Outflow: Visualizing Patient Flow by Symptoms and Outcome

Krist Wongsuphasawat, and David H. Gotz

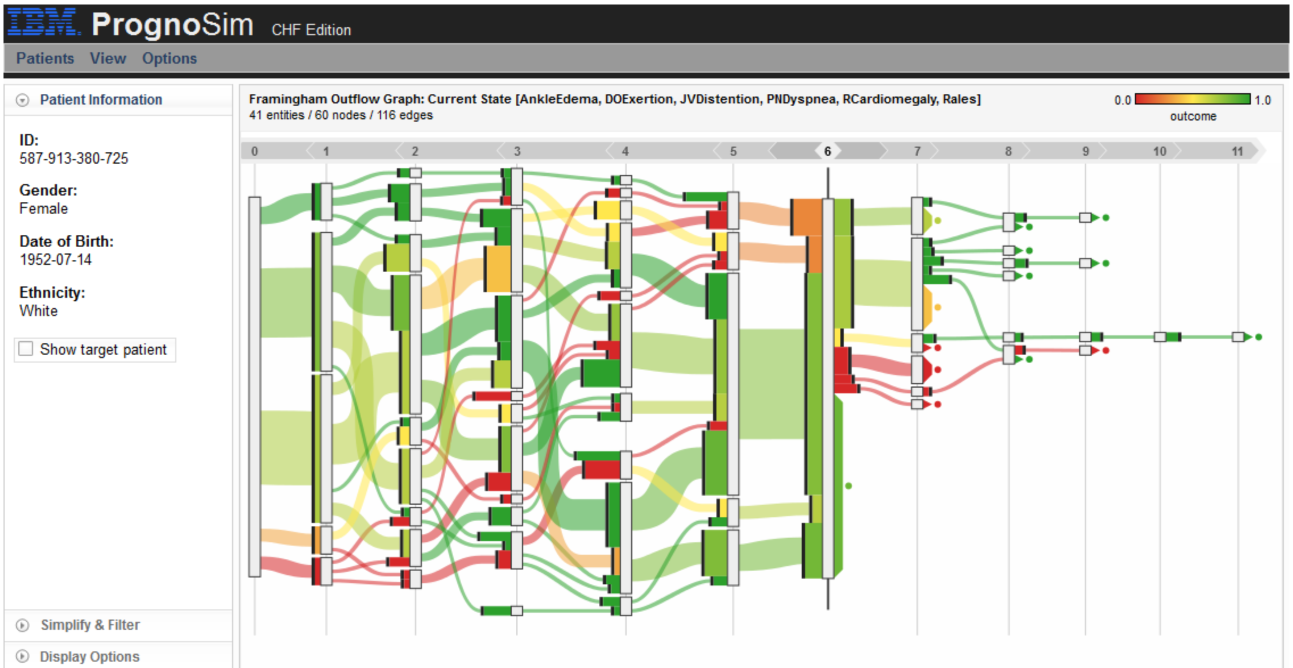

Fig. 1. Outflow aggregates temporal event data from a cohort of patients and visualizes alternative clinical pathways using color-coded edges that map to patient outcome. Interactive capabilities allow users to explore the data and uncover insights.

Abstract—Electronic Medical Record (EMR) databases contain a large amount of temporal events such as diagnosis dates for various symptoms. Analyzing disease progression pathways in terms of these observed events can provide important insights into how diseases evolve over time. Moreover, connecting these pathways to the eventual outcomes of the corresponding patients can help clinicians understand how certain progression paths may lead to better or worse outcomes. In this paper, we describe the Outflow visualization technique, designed to summarize temporal event data that has been extracted from the EMRs of a cohort of patients. We include sample analyses to show examples of the insights that can be learned from this visualization.

Index Terms—Outflow, Information Visualization, Temporal Event Sequences, State Diagram, State Transition

1 Introduction

Electronic medical records (EMRs) are proliferating throughout the healthcare system. At major medical institutions such as hospitals and large medical groups, these computer-based systems contain vast amounts of historical patient data complete with patient profile information, structured observational data such as diagnosis codes and medications, as well as unstructured physician notes. The information in these enormous databases can be useful in guiding the diagnosis of incoming patients or in clinical studies of a disease. However, the vast amount of information can be overwhelming and makes these datasets difficult to analyze. In particular, EMR databases contain a large amount of temporal disease events such as diagnosis dates and the onset dates for various symptoms. Analyzing disease progression pathways in terms of these observed events can provide important insights into how diseases evolve over time. Moreover, connecting these pathways to the eventual outcomes of the corresponding patients can help clinicians understand how certain progression paths may lead to better or worse outcomes.

