	0.65 [0.58, 0.71]	0.68 [0.63, 0.73]	0.72 [0.66, 0.78]
	Insert	0.60 [0.54, 0.67]	0.73 [0.66, 0.80]	0.72 [0.67, 0.77]	0.75 [0.69, 0.81]
	Substitute	0.60 [0.55, 0.66]	0.76 [0.70, 0.82]	0.77 [0.71, 0.82]	0.89 [0.84, 0.94]

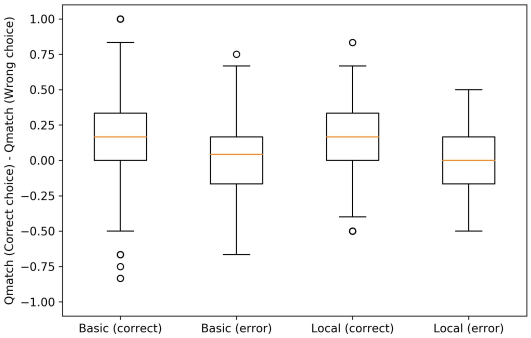
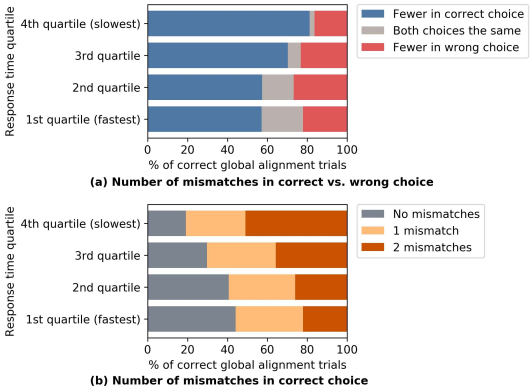


Fig. 12. The difference in Qmatch for the correct and wrong choice in each trial, for each combination of group (basic vs. local alignment) and whether participants were correct or made an error. Whiskers are 1.5 x inter-quartile range.



(a) Number of mismatches in correct vs. wrong choice

(b) Number of mismatches in correct choice

Fig. 13. Mismatches for trials in each quartile of response times for global alignment trials that participants answered correctly, showing: (a) the difference between the correct and wrong choice mismatches, and (b) the number of mismatches in the correct choice.

To investigate the problems, Qmatch was calculated for the correct choice and target, and for the wrong choice and target. This showed that, in 75% of trials that participants got correct, Qmatch was greater for the correct choice than the wrong choice. By contrast, for 50% of trials in which participants made an error, the wrong choice's Qmatch was greater (see Figure 12). That is consistent with participants being more likely to make an error if the wrong choice has a greater number of aligned events.

To investigate the global group's response times, the 20 slowest correct trials were manually inspected. In 50% of those, the correct choice and/or the wrong choice involved substitute events. This problem is not inherent to global alignment, because alignments depend on the ratio of the match bonus, mismatch penalty and gap penalty parameters. To investigate this for all of the global alignment trials, the trials were divided into quartiles according to their response times, and the number of mismatches was counted. One would expect participants to respond faster when there were fewer mismatches in the correct choice than the wrong choice, but the opposite occurred (see Figure 13a). However, when the correct choices were analyzed separately then a different pattern was revealed, which was that the correct choice tended to have a greater number of mismatches when participants responded more slowly (see Figure 13b). In other words, participants were slowed down by the difficulty of comparing each choice to the target, rather than directly comparing the two choices.

5 Conclusions and Future Work

This paper describes a controlled experiment which shows that global alignment allows users to judge the similarity of event sequences more accurately than if either local alignment or basic alignment is used. Global alignment is particularly helpful when users have to judge small differences between sequences, whereas basic alignment may be sufficient for coarser judgements. Our results also pave the way for visualization tools to better exploit human perception by adopting global alignment parameters that heavily penalize mismatches, so that they occur less often.

Global alignment is widely used for computational sequence alignment [46], but this is the first time that it has been investigated in a user experiment. Even though only a limited set of factors could be studied, the sequence lengths and number of event types were similar to some of the examples that are listed in Table 1. Other examples deal with complexity by adding computational or hierarchical elements to the workflow, and our results are directly applicable to where analysts use visualization in those workflows [5], [15], [18]. Beyond that, we anticipate that global alignment will be more beneficial if complexity arises from sequence length rather than the number of event types (e.g., [13], [20], [29], [30]).

Controlled, pre-registered experiments provide a gold standard for investigating users' performance but are time-consuming, which limits the factors that may be investigated. As well as describing a successful experiment, the present research showed how metrics may be combined with simulations to identify good parameter values for user experiments and also indicated how sequence alignment metrics could be improved by taking account of the number of mismatches, as well as length and the number of matches.

