

# Peekquence: Visual Analytics for Event Sequence Data

Bum Chul Kwon  
IBM T.J. Watson Research  
Center  
Yorktown Heights, NY, USA  
bumchul.kwon@us.ibm.com

Janu Verma  
IBM T.J. Watson Research  
Center  
Yorktown Heights, NY, USA  
jverma@us.ibm.com

Adam Perer  
IBM T.J. Watson Research  
Center  
Yorktown Heights, NY, USA  
adam.perer@us.ibm.com

## ABSTRACT

Exploring event sequences in big data is challenging. Though many mining algorithms have been developed to derive the most frequently occurring and the most meaningful sequential patterns, it is yet difficult to make sense of the results. To tackle the problem, we introduce a visual analytics approach, *Peekquence*. In this paper, we describe the design of *Peekquence*, which aims to increase the interpretability of machine learning-based sequence mining algorithms.

## CCS Concepts

•Human-centered computing → Visual analytics;

## Keywords

Event Sequence; Sequence Mining; Healthcare; Electronic Health Records

## 1. INTRODUCTION

Finding temporal patterns in longitudinal event sequences is a challenging task, as the volume and variety of events often make it difficult to extract salient patterns. In response to this challenge, data scientists have turned to machine learning, known as frequent sequence mining (FSM) techniques, to automatically detect the most common sequences of events to unearth interesting patterns. However, these algorithms often require users to specify a support threshold that, if too high, will yield only a few patterns, or if too low, will yield numerous patterns that may be difficult for data scientists to determine the interesting sequences from the mundane. In this work, we aim to make the results of frequent sequence mining algorithms more interpretable by giving end-users powerful ways to explore the data.

In particular, we propose several new techniques that include: 1) powerful ways to navigate the patterns by sorting with metrics relevant to users (variability, correlation to outcome, etc), 2) integration of patterns with patient time lines, so users can understand where the patterns occur in

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*KDD 2016 Workshop on Interactive Data Exploration and Analytics (IDEA'16) August 14th, 2016, San Francisco, CA, USA.*

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

the actual data, and 3) overviews the summarize the most common events in the patterns.

## 2. RELATED WORK

There are a large number of visual analytics tools designed to making temporal event sequences more interpretable. However, as a recently survey points out, many of them have difficulty handling the volume and variety of data [2].

Recently, there have been several approaches that integrate visualization with machine learning algorithms to surface the most interesting patterns, so only a manageable subset of patterns need to be visualized. Frequent Sequence Mining (FSM) is a popular data mining technique for finding sets of frequently occurring subsequences from a larger set of temporal event sequences. *Peekquence* uses SPAM (Sequential Pattern Mining) [1] as its FSM algorithm, which uses a bitmap-based representation for event sequences for efficiency reasons. Integrating visualization with the data mining algorithms is a promising approach, as it can help users understand algorithmic uncertainties, as well as trust the results of algorithms [9].

There have been other visualization systems that have integrated with FSM techniques. For instance, *Frequent* [5] integrates SPAM with visualization to support finding frequent patterns from longitudinal event sequences. This work was later extended and adapted to a medical context as *Care Pathway Explorer* [6]. However, the visualizations are similar to Sankey Diagrams [8], which have scalability issues when there are many patterns and large event dictionaries. Another system, *TimeStitch* [7], relies on the PrefixSpan [4] algorithm, which has several limitations, and is demonstrated on only small event sequences, generally composed of 2 or less events. *Peekquence* addresses these issues by having interactive sorting, clustering, and overviews to visualize thousands of patterns with large event dictionaries.

## 3. PEEKQUENCE

*Peekquence* is designed to make the results of the SPAM frequent sequence mining algorithm [1] more interpretable. To achieve the goal, the system has four views that present visual representations of the mining results. Figure 1 illustrates the four views: (A) the sequence network view, (B) the event co-occurrence histogram view, (C) the pattern list view, and (D) the patient timeline view. Using four coordinated views, users can interactively explore commonly occurring event sequences as well as their occurrences within patients' records.