In this paper, we describe the Outflow visualization technique. Outflow is designed to summarize temporal event data that has been extracted from the EMRs of a cohort of patients. We present a novel interactive visual design which combines multiple patient records into a graph-based visual presentation. Users can manipulate the visualization through direct interaction techniques (e.g., selection and brushing) and a series of control widgets. The interactions allow users to explore the data in search of insights. Throughout the paper we describe Outflow using a motivating problem related to the diagnosis of congestive heart failure. We include two sample analyses to show examples of the insights that can be learned from this visualization.

The rest of the paper are organized as follows. We describe our motivating problem in Section 2 and review related work in Section 3. We explain the design of Outflow in Section 4 and demonstrate preliminary analyses in Section 5. The paper concludes in Section 6.

- *Krist Wongsuphasawat is with University of Maryland. This work is part of his internship at IBM T.J. Watson Research Center. , E-mail: kristw@cs.umd.edu.*
- *David H. Gotz is with IBM T.J. Watson Research Center, E-mail: dgotz@us.ibm.com.*

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

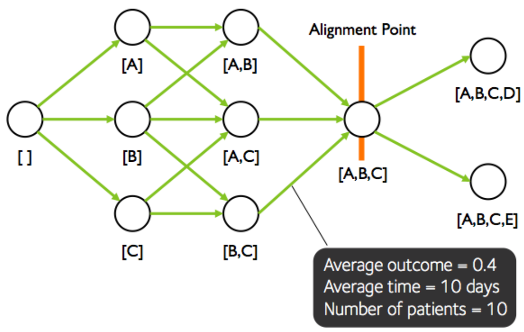

Fig. 2. Multiple medical records are aggregated into a representation called an Outflow graph. This structure is a directed acyclic graph (DAG) that captures the various event sequences that led to the alignment point and all the sequences that occurred after the alignment point. Aggregate patient statistics are then anchored to the graph to describe specific patient subpopulations.

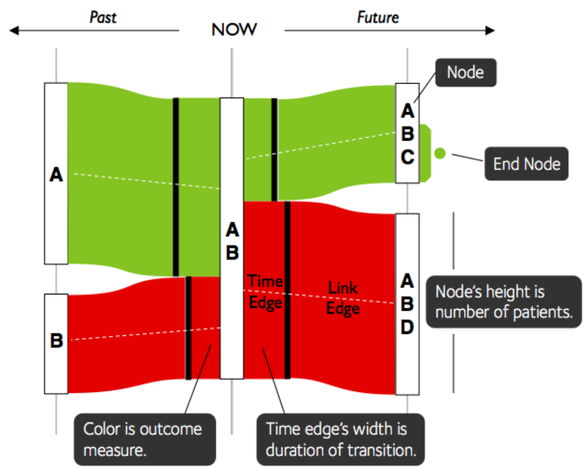

Fig. 3. Outflow visually encodes nodes in the Outflow graph using rectangles while edges are represented using two distinct visual marks: time edges and link edges. Color is used to encode average outcome.

2 Motivating Problem

Congestive heart failure (CHF) is generally defined as the inability of the heart to supply sufficient blood flow to meet the needs of the body. CHF is a common, costly, and potentially deadly condition that afflicts roughly 2% of adults in developed countries with rates growing to 6-10% for those over 65 years of age [12]. The disease is difficult to manage and no system of diagnostic criteria has been universally accepted as the gold standard.

One commonly used system comes from the *Framingham study* [11]. This system requires the simultaneous presence of at least two major symptoms (e.g., S3 gallop, Acute pulmonary edema, Cardiomegaly) or one major symptom in conjunction with two minor symptoms (e.g., Nocturnal cough, Pleural effusion, Hepatomegaly). In total, 18 distinct Framingham symptoms have been defined.