Finally, the above conclusions lead to three avenues that require further research. The first is user-centered, where there is a clear need for additional controlled experiments and combining the results to develop a cognitive model that accurately predicts users' accuracy and response time when judging sequence similarity. The second is to incorporate such models within user interfaces that adapt to the complexity of given sequence data and the types of judgment that users make. Such interfaces could then provide the core of new visual analytic tools for event analysis. The third is understanding the workflows that users adopt with those adaptive interfaces, to increase the volume and complexity of data that may be analyzed.

Acknowledgments

This research was supported by the Alexander von Humboldt Foundation, the Engineering and Physical Sciences Research Council (EP/N013980/1), and an Alan Turing Institute Fellowship awarded to R.A.R. Data associated with this paper is available from the University of Leeds at https://doi.org/10.5518/934 [52].

References

[1] Z. Liu, H. Dev, M. Dontcheva, and M. Hoffman, "Mining, pruning and visualizing frequent patterns for temporal event sequence analysis," in *IEEE VIS 2016 Workshop on Temporal & Sequential Event Analysis*, 2016.

[2] F. Du, B. Shneiderman, C. Plaisant, S. Malik, and A. Perer, "Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 6, pp. 1636–1649, 2017.

[3] T. F. Smith, M. S. Waterman *et al.*, "Identification of common molecular subsequences," *Molecular Biology*, vol. 147, no. 1, pp. 195–197, 1981.

[4] S. Needleman, "Needleman-Wunsch algorithm for sequence similarity searches," *Molecular Biology*, vol. 48, pp. 443–453, 1970.

[5] M. Hess, D. Jente, J. Wiemeyer, K. Hamacher, and M. Goesele, "Visual Analysis and Comparison of Multiple Sequence Alignments," in *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2016.

[6] C. J. A. Sigrist, L. Cerutti, E. de Castro, P. S. Langendijk-Genevaux, V. Bulliard, A. Bairoch, and N. Hulo, "Prosite, a protein domain database for functional characterization and annotation," *Nucleic Acids Research*, vol. 38, pp. D161–D166, 2010.

[7] X. Robert and P. Gouet, "Deciphering key features in protein structures with the new endscript server," *Nucleic Acids Research*, vol. 42, no. W1, pp. W320–W324, 2014.

[8] M. Larkin, G. Blackshields, N. Brown, R. Chenna, P. McGettigan, H. McWilliam, F. Valentin, I. Wallace, A. Wilm, R. Lopez, J. Thompson, T. Gibson, and D. Higgins, "Clustal w and clustal x version 2.0," *Bioinformatics*, vol. 23, no. 21, pp. 2947–2948, 2007.

[9] K. Katoh and D. M. Standley, "Mafft multiple sequence alignment software version 7: Improvements in performance and usability," *Molecular Biology and Evolution*, vol. 30, no. 4, pp. 772–780, 2013.

[10] C. L Anderson, C. Strope, and E. N Moriyama, "Suitemsa: Visual tools for multiple sequence alignment comparison and molecular sequence simulation," *BMC Bioinformatics*, vol. 12, p. 184, 2011.

[11] D. M. Goodstein, S. Shu, R. Howson, R. Neupane, R. D. Hayes, J. Fazo, T. Mitros, W. Dirks, U. Hellsten, N. Putnam, and D. S. Rokhsar, "Phytozome: a comparative platform for green plant genomics," *Nucleic Acids Research*, vol. 40, no. D1, pp. D1178–D1186, 2012.

[12] D. Albers, C. Dewey, and M. Gleicher, "Sequence surveyor: Leveraging overview for scalable genomic alignment visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2392–2401, 2011.

[13] H. Sakai, S. S. Lee, T. Tanaka, H. Numa, J. Kim, Y. Kawahara, H. Wakimoto, C.-c. Yang, M. Iwamoto, T. Abe, Y. Yamada, A. Muto, H. Inokuchi, T. Ikemura, T. Matsumoto, T. Sasaki, and T. Itoh, "Rice annotation project database (rap-db): An integrative and interactive database for rice genomics," *Plant and Cell Physiology*, vol. 54, no. 2, p. e6, 2013.

[14] M. P. Heß, "Visual search and analysis in molecular biology," Ph.D. dissertation, Technische Universität Darmstadt, Darmstadt, 2018. [Online]. Available: http://tubiblio.ulb.tu-darmstadt.de/106609/

[15] M. Hess, J. Wiemeyer, K. Hamacher, and M. Goesele, "Serious games for solving protein sequence alignments - combining citizen science and gaming," in *Games for Training, Education, Health and Sports*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, vol. 8395, pp. 175–185.