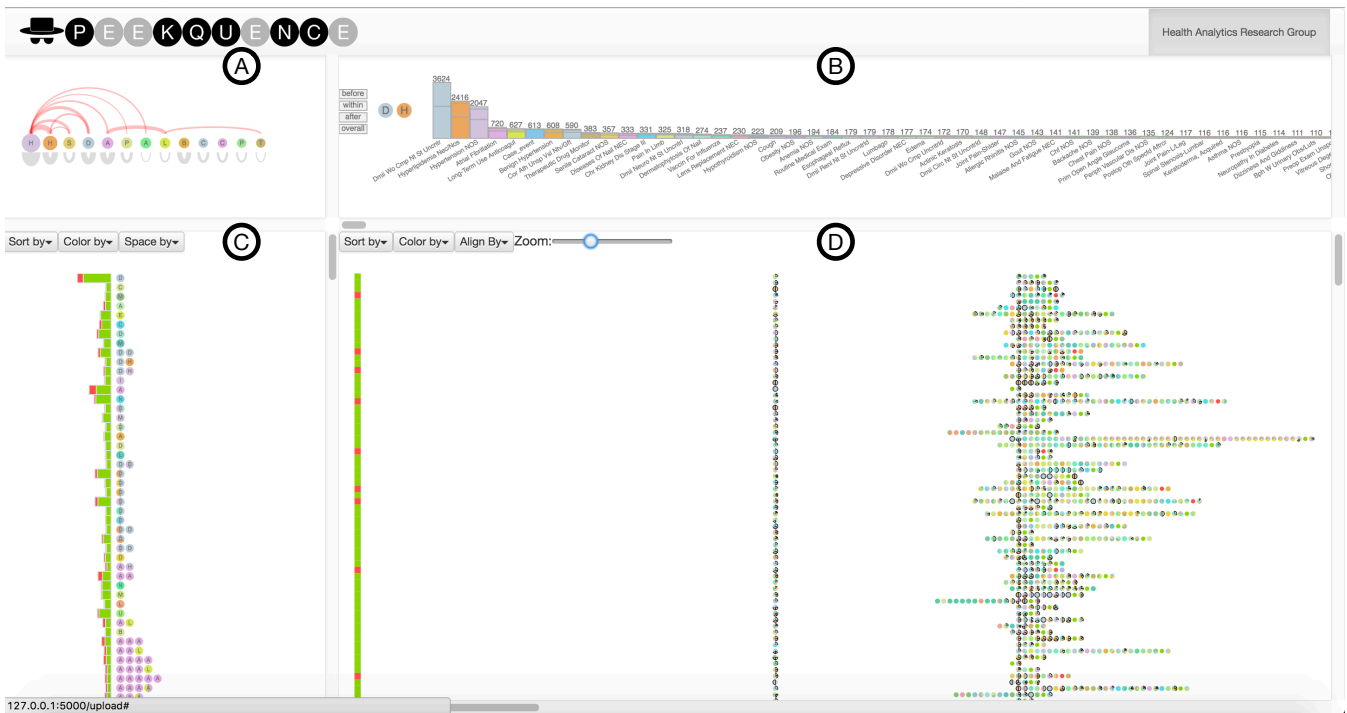


Figure 1: Peekquence consists of four views: (A) the sequence network view showing the frequency of event sequence occurrences within patterns mined from SPAM; (B) the event co-occurrence histogram view showing the frequency of events co-occurring with a pattern selected (“S”, “H” in this example); (C) the pattern list view showing patterns mined from SPAM with event sequences (colored circles with letters) as well as bars of patients with the ratio of case and control labels (diagnosis of a disease); (D) the patient timeline view showing patients’ event sequences aligned with respect to the pattern selected (“S” and “H” events are vertically aligned in this example).