While these symptoms are used regularly to diagnose CHF, our medical collaborators are interested in understanding how the various symptoms and their order of onset correlate with patient outcome. To examine this problem, we were given access to an anonymized dataset of 6,328 patient records. Each patient record includes timestamped entries for each time a patient was diagnosed with a Framingham symptom. For example:

Patient#1:(27 Jul 2009, Ankle edema), (14 Aug 2009, Pleural effusion), ...
Patient#2:(17 May 2002, S3 gallop), (1 Feb 2003, Cardiomegaly), ...

In line with the use of Framingham symptoms for diagnosis, we assume that once a symptom has been observed it applies perpetually. We therefore filter the event sequences for each patient to select only the first occurrence of a given symptom type. The filtered event sequences describe the *flow* for each patient through different disease states. For example, a filtered event sequence ***symptom A*** → ***symptom B*** indicates that the patient's flow is ***no symptom*** → ***symptom A*** → ***symptoms A and B***. The data also has an outcome for each patient (dead (0) or alive (1)).

Our analysis task, therefore, is to examine aggregated statistics for the flows of many patients to find common disease states and transitions between states. In addition, we wish to discover any correlations between these paths and patient outcome.

3 Related Work

3.1 Temporal Event Sequence Visualizations

Many researchers have explored visualization techniques for temporal event sequences. In the early years, many systems focused on visualizing a single record [1, 2, 6, 8, 9, 16]. The most common approach is to place the events on a horizontal timeline according to the time that events occurred. Later, attention shifted towards visualizing multiple records in parallel. One popular technique is to stack instances of single-record visualizations and to provide additional functionality for searching [7, 21, 22, 23, 26], filtering [23], and grouping [5, 14]. However, these approaches do not aggregate nor provide any abstraction of multiple event sequences. Most recently, a technique called *LifeFlow* [25] introduced a way to aggregate and provide an abstraction for multiple event sequences. However, LifeFlow's aggregation combines multiple event sequences into a tree, while Outflow's aggregation combines multiple event sequences into a graph.

3.2 State Diagram Visualizations

Our approach aggregates event sequences into an *Outflow graph* which is analogous to a state diagram [4] or state transition graph. State diagrams are used in computer science and related fields to represent a system of states and state changes. State diagrams are generally displayed as simple node-link diagrams where each state is depicted as a node and transitions are drawn as links [3]. Many visualizations of state diagrams have been developed [3, 17, 18, 20, 24]. These typically focus on multivariate graphs where a number of attributes are associated with every node. Some support exploration of sequences of three or more states. Variants on traditional state diagrams have also been explored, such as *Petri nets* (also known as a *place/transition* net or *P/T net*) [13] which offer a graphical notation for stepwise processes that include choice, iteration, and concurrent execution. However, to the best of our knowledge, these approaches do not display or allow easy comparison of the transition time, which is one of Outflow's design goals.

3.3 Flow & Parallel Coordinates Visualizations

Another group of visualizations called Sankey Diagrams [19] was designed to visualize flow quantities in process systems. However, they only focus on displaying the proportion of the flow that splits in different ways, without temporal information. The visual display of Outflow also looks similar to parallel coordinates [10], but the underlying data types are different. Parallel coordinates are used for categorical data while Outflow was designed for temporal event sequences.

4 Description of the Visualization

4.1 Data Aggregation

The first step in Outflow is data aggregation. We begin by selecting an *alignment point*. For example, we can align a set of patient event sequences around a state where all patients have the same three symptoms A, B and C and no other symptoms. After choosing an alignment point, we construct an *Outflow graph* (Figure 2) using data from all patients that satisfy the alignment point.

The Outflow graph is a state diagram represented using a directed acyclic graph (DAG). The states are the unique combinations of symptoms that were observed in the data. Edges capture symptom transitions. Each edge is annotated with the number of patients that make the corresponding transition, the average time gap between the states, and the average outcome of the patient group.

Therefore, the Outflow graph captures all event paths that led to the alignment point and all event paths that occur after the alignment point. Our prototype implementation lets users select a target patient from the database and uses the target patient's current state as the alignment point. This approach allows for the analysis of historical data when considering the possible future progression of symptoms for the selected target patient.

4.2 Visual Encoding

Based on the information contained in the Outflow graph, we have designed a rich visual encoding that displays (a) the time gap for each state change, (b) the cardinality of patients in each state and state transition, and (c) the average patient outcome for each state and transition. Drawing on prior work from FlowMap [15] and LifeFlow [25], we developed the visual encoding shown in Figure 3.