[16] I. Ovcharenko, G. G. Loots, B. M. Giardine, M. Hou, J. Ma, R. C. Hardison, L. Stubbs, and W. Miller, "Mulan: multiple-sequence local alignment and visualization for studying function and evolution," *Genome research*, vol. 15, no. 1, pp. 184–194, 2005.

[17] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao, "Eventthread: Visual summarization and stage analysis of event sequence data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 56–65, 2018.

[18] B. C. M. Cappers and J. J. van Wijk, "Exploring multivariate event sequences using rules, aggregations, and selections," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 532–541, 2018.

[19] S. Malik, F. Du, M. Monroe, E. Onukwugha, C. Plaisant, and B. Shneiderman, "Cohort comparison of event sequences with balanced integration of visual analytics and statistics," in *International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2015, pp. 38–49.

[20] K. Wongsuphasawat, C. Plaisant, M. Taieb-Maimon, and B. Shneiderman, "Querying event sequences by exact match or similarity search: Design and empirical evaluation," *Interacting with Computers*, vol. 24, no. 2, pp. 55–68, 2012.

[21] A. Perer, B. C. Kwon, and J. Verma, "The critical role of data mining for analyzing real-world event sequences," in *IEEE VIS2016 Workshop on Temporal & Sequential Event Analysis*, 2016.

[22] D. Gotz, F. Wang, and A. Perer, "A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data," *Biomedical Informatics*, vol. 48, pp. 148–159, 2014.

[23] Y. Zhang, S. Di Bartolomeo, F. Sheng, H. Jimison, and C. Dunne, "Evaluating alignment approaches in superimposed time-series and temporal event-sequence visualizations," in *2019 IEEE Visualization Conference*. IEEE, 2019, pp. 1–5.

[24] Y. Zhang, K. Chanana, and C. Dunne, "Idmvis: Temporal event sequence visualization for type 1 diabetes treatment decision support," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 512–522, 2019.

[25] S. Guo, Z. Jin, D. Gotz, F. Du, H. Zha, and N. Cao, "Visual progression analysis of event sequence data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 417–426, 2019.

[26] P. H. Nguyen, R. Henkin, S. Chen, N. Andrienko, G. Andrienko, O. Thonnard, and C. Turkay, "VASABI: Hierarchical user profiles for interactive visual user behaviour analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 77–86, 2019.

[27] X. Lu, D. Fahland, and W. van der Aalst, "Interactively exploring logs and mining models with clustering, filtering, and relabeling," ser. BPM 2016 Tool Demonstration Track, Rio de Janeiro, Brasil, 2016.

[28] P. H. Nguyen, C. Turkay, G. Andrienko, N. Andrienko, O. Thonnard, and J. Zouaoui, "Understanding user behaviour through action sequences: From the usual to the unusual," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 9, pp. 2838–2852, 2019.

[29] J. Li, S. Chen, K. Zhang, G. Andrienko, and N. Andrienko, "COPE: Interactive exploration of co-occurrence patterns in spatial time series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 8, pp. 2554–2567, 2018.

[30] S. Mahmood and K. Mueller, “Taxonomizer: Interactive construction of fully labeled hierarchical groupings from attributes of multivariate data,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2019.

[31] W. Jentner, D. Sacha, F. Stoffel, G. Ellis, L. Zhang, and D. A. Keim, “Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool,” *The Visual Computer*, vol. 34, no. 9, pp. 1225–1241, 2018.

[32] K. Vrotsou and A. Nordman, “Exploratory visual sequence mining based on pattern-growth,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 8, pp. 2597–2610, 2018.

[33] D. Cashman, A. Perer, R. Chang, and H. Strobelt, “Ablate, variate, and contemplate: Visual analytics for discovering neural architectures,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 863–873, 2019.

[34] B. Shneiderman, “The event quartet: How visual analytics works for temporal data,” in *IEEE VIS2016 Workshop on Temporal & Sequential Event Analysis*, 2016.

[35] R. Snowden, R. J. Snowden, P. Thompson, and T. Troscianko, *Basic vision: An introduction to visual perception*. Oxford Univ. Pr., 2012.

[36] T. Munzner, *Visualization analysis and design*. AK Peters, 2014.

[37] J. Bernard, D. Sessler, J. Kohlhammer, and R. Ruddle, “Using dashboard networks to visualize multiple patient histories: A design study on post-operative prostate cancer,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.