Figure 2: The design of visual elements: circle for unit time duration, pie for event, color and letter for event type.

All four views use a common visual element to visualize event sequences: an *event glyphs* that visually encodes each unique event type that occurs in the mined data. The event glyphs are visually encoded as circles, colored according to an categorical ontology, and labeled with an abbreviation of the event type’s name.

In the situation where multiple event types occur concurrently, the glyph is divided into multiple slices, similar to a pie chart, where each slice represents an event type. For example, Figure 2 shows a pattern consisting of three event types occurring sequentially: L, A, and L & A. In this medical dataset used throughout this paper, the colors represent the category of the clinical event according to ICD-9 (International Classification of Diseases) codes for classifying medical events.

The sequence network view in Figure 1 (A), also shown in Figure 3, acts as an overview, and shows the frequency of co-occurring event types within patterns mined from SPAM. The nodes indicate the types of events, and edges indicate

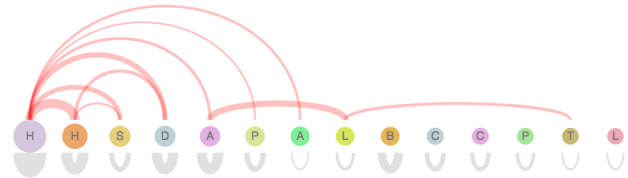


Figure 3: The sequence network view showing the most frequently occurring event types and their co-occurrences.

that two nodes co-occur within patterns. The size of nodes and the thickness of edges show the number of patients that include events and event sequences within their records, respectively. For example, the purple “H” event, representing Hypertension events (a clinical event type indicating high blood pressure), has the largest size and the most edges to other events, showing that many event sequences in mined patterns contain the event. Users can click on a node or an edge to filter the pattern list view (Figure 1 (C)).

The pattern list view in Figure 1 (C) shows all patterns mined from SPAM, aligned vertically. Each row shows a pattern, visualized as a sequence of circular event glyphs that describe the sequence of the mined pattern. In addition, the pattern’s association with outcome is represented by the stacked bar chart to the left of the sequence. In this medical example, the bars are divided into red and green, indicating the proportion of the case patients (patients diagnosed with the disease) and control patients (patients without the disease). This synchronization between pattern and outcome

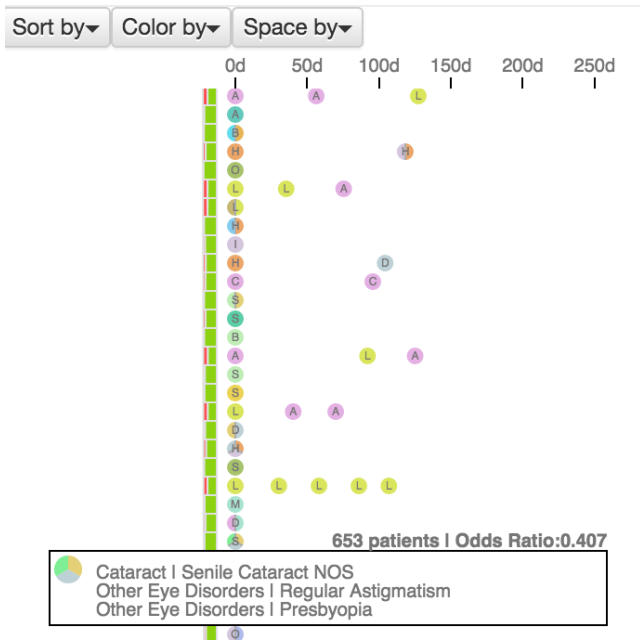


Figure 4: The pattern list view showing patterns of events spread out based on average time duration between events. allows users to understand the impact of each pattern. This view is interactive, so users can sort the pattern list view by various attributes of the pattern: 1) the number of patients that have the pattern; 2) length of the pattern; 3) odds ratio of outcome; 4) variability of events in sequences; 5) clusters based on sequence similarity. Users can choose to horizontally spread event glyphs so that spaces between events indicate the average duration of occurrences of the events within patient records. Figure 4 shows a list of patterns, in which events are spread out to show average duration between the events. Users can click on a pattern to populate patient information in the event co-occurrence histogram view (Figure 1 (B)) and the patient timeline view (Figure 1 (D)).