Node (State): Each node is represented by a rectangle which has its height proportional to the number of patients.

Layer: We slice the graph vertically into layers. Layer *i* contains all Outflow graph nodes with *i* symptoms. The layers are sorted from left to right, showing information from the past to the future. For example, in Figure 1, the first layer (layer 0) contains only one node, which represents patients that have no symptom. The next layer (layer 1) has five nodes, one for each first-occurring symptom in the patient cohort.

Edge (Transition): Each edge is displayed using two visual marks: a *time edge* and a *link edge*. Time edges are rectangles that whose width is proportional to the average time gap of the transition and height is proportional to the number of patients. Link edges connect nodes and time edges to convey sequentiality.

End Node: Each patient's path can stop in a different state. We use a trapezoid followed by a circle to mark these points. The height of the trapezoid is proportional to the number of patients whose path stops at a given point.

Color-coding: Colors assigned to edges and end nodes are used to encode the average outcome for the corresponding set of patients. The color scales linearly from red to green with red representing the worst and green representing the best outcomes.

4.3 Interactions

To allow interactive data exploration, we further designed Outflow to support the following user interaction capabilities.

Panning & Zooming: Users can pan and zoom to uncover detailed structure.

Filtering: Users can filter both nodes and edges based on the the number of associated patients to remove small subgroups.

Symptom Selection: Users can select which symptom types are used to construct the Outflow graph. This allows, for instance, for the omission of symptoms that users deem uninteresting. For example, a user can remove *Nocturnal Cough* if they deem it irrelevant to an analysis and the visualization will be recomputed dynamically.

Brushing: Hovering the mouse over a node or an edge will highlight all paths traveled by patients passing through the corresponding point in the outflow graph (see Figure 4).

Tooltips: Hovering also triggers the display of tooltips which provide more information about individual nodes and edges. Tooltips shows all symptoms associated with the corresponding node/edge, the average outcome, and the total number of patients in the subgroup (see Figure 4).

5 Preliminary Analysis

We have integrated the Outflow visualization technique into a prototype decision support system for CHF patients called *PrognoSim*. This system uses a patient similarity-based approach to provide medical intelligence. PrognoSim is a web-based application written using Java's J2EE platform and Apache Tomcat as the application server environment. The PrognoSim user interface is rendered using HTML and JavaScript. Dojo is used for traditional user interface widgets. The Outflow visualization component is rendered on an HTML 5 canvas via a scenegraph-based JavaScript visualization library named CVL.

We used Outflow within PrognoSim to view the evolution over time for a cohort of CHF patients similar to a clinician's current patient. Our initial analysis illuminates a number of interesting findings and highlights that various types of patients evolve differently. We share two such evolution patterns as examples of the type of analysis that can be performed using the Outflow technique.

Leading Indicators. In several scenarios, patient outcome is strongly correlated with certain leading indicators. For example, consider the patient cohort visualized in Figure 1. The strong red and green colors assigned to the first layer of edges in the visualization shows that the eventual outcome for patients in this cohort is strongly correlated with the very first symptom to appear. Similarly, the strong red and green colors assigned to the first layer of edges after the alignment point show that the next symptom to appear may be critical in determining patient outcome.

Progressive Complications. In contrast to the prior example, which showed strong outcome correlation with specific paths, the patient cohort in Figure 5 exhibits very different characteristics. At each time step, the outcomes across the different edges are relatively equal. However, the outcomes transition from green to red when moving left to right across the visualization. This implies that for this group of patients, no individual path is especially problematic historically. Instead, a general increase in co-occurring symptoms over time is the primary risk factor.

6 Conclusions and Future Work

We have introduced a novel visualization called Outflow that summarizes temporal event data extracted from multiple patient medical records to show aggregate disease evolution statistics for a cohort of patients. We described our motivating problem in the study of congestive heart failure and presented the main visual design concepts behind our visualization. We also described a number of interactive features in Outflow that allow more sophisticated analyses. Finally, we briefly shared two example analysis results which highlight some of the capabilities of our approach.

Due to these early promising results, we plan to continue work on this topic in the future. We believe that there are many promising directions to explore including integration with forecasting/prediction algorithms, the use of more sophisticated similarity measures, and deeper evaluation studies with practitioners. Moreover, the flexibility of Outflow's design means it can be used beyond our motivating problem and can be useful for a range of medical (and non-medical) problems which involve temporal event data.