[38] J. Bernard, M. Steiger, S. Mittelstädt, S. Thum, D. Keim, and J. Kohlhammer, “A survey and task-based quality assessment of static 2d colormaps,” in *SPIE Conference on Visualization and Data Analysis*, vol. 9397. SPIE Press, 2015, pp. 93 970M–93 970M–16.

[39] Ç. Demiralp, M. S. Bernstein, and J. Heer, “Learning perceptual kernels for visualization design,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1933–1942, 2014.

[40] A. M. Waterhouse, J. B. Procter, D. M. A. Martin, M. Clamp, and G. J. Barton, “Jalview version 2—a multiple sequence alignment editor and analysis workbench,” *Bioinformatics*, vol. 25, no. 9, pp. 1189–1191, 2009.

[41] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson, “Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 321–330, 2017.

[42] J. A. Fails, A. Karlson, L. Shahamat, and B. Shneiderman, “A visual interface for multivariate temporal data: Finding patterns of events across multiple histories,” in *2006 IEEE Symposium On Visual Analytics Science And Technology*, 2006, pp. 167–174.

[43] D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge Univ. Press, 1997.

[44] A. Darling, B. Mau, F. Blattner, and N. Perna, “Mauve: multiple alignment of conserved genomic sequence with rearrangements,” *Genome Research*, vol. 14, no. 7, pp. 1394–1403, 2004.

[45] S. Chappidi and S. Bereg, “Visualization of genome rearrangements using dcj operations,” in *International Workshop on Interactive and Spatial Computing*. ACM, 2018, pp. 15–22.

[46] D. E. Krane, *Fundamental concepts of bioinformatics*. Pearson Education India, 2003.

[47] H. Lücke-Tieke, M. Beuth, P. Schader, T. May, J. Bernard, and J. Kohlhammer, “Lowering the Barrier for Successful Replication and Evaluation,” in *BELIV IEEE VIS Workshop on Evaluation and Beyond - Methodological Approaches for Visualizations*. IEEE, 2018.

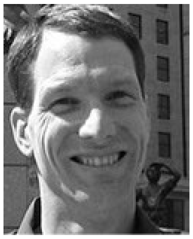
[48] J. Mackinlay, “Automating the design of graphical presentations of relational information,” *ACM Transactions on Graphics*, vol. 5, no. 2, pp. 110–141, 1986.

[49] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, “Voyager: Exploratory analysis via faceted browsing of visualization recommendations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 649–658, 2015.

[50] R. A. Ruddle, J. Bernard, T. May, H. Lücke-Tieke, and J. Kohlhammer, “Methods and a research agenda for the evaluation of event sequence visualization techniques,” in *IEEE VIS 2016 Workshop on Temporal & Sequential Event Analysis.*, 2016.

[51] M. Harrower and C. A. Brewer, “Colorbrewer. org: an online tool for selecting colour schemes for maps,” *The Cartographic Journal*, vol. 40, no. 1, pp. 27–37, 2003.

[52] R. A. Ruddle, J. Bernard, H. Lücke-Tieke, T. May, and J. Kohlhammer, “The effect of alignment on people’s ability to judge event sequence similarity. [dataset].” 2020.



Roy Ruddle is Professor of Computing at the University of Leeds. He has a multidisciplinary background, combining research and development in the software industry with a PhD in psychology. He conducts basic and applied research at the interface of computer graphics and human-computer interaction. His current research focuses on ultra-high-definition displays and visual analytic tools.

Jürgen Bernard is an Assistant Professor of Computer Science at the University of Zurich (UZH), Switzerland. His research is at the intersection between visual analytics and interactive machine learning, with an emphasis on the medical applications. His data-centered focus is on time-oriented data and high-dimensional data. Many of his solutions include techniques from cluster analysis, dimensionality reduction, similarity search, active learning, and classification.

Hendrik Lücke-Tieke is a researcher at the Fraunhofer Institute for Computer Graphics. His background includes network analysis as well as document analysis for industrial applications. He currently focuses on systems that guide users on developing evaluations and analyses, and conducting explorations in heterogenous search spaces.

Thorsten May is a researcher at the Fraunhofer Institute for Computer Graphics Research (IGD) with a background of mathematics and computer science. His research interests include the systematization of options to combine visualization with machine learning, with a particular focus on multivariate data and progressive visual analytics.

Jörn Kohlhammer is the head of the Competence Center for Information Visualization and Visual Analytics at Fraunhofer IGD. His research interests include decision-centered visualization and visual analytics. Kohlhammer has a PhD in computer science from the Technische Universität Darmstadt and is Honorary Professor for user-centered visual analytics at this university.