The event co-occurrence histogram view in Figure 1 (B) shows the summary of patient records which contain the selected pattern from the pattern list view. The summary is the histogram of events co-occurring with the selected pattern within patients. Each bar indicating a event type is divided into three blocks that show events occurring 1) before, 2) within, and 3) after the selected pattern, respectively. For example, Figure 1 (B) shows the histogram of the selected pattern of “D” and “H” events. The top block of each bar in indicates the number of occurrences of the corresponding event before the “D” event within patient records. Subsequently, the second block shows the number of co-occurring events on the same day of or later than the “D” event and before or on the same day of the “H” event. Lastly, the bottom

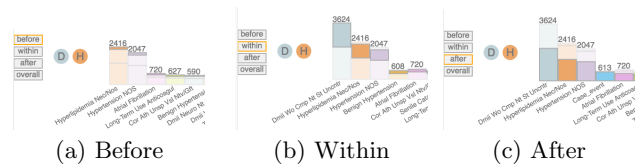


Figure 5: The histogram view sorted by before, within, and after the pattern.

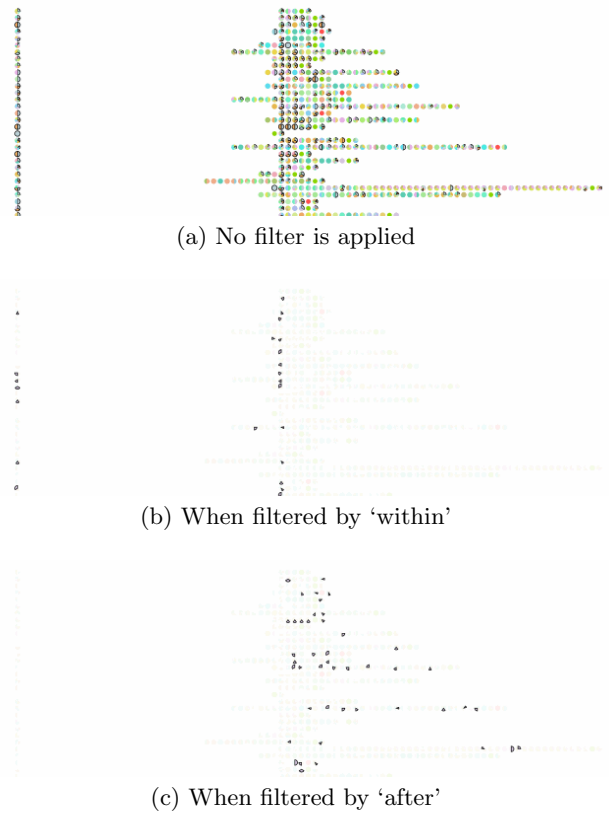


Figure 6: The patient timeline view 6a before filter, 6b filtered by within pattern, and 6c filtered by after pattern.

block indicates the number of co-occurring events later than the “H” event. Using the view, users can find the most commonly co-occurring events with the selected pattern. The view allows users to sort the histogram horizontally by the frequency of events before, within, and after the pattern as shown in Figure 5. In this view, users can select a block of histogram bar to highlight the events within patient records shown in the the patient timeline view (Figure 1 (D)).