Acknowledgments

The authors wish to thank Charalambos (Harry) Stavropoulos, Robert Sorrentino and Jimeng Sun for their help and suggestions.

References

[1] W. Aigner, S. Miksch, B. Thurnher, and S. Biffl. PlanningLines: Novel Glyphs for Representing Temporal Uncertainties and Their Evaluation. In *Proc. International Conf. Information Visualization (IV)*, pages 457–463, 2005.

[2] R. Bade, S. Schlechtweg, and S. Miksch. Connecting time-oriented data and information to a coherent interactive visualization. In *Proc. Annual SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pages 105–112, 2004.

[3] J. Blaas, C. P. Botha, E. Grundy, M. W. Jones, R. S. Laramee, and F. H. Post. Smooth graphs for visual exploration of higher-order state transitions. *IEEE Trans. Visualization and Computer Graphics*, 15(6):969–76, 2009.

[4] T. Booth. *Sequential Machines and Automata Theory.* , 1967.

[5] M. Burch, F. Beck, and S. Diehl. Timeline trees: visualizing sequences of transactions in information hierarchies. In *Proc. Working Conf. Advanced Visual Interfaces (AVI)*, pages 75–82, 2008.

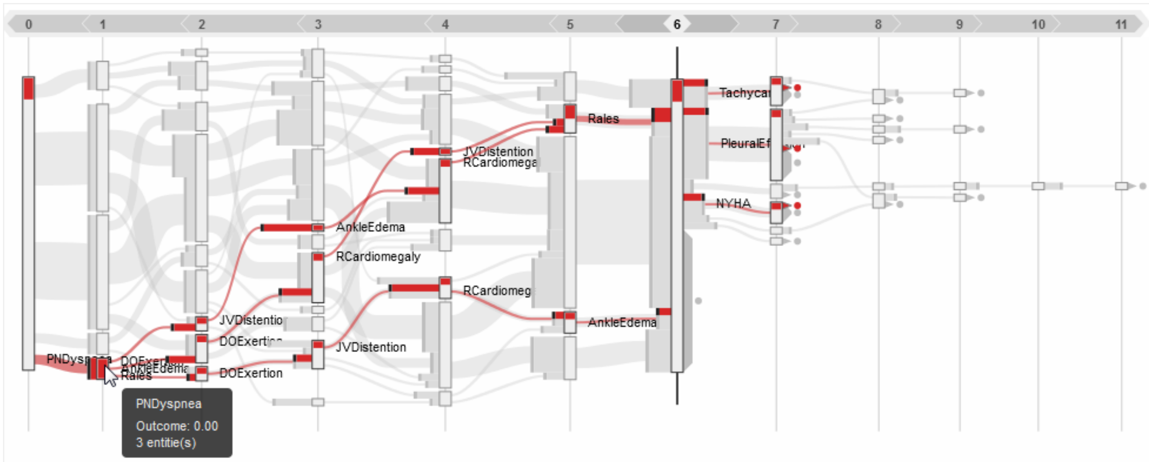

Fig. 4. Interactive brushing allows users to highlight paths emanating from specific nodes or edges in the visualization.

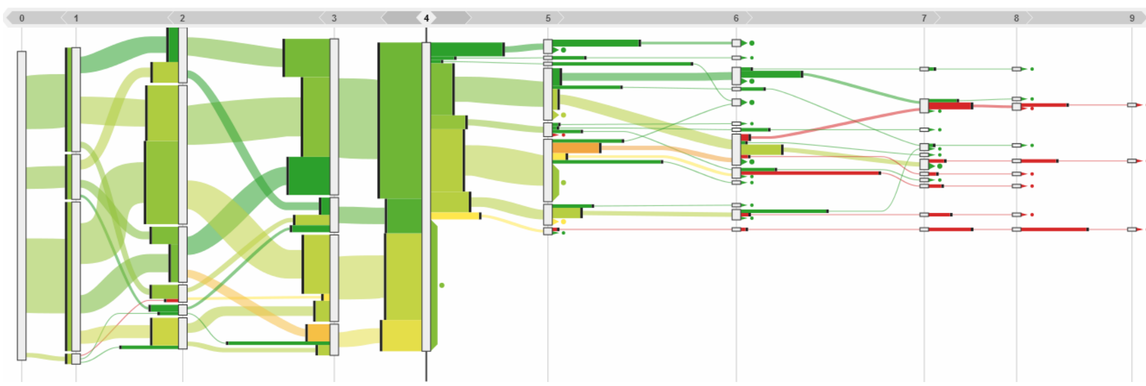

Fig. 5. The progression from green to red when moving left to right in this figure shows that patients with more symptoms exhibit worse outcomes.

[6] S. Cousins and M. Kahn. The visual display of temporal information. *Artificial Intelligence in Medicine*, 3(6):341–357, 1991.

[7] J. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A Visual Interface for Multivariate Temporal Data: Finding Patterns of Events across Multiple Histories. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST)*, pages 167–174, 2006.

[8] B. Harrison, R. Owen, and R. Baecker. Timelines: an interactive system for the collection and visualization of temporal data. In *Proc. Graphics Interface (GI)*, pages 141, 1994.

[9] G. Karam. Visualization using timelines. In *Proc. ACM SIGSOFT International Symp. Software Testing and Analysis*, pages 125–137, 1994.

[10] R. Kosara, F. Bendix, and H. Hauser. Parallel sets: interactive exploration and visual analysis of categorical data. *IEEE Trans. Visualization and Computer Graphics*, 12(4):558–68, 2006.

[11] P. A. McKee, W. P. Castelli, P. M. McNamara, and W. B. Kannel. The natural history of congestive heart failure: the Framingham study. *The New England journal of medicine*, 285(26):1441–6, 1971.

[12] J. J. V. McMurray and M. A. Pfeffer. Heart failure. *Lancet*, 365(9474):1877–1889, 2005.

[13] C. A. Petri. Communication with automata. Technical report, DTIC Research Report, 1966.

[14] D. Phan, A. Paepcke, and T. Winograd. Progressive multiples for communication-minded visualization. In *Proc. Graphics Interface (GI)*, page 225, 2007.

[15] D. Phan, L. Xiao, and R. Yeh. Flow map layout. In *Proc. IEEE Symp. Information Visualization*, pages 219–224, 2005.

[16] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman. LifeLines: using visualization to enhance navigation and analysis of patient records. In *Proc. AMIA Annual Symp.*, pages 76–80, 1998.

[17] A. J. Pretorius and J. J. van Wijk. Visual analysis of multivariate state transition graphs. *IEEE Trans. Visualization and Computer Graphics*, 12(5):685–92, 2006.

[18] A. J. Pretorius and J. J. van Wijk. Visual Inspection of Multivariate Graphs. *Computer Graphics Forum*, 27(3):967–974, 2008.

[19] P. Riehmann, M. Hanfler, and B. Froehlich. Interactive Sankey diagrams. In *Proc. IEEE Symp. Information Visualization*, pages 233–240, 2005.

[20] F. van Ham, H. van de Wetering, and J. van Wijk. Interactive visualization of state transition systems. *IEEE Trans. Visualization and Computer Graphics*, 8(4):319–329, 2002.

[21] K. Vrotsou, K. Ellegard, and M. Cooper. Everyday Life Discoveries: Mining and Visualizing Activity Patterns in Social Science Diary Data. In *Proc. International Conf. Information Visualization (IV)*, pages 130–138, 2007.

[22] K. Vrotsou, J. Johansson, and M. Cooper. ActiviTree: interactive visual exploration of sequences in event-based data using graph similarity. *IEEE Trans. Visualization and Computer Graphics*, 15(6):945–52, 2009.

[23] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman. Aligning temporal data by sentinel events: discovering patterns in electronic health records. In *Proc. Annual SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pages 457–466, 2008.

[24] M. Wattenberg. Visual exploration of multivariate graphs. In *Proc. Annual SIGCHI Conf. Human Factors in Computing Systems (CHI)*, page 811, 2006.

[25] K. Wongsuphasawat, J. Guerra Gómez, C. Plaisant, T. Wang, M. Taieb-Maimon, and B. Shneiderman. LifeFlow: Visualizing an Overview of Event Sequences. In *Proc. Annual SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pages 1747–1756, 2011.

[26] K. Wongsuphasawat and B. Shneiderman. Finding comparable temporal categorical records: A similarity measure with an interactive visualization. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST)*, pages 27–34, 2009.