The patient timeline view shows individual patient’s entire event sequences per row in Figure 1 (D). The sequences are aligned horizontally so that the selected pattern occurs at the same horizontal location. To do so, we shift patients’ records horizontally, which sometimes creates empty space between events. Thus, in Figure 1 (D), the horizontal distance between events of “D” and “H” indicates the maximum days of events that occurred between the “D” and “H” events within a patient’s record. As mentioned earlier, by clicking a block of the event co-occurrence histogram view, users can filter the patient timeline view. Figure 6 shows the patient timeline view filtered by the event “H”, shown as purple pies, 6b within and 6c after the selected pattern of “D” and “H”.

In Peekquence, the four views independently show information about patterns mined from an algorithm, and they also connect to each other by highlighting and filtering other views. The divided views ensure participants to gain new insights in different levels. At the same time, the interactive exploration enables users to progressively investigate event sequences from overview (top views) to details (bottom views) and vice versa. The sections also let users smoothly switch back and forth between pattern-level investigation (left views) and patient-level investigation (right

views). The design of Peekquence captures information in different granularities providing users with appropriate interpretation layers, which confirm the importance of paving the cow path of users' analysis pattern [3]. To increase the transparency of complex pattern mining algorithms like SPAM, we believe that it is important to provide users with visual channels to different modalities and depths of information through divided-but-connected views.

The current status of Peekquence shows the potential of visual analytics approach to make frequent sequence mining algorithms more interpretable. At the same time, we believe that much work remains to be done to improve the prototype. First, we need to allow users to run the SPAM algorithm with a subset of data as well as user-specified parameters. By doing so, users will have an ability to detect user-defined patterns. Second, we are investigating new methods for visually summarizing event sequences that share common events within them. As the number of patterns grow, it is difficult for users to explore patterns. Thus, visual aggregation will help users understand the difference and similarity between event sequences. Lastly, we are investigating ways to incorporate predictive models so that the model can provide the probability of having certain diseases based on event sequences of users.

## 4. CONCLUSION

In this paper, we presented our visual analytics approach, called Peekquence, which aims to increase the interpretability of frequent sequence mining algorithm such as SPAM. The four views combined with interactions provide useful functionalities for users to make sense of patterns as well as their occurrences within patients' records. In future work, we aim to integrate the visual representation with the algorithm so that users can iteratively run the algorithm with new parameters based on insights gained from previous runs. Work is also in progress to exploit the hierarchy of events and provide users the ability to run SPAM at different levels of detail.

## 5. ACKNOWLEDGMENTS

We would like to thank our colleagues who provided constructive feedback for the research.

## 6. REFERENCES

- [1] J. Ayres, J. E. Gehrke, T. Yiu, and J. Flannick. Sequential pattern mining using bitmaps. pages 429–435, 2002.
- [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*, In press.
- [3] B. C. Kwon, S.-H. Kim, S. Lee, J. Choo, J. Huh, and J. S. Yi. Visohc: Designing visual analytics for online health communities. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of the Visual Analytics Science and Technology)*, 22(1):71–80, 2016.
- [4] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach.

*IEEE Transactions on Knowledge and Data Engineering*, 16(11):1424–1440, 2004.

- [5] A. Perer and F. Wang. Frequency: Interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, pages 153–162, New York, NY, USA, 2014. ACM.
- [6] A. Perer, F. Wang, and J. Hu. Mining and exploring care pathways from electronic medical records with visual analytics. *Journal of Biomedical Informatics*, 56(C):369–378, Aug. 2015.
- [7] P. J. Polack Jr, S.-T. Chen, M. Kahng, M. Sharmin, and D. H. Chau. Timestitch: Interactive multi-focus cohort discovery and comparison. In *IEEE Proceedings of the Visual Analytics Science and Technology (VAST)*, pages 209–210. IEEE, 2015.
- [8] P. Riehmman, M. Hanfler, and B. Froehlich. Interactive sankey diagrams. In *IEEE InfoVis*, pages 233–240, 2005.
- [9] D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis, and D. A. Keim. The Role of Uncertainty, Awareness, and Trust in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of the Visual Analytics Science and Technology)*, 22(01):240–249, Jan. 2016.